INOVANCE

H5U系列

可编程逻辑控制器指令手册

前言

资料简介

H5U系列PLC是一款高性能小型PLC，主机结构设计紧凑，自带16路输入和14路输出。

本手册介绍产品编程应用时使用的编程基础知识、基本指令及指令示例、复杂应用指令及指令示例等。

面向的读者

本手册面向以下读者对象：

- 电气工程师
- 设备硬件工程师
- 软件工程师
- 系统工程师

初次使用

对于初次使用本产品的用户，应先认真阅读本手册。若对一些功能及性能方面有所疑惑，请咨询我公司的技术支持人员，以获得帮助，对正确使用本产品有利。

相关手册列表

手册类型	手册名称	手册编码
用户手册	*《H5U系列可编程逻辑控制器用户手册》*	19011155
指令手册	*《H5U系列可编程逻辑控制器指令手册》*	19011156（本手册）
编程与应用手册	*《H5U系列可编程逻辑控制器编程与应用手册》*	19011157

版本变更记录

修订日期	发布版本	变更内容
2019–12	A00	手册第一次发布。
2020–06	A01	新增轴组指令及指令详解。 更新错误代码列表。 修正上一版中的错误。
2020–08	A02	修正上一版中的错误。

关于手册获取

本手册不随产品发货，如需获取电子版PDF文件，可以通过以下方式获取：

- 登录汇川技术官网网站（http://www.inovance.com）下载PDF文件。
- 扫描手册二维码下载PDF文件。

目录

前言 1
1 指令概要 9
1.1 指令构成 9
1.2 软元件和变量一览表 9
1.3 软元件 12
1.3.1 位软元件 12
1.3.2 字软元件 12
1.3.3 特殊软元件 13
1.3.4 字元件的位操作 14
1.4 变量 14
1.4.1 自定义变量 14
1.4.2 定义变量 14
1.4.3 定义数组 16
1.4.4 定义结构体 16
1.4.5 变量的使用方法 17
1.5 图形块指令 17
1.5.1 图形块指令的构成 17
1.5.2 图形块指令编程 18
1.6 功能与函数（FB/FC） 19
1.6.1 功能块 (FB) 19
1.6.2 函数 (FC) 23
1.6.3 授权功能块 26
2 指令速查表 28
2.1 指令速查表 28
3 指令详解 38
3.1 程序逻辑指令 38
3.1.1 触点指令 38
3.1.1.1 指令列表 38
3.1.1.2 LD/LDI/LDP/LDF 38
3.1.1.3 AND/ANI/ANDP/ANDF 39
3.1.1.4 OR/ORI/ORP/ORF 40
3.1.1.5 MEP/MEF 41
3.1.2 输出控制指令 42
3.1.2.1 指令列表 42
3.1.2.2 OUT 43
3.1.2.3 SET 43
3.1.2.4 RST 44
3.1.2.5 ZSET 44
3.1.2.6 ZRST 45
3.1.2.7 PLS/PLF 46
3.1.2.8 ALT 47
3.1.3 能流控制指令 48
3.1.3.1 指令列表 48
3.1.3.2 ANB/ORB 49
3.1.3.3 MPS/MRD/MPP 49
3.1.3.4 INV 50
3.1.3.5 NOP 50

3.2 流程控制指令 51
3.2.1 指令列表 51
3.2.2 CJ 51
3.2.3 LBL 51
3.2.4 CALL 52
3.2.5 SRET 54
3.2.6 SSRET 55
3.2.7 IRET 55
3.2.8 EI/DI 56
3.2.9 WDT 56
3.2.10 FOR/NEXT 57
3.3 SFC指令 59
3.3.1 指令列表 59
3.3.2 STL 59
3.3.3 RET 59
3.3.4 OUTSTL/SETSTL/RSTSTL 60
3.4 触点指令 61
3.4.1 触点比较 61
3.4.1.1 指令列表 61
3.4.1.2 AND# 63
3.4.1.3 LD# 64
3.4.1.4 OR# 66
3.4.1.5 FLDD# 67
3.4.1.6 FAND# 68
3.4.1.7 FOR# 69
3.4.1.8 LDZ# 71
3.4.1.9 ANDZ# 72
3.4.1.10 ORZ# 73
3.4.2 触点逻辑运算 75
3.4.2.1 指令列表 75
3.4.2.2 LD# 75
3.4.2.3 AND# 76
3.4.2.4 OR# 77
3.5 数据运算指令 78
3.5.1 四则运算 78
3.5.1.1 指令列表 78
3.5.1.2 ADD 79
3.5.1.3 SUB 80
3.5.1.4 MUL 81
3.5.1.5 DIV 81
3.5.1.6 MOD 82
3.5.1.7 EADD 83
3.5.1.8 ESUB 84
3.5.1.9 EMUL 85
3.5.1.10 EDIV 86
3.5.1.11 INC 87
3.5.1.12 DEC 88
3.5.2 数据逻辑运算 89
3.5.2.1 指令列表 89
3.5.2.2 WAND 89
3.5.2.3 WOR 90
3.5.2.4 WXOR 91
3.5.2.5 NEG 92

3.5.2.6 ENEG 93
3.5.3 字的位运算 94
3.5.3.1 指令列表 94
3.5.3.2 BLD 95
3.5.3.3 BLDI 95
3.5.3.4 BAND 96
3.5.3.5 BANDI 97
3.5.3.6 BOR 98
3.5.3.7 BORI 98
3.5.3.8 BOUT 99
3.5.3.9 BSET 100
3.5.3.10 BRST 101
3.5.4 三角函数 102
3.5.4.1 指令列表 102
3.5.4.2 SIN 102
3.5.4.3 TAN 103
3.5.4.4 COS 103
3.5.4.5 ASIN 104
3.5.4.6 ACOS 105
3.5.4.7 ATAN 106
3.5.4.8 RAD 107
3.5.4.9 DEG 108
3.5.4.10 SINH 108
3.5.4.11 COSH 109
3.5.4.12 TANH 110
3.5.5 表格运算 110
3.5.5.1 指令列表 110
3.5.5.2 WSUM 111
3.5.5.3 MEAN 111
3.5.5.4 LIMIT 112
3.5.5.5 BZAND 114
3.5.5.6 ZONE 115
3.5.5.7 SCL 116
3.5.5.8 SCL2 118
3.5.6 指数运算 120
3.5.6.1 指令列表 120
3.5.6.2 EXP 120
3.5.6.3 LOG 121
3.5.6.4 LOGE 122
3.5.6.5 ESQR 123
3.5.6.6 SQR 123
3.5.6.7 POW 124
3.6 数据处理指令 125
3.6.1 数据转换 125
3.6.1.1 指令列表 125
3.6.1.2 INT 125
3.6.1.3 BCD 126
3.6.1.4 BIN 127
3.6.1.5 FLT 128
3.6.1.6 EBCD 129
3.6.1.7 EBIN 129
3.6.1.8 DABIN 130
3.6.1.9 BINDA 131
3.6.1.10 WBIT 133

3.6.1.11 UNI 134
3.6.1.12 DWTOW....134
3.6.1.13 DIS 135
3.6.1.14 BTOW....136
3.6.1.15 WTOB....137
3.6.1.16 BITW....138
3.6.1.17 WTODW....139
3.6.1.18 ASCI 140
3.6.1.19 HEX....142
3.6.2 数据传送与比较....144
3.6.2.1 指令列表 144
3.6.2.2 MOV 144
3.6.2.3 EMOV 145
3.6.2.4 BMOV 146
3.6.2.5 SMOV 147
3.6.2.6 FMOV 148
3.6.2.7 CML....149
3.6.2.8 CMP....150
3.6.2.9 ECMP 151
3.6.2.10 ZCP....152
3.6.2.11 EZCP....153
3.6.3 表格操作 154
3.6.3.1 指令列表 154
3.6.3.2 SORT 154
3.6.3.3 SORT2 155
3.6.3.4 SER 157
3.6.3.5 FDEL....159
3.6.3.6 FINS 160
3.6.3.7 POP....161
3.6.4 数据移位 163
3.6.4.1 指令列表 163
3.6.4.2 ROR....164
3.6.4.3 ROL....164
3.6.4.4 RCR....165
3.6.4.5 RCL 166
3.6.4.6 SFTR....167
3.6.4.7 SFTL 168
3.6.4.8 WSFR 169
3.6.4.9 WSFL....170
3.6.4.10 SFWR 171
3.6.4.11 SFRD 171
3.6.4.12 SFR 172
3.6.4.13 SFL 173
3.6.5 其他数据处理....175
3.6.5.1 指令列表 175
3.6.5.2 SWAP 175
3.6.5.3 BON 176
3.6.5.4 SUM 177
3.6.5.5 RAND 177
3.6.5.6 XCH....178
3.6.5.7 ABS 179
3.6.5.8 EABS....180
3.6.5.9 PID....181
3.6.5.10 CCD....181

3.6.5.11 CRC 182
3.6.5.12 LRC 184
3.6.5.13 RAMP 185
3.7 矩阵指令 186
3.7.1 矩阵运算 186
3.7.1.1 指令列表 186
3.7.1.2 BK+ 187
3.7.1.3 BK- 188
3.7.1.4 MAND 189
3.7.1.5 MOR 191
3.7.1.6 MXNR 192
3.7.1.7 MXOR 193
3.7.1.8 MINV 195
3.7.2 矩阵比较 195
3.7.2.1 指令列表 195
3.7.2.2 BKCMP# 196
3.8 字符串指令 198
3.8.1 指令列表 198
3.8.2 STR 198
3.8.3 STRMOV 202
3.8.4 VAL 203
3.8.5 ESTR 207
3.8.6 EVAL 209
3.8.7 $ADD 213
3.8.8 LEN 214
3.8.9 INSTR 215
3.8.10 RIGHT 216
3.8.11 LEFT 218
3.8.12 MIDW 219
3.8.13 MIDR 221
3.8.14 $MOV 222
3.9 时钟指令 223
3.9.1 指令列表 223
3.9.2 TCMP 224
3.9.3 TZCP 225
3.9.4 TADD 226
3.9.5 TSUB 227
3.9.6 HTOS 228
3.9.7 STOH 229
3.9.8 TRD 231
3.9.9 TWR 232
3.9.10 HOUR 233
3.10 高速计数器指令 234
3.10.1 指令列表 234
3.10.2 HC_Preset 234
3.10.3 HC_Counter 237
3.10.4 HC_TouchProbe 240
3.10.5 HC_Compare 245
3.10.6 HC_ArrayCompare 247
3.10.7 HC_StepCompare 249
3.10.8 指令错误代码 253
3.11 定时器指令 253
3.11.1 定时器指令参数 253

3.11.2 指令列表 254
3.11.3 TPR 254
3.11.4 TONR 256
3.11.5 TOFR 257
3.11.6 TACR 259
3.12 指针指令 260
3.12.1 指令列表 260
3.12.2 PTGET 261
3.12.3 PTINC 262
3.12.4 PTDEC 263
3.12.5 PTADD 263
3.12.6 PTSUB 264
3.12.7 PTMOV 265
3.12.8 PT# 266
3.13 通信协议指令 267
3.13.1 指令列表 267
3.13.2 SerialSR 268
3.13.3 面向连接的套接字TCP通信 270
3.13.4 TCP_Listen 270
3.13.5 TCP_Accept 272
3.13.6 TCP_Connect 273
3.13.7 TCP_Close 275
3.13.8 TCP_Send 276
3.13.9 TCP_Receive 278
3.13.10 TCP服务器通信实例 279
3.13.11 TCP客户端通信实例 280
3.13.12 面向无连接的套接字UDP通信 281
3.13.13 UDP_Receive 282
3.13.14 UDP_Bind 284
3.13.15 UDP_Send 285
3.13.16 UDP通信实例 287
3.13.17 套接字通信指令错误代码 287
3.13.18 ECT_ReadParameter_CoE 288
3.13.19 ECT_WriteParameter_CoE 290
3.14 运动控制轴指令 292
3.14.1 指令列表 292
3.14.2 MC轴状态机简介 293
3.14.3 MC_Power 294
3.14.4 MC_Reset 296
3.14.5 MC_ReadStatus 298
3.14.6 MC_ReadAxisError 301
3.14.7 MC_ReadDigitalInput 302
3.14.8 MC_ReadActualPosition 304
3.14.9 MC_ReadActualTorque 306
3.14.10 MC_ReadActualVelocity 307
3.14.11 MC_SetPosition 308
3.14.12 MC_TouchProbe 311
3.14.13 MC_MoveRelative 318
3.14.14 MC_MoveVelocity 324
3.14.15 MC_MoveAbsolute 327
3.14.16 MC_Jog 334
3.14.17 MC_TorqueControl 339
3.14.18 MC_Home 344

3.14.19 MC_Stop 347
3.14.20 MC_Halt 350
3.14.21 MC_MoveFeed 353
3.14.22 MC_MoveBuffer 361
3.14.23 MC_ImmediateStop 366
3.14.24 轴故障码 370
3.15 轴组控制指令 374
3.15.1 指令列表 374
3.15.2 MC_MoveLinear 375
3.15.3 MC_MoveCircular 383
3.15.4 MC_GroupStop 387
3.15.5 MC_GroupPause 389
3.15.6 故障码 391
3.16 CANopen轴控指令 393
3.16.1 指令列表 393
3.16.2 MC_Power_CO 394
3.16.3 MC_Reset_CO 395
3.16.4 MC_ReadActualVelocity_CO 397
3.16.5 MC_ReadActualPosition_CO 397
3.16.6 MC_Halt_CO 398
3.16.7 MC_Stop_CO 400
3.16.8 MC_MoveVelocity_CO 401
3.16.9 MC_MoveRelative_CO 403
3.16.10 MC_MoveAbsolute_CO 405
3.16.11 MC_Home_CO 407
3.16.12 MC_Jog_CO 409
3.16.13 MC_WriteParameter_CO 411
3.16.14 MC_ReadParameter_CO 413
3.16.15 CANopen轴控指令错误代码 414
4 附录 416
4.1 ASCII码转换表 416
4.2 错误代码列表 419

1 指令概要

1.1 指令构成

H5U可使用的指令由指令部及操作数构成。指令部及软元件部的用途如下所示。

- 指令部: 指令功能描述。
- 操作数: 指令中使用的数据。

操作数包含输入数据、输出数据、数值数据。

输入（S）

输入是运算中使用的数据。

根据各指令中指定的变量及软元件，输入数据的用法如下所示。

表1–1 输入数据

类别	描述
常数	指定运算中使用的数值。 由于是在创建程序时进行设置，因此在程序执行中不能更改。
软元件 变量	程序执行中，通过更改指定的软元件中存储的数据，可以更改该指令中使用的数据。

输出（D）

输出中存储运算后的数据。根据具体指令，有时在运算前需要在目标中存储运算中使用的数据。

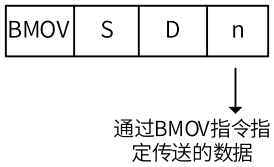
示例 — INT位数据的加法运算：

D元件必须设置用于存储数据的变量及软元件

软元件数/ 传送数/ 数据数/ 字符串数等的数值（n）

在指定使用多个软元件的指令及指定重复次数、指定处理的数据数、字符串数等的指令中、指定软元件数、传送数、数据数、字符串数等时、使用数值。

示例 — 块传送指令：



1.2 软元件和变量一览表

H5U系列PLC支持位软元件、字软元件、特殊软元件、变量、数组、结构体以及自定义变量，详情如下：

位软元件

类型	范围	点数	数据类型	描述
X	X0~X1777	1024点， 8进制编码	BOOL	掉电不保存
Y	Y0~Y1777	1024点， 8进制编码	BOOL	掉电不保存
M	M0-M7999	8000点	BOOL	M0-M999掉电不保存， M1000及之后掉电保存；
S	S0-S4095	4096点	BOOL	S0-S999掉电不保存， S1000及之后掉电保存；
B	B0-B32767	32768点	BOOL	B0-B999掉电不保存， B1000及之后掉电保存；

字软元件

类型	范围	点数	数据类型	描述
D	D0-D7999	8000点	BOOL/ INT/ DINT/ REAL	D0-D999掉电不保存， D1000之后掉电保存；
R	R0-R32767	32768点	BOOL/ INT/ DINT/ REAL	R0-R999掉电不保存， R1000及之后掉电保存；
W	W0-W32767	32768点	BOOL/ INT/ DINT/ REAL	W0-W999掉电不保存， W1000及之后掉电保存；

自定义变量

类型	范围	点数	数据类型	描述
Pointer	-	4096点（32位）	BOOL/ INT/ DINT/ REAL数组	掉电不保存；
BOOL INT DINT REAL	-	2MB（8位）	INTDINTREAL变量， INTDINTREAL数组， INTDINTREAL组合结构体	256KB掉电保存， 其他掉电不保存；

特殊软元件

类型	功能	范围	点数	描述
L	跳转标签	L0-L1023	1024点	与CJ指令、LBL指令配套使用；
K	10进制	K-32,768 ~ K32,767 (16 位)， K-2,147,483,648 ~ K2,147,483,647 (32 位)	-	-
H	16进制	H0000 ~ HFFFF (16 位)， H00000000 ~ HFFFFFFFF (32 位)	-	-

类型	功能	范围	点数	描述
E	浮点数、实数	0，-1.0*2e128 ~ -1.0*2e-126，1.0*2e-126 ~ 1.0*2e128(32位)	-	-
字符	字符、字符串	-	-	字符、字符串，作为指令参数

M8000和D8000特殊元件

特殊软元件	功能描述	访问权限
M8000	用户程序运行时置为ON状态	只读
M8001	M8000状态取反	只读
M8002	用户程序开始运行的第一个周期为ON	只读
M8003	M8002状态取反	只读
-	-	-
M8011	10ms时钟周期的振荡时钟	只读
M8012	100ms时钟周期的振荡时钟	只读
M8013	1S时钟周期的振荡时钟	只读
M8014	1分钟时钟周期的振荡时钟	只读
-	-	-
M8020	运算零标志	只读
M8021	运算借位标志	只读
M8022	运算进位标志	只读
M8029	多周期指令执行完成标志位，适用于RAMP、SORT、SORT2指令	只读
-	-	-
M8040	SFC，禁止SFC状态转移标志位	读写
-	-	-
M8161	OFF-16位模式，ON-8位模式； ASCII / HEX / CCD / LRC / CRC / RS的位处理模式	读写
M8163	BINDA指令输出字符切换标志，保持还是切换为0000h	读写
M8165	SORT2指令降序排序使能标志	读写
M8168	SMOV指令数据格式设置，OFF-BCD模式，ON-HEX模式	读写
M8333	BKCMP指令矩阵比较结果全为1标志	只读

其它未定义的M8000之后的软元件，不能在程序中使用。

特殊软元件	功能描述	访问权限
D8066	用户程序和指令的严重错误（触发式，不复位）	只读
D8067	用户程序和指令的一般错误（触发式，不复位）	只读

访问权限说明

- 只读：PLC控制输出，用户只读，用户写入会被覆盖；
- 读写：作为PLC控制的输入，用户可读可写。

1.3 软元件

1.3.1 位软元件

H5U系列PLC编程支持位软元件，位软元件具体类型、范围、点数和相关说明如下表所示：

类型	范围	点数	数据类型	描述
X	X0~X1777	1024点， 8进制编码	BOOL	输入
Y	Y0~Y1777	1024点， 8进制编码	BOOL	输出
M	M0-M7999	8000点	BOOL	M0-M999掉电不保存，M1000及之后掉电保存；
S	S0-S4095	4096点	BOOL	S0-S999掉电不保存，S1000及之后掉电保存；
B	B0-B32767	32768点	BOOL	B0-B999掉电不保存，B1000及之后掉电保存；

说明

- 位软元件作为BOOL类型使用，可应用于指令BOOL类型输入或输出参数中。
- X和Y分别作为输入和输出元件使用。
- 作为普通编程使用的M元件仅支持最大到M7999，部分M8000后软元件作为特殊功能可在程序中直接编程使用，M8000特殊功能详见特殊软元件。

1.3.2 字软元件

H5U系列PLC编程支持字软元件，字软元件具体类型、范围、点数和相关说明如下表所示：

类型	范围	点数	数据类型	描述
D	D0-D7999	8000点	BOOL/ INT/ DINT/ REAL	D0-D999掉电不保存，D1000及之后掉电保存；
R	R0-R32767	32768点	BOOL/ INT/ DINT/ REAL	R0-R999掉电不保存，R1000及之后掉电保存；
W	W0-W32767	32768点	BOOL/ INT/ DINT/ REAL	W0-W999掉电不保存，W1000及之后掉电保存；

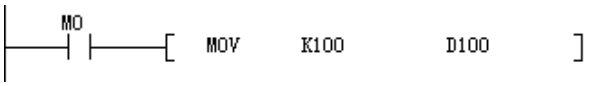
- 掉电保持范围不可更改。
- 字软元件可作为整数或浮点数使用，软元件本身不具有数据类型属性，根据指令的参数属性，将元件解释为整数或浮点数。
- 字软元件作为整数使用时，根据指令参数，作为16位或32位数据使用。作为16位数据使用时，占用1个软元件；作为32位数据使用时，占用2个软元件。
- 字软元件作为浮点数使用时，占用2个软元件。

举例

示例

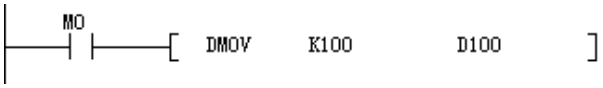
1. 字软元件作为16位整数使用

使用16位赋值指令，将值100赋给字软元件D100，占用D100。



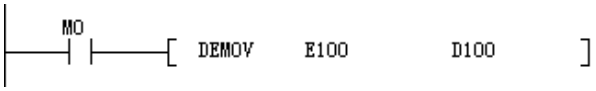
2. 字软元件作为32位整数使用

使用32位赋值指令，将值100赋给字软元件D100，占用D100（低位）和D101（高位）。



3. 字软元件作为浮点数使用

使用浮点数指令，将值100赋给字软元件D100，占用D100和D101。



1.3.3 特殊软元件

H5U系列PLC编程支持特殊软元件，特殊软元件功能、范围和相关说明如下表所示：

类型	功能	范围	点数	描述
SBR	子程序标号	SBR0–SBR1023	1024	CALL指令用，调用SBR子程序，子程序属性可以被设置为普通子程序、加密子程序；共同占用系统程序区容量。
L	跳转标签	L0-L1023	1024点	与CJ指令、LBL指令配套使用
I	外部中断	—	4	中断标号，X端口上升沿、下降沿、上升下降沿
	定时中断	—	4	定时时间ms
	比较中断	—	16	受限于内部编码器轴（高速计数器）数量
K	10进制	K-32,768 ~ K32,767 (16 位)，K-2,147,483,648 ~ K2,147,483,647 (32 位)	-	-
H	16进制	H0000 ~ HFFFF (16 位)，H00000000 ~ HFFFFFFFF (32 位)	-	-
E	浮点数、实数	$-3.402823e^{+38}$ ~ $-1.175495e^{-38}$，0，$+1.175495e^{-38}$ ~ $+3.402823e^{+38}$	-	单精度浮点数最多7位十进制有效数字，超出部分会自动四舍五入
字符	字符、字符串	-	-	字符、字符串，作为指令参数

举例

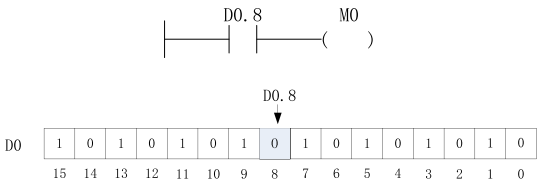
例如，单精度浮点数最多7位十进制有效数字，如果将9位二进制浮点数1234567.89传输到目标位置D0，但D0实际值为1234567.9，会丢失精度。

1.3.4 字元件的位操作

字元件可以通过(.)方式对元件的位操作，如编程写入D0.8表示对D0字元件的第8位操作。

举例

示例：



字元件的位从第0位开始数：D0的第8位为0时，输出M0为OFF；D0的第8位为1时，输出M0为ON。

1.4 变量

1.4.1 自定义变量

在H5U的编程体系，除了编程时直接使用直接地址，如X、Y、M、D、R等元件进行编程，也可以在没有具体的存储地址的情况下，以「变量」的方式进行编程，实现所需的控制逻辑，或应用对象的完整控制工艺，这样提高代码编写的便利性、可复用性。

表1–2 支持的自定义变量

类型	点数	数据类型	描述
Pointer	4096点（32位）	BOOL/ INT/ DINT/ REAL数组	指针变量； 掉电不保存；
BOOL	2MB（8位）	INT/ DINT/ REAL变量， INT/ DINT/ REAL数组， INT/ DINT/ REAL组合结构体	256KB掉电保存， 其他掉电不保存；
INT			
DINT			
REAL			

1.4.2 定义变量

H5U支持自定义变量，用户可以通过定义全局变量，在程序中直接使用变量名编程。定义全局变量名称时需要遵循以下规则：

1. 只能由“_、字母、数字、中文字符”组成且不能以“_、数字”开头；

2. 不能与"软元件形式、常数、标准数据类型、指令"重名；
3. 不能为“ARRAY、TRUE、FALSE、ON、OFF、NULL”等关键字。

变量数据类型

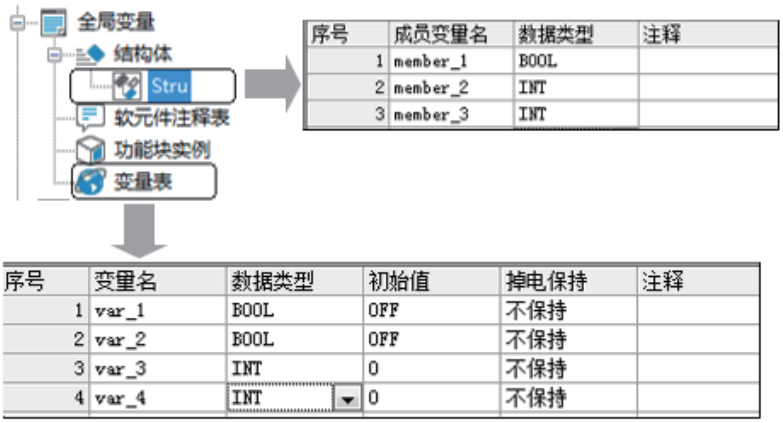
变量定义支持结构体和数组，变量数据类型支持如下：

表1–3 变量数据类型

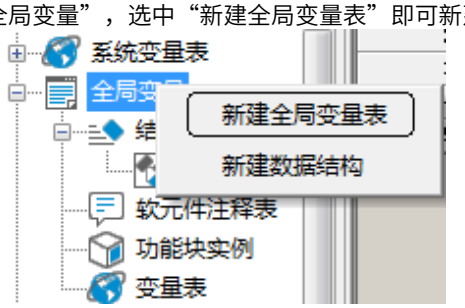
数据类型	描述
BOOL	布尔类型
INT	单字节整数类型
DINT	双字整数类型
REAL	实数类型

定义全局变量

AutoShop编程软件工程管理栏中的“全局变量”用于变量管理，可以实现变量的添加、删除和编辑操作。



1. 添加变量表和变量：右键点击“全局变量”，选中“新建全局变量表”即可新建全局变量表。



2. 双击变量表，进入变量编辑界面

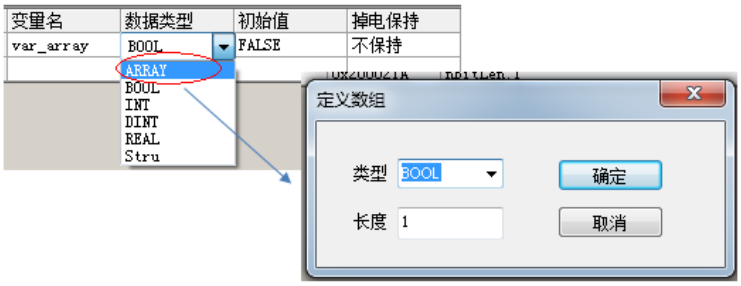
- 在变量表中，鼠标右键点击弹出的菜单中，可以插入或删除变量。
- 如果在变量表中的变量名一栏输入自定义变量名，编程时可以直接使用变量名编程。
- 数据类型可以选择BOOL、INT、DINT、REAL以及数组和结构体（需要事先定义好结构体）。选择数组作为数据类型时，在弹出发对话框中可设置数组变量类型和长度，选择事先定义好结构体时，即可定义结构体变量。
- 初始值一栏可以给变量定义初始值，数组和结构体可以单独定义每一个元素的初始值
- 掉电保持可以选择保持和非保持两种类型，初始值的设置只对非保持变量有效。

序号	变量名	数据类型	初始值	掉电保持	注释
1	Axis0	Axis		不支持	
2	Axis1	Axis		不支持	
3	bMovVel	BOOL	FALSE	不支持	
4	bMovStart	BOOL	FALSE	不支持	
5	bMovStop	BOOL	FALSE	支持	
6	bMcReset	BOOL	FALSE	不支持	
7	bMovAbsDone	BOOL	FALSE	不支持	
8	bAbsCmdAbort	BOOL	FALSE	不支持	
9	bAbsBusy	BOOL	FALSE	不支持	
10	bAbsErr	BOOL	FALSE	不支持	
11	iAbsErrID	INT	0	不支持	
12	bMovRelDone	BOOL	FALSE	不支持	
13	bRelCmdAbort	BOOL	FALSE	不支持	
14	bRelBusy	BOOL	FALSE	不支持	
15	bRelErr	BOOL	FALSE	不支持	
16	iRelErrID	INT	0	不支持	
17	bAutoRun	BOOL	FALSE	不支持	
18	var_1	BOOL[10]		不支持	
19	var_2	BOOL	FALSE	不支持	

1.4.3 定义数组

用户编程时，如果选择的数据类型为ARRAY，则可以定义数组。

1. 在弹出的对话框中选择数组变量的类型和长度，点击“确定”后即可进行数组定义。



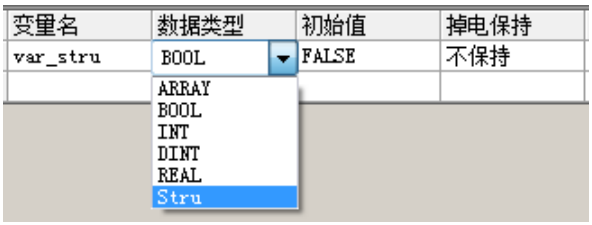
2. 点击数组变量的初始值一栏，进入数组变量的初始值设置界面：

1.4.4 定义结构体

变量定义中如果需要定义结构体变量，需要事先定义好结构体的数据结构。

右键点击“全局变量”下的“结构体”，选择新建数据结构，输入结构体名称，即定义好结构体，在变量表定义数据类型时，即可选择结构体名，将变量定义为结构体变量。

建立结构体和成员变量后，可以在变量定义的数据类型中选择结构体，定义结构体变量。



点击结构体变量的初始值一栏，进入结构体变量的初始值设置界面，可以设置结构体变量成员的初始值。

1.4.5 变量的使用方法

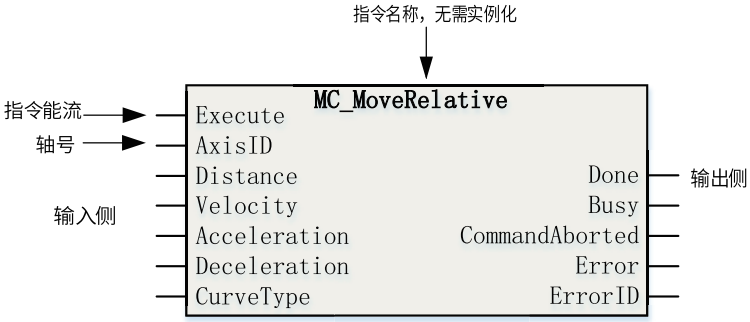
定义好变量后，在程序编程时，可以直接使用变量名进行编程，不需要再分配软元件。

- 直接变量编程操作。
- 使用数组变量时，编程用用 “[编号]” 表示数组元素，编号从0开始。
- 使用结构体变量时，编程用 “结构体变量名.成员变量” 表示结构体成员。

1.5 图形块指令

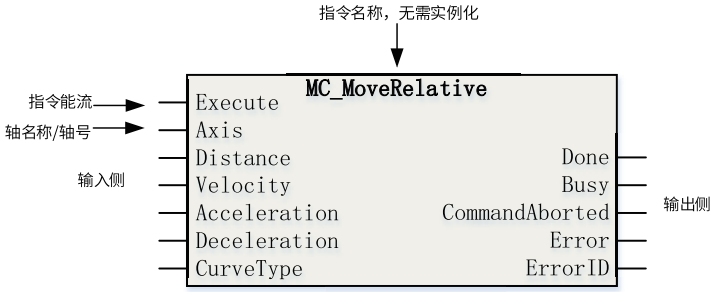
1.5.1 图形块指令的构成

H5U的部分指令支持图形块编程，图形块指令由指令名称、能流信号、输入侧和输出侧构成。以运动控制轴图形块指令为例，具体构成如下图所示：



指令中目标位置和目标速度等浮点数采用的是单精度浮点型，因此在PLC程序中处理时指令的值要符合单精度浮点型的范围和精度，即其数值范围为-3.4E38～3.4E38，精度位7位有效数字，如果某个数的有效数字位数超过7位，超出的部分会自动四舍五入。

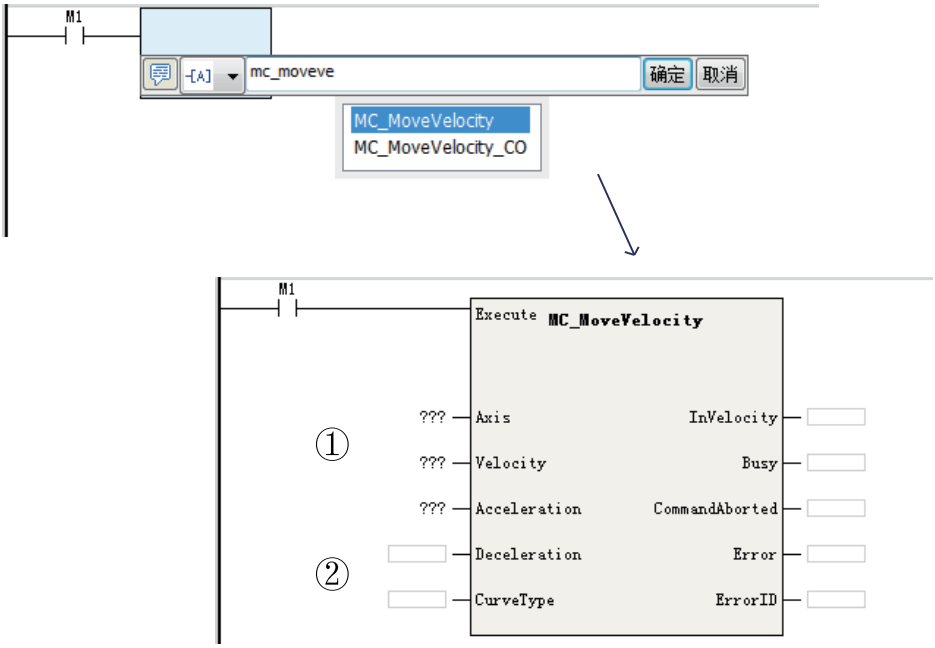
从Autoshop 4.0.0.0版本搭配单板软件 3.0.0.0开始，图形块的运动控制轴控指令（EterCAT\脉冲输出、脉冲输入）新增支持通过轴名称的方式访问，其中的AxisID改成Axis，更改后原有的通过轴号访问的方式仍然支持。



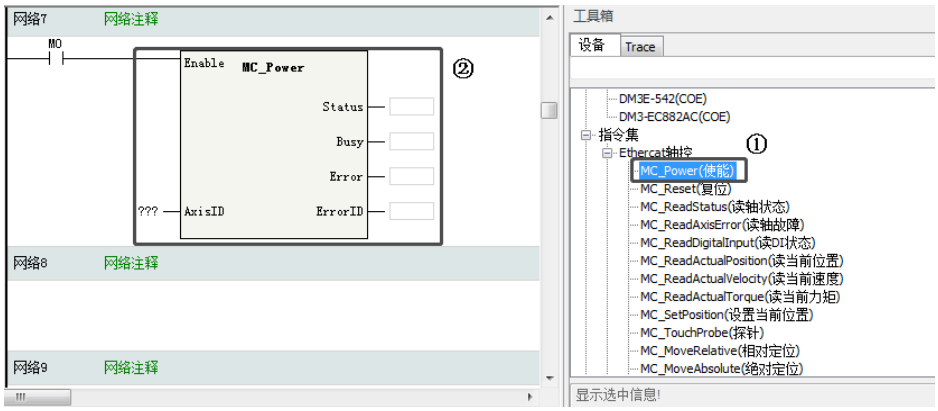
1.5.2 图形块指令编程

用户编程时，输入图形块指令名后，只需按回车键即可将图形块指令加入程序网络中，图形块支持直接编辑指令参数。

- 在梯形图编辑中，输入指令名或根据指令提示选择指令名后确定，梯形图网络中接入图形块指令。



- 在图形块指令中输入参数，可以完成图形块指令的编辑。
  指令中①显示“???”为必须使用参数，②可选择是否使用参数，如未使用参数，指令输入由指令中自动默认参数值，指令输出在程序中或监控调试时不能获取指令中状态。
- 工具箱指令集节点下的所有指令为图形块模式，编程时可以直接双击工具箱指令集节点下的指令，将指令添加到梯形图当前焦点位置。



①：双击指令添加到梯形图中；②：指令添加成功

1.6 功能与函数（FB/FC）

1.6.1 功能块 (FB)

功能块 (FB) 可以把程序中重复使用的部分抽象封装成一个通用程序块，在程序中可以被重复调用。在编程中使用封装的功能块，可提高程序的开发效率，减少编程错误，改善程序质量。
功能块在执行时能够产生一个或多个值，功能块保留有自己特殊的内部变量，控制器执行系统给功能块内部状态变量分配内存，这些内部变量构成自身的状态特征。对于相同参数的输入变量值，可能存在在不能的内部状态变量，会得到不同的计算结果。

功能块的基本使用步骤为：新建功能块 -> 功能块编程 -> 功能块实例化 -> 运行功能块 -> 封装功能块 -> 导入功能块。

新建功能块

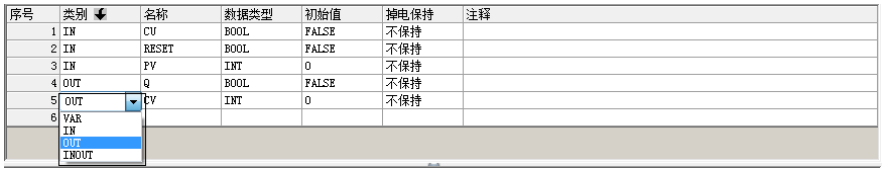
通过AutoShop软件，可以新建功能块。

在“编程”节点下右键单击“功能块（FB）”，选择新建，在弹出的对话框中输入功能块名，点击“确定”即完成功能块新建。

①:新建FB；②：命名功能块的名称；③新建完成的FB

功能块编程

功能块仅支持梯形图编程。在“功能块（FB）”节点下鼠标双击新建的功能块，进入功能块程序编辑界面。功能块程序编辑界面与普通程序编辑相比，多了一个输入输出和局部变量定义窗口。



①：输入输出和局部变量定义窗口

1. 类别：功能块变量的属性。

变量类别	类别说明	描述
IN	输入变量	由调用它的逻辑块提供参数，输入传递给逻辑块的指令
OUT	输出变量	向调用它的逻辑块提供参数，即从逻辑块输出结构数据
INOUT	输入输出变量	输入输出变量不仅可以传入被调用的逻辑块内，并且可以在被调用的逻辑块内部修改。
VAR	局部变量	仅在本逻辑块中有效，不能被外部访问。

2. 名称：变量的名称。
3. 数据类型

   变量数据类型支持BOOL、INT、DINT和REAL，可以定义数组变量和结构体。如使用结构体变量，需在全局变量的结构体中建立结构体成员。

4. 初始值

   设置变量执行开始时的初始数据。

5. 掉电保持

   掉电保持属性可将变量设置为保持或非保持属性。

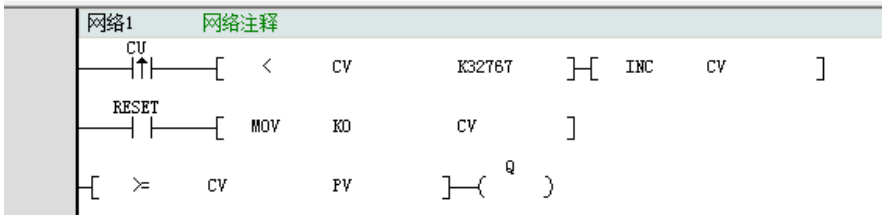
- 「非保持」，上电后变量恢复为初始值中设定的值；
- 「保持」，如在系统参数中勾选“下载时，初始化掉电保持型变量”，程序下载时变量恢复为初始值中设定的值，否则保持上一次运行值。

功能块程序使用梯形图编程，在功能块程序里面，可以调用函数（FC）或功能块（FB），最大支持8级嵌套调用。

功能块程序除使用变量外，可将H5U支持的软元件作为全局变量使用，如M8000。

示例：用FB封装增计数

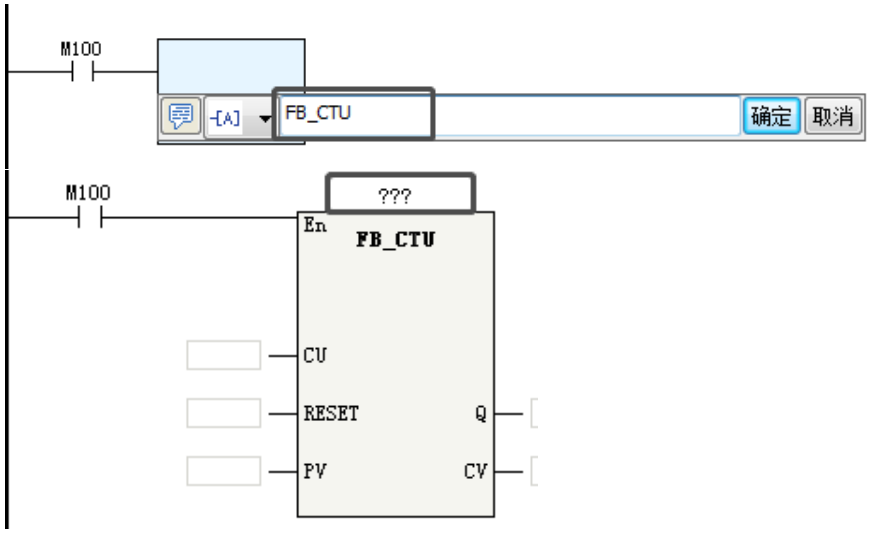
序号	类别	名称	数据类型	初始值	掉电保持	注释
1	IN	CU	BOOL	FALSE	不保持	
2	IN	RESET	BOOL	FALSE	不保持	
3	IN	PV	INT	0	不保持	
4	OUT	Q	BOOL	FALSE	不保持	
5	OUT	CV	INT	0	不保持	



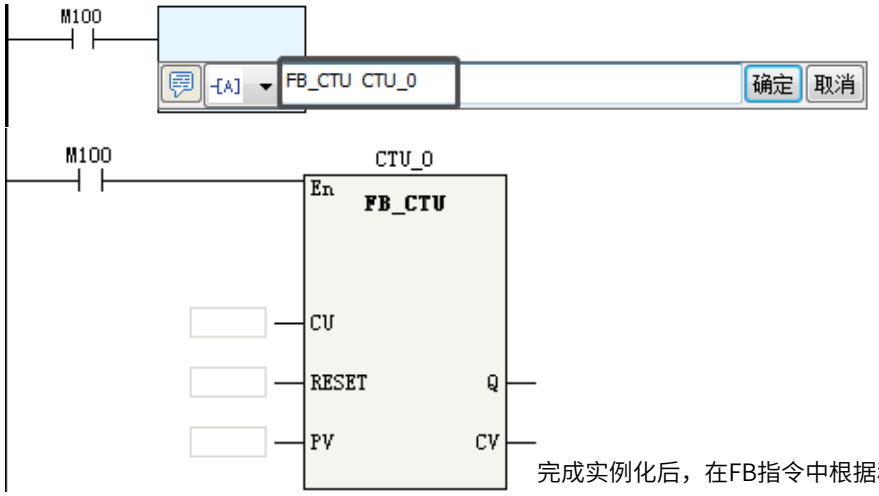
功能块实例化调用

编写好FB程序后，在应用程序中使用，需要对功能块实例化调用。

- 方法一：在梯形图应用程序中，直接输入FB名称，在功能块指令顶部的“???”中输入实例化名，完成功能块实例化。



- 方法二：在梯形图应用程序中，直接输入FB名+实例化名，点击确定后完成功能块实例化。

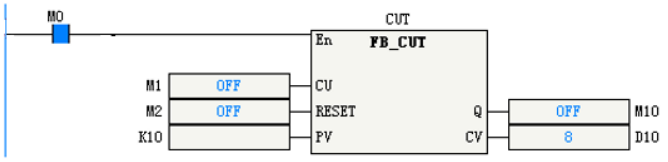


完成实例化后，在FB指令中根据程序需要编辑指令参数，完成功能块的实例化调用。

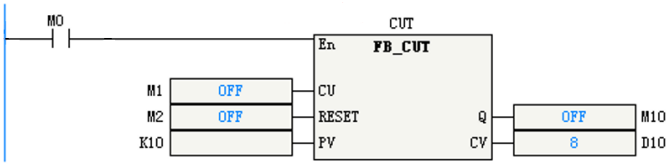
- 方法三：在工具箱的FB节点下，双击FB指令，可以将FB指令添加到梯形图选中的位置，添加后再图形块指令中输入实例化名称完成实例化定义。

运行功能块

功能块实例化后，功能块的En连接梯形图网络。En网络能流有效（ON）时，功能块程序被执行，功能块的输出根据输入状态和内部变量状态刷新变化。En网络能流有效（OFF）时，功能块程序不执行，功能块输出不刷新。



计数器功能块CUT能流条件为ON，功能块执行，输入条件CU上升沿变化时，输出CV加1。



计数器功能块CUT能流条件为OFF，功能块不执行，输入条件CU上升沿变化时，输出CV不刷新。

说明

功能块En能流无效，仅代表功能块不执行，表示功能块程序不扫描刷新。因为功能块可以包含持续运行的指令，比如轴相对定位运动，在相对运动指令执行中未达目标位置时，即使指令不扫描，也可以继续运动。因此，要停止对功能块内部控制的动作，需要通过功能块逻辑实现，不能通过功能块En能流控制。

封装功能块

编辑调试好的功能块可以封装成库。封装成库的功能块，可以通过AutoShop的库管理，实现在不同程序中的复用。

①:导出库；

②：选择需要封装的功能块；

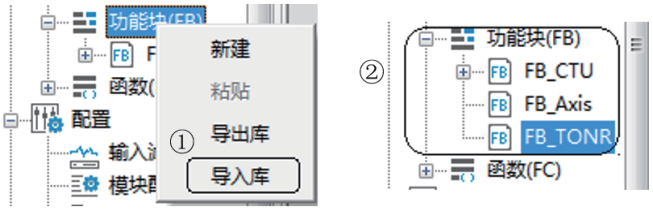
③：设置版本号；

④：选择是否源码可见；如选择源码可见，在工程中导入后，可打开调试或修改功能块程序。如不选择源码可见，则在库导入后，在工程中试无法查看修改功能块程序的，只能调用。

导入功能块

功能块作为库导出后，可以通过导入的方式，在其他程序中调用。可以通过两种方法导入功能块库。

- 方法一：在工程管理的功能块节点下右击选择导入库，作为工程导入使用。
  这种方法仅对源码可见的功能块进行导入，导入后的功能块程序可以双击打开，可以编辑调试功能块程序。这种方法导入的功能块库随工程管理，新建工程后如需调用这些功能块，需要重新导入。



- 方法二：工具箱的库节点下右击选择导入库，作为库导入使用。
  这种方式可对源码可见或源码不可见方式导入，导入后作为用户自定义库管理，新建工程后这些库里面的功能块都可以直接使用，不需要重新导入。在工具箱导入的功能块库，双击后作为指令可直接添加到梯形图程序中，如需要查看或修改源码可见的功能块程序，需要在工程管理中导入。

1.6.2 函数 (FC)

函数（FC）是独立封装的程序块，程序块可以定义输入/输出类型参数，可以定义非静态内部变量，即使用相同的输入参数调用某一函数时，得到的输出结果是相同的。函数的重要特点是他的内部变量是静态的，没有内部状态存储，相同的输入参数能得到相同的输出，这是函数（FC）与功能块（FB）之间的主要区别。
函数（FC）作为基本算法单元，常用于各种数学运算函数，比如sin(x)、sqrt(x)等就是典型的函数类型。

函数的基本使用步骤为：新建函数 -> 函数编程 -> 调用函数 -> 运行函数 -> 封装函数。

新建函数

在“编程”节点下右键单击“函数（FC）”，选择新建，在弹出的对话框中输入函数名，点击“确定”即完成函数新建。

函数编程

函数仅支持梯形图编程。在“函数（FC）”节点下鼠标双击新建的函数，进入函数程序编辑界面。函数程序编辑界面与功能块类似，和普通程序编辑相比，多了一个输入输出和局部变量定义窗口。

在输入输出和局部变量定义窗口，可以定义功能块的输入（IN）、输出（OUT）、输入输出（INTOUT）和局部变量（VAR）。变量数据类型支持BOOL、INT、DINT和REAL，可以定义数组变量和结构体。如使用结构体变量，需在全局变量的结构体中建立结构体成员。

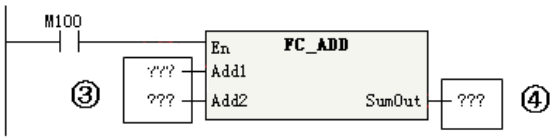
- 与功能块的变量相比，函数变量不能定义初始值，且所有局部变量都是非保持型的。
- 函数程序使用梯形图编程，在函数程序里面，可以调用函数（FC）。函数本身可以被其它函数、功能块、程序调用。
- 函数程序除使用变量外，可将M8000作为常ON变量使用。
- 函数程序中，不能使用和状态相关或多周期执行的指令，如LDP、MC_Power等指令。

示例：用FB封装加法函数

调用函数

编写好FC程序后，可以在应用程序中使用，也可以直接调用使用。

- 方法一：在梯形图应用程序中，直接输入FC名称，回车确定后在图形块指令中编辑输入输出参数。

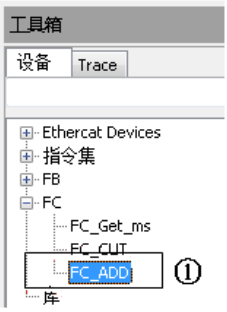


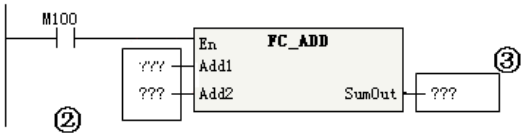
①：输入调用的FC名称；

②：确定；

③/④：添加输入输出变量；

- 方法二：新建FC程序后，在工具箱的FC节点下会生成相应的指令，直接双击工具箱FC节点中的FC指令，可以将FC指令添加到梯形图选中的位置。





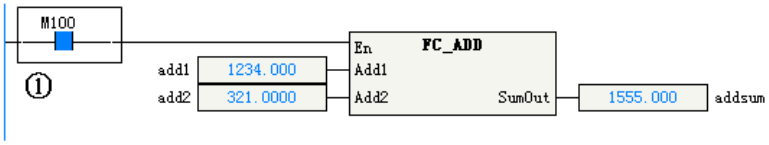
①：双击添加；

②：添加输入参数；

③：添加输出参数；

运行函数

函数调用后，函数的En连接梯形图网络。En网络能流有效（ON）时，函数程序被执行，函数的输出根据输入状态运算刷新输出结果。En网络能流有效（OFF）时，函数程序不执行，功能块输出不刷新。



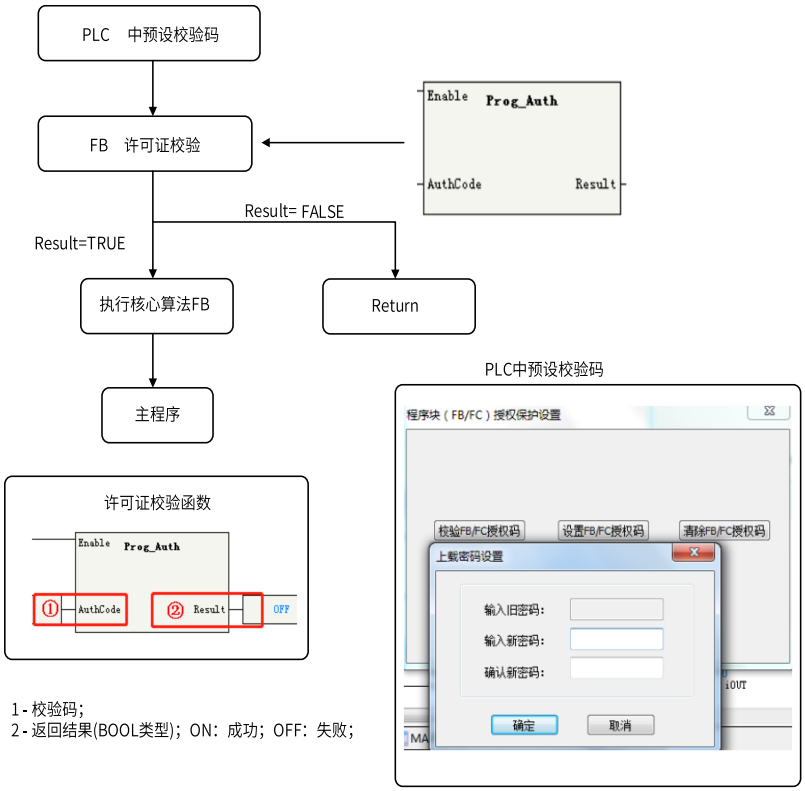
①：ON时函数执行

封装函数

函数的封装步骤与功能块类似，请参考“封装功能块”章节。

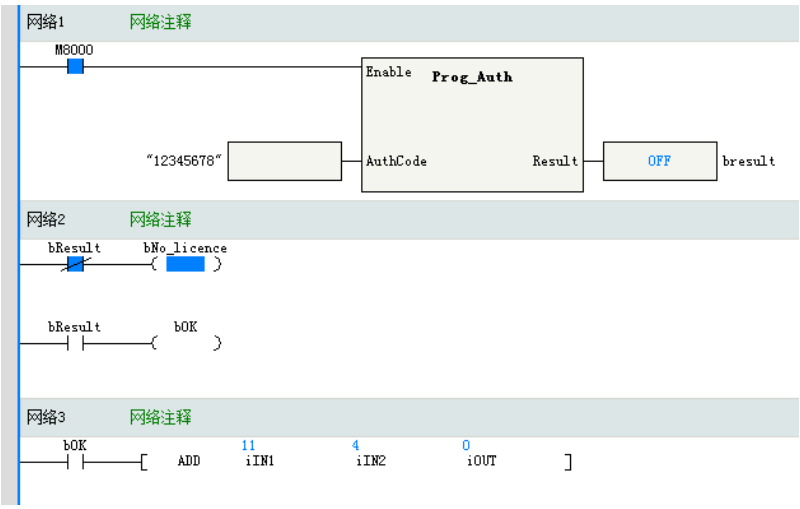
1.6.3 授权功能块

H5U支持Prog_Auth函数，将核心的算法功能块通过此函数输出控制，编译生成库文件，只有授权的PLC通过校验后才能使用此库文件，保护设备厂商知识产权。



示例：

由于通过函数Prog_Auth获取的校验码和PLC中预设校验码不一致，返回值=OFF，所以下列程序ADD指令没有执行。



2 指令速查表

2.1 指令速查表

H5U系列支持的全部指令汇总在指令速查表中，并按相应功能类别分类。

指令类别	名称	功能描述
触点指令	LD	加载常开触点
	LDI	加载常闭触点
	AND	串联常开触点
	ANI	串联常闭触点
	OR	并联常开触点
	ORI	并联常闭触点
	LDP	取脉冲上升沿
	LDF	取脉冲下降沿
	ANDP	与脉冲上升沿检测串行连接
	ANDF	与脉冲(F)下降沿检测串行连接
	ORP	或脉冲上升沿检测并行连接
	ORF	或脉冲(F)下降沿检测并行连接
	MEP	运算结果上升沿脉冲化
	MEF	运算结果下降沿脉冲化
输出控制指令	OUT	驱动线圈
	SET	置位动作保存线圈指令
	RST	接点或缓存器清除
	ZSET	全部数据置位
	ZRST	全部数据复位
	PLS	脉冲上升沿检测线圈指令
	PLF	脉冲(F)下降沿检测线圈指令
	ALT	交替输出
能流控制指令	ANB	串联回路方块
	ORB	并联回路方块
	MPS	存入堆栈
	MRD	读出堆栈
	MPP	读出堆栈
	INV	运算结果取反
	NOP	无动作

指令类别	名称	功能描述
流程控制指令	ICJ	条件跳转指令
	ILBL	标号指令
	CALL	子程序调用指令
	SRET	子程序返回指令
	ISSRET	子程序待条件返回
	IRET	中断返回
	EI	中断许可
	DI	中断禁止
	WDT	监视定时器复位指令
	FOR	循环范围开始指令
	NEXT	循环范围结束指令
SFC指令	ISTL	程序跳至副母线
	IRET	程序返回主母线
	IOUTSTL	输出程序跳至副母线
	SETSTL	置位程序跳至副母线
	RSTSTL	清除程序跳至副母线

指令类别	名称	功能描述
触点比较	LD=	LD触点比较等于
	LD>	LD触点比较大于
	LD<	LD触点比较小于
	LD<>	LD触点比较不等于
	LD>=	LD触点比较大于等于
	LD<=	LD触点比较小于等于
	AND=	AND触点比较等于
	AND>	AND触点比较大于
	AND<	AND触点比较小于
	AND<>	AND触点比较不等于
	AND>=	AND触点比较大于等于
	AND<=	AND触点比较小于等于
	OR=	OR触点比较等于
	OR>	OR触点比较大于
	OR<	OR触点比较小于
	OR<>	OR触点比较不等于
	OR>=	OR触点比较大于等于
	OR<=	OR触点比较小于等于
	LD&	LD逻辑与运算
	LD|	LD逻辑或运算
	LD^	LD逻辑异或运算
	AND&	AND逻辑与运算
	AND|	AND逻辑或运算
	AND^	AND逻辑异或运算
	OR&	OR逻辑与运算
	OR|	OR逻辑或运算
	OR^	OR逻辑异或运算
	FLDD >	浮点数>比较的状态触点S1 ＞ S2时导通
	FLDD >=	浮点数>=比较的状态触点S1 ≧ S2时导通
	FLDD <	浮点数<比较的状态触点S1 ＜ S2时导通
	FLDD <=	浮点数<=比较的状态触点S1 ≦ S2时导通
	FLDD =	浮点数=比较的状态触点S1 ＝ S2时导通
	FLDD <>	浮点数<>比较的状态触点S1 ≠ S2时导通
	FAND>	浮点数>比较的与状态触点S1 ＞ S2时导通
	FAND>=	浮点数>=比较的与状态触点S1 ≧ S2时导通
	FAND<	浮点数<比较的与状态触点S1 ＜ S2时导通
	FAND<=	浮点数<=比较的与状态触点S1 ≦ S2时导通
	FAND=	浮点数=比较的与状态触点S1 ＝ S2时导通
	FAND<>	浮点数<>比较的与状态触点S1 ≠ S2时导通

指令类别	名称	功能描述
触点比较	FOR>	浮点数>比较的或状态触点S1 ＞ S2时导通
	FOR>=	浮点数>=比较的或状态触点S1 ≧ S2时导通
	FOR<	浮点数<比较的或状态触点S1 ＜ S2时导通
	FOR<=	浮点数<=比较的或状态触点S1 ≦ S2时导通
	FOR=	浮点数=比较的或状态触点S1 ＝ S2时导通
	FOR<>	浮点数<>比较的或状态触点S1 ≠ S2时导通
	LDZ>	绝对值>比较的状态触点|S1 - S2|＞| S3 |时导通
	LDZ>=	绝对值>=比较的状态触点|S1 - S2|≧| S3 |时导通
	LDZ<	绝对值<比较的状态触点|S1 - S2|＜| S3 |时导通
	LDZ<=	绝对值<=比较的状态触点|S1 - S2|≦| S3 |时导通
	LDZ=	绝对值=比较的状态触点|S1 - S2|＝| S3 |时导通
	LDZ<>	绝对值<>比较的状态触点|S1 - S2|≠| S3 |时导通
	ANDZ>	绝对值>比较的与状态触点|S1 - S2|＞| S3 |时导通
	ANDZ>=	绝对值>=比较的与状态触点|S1 - S2|≧| S3 |时导通
	ANDZ<	绝对值<比较的与状态触点|S1 - S2|＜| S3 |时导通
	ANDZ<=	绝对值<=比较的与状态触点|S1 - S2|≦| S3 |时导通
	ANDZ=	绝对值=比较的与状态触点|S1 - S2|＝| S3 |时导通
	ANDZ<>	绝对值<>比较的与状态触点|S1 - S2|≠| S3 |时导通
	ORZ>	绝对值>比较的或状态触点|S1 - S2|＞| S3 |时导通
	ORZ>=	绝对值>=比较的或状态触点|S1 - S2|≧| S3 |时导通
	ORZ<	绝对值<比较的或状态触点|S1 - S2|＜| S3 |时导通
	ORZ<=	绝对值<=比较的或状态触点|S1 - S2|≦| S3 |时导通
	ORZ=	绝对值=比较的或状态触点|S1 - S2|＝| S3 |时导通
	ORZ<>	绝对值<>比较的或状态触点|S1 - S2|≠| S3 |时导通
四则运算	ADD	二进制数据加法
	SUB	二进制数据减法
	MUL	二进制数据乘法
	DIV	二进制数据除法
	MOD	二进制除法求余
	EADD	二进制浮点加法
	ESUB	二进制浮点减法
	EMUL	二进制浮点乘法
	EDIV	二进制浮点除法
	INC	二进制数据加一
	DEC	二进制数据减一
数据逻辑运算	WAND	二进制数据逻辑与
	WOR	二进制数据逻辑或
	WXOR	二进制数据逻辑异或
	NEG	二进制数据求补
	ENEG	二进制浮点数符号取反

指令类别	名称	功能描述
字的位运算	BLD	字或双字位数据触点指令
	BLDI	字或双字位数据反触点指令
	BAND	字或双字位数据与触点指令
	BANDI	字或双字位数据与非触点指令
	BOR	字或双字位数据或触点指令
	BORI	字或双字位数据或非触点指令
	BOUT	字或双字位数据输出指令
	BSET	字或双字位数据输出指令
	BRST	字或双字位数据输出指令
三角函数	SIN	浮点SIN运算指令
	COS	浮点COS运算指令指令
	TAN	浮点TAN运算指令
	ASIN	二进制浮点数ARCSIN运算
	ACOS	二进制浮点数ARCCOS运算
	ATAN	二进制浮点数ARCTAN运算
	RAD	二进制浮点数角度→弧度的转换
	DEG	二进制浮点数弧度→角度的转换
	SINH	二进制浮点数SINH运算
	COSH	二进制浮点数COSH运算
	TANH	二进制浮点数TANH运算
表格运算	WSUM	算出数据合计值
	MEAN	平均值计算
	LIMIT	上下限限位控制
	BZAND	死区控制
	ZONE	区域控制
	SCL	定坐标(不同点坐标数据)
	SCL2	定坐标2(X/Y坐标数据)
指数运算	EXP	二进制浮点数指数运算
	LOGE	二进制浮点数自然对数运算
	LOG	二进制浮点数以10为低的对数运算
	ESQR	二进制浮点开方运算
	SQR	二进制数据开方运算
	POW	浮点数权值指令

指令类别	名称	功能描述
数据转换	INT	二进制浮点→BIN整数变换
	BCD	二进制数据转换BCD数据
	BIN	BCD数据转换二进制数据
	FLT	二进制数据→二进制浮点数转换
	EBCD	二进制浮点→十进制浮点转换
	EBIN	十进制浮点→二进制浮点转换
	DABIN	10进制ASCII→BIN的转换
	BINDA	BIN→10进制ASCII的转换
	WTOB	字节单位的数据分离
	BITW	位元件赋值给字元件
	BTOW	字节单位的数据结合
	WBIT	字元件赋值给位元件
	WTODW	16位字元件转换成32位字元件
	DWTOW	32位字元件转换成16位字元件
	UNI	16位数据的4位结合
	DIS	16位数据的4位分离
	ASCI	HEX→ASCII转换
	HEX	ASCII→HEX转换
数据传送	MOV	赋值传送
	EMOV	二进制浮点传送
	SMOV	移位传送
	BMOV	数据成批传送
	FMOV	数据一对多传送
	CML	数据取反传送
	CMP	数据比较
	ECMP	浮点比较指令
	ZCP	区域比较
	EZCP	浮点区域比较指令
	SORT	数据排序
	SORT2	数据排序2
	SER	数据查找
	FDEL	数据表的数据删除
	FINS	数据表的数据插入
	POP	后入数据的读取

指令类别	名称	功能描述
数据移位	ROR	循环右移
	ROL	循环左移
	RCR	带进位的循环右移
	RCL	带进位的循环左移
	SFTR	位右移
	SFTL	位左移
	WSFR	字右移
	WSFL	字左移
	SFWR	先进先出的数据写入
	SFRD	先进先出的数据读出
	SFR	16位数据n位右移(带进位)
	SFL	16位数据n位左移(带进位)
其他数据处理	SWAP	高低字节交换
	BON	ON位判断
	SUM	统计ON位总数
	RAND	产生有范围限制的随机数据
	XCH	数据交换
	ABS	绝对值指令
	EABS	浮点绝对值指令
	PID	PID运算指令
	CCD	校验码
	CRC	CRC校验码计算
	LRC	LRC校验码计算
	RAMP	斜坡指令
矩阵运算	BK+	数据块加法运算
	BK-	数据块减法运算
	MAND	矩阵与运算
	MOR	矩阵或运算
	MXOR	矩阵异或运算
	MXNR	矩阵同或运算
	MINV	矩阵反相运算
矩阵比较	BKCMP=	矩阵等于比较(S1＝S2)
	BKCMP>	矩阵大于比较(S1＞S2)
	BKCMP<	矩阵小于比较(S1＜S2)
	BKCMP<>	矩阵不等于比较(S1≠S2)
	BKCMP<=	矩阵小于等于比较(S1≤S2)
	BKCMP>=	矩阵大于等于比较(S1≥S2)

指令类别	名称	功能描述
字符串指令	STR	整数→字符串的转换
	STRMOV	字符串直接赋值指令
	VAL	字符串→整数的转换
	ESTR	二进制浮点数→字符串的转换
	EVAL	字符串→二进制浮点数的转换
	$ADD	字符串的组合
	LEN	检出字符串的长度
	INSTR	字符串的检索
	RIGHT	从字符串右侧开始取出
	LEFT	从字符串左侧开始取出
	MIDR	从字符串中任意取出
	MIDW	字符串中任意替换
	$MOV	字符串的传送
时钟指令	TCMP	时钟数据比较
	TZCP	时钟数据区间比较
	TADD	时钟数据加法运算
	TSUB	时钟数据减法运算
	HTOS	时，分，秒数据的秒转换
	STOH	秒对时，分，秒数据的转换
	TRD	时钟数据读取
	TWR	时钟数据写入
	HOUR	计时表
高速计数器	HC_Counter	高速计数器使能
	HC_Preset	高速计数器预置值
	HC_TouchProbe	HC_TouchProbe
	HC_Compare	高速计数器比较
	HC_ArrayCompare	高速计数器数组比较
	HC_StepCompare	高速计数器等间距离比较
定时器	TPR	脉冲定时器
	TONR	接通延时定时器
	TOFR	关断延时定时器
	TACR	时间累加定时器

指令类别	名称	功能描述
指针指令	PTGET	指针变量赋值指令
	PTINC	指针变量地址增1指令
	PTDEC	针变量地址减1指令
	PTADD	指针变量地址加法指令
	PTSUB	指针变量地址减法指令
	PTMOV	指针变量相互赋值指令
	PTLD>	指针变量触点比较指令
	PTLD>=	指针变量触点比较指令
	PTLD<=	指针变量触点比较指令
	PTLD=	指针变量触点比较指令
	PTLD<>	指针变量触点比较指令
	PTAND>	指针变量触点比较指令
	PTAND>=	指针变量触点比较指令
	PTAND<	指针变量触点比较指令
	PTAND<=	指针变量触点比较指令
	PTAND=	指针变量触点比较指令
	PTAND<>	指针变量触点比较指令
	PTOR>	指针变量触点比较指令
	PTOR>=	指针变量触点比较指令
	PTOR<	指针变量触点比较指令
	PTOR<=	指针变量触点比较指令
	PTOR=	指针变量触点比较指令
	PTOR<>	指针变量触点比较指令
FB/FC指令	lPROG_AUTH	程序块（FB/FC等）授权校验指令
通讯协议指令	l SerialSR	串口自由协议发送接收
	TCP_Listen	TCP监听指令
	TCP_Accept	TCP接受连接请求指令
	TCP_Connect	TCP发起连接请求指令
	TCP_Close	CP关闭连接指令
	TCP_Send	CP发送数据指令
	TCP_Receive	TCP接收数据指令
	UDP_Bind	UDP套接字绑定指令
	UDP_Send	UDP发送数据指令
	UDP_Receive	UDP接收数据指令
	ECT_ReadParameter_CoE	读取从站的SDO参数
	ECT_WriteParameter_CoE	写入从站的SDO参数

指令类别	名称	功能描述
运动控制轴	MC_Power	启用/禁用轴
	MC_Reset	复位/确认错误
	MC_ReadStatus	读取状态
	MC_ReadAxisError	读取轴错误
	MC_ReadDigitalInput	读取数字量输入
	MC_ReadActualPosition	读取实际为准
	MC_ReadActualVelocity	读取实际速度
	MC_ReadActualTorque	读取实际转矩
	MC_SetPosition	设定位置
	MC_TouchProbe	探针指令
	MC_MoveRelative	相对定位
	MC_MoveAbsolute	绝对定位
	MC_MoveVelocity	以预定义速度运动
	MC_Jog	点动运动
	MC_ProfileTorque	轮廓力矩
	MC_Home	回原点
	MC_Stop	停止轴
	MC_Halt	暂停轴
	MC_MoveFeed	中断定长
	MC_MoveBuffer	多段位置定位
轴组控制	MC_MoveLinear	直线插补
	MC_MoveCircular	圆弧插补
	MC_GroupStop	停止轴组运行
	MC_GroupPause	暂停轴组运行
CANopen轴	MC_Power_CO	通讯控制伺服轴使能
	MC_Reset_CO	通讯控制伺服轴故障复位
	MC_ReadActualPosition_CO	通讯控制读取轴当前实际位置
	MC_ReadActualVelocity_CO	通讯控制读取轴当前实际速度
	MC_Halt_CO	通讯控制伺服轴终止运动
	MC_Stop_CO	通讯控制伺服轴停止
	MC_MoveAbsolute_CO	通讯控制轴绝对定位
	MC_MoveRelative_CO	通讯控制轴相对定位
	MC_MoveVelocity_CO	通讯控制轴速度运行模式
	MC_Jog_CO	通讯控制轴点动
	MC_Home_CO	通讯控制轴原点回归
	MC_WriteParameter_CO	通讯控制写入轴参数
	MC_ReadParameter_CO	通讯控制读取轴参数

3 指令详解

3.1 程序逻辑指令

3.1.1 触点指令

3.1.1.1 指令列表

触点指令涵盖以下指令条目：

指令类别	名称	功能
触点指令	LD	加载常开触点
	LDI	加载常闭触点
	AND	串联常开触点
	ANI	串联常闭触点
	OR	并联常开触点
	ORI	并联常闭触点
	LDP	取脉冲上升沿
	LDF	取脉冲下降沿
	ANDP	与脉冲上升沿检测串行连接
	ANDF	与脉冲(F)下降沿检测串行连接
	ORP	或脉冲上升沿检测并行连接
	ORF	或脉冲(F)下降沿检测并行连接
	MEP	运算结果上升沿脉冲化
	MEF	运算结果下降沿脉冲化

3.1.1.2 LD/LDI/LDP/LDF

LD — 加载常开触点

LDI — 加载常闭触点

LDP — 取脉冲上升沿

LDF — 取脉冲下降沿

16位指令	LD 连续执行			
32位指令	—			
16位指令	LDI 连续执行			
32位指令	—			
16位指令	LDP 连续执行			
32位指令	—			
16位指令	LDF 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	位元件	待判断能流状态的软元件或变量	-	BOOL

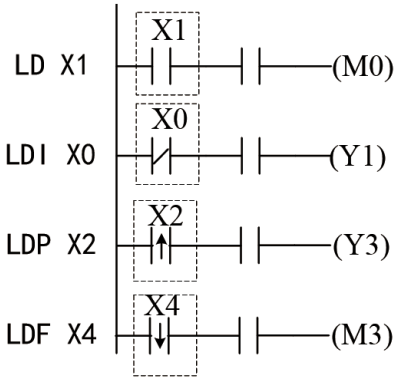
表3–1 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	√	-	-	-

功能和指令说明

LD/LDI/LDP/LDF指令用于左母线开始的接点，其中：

- LD/LDI指令分别是把A接点和B接点的当前能流状态保存，同时把取来的接点状态存入累计缓存器内。
- LDP指令用于取用接点信号的上升沿，若本次扫描中检测到对应信号的上升跳变，则触点有效，下一次扫描时，触点即变成无效。
- LDF指令用于取用接点信号的下降沿，若本次扫描中检测到对应信号的下降跳变，则触点有效，下一次扫描时，触点即变成无效。



3.1.1.3 AND/ANI/ANDP/ANDF

AND — 串联常开触点

ANI — 串联常闭触点

ANDP — 与脉冲上升沿检测串行连接

ANDF — 与脉冲(F)下降沿检测串行连接

16位指令	AND 连续执行			
32位指令	—			
16位指令	ANDI 连续执行			
32位指令	—			
16位指令	ANDP 连续执行			
32位指令	—			
16位指令	ANDF 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	位元件	待判断能流状态的软元件或变量	-	BOOL

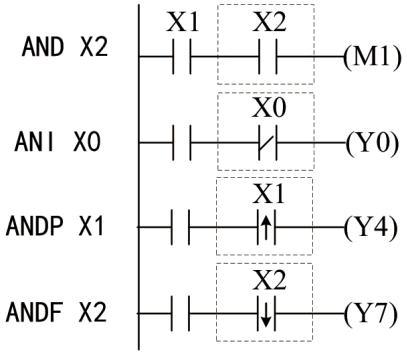
表3-2 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	√	-	-	-
S	√	√	√	-	-	√	-	-	-

功能和指令说明

AND/ANI/ANDP/ANDF指令用于串联接点的状态运算，其操作是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“与”（AND）的运算，并将结果存入累计缓存器内。

- AND/ANI指令分别是将A接点和/B接点的状态参与AND运算。
- ANDP指令是将接点的上升沿跳变状态参与AND运算。
- ANDF指令是将接点的下降沿跳变状态参与AND运算。



3.1.1.4 OR/ORI/ORP/ORF

OR — 并联常开触点

ORI — 并联常闭触点

ORP — 与脉冲上升沿检测并行连接

ORF — 与脉冲(F)下降沿检测并行连接

16位指令	OR 连续执行			
32位指令	—			
16位指令	ORI 连续执行			
32位指令	—			
16位指令	ORP 连续执行			
32位指令	—			
16位指令	ORF 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	位元件	待判断能流状态的软元件或变量	-	BOOL

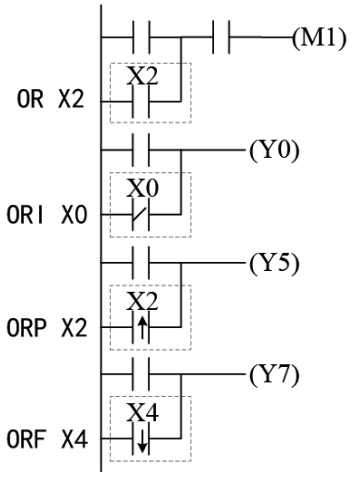
表3-3 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	√	-	-	-
S	√	√	√	-	-	√	-	-	-

功能和指令说明

OR/ORI指令用于并联接点的状态运算，其操作是先读取目前所指定接点的状态，再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累计缓存器内。

- OR/ORI指令分别是将A接点和/B接点的状态参与OR运算。
- ORP指令是将接点的上升沿跳变状态参与OR运算。
- ORF指令是将接点的下降沿跳变状态参与OR运算。



3.1.1.5 MEP/MEF

MEP — 运算结果上升沿脉冲化

MEF — 运算结果下降沿脉冲化

16位指令	MEP 连续执行			
32位指令	—			
16位指令	MEF 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	BOOL

功能和指令说明

MEP

在到MEP指令为止的运算结果，从OFF→ON时变为导通状态。

如果使用MEP指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理。

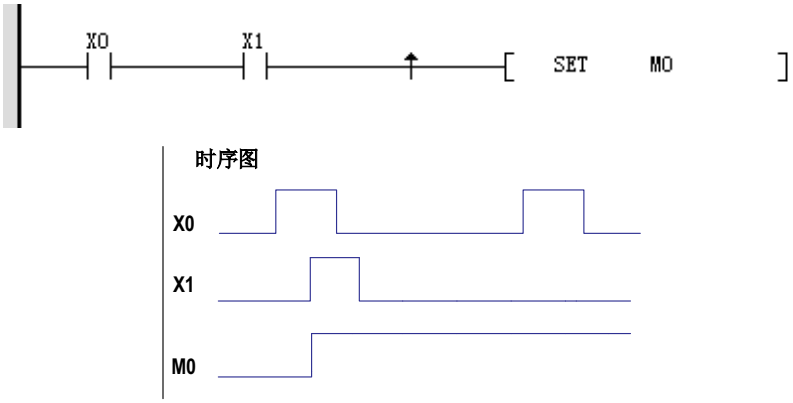
MEF

在到MEF指令为止的运算结果，从ON→OFF时变为导通状态。

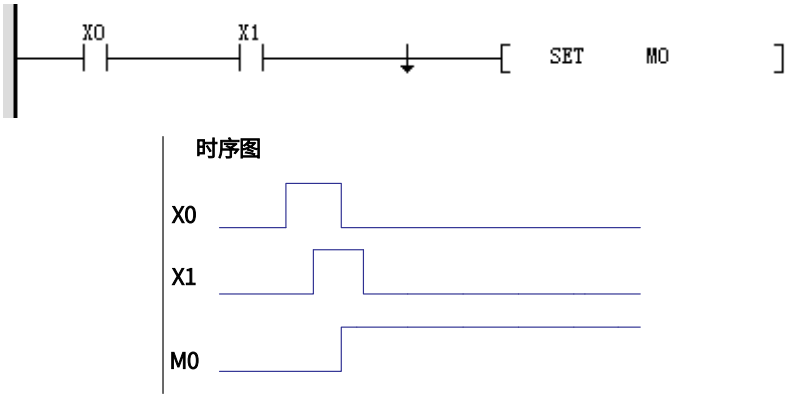
如果使用MEF指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理

指令示例

- MEP指令（运算结果上升沿为ON）



- MEF指令（运算结果下降沿为ON）



3.1.2 输出控制指令

3.1.2.1 指令列表

输出控制指令涵盖以下指令条目：

指令类别	名称	功能
输出控制指令	OUT	驱动线圈
	SET	置位动作保存线圈指令
	RST	接点或缓存器清除
	ZSET	全部数据置位
	ZRST	全部数据复位
	PLS	脉冲上升沿检测线圈指令
	PLF	脉冲(F)下降沿检测线圈指令
	ALT	交替输出

3.1.2.2 OUT

OUT — 驱动线圈

16位指令	OUT 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	输出元件	要输出的位元件或变量	-	BOOL

表3–4 软元件列表

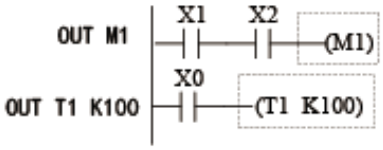
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	√	-	-	-

说明 [1]不支持X元件；S元件是单独的指令，请参见SFC指令。

功能和指令说明

将OUT 指令之前的逻辑运算结果输出至指定的元件。

指令示例



3.1.2.3 SET

SET — 置位动作保存线圈

16位指令	SET 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	输出元件	要输出的位元件或变量	-	BOOL

表3–5 软元件列表

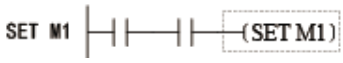
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	√	-	-	-

说明 [1]不支持X元件；S元件是单独的指令，请参见SFC指令。

功能和指令说明

当SET 指令被驱动，其指定的组件被设定为ON，且被设定的组件会维持ON，不管SET指令是否仍被驱动。可利用RST指令将该组件设为OFF。

指令示例



3.1.2.4 RST

RST — 接点或缓存器清除

16位指令	RST（位） 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	输出元件	要输出的位元件或变量	-	BOOL

表3–6 软元件列表

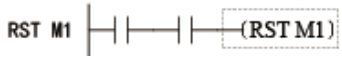
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	√	-	-	-

说明 [1]不支持X元件；S元件是单独的指令，请参见SFC指令。

功能和指令说明

当RST 指令被驱动，其指定的组件被设定为OFF，且被设定的组件会维持OFF，不管RST 指令是否仍被驱动。可利用SET 指令将该组件设为ON。

指令示例



3.1.2.5 ZSET

ZSET — 批量置位指令

16位指令	ZSET（位） 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D1	元件首地址	批量置位的元件或变量首地址	-	BOOL， 数组*（D2-D1+1）
D2	元件末地址	批量置位的元件或变量末地址	-	BOOL

表3-7 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D1	√[1]	√	√	-	-	-	-	-	-
D2	√[1]	√	√	-	-	-	-	-	-

说明 [1]不支持X元件。

功能和指令说明

将D1至D2区间的变量全部置1。D1和D2可指定为Y、M、S位变量。

其中要求：

D1和D2必须为同一类型的软元件；

编号D1应不大于D2，若两者相同时，仅置位指定的软元件；

指令示例

补充说明

位元件Y、M、S可使用SET指令来单独置位。

3.1.2.6 ZRST

ZRST — 全部数据复位

16位指令	ZRST 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D1	元件首地址	批量复位的元件或变量首地址	-	BOOL，INT,DINT 数组*（D2-D1+1）
D2	元件末地址	批量复位的元件或变量末地址	-	BOOL

表3-8 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D1	√[1]	√	√	√	√	-	-	-	-
D2	√[1]	√	√	√	√	-	-	-	-

说明 [1]不支持X元件。

功能和指令说明

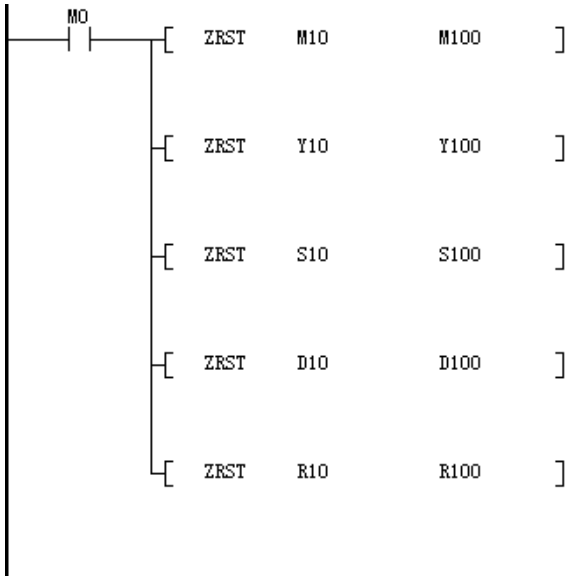
将D1至D2区间的变量全部清0。D1和D2可指定字变量，也可为Y、M、S位变量。

其中要求：

D1和D2必须为同一类型的软元件；

编号D1应不大于D2，若两者相同时，仅复位指定的软元件；

指令示例



补充说明

位元件Y、M、S可使用RST指令来单独复位，字元件可用MOV、DMOV等指令来单独复位。

3.1.2.7 PLS/PLF

PLS — 脉冲上升沿检测线圈指令

16位指令	PLS 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	输出元件	要输出的位元件或变量	-	BOOL

表3-9 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	-	-	-	-

说明 [1]不支持X元件。

PLF — 脉冲下降沿检测线圈指令

16位指令	PLF 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	输出元件	要输出的位元件或变量	-	BOOL

表3-10 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	-	-	-	-

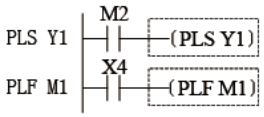
说明 [1]不支持X元件。

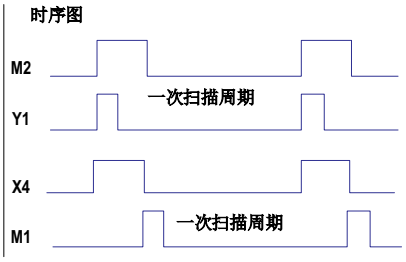
功能和指令说明

当PLS 指令被上升沿驱动时，其指定的元件被设定为ON状态，该ON状态仅持续1个扫描周期；

当PLF 指令被下降沿驱动时，其指定的元件被设定为ON状态，该ON状态仅持续1个扫描周期。

指令示例





3.1.2.8 ALT

ALT — 交替输出指令

当驱动条件成立时，对位元件D执行ON/OFF反转

16位指令	ALT 连续执行/ALTP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S/D	执行装置	位元件	-	BOOL

表3–11 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S/D	√[1]	√	√	-	-	-	-	-	-

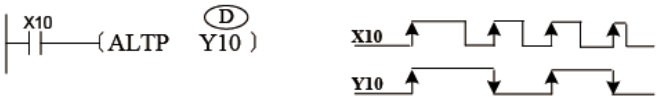
说明 [1]不支持X元件。

功能和指令说明

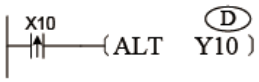
该指令的功能是能流有效时，将D元件的状态反转。其中D为位变量元件。

一般使用脉冲执行型ALTP指令。

指令示例



如下指令产生的动作与之相同



3.1.3 能流控制指令

3.1.3.1 指令列表

能流控制指令涵盖以下指令条目：

能流控制指令	ANB	串联回路方块
	ORB	并联回路方块
	MPS	存入堆栈
	MRD	读出堆栈
	MPP	读出堆栈
	INV	运算结果取反
	NOP	无动作

3.1.3.2 ANB/ORB

ANB — 串联回路方块

ORB — 并联回路方块

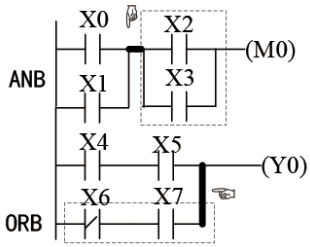
16位指令	ANB 连续执行			
32位指令	—			
16位指令	ORB 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

说明 Autoshop自动添加，不需用户输入。

功能和指令说明

ORB操作数是最近两次LD（或LDI/LDP/LDF）区间的计算能流。

ANB和ORB是将前一保存的逻辑结果与目前累计缓存器旳内容作“与”和“或”的运算。



3.1.3.3 MPS/MRD/MPP

MPS — 存入堆栈

MRD — 读出堆栈

MPP — 读出堆栈

16位指令	MPS 连续执行			
32位指令	—			
16位指令	MRD 连续执行			
32位指令	—			
16位指令	MPP 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

说明 Autoshop自动添加，不需用户输入。

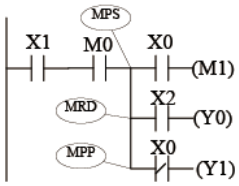
功能和指令说明

MPS：将目前累计缓存器的内容存入堆栈（堆栈指针加一）。

MRD：读取堆栈内容存入累计缓存器（堆栈指针不动）。

MPP：自堆栈取回前一保存的逻辑运算结果，存入累计缓存器（堆栈指针减一）。

指令示例




```
LD X0
OR X1
LD X2
OR X3
ANB
OUT M0
```

```
LD X4
AND X5
LDI  X6
AND  X7
ORB
OUT Y0
```

3.1.3.4 INV

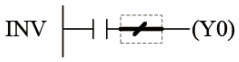
INV — 运算结果取反

16位指令	INV 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

将INV指令之前的逻辑运算结果反相后存入累计缓存器内。当INV指令之前能流为ON，经过INV后能流变为OFF；反之，变为ON。

指令示例



3.1.3.5 NOP

NOP — 无动作

16位指令	NOP 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

指令NOP在程序不做任何运算，因此执行后仍会保持原逻辑运算结果，没有实际操作，在AutoShop编译时，会自动将之删除，减少程序空间的浪费，加快运行速度。

3.2 流程控制指令

3.2.1 指令列表

流程控制指令涵盖以下指令条目：

指令类别	名称	功能
流程控制指令	CJ	条件跳转指令
	LBL	标号指令
	CALL	子程序调用指令
	SRET	子程序返回指令
	SSRET	子程序待条件返回
	IRET	中断返回
	EI	中断许可
	DI	中断禁止
	WDT	监视定时器复位指令
	FOR	循环范围开始指令
	NEXT	循环范围结束指令

3.2.2 CJ

CJ — 条件跳转指令

16位指令	CJ 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	目标标签	要跳转到的目标标签	-	-

表3–12 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	-	-	L[1]

说明 [1]仅支持L元件。

3.2.3 LBL

LBL — 标号指令

16位指令	LBL连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	标签号	标注当前标签号	-	-

表3-13 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	-	-	L[1]

说明 [1]仅支持L元件。

功能和指令说明

- 当能流有效时，程序自动从CJ指令的地址跳转至由L***指定的地址后继续执行，中间地址的程序指令被跳过，不予执行；
- 当能流无效时，程序继续往下执行，此时CJ指令不被执行。

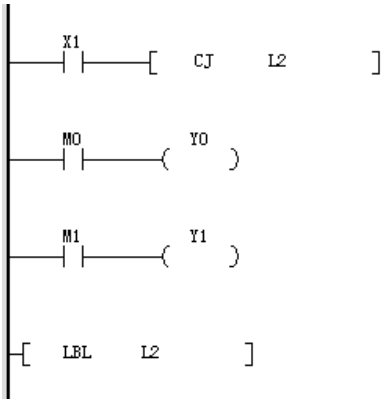
对L标签的要求如下：

- CJ指令必须与LBL指令配套使用，且目的标号必须位于当前程序块中，不能跨程序块跳转；
- L标签的定义地址在同一个程序块中不能有重复；
- 当使用者希望某一部份程序不需要执行时，或者想使用两个线圈输出时，为避免双线圈的出现。可使用此指令；
- CJ指令可重复指定同一标签L。

说明 跳转指令不能在子程序、中断子程序、FB、FC中使用。

指令示例

在AutoShop编程环境中，跳转指令用法如下：



3.2.4 CALL

CALL — 子程序调用

16位指令	CALL 连续执行
32位指令	—

操作数	名称	描述	范围	数据类型
S	子程序	要调用的目标子程序序号	-	-

表3–14 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	-	-	P[1]

说明

- [1]仅支持PL元件。
- Autoshop自动添加，不需用户输入。
- 子程序嵌套层次最多6层（含主程序层）。

功能和指令说明

当能流有效时，程序调用由P***指定的子程序。子程序执行完毕，会返回到该CALL（或CALLP）语句的下一指令，继续执行后续语句。

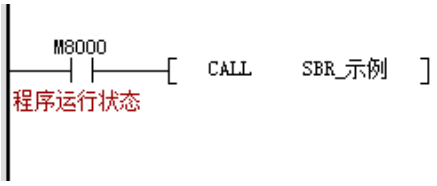
对P***地址指针的要求如下：

- 由P***开始的子程序，必须在主程序结束（FEND指令）之后的范围；
- 子程序必须以SRET语句结束；
- P***的子程序可被多处调用，也可被其他子程序调用，但嵌套层数不得超过 5 层；
- 在子程序内不得调用自身，防止死循环或程序运行超时。
- AutoShop编程环境中，子程序在单独窗口编写，因此没有FEND、SRET等指令的问题存在，且子程序名可任意修改（包括中文）。

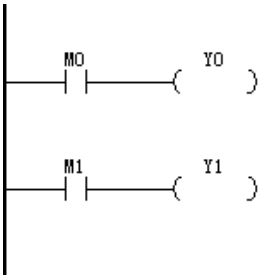
指令示例

在AutoShop编程环境中，CALL指令用法如下：

- 主程序



- 子程序：SBR_示例



3.2.5 SRET

SRET — 子程序返回

16位指令	SRET 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

说明

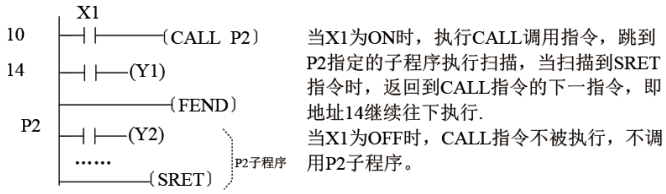
Autoshop自动添加，不需用户输入。

功能和指令说明

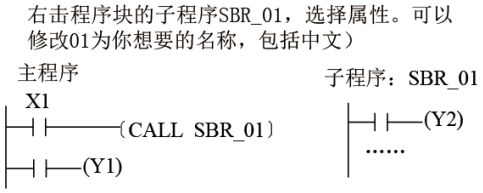
SRET指令位于子程序结束处，执行该指令后会退回调用该子程序的语句处，继续执行随后的程序。在AutoShop编程环境中，子程序最后无需编写SRET。

指令示例

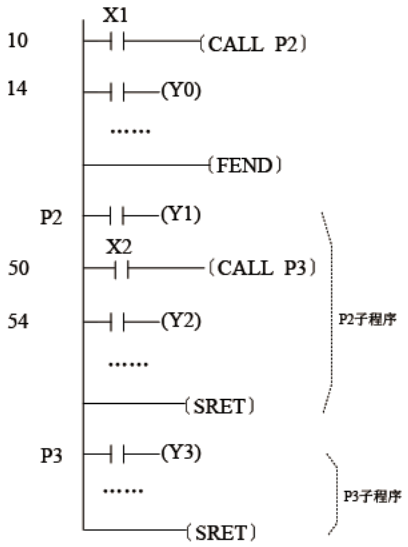
- 示例一



在AutoShop编程环境中“指令示例一”的格式如下：



- 示例二



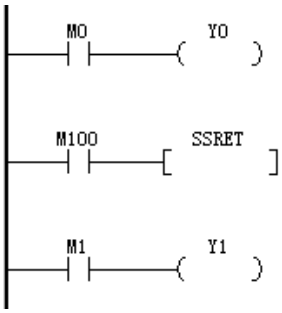
3.2.6 SSRET

SSRET — 子程序带条件返回

16位指令	SSRET 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

指令示例

如下图，当M100位ON时，直接返回执行主程序（该指令只能在子程序中执行）。



3.2.7 IRET

IRET — 中断返回

16位指令	IRET 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

说明 Autoshop自动添加，不需用户输入。

3.2.8 EI/DI

EI — 中断许可

DI — 中断禁止

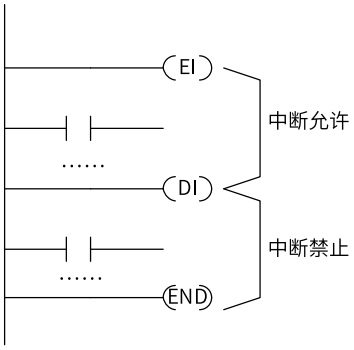
16位指令	EI 连续执行			
32位指令	—			
16位指令	DI 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

PLC程序开始运行时，默认为中断禁止状态；执行了EI语句后，中断功能允许；当中断为允许状态，执行了DI语句后，即进入中断禁止状态。在程序中如果没有中断插入禁止的区间时，可以不使用DI指令。

如需了解中断子程序的介绍和用法，请参见H5U系列可编程逻辑控制器编程与应用手册中的“中断”与“子程序”章节。

中断的种类：外部信号输入中断、高速计数比较中断、定时器中断。



3.2.9 WDT

WDT — 监视定时器复位指令

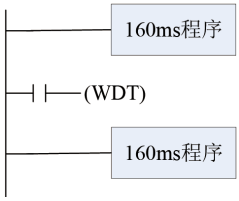
16位指令	WDT 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

PLC系统内有用于监视用户程序执行一次的时间是否超时的定时器，若超时即会停止用户程序的执行并报警，执行WDT指令即可将该监视定时器复位，让监视定时器重新开始计时，避免超时错误。

若用户程序所执行的操作过于复杂（例如过多的循环计算），执行时有可能出现运行超时错误，编程时若必要，可用WDT指令（例如在FOR~NEXT指令之间中插入该指令）；

指令示例



如果上述程序监视定时器设定是200ms，此程序扫描时间是320ms，直接运行会导致监视器超时。用WDT指令可将程序分割为2部分，使得每部分程序扫描时间都在200ms以下。

3.2.10 FOR/NEXT

FOR — 循环范围开始

16位指令	WDT 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

FOR指令用于一个循环的起始，同时指明循环执行的次数，必须与NEXT指令配套使用。其中：S为循环次数控制变量。

NEXT — 循环范围结束

16位指令	NEXT 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

功能和指令说明

NEXT指令用于指示循环区域的尾部。由FOR指令指定FOR~NEXT循环来回执行N次后跳出FOR~NEXT循环往下继续执行。

在FOR~NEXT指令的循环区间，可以嵌入另一个FOR~NEXT循环，但规定：从最外层的FOR~NEXT计算，最多可内嵌6层FOR~NEXT循环。运行时PLC会以各FOR~NEXT层对应解析执行。但需要注意当循环次数过多时，会使PLC扫描周期延长，可能造成逾时监视定时器动作而导致错误产生。可在FOR~NEXT指令之间使用WDT指令来改善。

说明 for next最多嵌套6层。

报错

出现下列情况时，会报错：

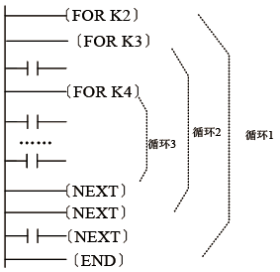
NEXT指令在FOR指令之前；

有FOR指令而无NEXT指令；

FOR指令与NEXT指令个数不一致等。

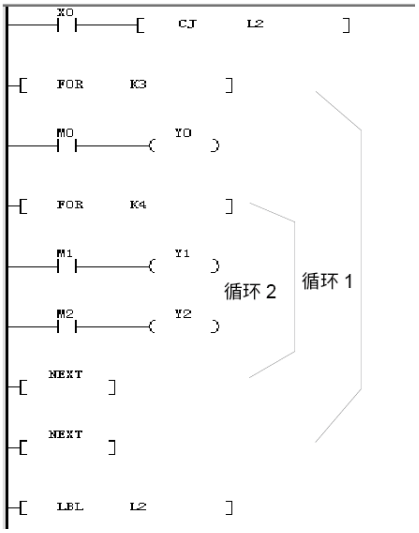
指令示例

- 示例一



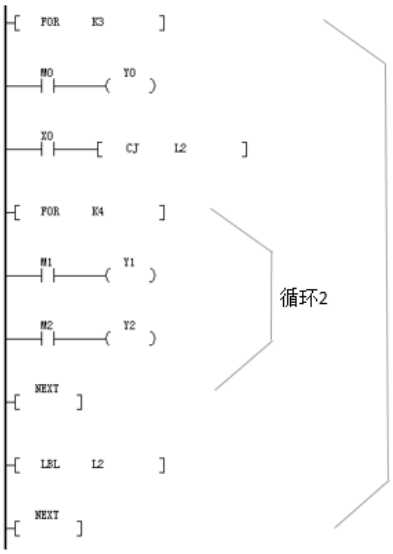
循环1执行2次后在到NEXT指令以后的程序继续执行，而循环1每执行一次循环2会执行3次，而循环2每执行一次循环3又执行4次，所以循环3共执行2*3*4=24次，循环2共执行2*3=6次

- 示例二



想跳过 FOR~NEXT 指令时，可用 CJ 跳转指令实现，范例中当 X0 为 OFF 时，执行循环 1 和循环 2，当 X0 为 ON 时，CJ 指令跳至 L2 处，循环 1 和循环 2 之间的程序不被执行。

- 示例三



想跳过循环内嵌套的FOR~NEXT指令时，也可用CJ跳转指令实现，范例中当X0为OFF时，执行循环1内的循环2，当X0为ON时，CJ指令跳转至L2处，循环1内嵌套的循环2FOR~NEXT被CJ指令跳过不执行。

3.3 SFC指令

3.3.1 指令列表

SFC指令涵盖以下指令条目：

指令类别	名称	功能
SFC指令	STL	程序跳至副母线
	RET	程序返回主母线
	OUTSTL	输出程序跳至副母线
	SETSTL	置位程序跳至副母线
	RSTSTL	清除程序跳至副母线

说明

SFC指令仅用于主程序，不能用于子程序和中断子程序等。

3.3.2 STL

STL — 程序跳至副母线

16位指令	STL 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	STL序号	要执行的STL语句的S序号	-	BOOL

表3–15 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	-	-	S[1]

说明

[1]仅支持S元件。

3.3.3 RET

RET — 程序返回主母线

16位指令	RET连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
-	-	-	-	-

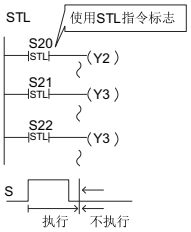
功能和指令说明

步进梯形图是一种根据被控设备的运行过程，分解为若干个状态或工序，针对每一个状态进行逻辑编程的方式，再根据信号条件进行状态间的切换。编程时采用STL梯形图，这种编程方法思路清晰，简化了逻辑设计，方便调试和维护。

步进梯形图指令可用梯形图表示，在步进梯形图中，将状态（S）看作为一个控制工序，从中将输入条件与输出控制按顺序编程。这种控制最大的特点是在工序进行时，与前一工序不接通，以各道工序的简单顺序，即可控制设备。

步进梯形图有相应的编程规则，既包含了普通梯形图的编程方法，又与普通的梯形图编程有一定的差异，说明如下：

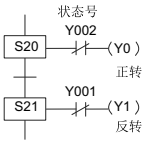
步进梯形图程序以STL指令开始（注意与普通梯形图中S不同），以RET指令结束，中间的程序以S状态引导，后续该S状态的所有操作逻辑，包括条件满足时切换为下一状态的操作。



如果STL触点S接通，则与其相连的回路动作；若S触点断开，则与其相连的回路不动作。但是在一个扫描周期以后，不再执行指令（跳转状态）。

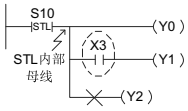
在不同的状态S，可对同样的输出软元件（如Y3）。此时，S21或S22接通时，Y3被输出。但在同一S状态中同样存在“双线圈”处理问题，使用中请注意。

状态S号编号不可重复使用。



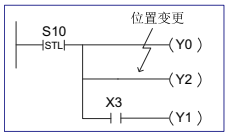
☆ **输出的互锁问题：**

在状态的转移过程中，会存在两种状态同时接通瞬间（一个扫描周期）。因此，为避免不能同时接通的一对输出同时接通，需在可编程控制器外部设置互锁，同时要在相应程序上设置互锁。

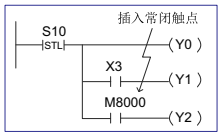


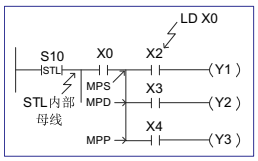
☆ **输出驱动方法**

从状态内的母线，一旦写入LD或LDI指令后，不能再使用不需要触点的指令（如左图所示），需要按下图方法改变这样的回路。



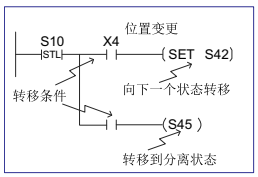
或





☆ 栈操作指令MPS/MRD/MPP的位置

在状态内，不能从STL内母线中直接使用MPS/MRD/MPP指令。而是须在LD或LDI指令以后编制程序。如左图所示。



☆ **状态的转移方法**

OUT指令与SET指令对于STL指令后的状态（S）只有同样的功能，都将自动复位转移源。此外，还有自保持功能。

但使用OUT指令时，在SFC图中用于向分离的状态转移。

3.3.4 OUTSTL/SETSTL/RSTSTL

OUTSTL — 输出程序跳至副母线

SETSTL — 置位程序跳至副母线

RSTSTL — 清除程序跳至副母线

16位指令	OUTSTL 连续执行			
32位指令	—			
16位指令	SETSTL 连续执行			
32位指令	—			
16位指令	RSTSTL 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	STL序号	要执行的STL语句的S序号	-	BOOL

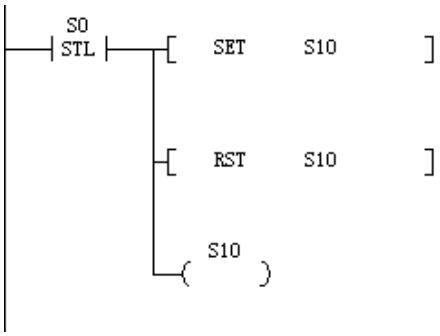
表3–16 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	-	-	-	-	-	S[1]

说明

[1]仅支持S元件。

指令示例



3.4 触点指令

3.4.1 触点比较

3.4.1.1 指令列表

触点比较指令涵盖以下指令条目：

指令类别	名称	功能
触点比较	LD＝	LD触点比较等于
	LD>	LD触点比较大于
	LD<	LD触点比较小于
	LD<>	LD触点比较不等于
	LD>＝	LD触点比较大于等于
	LD<＝	LD触点比较小于等于
	AND＝	AND触点比较等于
	AND>	AND触点比较大于
	AND<	AND触点比较小于
	AND<>	AND触点比较不等于
	AND>＝	AND触点比较大于等于
	AND<＝	AND触点比较小于等于
	OR＝	OR触点比较等于
	OR>	OR触点比较大于
	OR<	OR触点比较小于
	OR<>	OR触点比较不等于
	OR>＝	OR触点比较大于等于
	OR<＝	OR触点比较小于等于
	FLDD>	浮点数>比较的状态触点S1 ＞ S2时导通
	FLDD>=	浮点数>=比较的状态触点S1 ≧ S2时导通
	FLDD<	浮点数<比较的状态触点S1 ＜ S2时导通
	FLDD<=	浮点数<=比较的状态触点S1 ≦ S2时导通
	FLDD=	浮点数=比较的状态触点S1 ＝ S2时导通
	FLDD<>	浮点数<>比较的状态触点S1 ≠ S2时导通
	FAND>	浮点数>比较的与状态触点S1 ＞ S2时导通
	FAND>=	浮点数>=比较的与状态触点S1 ≧ S2时导通
	FAND<	浮点数<比较的与状态触点S1 ＜ S2时导通
	FAND<=	浮点数<=比较的与状态触点S1 ≦ S2时导通
	FAND=	浮点数=比较的与状态触点S1 ＝ S2时导通
	FAND<>	浮点数<>比较的与状态触点S1 ≠ S2时导通
	FOR>	浮点数>比较的或状态触点S1 ＞ S2时导通
	FOR>=	浮点数>=比较的或状态触点S1 ≧ S2时导通
	FOR<	浮点数<比较的或状态触点S1 ＜ S2时导通
	FOR<=	浮点数<=比较的或状态触点S1 ≦ S2时导通
	FOR=	浮点数=比较的或状态触点S1 ＝ S2时导通
	FOR<>	浮点数<>比较的或状态触点S1 ≠ S2时导通
	LDZ>	绝对值>比较的状态触点|S1 - S2|＞| S3 |时导通
	LDZ>=	绝对值>=比较的状态触点|S1 - S2|≧| S3 |时导通
	LDZ<	绝对值<比较的状态触点|S1 - S2|＜| S3 |时导通
	LDZ<=	绝对值<=比较的状态触点|S1 - S2|≦| S3 |时导通
	LDZ=	绝对值=比较的状态触点|S1 - S2|＝| S3 |时导通
	LDZ<>	绝对值<>比较的状态触点|S1 - S2|≠| S3 |时导通

指令类别	名称	功能
触点比较	ANDZ>	绝对值>比较的与状态触点|S1 - S2|>| S3 |时导通
	ANDZ>=	绝对值>=比较的与状态触点|S1 - S2|≧| S3 |时导通
	ANDZ<	绝对值<比较的与状态触点|S1 - S2|<| S3 |时导通
	ANDZ<=	绝对值<=比较的与状态触点|S1 - S2|≦| S3 |时导通
	ANDZ=	绝对值=比较的与状态触点|S1 - S2|=| S3 |时导通
	ANDZ<>	绝对值<>比较的与状态触点|S1 - S2|≠| S3 |时导通
	ORZ>	绝对值>比较的或状态触点|S1 - S2|>| S3 |时导通
	ORZ>=	绝对值>=比较的或状态触点|S1 - S2|≧| S3 |时导通
	ORZ<	绝对值<比较的或状态触点|S1 - S2|<| S3 |时导通
	ORZ<=	绝对值<=比较的或状态触点|S1 - S2|≦| S3 |时导通
	ORZ=	绝对值=比较的或状态触点|S1 - S2|=| S3 |时导通
	ORZ<>	绝对值<>比较的或状态触点|S1 - S2|≠| S3 |时导通

3.4.1.2 AND#

数据比较指令 — 将两个操作数进行比较，将比较结果以逻辑状态输出，参与比较的变量都按有符号数处理。

AND= — AND触点比较等于

AND> — AND触点比较大于

AND< — AND触点比较小于

AND<> — AND触点比较不等于

AND>= — AND触点比较大于等于

AND<= — AND触点比较小于等于

16位指令	AND= 连续执行			
32位指令	ANDD= 连续执行			
16位指令	AND> 连续执行			
32位指令	ANDD> 连续执行			
16位指令	AND< 连续执行			
32位指令	ANDD< 连续执行			
16位指令	AND<> 连续执行			
32位指令	ANDD<> 连续执行			
16位指令	AND<= 连续执行			
32位指令	ANDD<= 连续执行			
16位指令	AND>= 连续执行			
32位指令	ANDD>= 连续执行			
操作数	名称	描述	范围	数据类型
S1	比较数1	待比较的数据源或数据变量单元1	-	INT/DINT
S2	比较数2	待比较的数据源或数据变量单元2	-	INT/DINT

说明

- #号为=、>、<、<>、>=、<=之一；
- 该指令之前已有其他逻辑操作；
- 该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的运算，指令中参与比较的变量都按有符号数处理；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“LD*/LDD*”，由后台自动生成对应指令。

表3-17 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-

功能和指令说明

AND触点型比较方式如下表所示：

表3-18

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
AND=	232	ANDD=	S1=S2	S1≠S2
AND>	233	ANDD>	S1>S2	S1<=S2
AND<	234	ANDD<	S1<S2	S1>=S2
AND<>	236	ANDD<>	S1<>S2	S1=S2
AND<=	237	ANDD<=	S1<=S2	S1>S2
AND>=	238	ANDD>=	S1>=S2	S1<S2

指令示例

若参与比较的变量为32bit，应使用32bit指令ANDD#，否则会出错。

3.4.1.3 LD#

触点比较指令 — 将两个操作数进行比较，将比较结果以逻辑状态输出，参与比较的变量都按有符号数处理。

LD= — 触点比较等于

LD> — 触点比较大于

LD< — 触点比较小于

LD<> — 触点比较不等于

LD>= — 触点比较大于等于

LD<= — 触点比较小于等于

16位指令	LD= 连续执行			
32位指令	LDD= 连续执行			
16位指令	LD> 连续执行			
32位指令	LDD> 连续执行			
16位指令	LD< 连续执行			
32位指令	LDD< 连续执行			
16位指令	LD<> 连续执行			
32位指令	LDD<> 连续执行			
16位指令	LD<= 连续执行			
32位指令	LDD<= 连续执行			
16位指令	LD>= 连续执行			
32位指令	LDD>= 连续执行			
操作数	名称	描述	范围	数据类型
S1	比较数1	待比较的数据源或数据变量单元1	-	INT/DINT
S2	比较数2	待比较的数据源或数据变量单元2	-	INT/DINT

说明

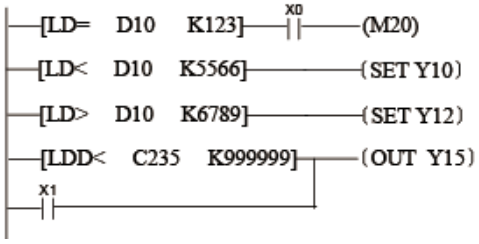
- #号为=、>、<、<>、>=、<=之一；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“LD*/LDD*”，由后台自动生成对应指令。

功能和指令说明

LD触点型比较方式如下表所示：

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
LD=	224	LDD=	S1=S2	S1≠S2
LD>	225	LDD>	S1S2	S1<=S2
LD<	226	LDD<	S1<S2	S1=S2
LD<>	228	LDD<>	S1<>S2	S1=S2
LD<=	229	LDD<=	S1<=S2	S1S2
LD>=	230	LDD>=	S1=S2	S1<S2

指令示例



若参与比较的变量为32bit，应使用32bit指令LDD#，否则出错。

3.4.1.4 OR#

数据比较指令 — 将两个操作数进行比较，将比较结果以逻辑状态输出，参与比较的变量都按有符号数处理。

OR= — OR触点比较等于

OR> — OR触点比较大于

OR< — OR触点比较小于

OR<> — OR触点比较不等于

OR>= — OR触点比较大于等于

OR<= — OR触点比较小于等于

16位指令	OR= 连续执行			
32位指令	ORD= 连续执行			
16位指令	OR> 连续执行			
32位指令	ORD> 连续执行			
16位指令	OR< 连续执行			
32位指令	ORD< 连续执行			
16位指令	OR<> 连续执行			
32位指令	ORD<> 连续执行			
16位指令	OR<= 连续执行			
32位指令	ORD<= 连续执行			
16位指令	OR>= 连续执行			
32位指令	ORD>= 连续执行			
操作数	名称	描述	范围	数据类型
S1	比较数1	待比较的数据源或数据变量单元1	-	INT/DINT
S2	比较数2	待比较的数据源或数据变量单元2	-	INT/DINT

说明

- #号为=、>、<、<>、>=、<=之一；
- 该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的运算，指令中参与比较的变量都按有符号数处理；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“LD*/LDD*”，由后台自动生成对应指令。

表3–19 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-

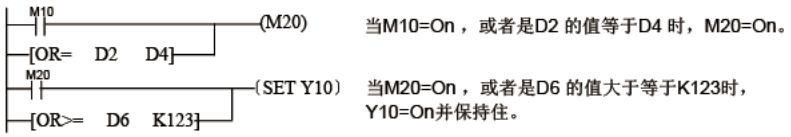
功能和指令说明

OR触点型比较方式如下表所示：

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
OR=	240	ORD=	S1=S2	S1≠S2
OR>	241	ORD>	S1>S2	S1<=S2

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
OR<	242	ORD<	S1<S2	S1>=S2
OR<>	244	ORD<>	S1<>S2	S1=S2
OR<=	245	ORD<=	S1<=S2	S1>S2
OR>=	246	ORD>=	S1>=S2	S1<S2

指令示例



若参与比较的变量为32bit，应使用32bit指令ORD#，否则会出错。

3.4.1.5 FLDD#

浮点数触点比较 - 比较两浮点操作数大小，根据比较结果把触点置ON或者OFF，与左母线直接相连的节点。

FLDD= — 浮点数触点比较等于

FLDD> — 浮点数触点比较大于

FLDD< — 浮点数触点比较小于

FLDD<> — 浮点数触点比较不等于

FLDD>= — 浮点数触点比较大于等于

FLDD<= — 浮点数触点比较小于等于

16位指令	—			
32位指令	FLDD> 连续执行			
16位指令	—			
32位指令	FLDD>= 连续执行			
16位指令	—			
32位指令	FLDD< 连续执行			
16位指令	—			
32位指令	FLDD<= 连续执行			
16位指令	–			
32位指令	FLDD= 连续执行			
16位指令	—			
32位指令	FLDD<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	REAL
S2	数据2	源数据2软元件编号	-	REAL

说明

- #号为=、>、<、< >、<=、>=之一；
- FLDD*、FANDD*、FORD*指令，输入方式统一为“FLDD*”，由后台自动生成对应指令。

表3-20 软元件列表

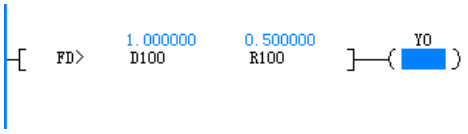
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-

功能和指令说明

[S1]与[S2]比较的指令。条件满足时导通，否则不导通。

32位指令	导通条件	非导通条件
FLDD>	S1>S2	S1<=S2
FLDD>=	S1>=S2	S1<S2
FLDD<	S1<S2	S1>=S2
FLDD<=	S1<=S2	S1>S2
FLDD=	S1=S2	S1< >S2
FLDD< >	S1< >S2	S1=S2

指令示例



3.4.1.6 FAND#

浮点数与触点比较 - 比较两浮点操作数大小，根据比较结果把触点置ON或者OFF，与其它节点串联连接的节点。

FAND= — 浮点数与触点比较等于

FAND> — 浮点数与触点比较大于

FAND< — 浮点数与触点比较小于

FAND<> — 浮点数与触点比较不等于

FAND>= — 浮点数与触点比较大于等于

FAND<= — 浮点数与触点比较小于等于

16位指令	—
32位指令	FAND> 连续执行
16位指令	—
32位指令	FAND>= 连续执行
16位指令	—
32位指令	FAND< 连续执行
16位指令	—
32位指令	FAND<= 连续执行
16位指令	–
32位指令	FAND= 连续执行

16位指令	—			
32位指令	FAND<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	REAL
S2	数据2	源数据2软元件编号	-	REAL

说明

- #号为=、>、<、< >、<=、>=之一；
- FLDD*、FANDD*、FORD*指令，输入方式统一为“FLDD*”，由后台自动生成对应指令。

表3–21 软元件列表

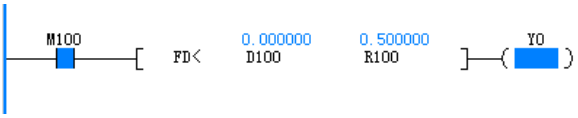
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-

功能和指令说明

[S1]与[S2]比较的指令。条件满足时导通，否则不导通。

32位指令	导通条件	非导通条件
FANDD>	S1>S2	S1<=S2
FANDD>=	S1>=S2	S1<S2
FANDD<	S1<S2	S1>=S2
FLD<=	S1<=S2	S1>S2
FANDD=	S1=S2	S1< >S2
FANDD< >	S1< >S2	S1=S2

指令示例



3.4.1.7 FOR#

浮点数或触点比较 - 比较两浮点操作数大小，根据比较结果把触点置ON或者OFF，与其它节点并联连接的节点。

FOR= — 浮点数或触点比较等于

FOR> — 浮点数或触点比较大于

FOR< — 浮点数或触点比较小于

FOR<> — 浮点数或触点比较不等于

FOR>= — 浮点数或触点比较大于等于

FOR<= — 浮点数或触点比较小于等于

16位指令	—			
32位指令	FOR> 连续执行			
16位指令	—			
32位指令	FOR>= 连续执行			
16位指令	—			
32位指令	FOR< 连续执行			
16位指令	—			
32位指令	FOR<= 连续执行			
16位指令	–			
32位指令	FOR= 连续执行			
16位指令	—			
32位指令	FOR<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	REAL
S2	数据2	源数据2软元件编号	-	REAL

说明

- #号为=、>、<、< >、<=、>=之一；
- FLDD*、FANDD*、FORD*指令，输入方式统一为“FLDD*”，由后台自动生成对应指令。

表3–22 软元件列表

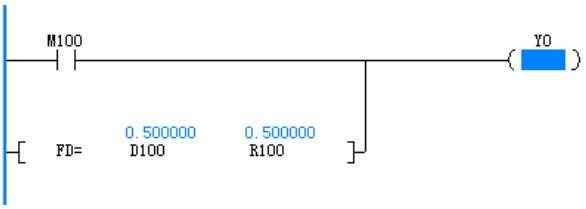
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-

功能和指令说明

[S1]与[S2]比较的指令。条件满足时导通，否则不导通。

32位指令	导通条件	非导通条件
FORD>	S1>S2	S1<=S2
FORD>=	S1>=S2	S1<S2
FORD<	S1<S2	S1>=S2
FORD<=	S1<=S2	S1>S2
FORD=	S1=S2	S1< >S2
FORD< >	S1< >S2	S1=S2

指令示例



3.4.1.8 LDZ#

绝对值比较触点 - 将S1与S2相减后结果的绝对值与S3的绝对值进行比较，根据比较结果把触点置ON或者OFF，与左母线直接相连的节点。

LDZ= — 绝对值触点比较等于

LDZ> — 绝对值触点比较大于

LDZ< — 绝对值触点比较小于

LDZ<> — 绝对值触点比较不等于

LDZ>= — 绝对值触点比较大于等于

LDZ<= — 绝对值触点比较小于等于

16位指令	LDZ> 连续执行			
32位指令	LDDZ> 连续执行			
16位指令	LDZ>= 连续执行			
32位指令	LDDZ>= 连续执行			
16位指令	LDZ< 连续执行			
32位指令	LDDZ< 连续执行			
16位指令	LDZ<= 连续执行			
32位指令	LDDZ<= 连续执行			
16位指令	LDZ= 连续执行			
32位指令	LDDZ= 连续执行			
16位指令	LDZ<> 连续执行			
32位指令	LDDZ<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	被减数	被减数来源元件	-	INT/DINT
S2	减数	减数来源元件	-	INT/DINT
S3	比较值	比较值来源元件	-	INT/DINT

说明

- #号为=、>、<、< >、<=、>=之一；
- LDZ*/LDDZ*、ANDZ*/ANDDZ*、ORZ*/ORDZ*指令，输入方式统一为“LDZ*/LDDZ*”，由后台自动生成对应指令。

表3–23 软元件列表

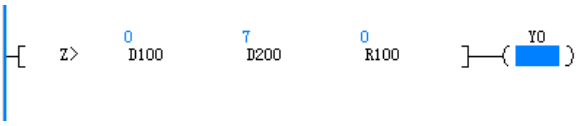
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
S3	-	-	-	√	√	√	√	-	-

功能和指令说明

[S1]与[S2]相减后结果的绝对值与[S3]的绝对值作比较的指令。条件满足时导通，否则不导通。

16位指令	32位指令	导通条件	非导通条件
LDZ>	LDDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
LDZ>=	LDDZ>=	|S1-S2|>=|S3|	|S1-S2|<|S3|
LDZ<	LDDZ<	|S1-S2|<|S3|	|S1-S2|>=|S3|
LDZ<=	LDDZ<=	|S1-S2|<=|S3|	|S1-S2|>|S3|
LDZ=	LDDZ=	|S1-S2|=|S3|	|S1-S2|< >|S3|
LDZ< >	LDDZ< >	|S1-S2|< >|S3|	|S1-S2|=|S3|

指令示例



3.4.1.9 ANDZ#

绝对值比较与触点 - 将S1与S2相减后结果的绝对值与S3的绝对值进行比较，根据比较结果把触点置ON或者OFF，与其它节点串联连接的节点。

ANDZ= — 绝对值与触点比较等于

ANDZ> — 绝对值与触点比较大于

ANDZ< — 绝对值与触点比较小于

ANDZ<> — 绝对值与触点比较不等于

ANDZ>= — 绝对值与触点比较大于等于

ANDZ<= — 绝对值与触点比较小于等于

16位指令	ANDZ> 连续执行			
32位指令	ANDDZ> 连续执行			
16位指令	ANDZ>= 连续执行			
32位指令	ANDDZ>= 连续执行			
16位指令	ANDZ< 连续执行			
32位指令	ANDDZ< 连续执行			
16位指令	ANDZ<= 连续执行			
32位指令	ANDDZ<= 连续执行			
16位指令	ANDZ= 连续执行			
32位指令	ANDDZ= 连续执行			
16位指令	ANDZ<> 连续执行			
32位指令	ANDDZ<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	被减数	被减数来源元件	-	INT/DINT
S2	减数	减数来源元件	-	INT/DINT
S3	比较值	比较值来源元件	-	INT/DINT

说明

- #号为=、>、<、<>、<=、>=之一；
- LDZ*/LDDZ*、ANDZ*/ANDDZ*、ORZ*/ORDZ*指令，输入方式统一为“LDZ*/LDDZ*”，由后台自动生成对应指令。

表3–24 软元件列表

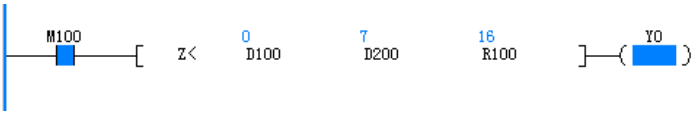
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
S3	-	-	-	√	√	√	√	-	-

功能和指令说明

[S1]与[S2]相减后结果的绝对值与[S3]的绝对值作比较的指令。条件满足时导通，否则不导通。

16位指令	32位指令	导通条件	非导通条件
ANDZ>	ANDDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
ANDZ >=	ANDDZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ANDZ <	ANDDZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ANDZ <=	ANDDZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|
ANDZ =	ANDDZ =	|S1-S2|=|S3|	|S1-S2|< >|S3|
ANDZ < >	ANDDZ < >	|S1-S2|< >|S3|	|S1-S2|=|S3|

指令示例



3.4.1.10 ORZ#

绝对值比较或触点 - 将S1与S2相减后结果的绝对值与S3的绝对值进行比较，根据比较结果把触点置ON或者OFF，与其它节点并联连接的节点进行或运算。

ORZ= — 绝对值或触点比较等于

ORZ> — 绝对值或触点比较大于

ORZ< — 绝对值或触点比较小于

ORZ<> — 绝对值或触点比较不等于

ORZ>= — 绝对值或触点比较大于等于

ORZ<= — 绝对值或触点比较小于等于

16位指令	ORZ> 连续执行
32位指令	ORDZ> 连续执行
16位指令	ORZ>= 连续执行
32位指令	ORDZ>= 连续执行

16位指令	ORZ< 连续执行			
32位指令	ORDZ< 连续执行			
16位指令	ORZ<= 连续执行			
32位指令	ORDZ<= 连续执行			
16位指令	ORZ= 连续执行			
32位指令	ORDZ= 连续执行			
16位指令	ORZ<> 连续执行			
32位指令	ORDZ<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	被减数	被减数来源元件	-	INT/DINT
S2	减数	减数来源元件	-	INT/DINT
S3	比较值	比较值来源元件	-	INT/DINT

说明

- #号为=、>、<、< >、<=、>=之一；
- LDZ*/LDDZ*、ANDZ*/ANDDZ*、ORZ*/ORDZ*指令，输入方式统一为“LDZ*/LDDZ*”，由后台自动生成对应指令。

表3-25 软元件列表

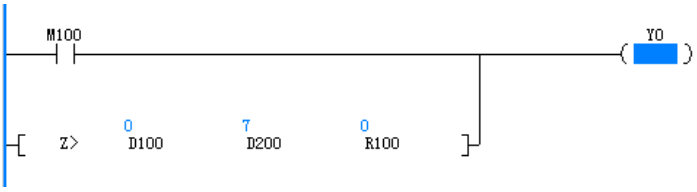
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
S3	-	-	-	√	√	√	√	-	-

功能和指令说明

[S1]与[S2]相减后结果的绝对值与[S3]的绝对值作比较的指令。条件满足时导通，否则不导通。

16位指令	32位指令	导通条件	非导通条件
ORZ>	ORDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
ORZ >=	ORDZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ORZ <	ORDZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ORZ <=	ORDZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|
ORZ =	ORDZ =	|S1-S2|=|S3|	|S1-S2|< >|S3|
ORZ < >	ORDZ < >	|S1-S2|< >|S3|	|S1-S2|=|S3|

指令示例



3.4.2 触点逻辑运算

3.4.2.1 指令列表

触点逻辑运算指令涵盖以下指令条目：

指令类别	名称	功能
触点逻辑运算	LD&	LD逻辑与运算
	LD|	LD逻辑或运算
	LD^	LD逻辑异或运算
	AND&	AND逻辑与运算
	AND|	AND逻辑或运算
	AND^	AND逻辑异或运算
	OR&	OR逻辑与运算
	OR|	OR逻辑或运算
	OR^	OR逻辑异或运算

3.4.2.2 LD#

LD逻辑运算指令 — 位逻辑运算结果作为该触点导通状态，与左母线直接相连的节点

LD&— LD逻辑与运算

LD|— LD逻辑或运算

LD^ — LD逻辑异或运算

16位指令	LD& 连续执行			
32位指令	LDD& 连续执行			
16位指令	LD| 连续执行			
32位指令	LDD| 连续执行			
16位指令	LD^ 连续执行			
32位指令	LDD^ 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	INT/DINT
S2	数据2	源数据2软元件编号	-	INT/DINT

说明

- #号为&、|、^之一；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“LD*/LDD*”，由后台自动生成对应指令。

表3–26 软元件列表

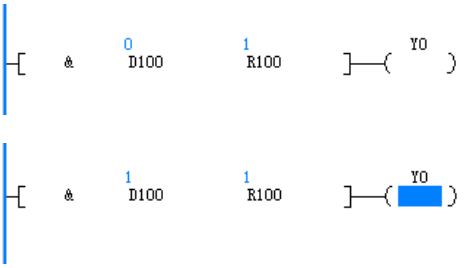
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-

功能和指令说明

[S1]与[S2]的内容进行逻辑运算(“与&”、“非|”、“异或^”)的指令，结果不为0时，该指令导通；比较结果为0时，该指令不导通。执行结果如下：

16位指令	32位指令	导通条件	非导通条件
LD&	LDD&	S1&S2≠0	S1&S2 = 0
LD|	LDD|	S1|S2 ≠0	S1|S2 = 0
LD^	LDD^	S1^S2 ≠0	S1^S2 = 0

指令示例



3.4.2.3 AND#

位逻辑运算结果作为该触点导通状态，与其它节点串联连接的节点

AND&— AND逻辑与运算

AND|— AND逻辑或运算

AND^ — AND逻辑异或运算

16位指令	AND& 连续执行			
32位指令	ANDD& 连续执行			
16位指令	AND| 连续执行			
32位指令	ANDD| 连续执行			
16位指令	AND^ 连续执行			
32位指令	ANDD^ 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	INT/DINT
S2	数据2	源数据2软元件编号	-	INT/DINT

说明

- #号为&、|、^之一；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“AND*/ANDD*”，由后台自动生成对应指令。

表3-27 软元件列表

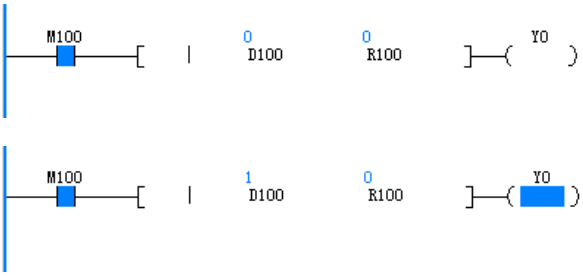
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-

功能和指令说明

[S1]与[S2]的内容进行逻辑运算(“与&”、“非|”、“异或^”)的指令，结果不为0时，该指令导通；比较结果为0时，该指令不导通。执行结果如下：

16位指令	32位指令	导通条件	非导通条件
AND&	ANDD&	S1&S2≠0	S1&S2 = 0
AND|	ANDD|	S1|S2 ≠0	S1|S2 = 0
AND^	ANDD^	S1^S2 ≠0	S1^S2 = 0

指令示例



3.4.2.4 OR#

位逻辑运算结果作为该触点导通状态，与其它节点并联连接的节点。

OR&— OR逻辑与运算

OR|— OR逻辑或运算

OR^ — OR逻辑异或运算

16位指令	OR& 连续执行			
32位指令	ORD& 连续执行			
16位指令	OR| 连续执行			
32位指令	ORD| 连续执行			
16位指令	OR^ 连续执行			
32位指令	ORD^ 连续执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	INT/DINT
S2	数据2	源数据2软元件编号	-	INT/DINT

说明

- #号为&、|、^之一；
- LD*/LDD*、AND*/ANDD*、OR*/ORD*指令，输入方式统一为“AND*/ANDD*”，由后台自动生成对应指令。

表3-28 软元件列表

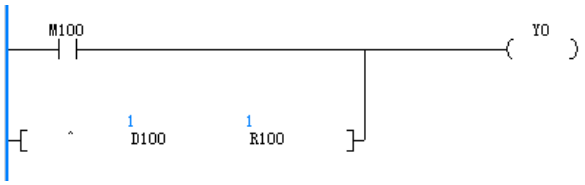
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-

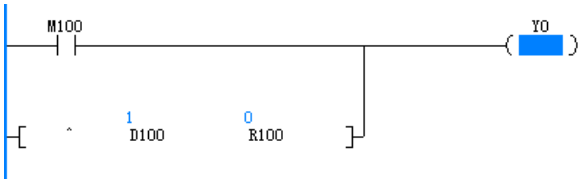
功能和指令说明

[S1]与[S2]的内容进行逻辑运算(“与&”、“非|”、“异或^”)的指令，结果不为0时，该指令导通；比较结果为0时，该指令不导通。执行结果如下：

16位指令	32位指令	导通条件	非导通条件
OR&	ORD&	S1&S2≠0	S1&S2 = 0
OR|	ORD|	S1|S2 ≠0	S1|S2 = 0
OR^	ORD^	S1^S2 ≠0	S1^S2 = 0

指令示例





3.5 数据运算指令

3.5.1 四则运算

3.5.1.1 指令列表

四则运算指令涵盖以下指令条目：

指令类别	名称	功能
四则运算	ADD	二进制数据加法
	SUB	二进制数据减法
	MUL	二进制数据乘法
	DIV	二进制数据除法
	MOD	二进制除法求余
	EADD	二进制浮点加法
	ESUB	二进制浮点减法
	EMUL	二进制浮点乘法
	EDIV	二进制浮点除法
	INC	二进制数据加一
	DEC	二进制数据减一

3.5.1.2 ADD

ADD — 二进制数据加法

16位指令	ADD 连续执行/ADDP 脉冲执行			
32位指令	DADD 连续执行/DADDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被加数	数据或数据存储字软元件地址	-	INT/DINT
S2	加数	数据或数据存储字软元件地址	-	INT/DINT
D	和	数据存储字软元件地址	-	INT/DINT

表3–29 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

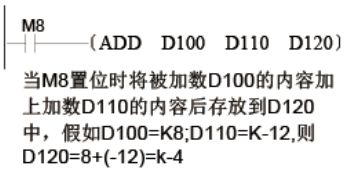
功能和指令说明

需要触点驱动，有3个操作变量，将S1和S2的值进行BIN代数相加后存入D中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

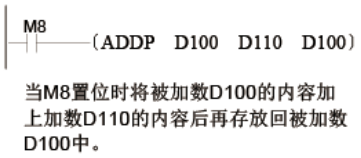
- 若计算结果为0，则0标志(M8020)会置位；
- 若计算结果超过32,767（16bit运算）或2，147，483，647（32bit运算）时，进位标志(M8022)会置位；
- 若计算结果不满-32,768（16bit运算）或-2,147,483,648（32bit运算）时，借位标志(M8021)会置位；
- 进行32bit运算时，指令中变量地址为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令示例

指令举例一：



指令举例二：



3.5.1.3 SUB

SUB — 二进制数据减法

16位指令	SUB 连续执行/SUBP 脉冲执行			
32位指令	DSUB 连续执行/DSUBP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被减数	数据或数据存储字软元件地址	-	INT/DINT
S2	减数	数据或数据存储字软元件地址	-	INT/DINT
D	差	数据存储字软元件地址	-	INT/DINT

表3–30 软元件列表

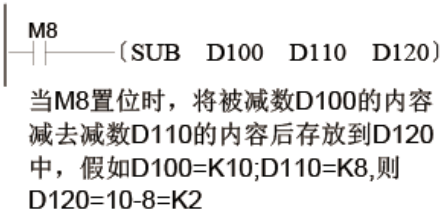
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

需要触点驱动，有3个操作变量，将S1和S2的值进行BIN代数相减后存入D中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

- 若计算结果为0，则0标志(M8020)会置位；
- 若计算结果超过32,767（16bit运算）或-2,147,483，647（32bit运算）时，进位标志(M8022)会置位；
- 若计算结果不满 -32,768（16bit运算）或-2,147,483,648（32bit运算）时，借位标志(M8021)会置位；
- 进行32bit运算时，指令中变量地址位为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令示例



3.5.1.4 MUL

MUL — 二进制数据乘法

16位指令	MUL 连续执行/MULP 脉冲执行			
32位指令	DMUL 连续执行/DMULP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被乘数	数据或数据存储字软元件地址	-	INT/DINT
S2	乘数	数据或数据存储字软元件地址	-	INT/DINT
D	积	数据存储字软元件地址	-	DINT

表3–31 软元件列表

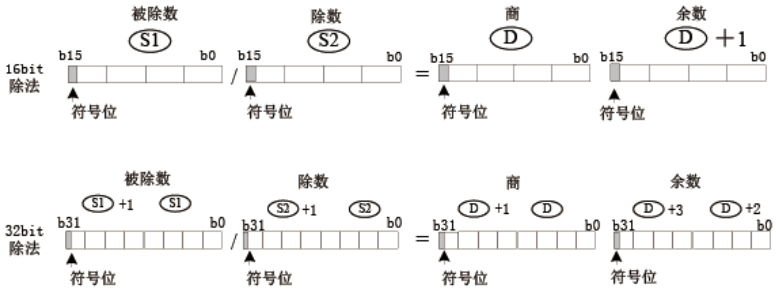
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

需要触点驱动，有3个操作变量，将被除数S1和除数S2的值进行BIN代数相除后存入D中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

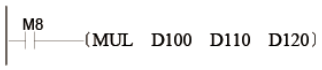
进行32bit运算时，指令中S1和S2变量地址为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算所得的商存入D、D+1所指单元。

若除数S2为0，会发生计算错误。



指令示例

梯形图



指令列表

```
LD   M8
MUL  D100  D110  D120
```

当M8置位时将被乘数D100的内容乘以乘数D110的内容后存放到D120中.
假如D100=K5;D110=K9,则D120=5×9=K45
假如D100=K1234;D110=K5678,则D120,d121=1234×5678=K7006652,需注意此时积大于16bit,需用到D的相邻高位D121,D120

3.5.1.5 DIV

DIV — 二进制除法指令

16位指令	DIV 连续执行/DIVP 脉冲执行			
32位指令	DDIV 连续执行/DDIVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被除数	数据或数据存储字软元件地址	-	INT/DINT
S2	除数	数据或数据存储字软元件地址	-	INT/DINT
D	商及余数	数据存储字软元件地址	-	INT/DINT

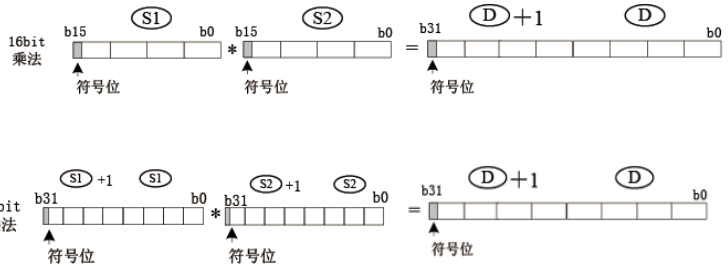
表3–32 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

需要触点驱动，有3个操作变量，将S1和S2的值进行BIN代数相乘后存入D中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

进行32bit运算时，指令中变量地址位为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算的结果只能为32bit，对于超出32bit范围的计算，最好采用浮点运算指令EMUL进行计算。



指令示例

梯形图

指令列表

LD M8

DIV D100 D110 D120

当M8置位时，将被除数D100的内容除以除数D110的内容后存放到D120中。

3.5.1.6 MOD

整数相除求取余数的指令。

MOD — 除法求余指令

16位指令	MOD 连续执行/MODP 脉冲执行			
32位指令	DMOD 连续执行/DMODP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	数据1	源数据1软元件编号	-	INT/DINT

S2	数据2	源数据2软元件编号	-	INT/DINT
D	运算结果	运算结果保存元件起始元件	-	INT/DINT

表3–33 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

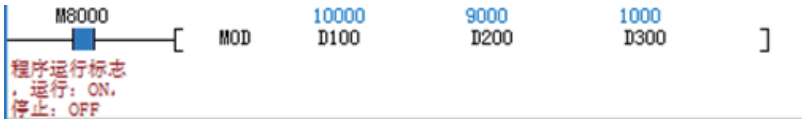
功能和指令说明

需要触点驱动，有3个操作变量，将被除数S1和除数S2的值进行相除后余数存入D中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

进行32bit运算时，指令中S1和S2变量地址为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算所得的余数存入D、D+1所指单元。

若除数S2为0，会发生计算错误；若被除数为负数，余数即为负数。

指令示例



	元件名称	数据类型	显示格式	当前值
1	D100	INT	十进制	10000
2	D200	INT	十进制	9000
3	D300	INT	十进制	1000
4				
5				

3.5.1.7 EADD

进行二进制浮点的加法运算。

EADD — 浮点加法指令

16位指令	—			
32位指令	DEADD 连续执行/DEADDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被加数	二进制浮点的被加数	-	REAL
S2	加数	二进制浮点的加数	-	REAL
D	和	二进制浮点加法和的存放单元	-	REAL

表3-34 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行二进制浮点的加法运算。其中：

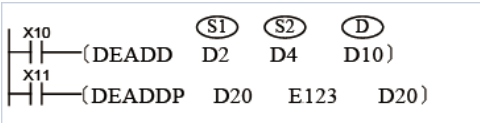
- S1和S2分别为二进制浮点的被加数和加数；
- D为二进制浮点加法和的存放单元。

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令示例



示例说明

当X10＝ON时，二进制浮点数（D3，D2）与二进制浮点数（D5，D4）相加后，二进制浮点数和存放于（D11，D10）；

当X11由OFF变为ON时，二进制浮点数（D21，D20）的值增大123。

和的存放单元可以与加数或被加数为同一单元，此时请使用脉冲执行型指令DEADDP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

3.5.1.8 ESUB

进行二进制浮点的减法运算。

ESUB — 浮点减法指令

16位指令	—			
32位指令	DESUB 连续执行/DESUBP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被减数	二进制浮点的被减数	-	REAL
S2	减数	二进制浮点的减数	-	REAL
D	差	二进制浮点减法差的存放单元	-	REAL

表3-35 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行二进制浮点的减法运算。其中：

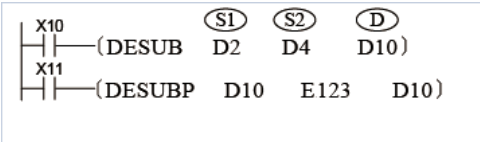
- S1和S2分别为二进制浮点的被减数和减数；
- D为二进制浮点减法差的存放单元。

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令示例



示例说明

当X10＝ON时，二进制浮点数（D3，D2）减去二进制浮点数（D5，D4）后，二进制浮点数差存放于（D11，D10）；

当X11由OFF变为ON时，二进制浮点数（D11，D10）的值减小123；

差值的存放单元可以与减数或被减数为同一单元，此时请使用脉冲执行型指令DESUBP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

3.5.1.9 EMUL

进行二进制浮点的乘法运算。

EMUL — 浮点乘法指令

16位指令	—			
32位指令	DEMUL 连续执行/DEMULP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被乘数	二进制浮点的被乘数	-	REAL
S2	乘数	二进制浮点的乘数	-	REAL
D	积	二进制浮点乘法积的存放单元	-	REAL

表3-36 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行二进制浮点的乘法运算。其中：

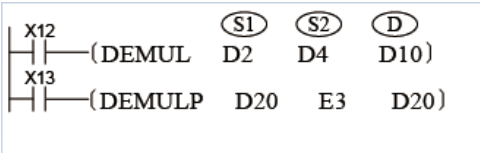
- S1和S2分别为二进制浮点的被乘数和乘数；
- D为二进制浮点乘法积的存放单元。

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令示例



示例说明

当X12＝ON时，二进制浮点数（D3，D2）乘以二进制浮点数（D5，D4）后，二进制浮点数积存放于（D11，D10）；

当X13由OFF变为ON时，二进制浮点数（D21，D20）的值乘以3倍后存回（D21，D20）。

积的存放单元可以与乘数或被乘数为同一单元，此时请使用脉冲执行型指令DEMULP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

3.5.1.10 EDIV

进行二进制浮点的除法运算。

EDIV — 浮点除法指令

16位指令	—			
32位指令	DEDIV 连续执行/DEDIVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	被除数	二进制浮点的被除数	-	REAL
S2	除数	二进制浮点的除数	-	REAL
D	商	二进制浮点除法商的存放单元启始地址	-	REAL

表3-37 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行二进制浮点的除法运算。其中：

- S1和S2分别为二进制浮点的被除数和除数；
- D为二进制浮点除法商的存放单元启始地址。

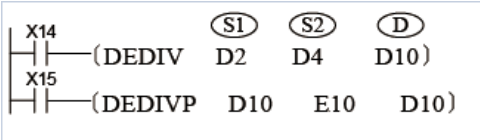
若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

除数不得为0，否则计算出错。

指令示例



示例说明

当X14＝ON时，二进制浮点数（D3，D2）除以二进制浮点数（D5，D4）后，二进制浮点数商存放于（D11，D10）；

当X15由OFF变为ON时，二进制浮点数（D11，D10）的值除以10后存回（D11，D10）；

商的存放单元可以与除数数或被除数为同一单元，此时请使用脉冲执行型指令DEDIVP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

3.5.1.11 INC

二进制数据加一。

INC - 加一指令

16位指令	INC 连续执行/INCP 脉冲执行			
32位指令	DINC 连续执行/DINCP 脉冲执行			
操作数	名称	描述	范围	数据类型
S/D	累加结果	数据存储字软元件地址	–	INT/DINT

表3-38 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S/D	-	-	-	√	√	√	-	-	-

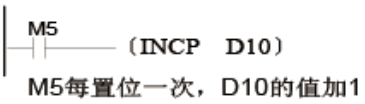
功能和指令说明

指令每执行一次，D中的数值增加1。

16位运算时，32,767再加1变为-32,768；32位运算时，2，147，483，647再加1变为-2,147,483,648。

本指令对0标志、进位、借位标志都不刷新。

指令示例



3.5.1.12 DEC

二进制数据减一。

DEC — 减一指令

16位指令	DEC连续执行/DECP 脉冲执行			
32位指令	DDEC 连续执行/DDECP 脉冲执行			
操作数	名称	描述	范围	数据类型
S/D	累减结果	累减结果数据存储字软元件地址	–	INT/DINT

表3-39 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	√	-	-

功能和指令说明

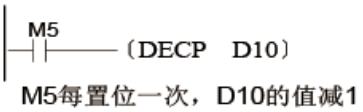
该指令每执行一次，D的数值减1：

本指令对0标志、进位、借位标志都不刷新。

16位运算时，-32,768再减1变为32,767；32位运算时，-2,147,483,648再减1变为2，147，483，647。

进行32bit运算时，指令中D变量地址位为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令示例



3.5.2 数据逻辑运算

3.5.2.1 指令列表

数据逻辑运算指令涵盖以下指令条目：

指令类别	名称	功能
数据逻辑运算	WAND	二进制数据逻辑与
	WOR	二进制数据逻辑或
	WXOR	二进制数据逻辑异或
	NEG	二进制数据求补
	ENEG	二进制浮点数符号取反

3.5.2.2 WAND

驱动条件成立时，将S1和S2按位进行逻辑与运算，并将结果存于D。

WAND - 逻辑与指令

16位指令	WAND 连续执行/WANDP 脉冲执行			
32位指令	DWAND 连续执行/DWANDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	数据1	参与与运算的数据或数据存储字软元件地址	–	INT/DINT
S2	数据2	参与与运算的数据或数据存储字软元件地址	–	INT/DINT
D	运算结果	运算结果数据存储字软元件地址	–	INT/DINT

表3–40 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

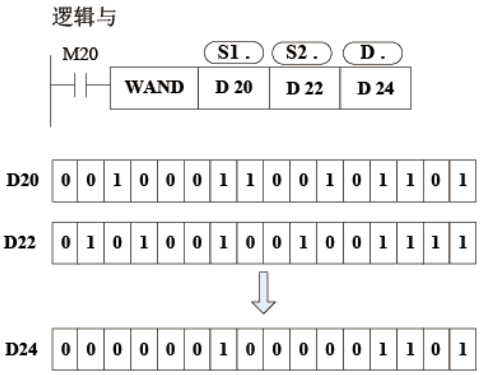
功能和指令说明

本指令执行时，将S1和S2中BIN值的各位对应作“逻辑与”运算，将结果存入D变量。

逻辑的‘与’（AND）运算的规则为任一为0结果为0。

1·1=1　1·0=0　0·1=0　0·0=0

指令示例



3.5.2.3 WOR

驱动条件成立时，将S1和S2按位进行逻辑或运算，并将结果存于D。

WOR — 逻辑或指令

16位指令	WOR 连续执行/WORP 脉冲执行			
32位指令	DWOR 连续执行/DWORP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	数据1	参与或运算数据或数据存储字软元件地址	–	INT/DINT
S2	数据2	参与或运算数据或数据存储字软元件地址	–	INT/DINT
D	运算结果	运算结果数据存储字软元件地址	–	INT/DINT

表3–41 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

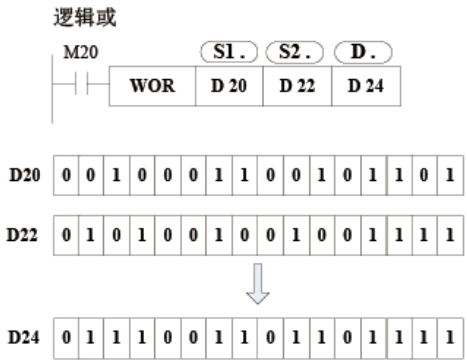
功能和指令说明

本指令执行时，将S1和S2中BIN值的各位对应作“逻辑或”运算，将结果存入D变量。

逻辑的‘或’（OR）运算的规则为任一为1结果为1。

1＋1=1　1＋0=1　0＋1=1　0＋0=0

指令示例



3.5.2.4 WXOR

驱动条件成立时，将S1和S2按位进行逻辑异或运算，并将结果存于D。

WXOR — 逻辑异或指令

16位指令	WXOR 连续执行/WXORP 脉冲执行			
32位指令	DWXOR 连续执行/DWXORP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	数据1	参与异或运算的数据或数据存储字软元件地址	–	INT/DINT
S2	数据2	参与异或运算的数据或数据存储字软元件地址	–	INT/DINT
D	运算结果	运算结果数据存储字软元件地址	–	INT/DINT

表3–42 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

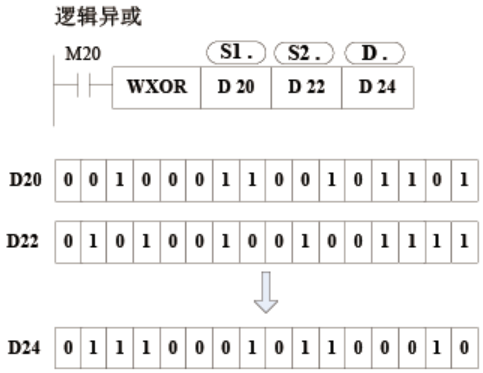
功能和指令说明

本指令执行时，将S1和S2中BIN值的各位对应作“逻辑异或”运算，将结果存入D变量。

逻辑的‘异或’（XOR）运算的规则为两者相同结果为0，两者不同结果为1。

11=0　10=1　01=1　00=0

指令示例



3.5.2.5 NEG

驱动条件成立时，对D进行逐位取反、再加1，并将结果写入D

NEG - 求补指令

16位指令	NEG 连续执行/NEGP 脉冲执行			
32位指令	DNEG 连续执行/DNEGP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	运算结果	数据存储字软元件地址	–	INT/DINT

表3–43 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-

功能和指令说明

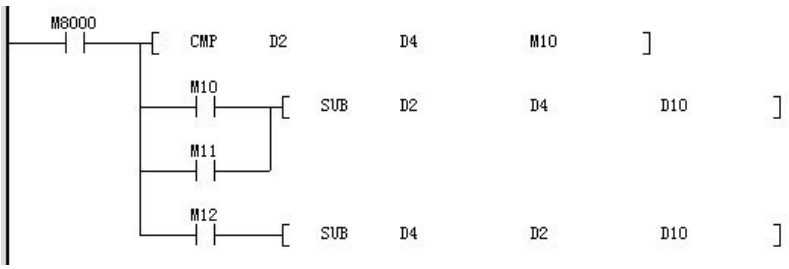
需要触点驱动，有1个操作变量。将D的数值逐位取反、再加1，存回D中。

此指令一般用脉冲执行型指令。

使用NEG指令，可得到与负的BIN值相对应的绝对值。

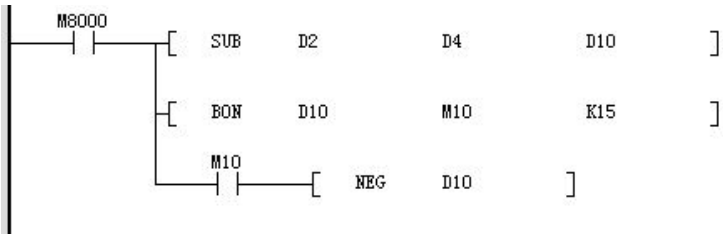
指令示例

减法运算的差取绝对值



若D2>D4时，M10=On。若D2=D4时，M11=On。若D2<D4时，M12=On。由此可保证D10为正值。

此程序可用下列的程序来表示：



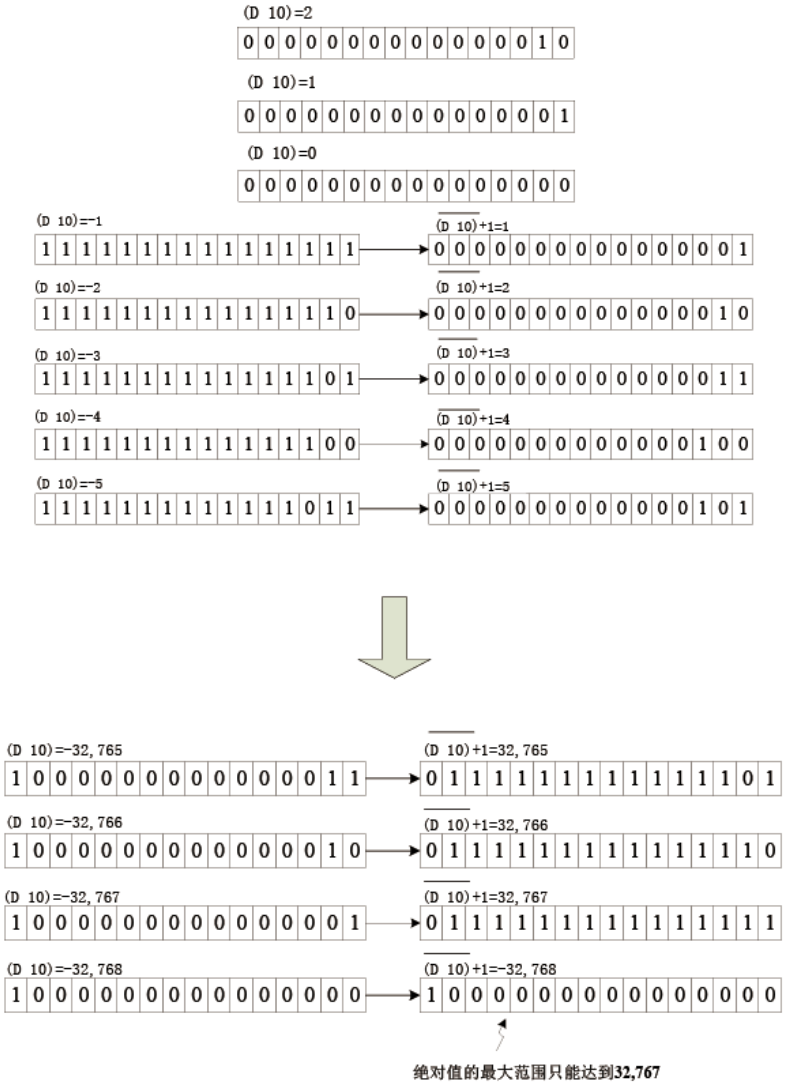
当D10的bit15为“1”时（表示D10为负数），M10=On，用NEG指令将D10取补码可得到D10的绝对值。

上述两例中假如D2=K4，D4=K8；或者D2=K8，D4=K4，D10的结果均为K4。

补充说明

正负数是以寄存器最上位（最左边）的位内容来表现，为“0”时为正数、为“1”时为负数。

最高位为1时，可使用NEG指令将它转成绝对值。



3.5.2.6 ENEG

将2进制浮点数(实数)的符号取反的指令。

ENEG — 浮点数符号取反

16位指令	—			
32位指令	DENEG 连续执行/DENEGP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	操作数	保存要执行符号翻转的2进制浮点数数据的软元件的起始编号	-	REAL

表3-44 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-

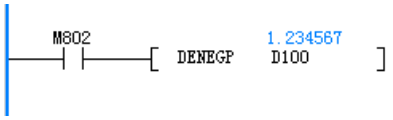
功能和指令说明

将[D+1，D]的2进制浮点数的符号取反，结果保存在[D+1，D]中。通常使用脉冲型指令。

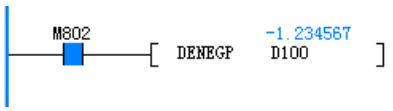
指令示例

将D100、D101里面的数据进行取反，存储于D100、D101。

- 指令执行前



- 指令执行后



3.5.3 字的位运算

3.5.3.1 指令列表

字的位运算指令涵盖以下指令条目：

指令类别	名称	功能
字的位运算	BLD	字或双字位数据触点指令
	BLDI	字或双字位数据反触点指令
	BAND	字或双字位数据与触点指令
	BANDI	字或双字位数据与非触点指令
	BOR	字或双字位数据或触点指令
	BORI	字或双字位数据或非触点指令
	BOUT	字或双字位数据输出指令
	BSET	字或双字位数据输出指令
	BRST	字或双字位数据输出指令

3.5.3.2 BLD

根据源数据指定位的状态ON(OFF)确定该节点状态ON(OFF)，与左母线直接相连的A节点。

BLD — 字或双字位数据位触点指令

16位指令	BLD 连续执行			
32位指令	DBLD 连续执行			
操作数	名称	描述	范围	数据类型
S	源数据	源数据软元件编号	-	INT/DINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

表3–45 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将字变量的指定bit位的状态作为触点输出。

指令示例

- n=3情况如下

- n=4情况如下

3.5.3.3 BLDI

根据源数据指定位的状态ON(OFF)确定该节点状态OFF(ON)，与左母线直接相连的B节点。

BLDI — 字或双字位数据位反触点指令

16位指令	BLDI 连续执行			
32位指令	DBLDI 连续执行			
操作数	名称	描述	范围	数据类型

S	源数据	源数据软元件编号	-	INT/DINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

表3–46 软元件列表

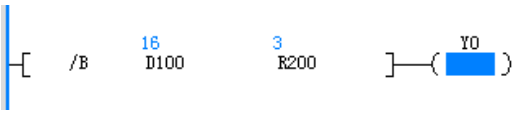
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

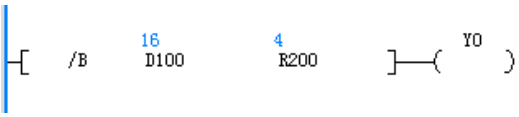
将字变量的指定bit位的状态作为触点输出。

指令示例

- n=3情况如下



- n=4情况如下



3.5.3.4 BAND

根据源数据指定位的状态ON(OFF)确定该节点状态ON(OFF)，与其它节点串联连接的A节点。

BAND — 字或双字位数据位与触点指令

16位指令	BAND 连续执行			
32位指令	DBAND 连续执行			
操作数	名称	描述	范围	数据类型
S	源数据	源数据软元件编号	-	INT/DINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

表3-47 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将字变量的指定bit位的状态作为触点输出。

指令示例

3.5.3.5 BANDI

根据源数据指定位的状态ON(OFF)确定该节点状态OFF(ON)，与其它节点串联连接的B节点。

BANDI — 字或双字位数据位与非触点指令

16位指令	BANDI 连续执行			
32位指令	DBANDI 连续执行			
操作数	名称	描述	范围	数据类型
S	源数据	源数据软元件编号	-	INT/DINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

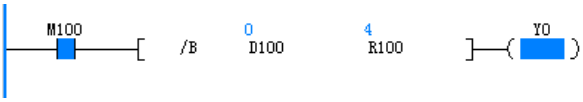
表3-48 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将字变量的指定bit位的状态作为触点输出。

指令示例



3.5.3.6 BOR

根据源数据指定位的状态ON(OFF)确定该节点状态ON(OFF)，与其它节点并联连接的A节点。

BOR — 字或双字位数据位或触点指令

16位指令	BOR 连续执行			
32位指令	DBOR 连续执行			
操作数	名称	描述	范围	数据类型
S	源数据	源数据软元件编号	-	INT/DINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

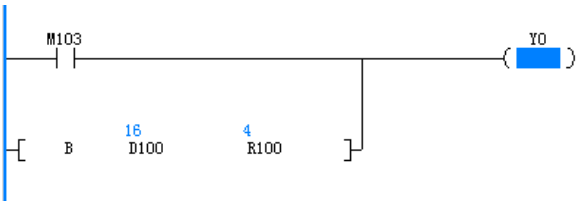
表3–49 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将字变量的指定bit位的状态作为触点输出。

指令示例



3.5.3.7 BORI

根据源数据指定位的状态ON(OFF)确定该节点状态OFF(ON)，与其它节点并联连接的B节点。

BORI — 字或双字位数据位或非触点指令

16位指令	BORI 连续执行			
32位指令	DBORI 连续执行			
操作数	名称	描述	范围	数据类型
S	源数据	源数据软元件编号	-	INT/DDINT
n	加载位	加载指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DDINT

说明

BLD/DBLD、BLDI/DBLDI、BAND/DBAND、BANDI/DBANDI、BOR/DBOR、BORI/DBORI指令，输入方式统一为“BLD/DBLD、BLDI/DBLDI”，由后台自动生成对应指令。

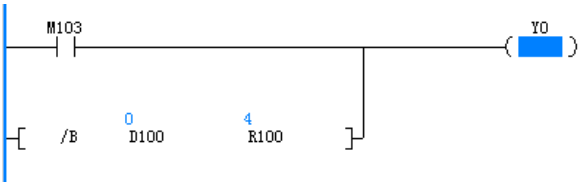
表3–50 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将字变量的指定bit位的状态作为触点输出。

指令示例



3.5.3.8 BOUT

将BOUT指令之前的逻辑运算结果输出至指定的位。

BOUT — 字或双字位数据输出指令

16位指令	BOUT 连续执行			
32位指令	DBOUT 连续执行			
操作数	名称	描述	范围	数据类型
D	输出数据	输出数据软元件编号	-	INT/DINT
n	输出位	输出指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

表3–51 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

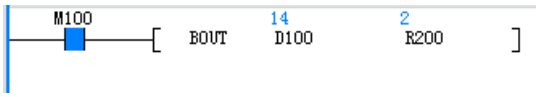
功能和指令说明

将BOUT指令之前的能流输出到字变量的指定bit位。

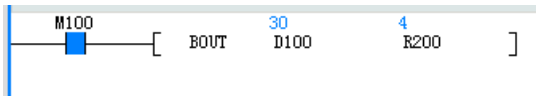
指令示例

D100初始值=2#1010（十进制K10）；

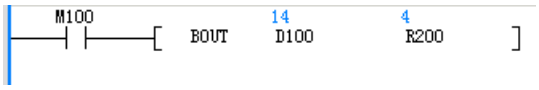
1. R200=2且M100=ON情况如下，D100的bit2被置位，结果D100=2#1110（十进制K14）。



2. R200=4且M100=ON情况如下，D100的bit4被置位，结果D100= 2#11110（十进制K30）。



3. 接着M100=OFF情况如下，D100的bit4被复位，结果D100= 2#1110（十进制K14）。



3.5.3.9 BSET

当BSET指令被驱动，其指定的位被设定为ON。

BSET — 字或双字位数据置位指令

16位指令	BSET 连续执行			
32位指令	DBSET 连续执行			
操作数	名称	描述	范围	数据类型
D	输出数据	输出数据软元件编号	-	INT/DINT
n	输出位	输出指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

表3–52 软元件列表

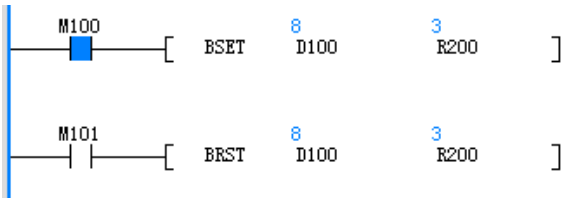
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

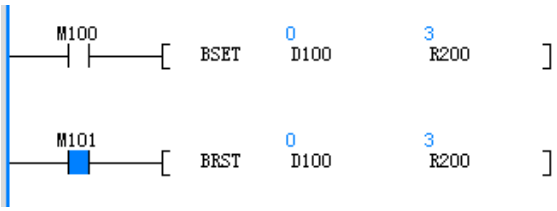
将字变量的指定bit位置位。

指令示例

- M100=ON时如下：



- M100=OFF时如下：



3.5.3.10 BRST

当BRST指令被驱动，其指定的位被设定为OFF。

BRST — 字或双字位数据输出指令

16位指令	BRST 连续执行			
32位指令	DBRST 连续执行			
操作数	名称	描述	范围	数据类型
D	输出数据	输出数据软元件编号	-	INT/DINT
n	输出位	输出指定位，范围为0-15(16位指令)或 0-31(32位指令)	0-15/31	INT/DINT

表3–53 软元件列表

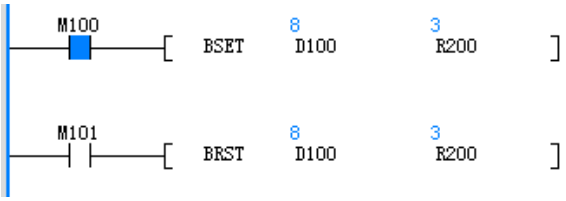
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

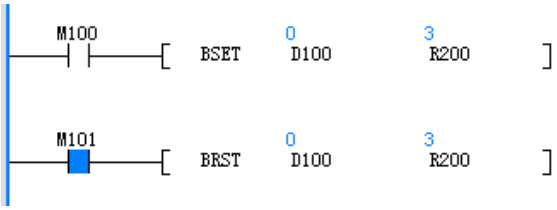
将字变量的指定bit位复位。

指令示例

- M100=ON时如下：



- M100=OFF时如下：



3.5.4 三角函数

3.5.4.1 指令列表

三角函数指令涵盖以下指令条目：

指令类别	名称	功能
三角函数	SIN	浮点SIN运算指令
	COS	浮点COS运算指令指令
	TAN	浮点TAN运算指令
	ASIN	二进制浮点数ARCSIN运算
	ACOS	二进制浮点数ARCCOS运算
	ATAN	二进制浮点数ARCTAN运算
	RAD	二进制浮点数角度→弧度的转换
	DEG	二进制浮点数弧度→角度的转换
	SINH	二进制浮点数SINH运算
	COSH	二进制浮点数COSH运算
	TANH	二进制浮点数TANH运算

3.5.4.2 SIN

求指定角度（RAD，弧度）的SIN（正弦）值，变量为二进制浮点存储格式。

SIN — 浮点SIN运算指令

16位指令	—			
32位指令	DSIN 连续执行/DSINP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求正弦值的角度变量，RAD单位，以二进制浮点数表示。取值范围0<=α<=2π	-	REAL
D	运算结果	变换后SIN计算结果的存储单元，二进制浮点数格式	-	REAL

表3–54 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是求指定角度（RAD，弧度）的SIN（正弦）值，变量为二进制浮点存储格式。其中：

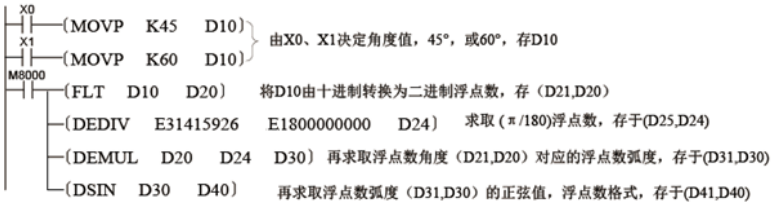
- S为待求正弦值的角度变量，RAD单位，以二进制浮点数表示。取值范围0≤α≤2π；
- D为变换后SIN计算结果的存储单元，二进制浮点数格式。

指令示例一

这里计算的源数据、SIN结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度360°对应的弧度＝360°×π/180°＝2π。

指令示例二



3.5.4.3 TAN

求指定角度（RAD，弧度）的TAN（正切）值，变量为二进制浮点存储格式。

TAN — 浮点TAN运算指令

16位指令	—			
32位指令	DTAN 连续执行/DTANP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求正切值的角度变量，RAD单位，以二进制浮点数表示。取值范围0≤α<2π	-	REAL
D	运算结果	变换后TAN计算结果的存储单元，二进制浮点数格式	-	REAL

表3–55 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是求指定角度（RAD，弧度）的TAN（正切）值，变量为二进制浮点存储格式。

指令示例

这里计算的源数据、TAN结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度360°对应的弧度＝360°×π/180°＝2π。

3.5.4.4 COS

求指定角度（RAD，弧度）的COS（余弦）值，变量为二进制浮点存储格式。

COS — 浮点COS运算指令指令

16位指令	—			
32位指令	DCOS 连续执行/DCOSP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求余弦值的角度变量，RAD单位，以二进制浮点数表示。取值范围0≤α≤2π	-	REAL
D	运算结果	变换后COS计算结果的存储单元，二进制浮点数格式	-	REAL

表3–56 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是求指定角度（RAD，弧度）的COS（余弦）值，变量为二进制浮点存储格式。

指令示例

这里计算的源数据、COS结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度360°对应的弧度＝360°×π/180°＝2π。

3.5.4.5 ASIN

根据SIN值求出对应弧度的运算。

ASIN 浮点数SIN^{-1}运算

16位指令	—			
32位指令	DASIN 连续执行/DASINP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求ARCSIN（反正弦）的2进制浮点数变量；	-	REAL
D	运算结果	为计算结果的存储单元(-л/2～л/2)	-	REAL

表3–57 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

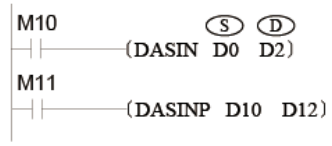
功能和指令说明

该指令是从SIN值求出对应弧度的运算。

说明

S中的值不在-1.0~1.0范围内时将发生运算错误。

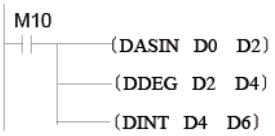
指令示例一



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行SIN $^{-1}$运算后保存到(D3、D2)中。

SIN $^{-1}$(D1、D0) ⟹ (D3、D2)

指令示例二



假设(D1、D0)为0.707106781，当M10由OFF到ON的时候，（D3、D2）内容为0.78539815，（D5、D4）的内容为45，（D7、D6）的内容为45。

3.5.4.6 ACOS

根据COS值求出对应弧度的运算。

ACOS — 浮点数COS^{-1}运算

16位指令	—			
32位指令	DACOS 连续执行/DACOSP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求ARCCOS（反余弦）的2进制浮点数变量	-	REAL
D	运算结果	计算结果的存储单元(0~л)	-	REAL

表3–58 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

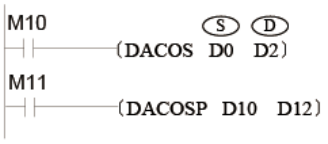
功能和指令说明

该指令是从COS值求出对应弧度的运算。

说明

S中的值不在-1.0~1.0范围内时将发生运算错误。

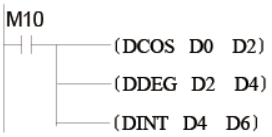
指令示例一



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行COS^{-1}运算后保存到(D3、D2)中。

COS^{-1}(D1、D0) ⟹ (D3、D2)

指令示例二



假设(D1、D0)为0.866025404，当M10由OFF到ON的时候，（D3、D2）内容为0.52359877，（D5、D4）的内容为30，（D7、D6）的内容为30。

3.5.4.7 ATAN

根据TAN值求出对应弧度的运算。

ATAN — 浮点数TAN^{-1}运算

16位指令	—			
32位指令	DATAN 连续执行/DATANP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求ARCTAN（反正切）的2进制浮点数变量	-	REAL
D	运算结果	计算结果的存储单元(-π/2~π/2)	-	REAL

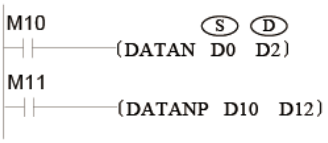
表3–59 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是从TAN值求出对应弧度的运算。

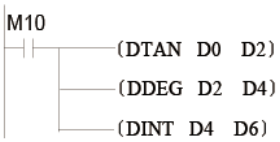
指令示例一



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行TAN^{-1}运算后保存到(D3、D2)中。

TAN^{-1}(D1、D0) ⟹ (D3、D2)

指令示例二



假设(D1、D0)为1.732050808，当M10由OFF到ON的时候，（D3、D2）内容为1.04719753，（D5、D4）的内容为60，（D7、D6）的内容为60。

3.5.4.8 RAD

进行2进制浮点数角度转换成弧度的运算。

RAD — 浮点数角度转换弧度

16位指令	—			
32位指令	DRAD 连续执行/DRADP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求弧度的2进制浮点数角度变量	-	REAL
D	运算结果	计算结果的存储单元	-	REAL

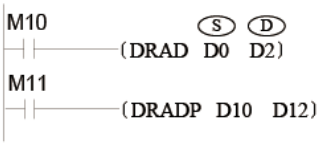
表3–60 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

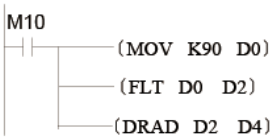
该指令是进行2进制浮点数角度转换成弧度的运算。其计算公式为[弧度单位=角度单位×л/180]。

指令示例一



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行角度到弧度的运算后保存到(D3、D2)中。

指令示例二



当M10由OFF到ON的时候，将90赋给D0，D0经过整数到浮点数的运算赋给（D3、D2），（D3、D2）经过角度到弧度的运算赋给（D5、D4），最后得（D3、D2）的值为π/2，即为1.570796。

3.5.4.9 DEG

进行2进制浮点数弧度转换成角度的运算。

DEG — 浮点数弧度转换角度

16位指令	—			
32位指令	DDEG 连续执行/DDEGP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求角度的2进制浮点数弧度变量	-	REAL
D	运算结果	计算结果的存储单元	-	REAL

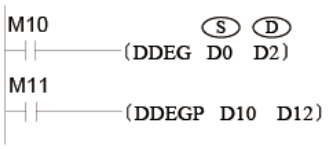
表3–61 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

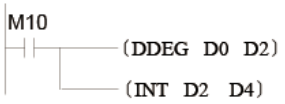
该指令是进行2进制浮点数弧度转换成角度的运算。其计算公式为[角度单位=弧度单位×180/π]。

指令示例一



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行弧度到角度的运算后保存到(D3、D2)中。

指令示例二



假设（D1、D0）为3.1415926，当M10由OFF到ON的时候，（D3、D2）为180，经过浮点数变整数的运算后得到（D5、D4）的值为180。

3.5.4.10 SINH

进行2进制浮点数取SINH值。

SINH — 浮点数SINH运算

16位指令	—			
32位指令	DSINH 连续执行/DSINHP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求SINH值的2进制浮点数变量	-	REAL
D	运算结果	计算结果的存储单元（运算结果D超出浮点数范围时会报错6706）	-	REAL

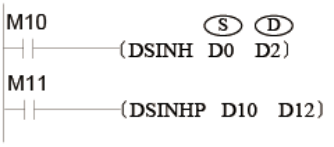
表3-62 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行2进制浮点数取SINH值。其计算公式为sinh值= （e^{s}- e^{-s})/ 2。

指令示例



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取SINH值后保存到(D3、D2)中。

3.5.4.11 COSH

进行2进制浮点数取COSH值。

COSH — 浮点数COSH运算

16位指令	—			
32位指令	DCOSH 连续执行/DCOSHP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求COSH值的2进制浮点数变量	-	REAL
D	运算结果	计算结果的存储单元	-	REAL

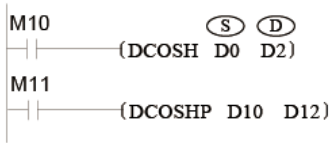
表3-63 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行2进制浮点数取COSH值。其计算公式为COSh值= （e^{s}+e^{-s})/ 2。

指令示例



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取COSH值后保存到(D3、D2)中。

3.5.4.12 TANH

进行2进制浮点数取TANH值。

TANH — 浮点数TANH运算

16位指令	—			
32位指令	DTANH 连续执行/DTANHP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	存放待求TANH值的2进制浮点数变量	-	REAL
D	运算结果	计算结果的存储单元	-	REAL

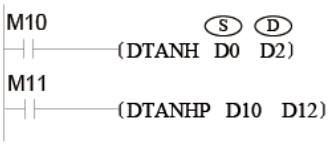
表3–64 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行2进制浮点数取TANH值。其计算公式为TANH值= （e^{s}- e^{-s} ）（/ e^{s}+e^{-s})。

指令示例



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取TANH值后保存到(D3、D2)中。

3.5.5 表格运算

3.5.5.1 指令列表

表格运算指令涵盖以下指令条目：

指令类别	名称	功能
表格运算	WSUM	算出数据合计值
	MEAN	平均值计算
	LIMIT	上下限限位控制
	BZAND	死区控制
	ZONE	区域控制
	SCL	定坐标(不同点坐标数据)
	SCL2	定坐标2(X/Y坐标数据)

3.5.5.2 WSUM

该指令可计算出连续的16位或是32位数据的合计值。

WSUM — 算出数据合计值

16位指令	WSUM 连续执行/WSUMP 脉冲执行			
32位指令	DWSUM 连续执行/DWSUMP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	保存要算出合计值的数据的软元件起始编号	-	INT/DINT，数组*n
D	结果	保存合计值的软元件起始编号	-	INT/DINT，数组*2
n	数据个数	数据个数	2-256	INT/DINT

表3–65 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

- 16位指令
  将[S]开始的n点16位数据的求和，结果以32位数据形式保存在[D+1，D]中。
- 32位指令
  将[S+1，S]开始的n点32位数据的求和，结果以64位数据形式保存在[D+3，D+2，D+1，D]中。

报错

以下一些情况会报故障：

如果[S]开始的n点软元件超出了其合法范围，报错误。

在存储数据时，如果[D]超出了其合法范围，报错误。

操作数n的有效范围：n≤0时，报错误。

指令示例

3.5.5.3 MEAN

当驱动条件成立时，求以S为起始的K个数据的平均值，结果存放到D。

MEAN — 平均值计算指令

16位指令	MEAN 连续执行/MEANP 脉冲执行
32位指令	DMEAN 连续执行/DMEANP 脉冲执行

操作数	名称	描述	范围	数据类型
S	数据首址	求平均值的数据存储字软元件首址	–	INT/DINT，数组*n
D	平均值	平均值数据存储字软元件地址	–	INT/DINT
n	数据长度	立即数	1-256	INT/DINT

表3–66 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

求取由S开始的N个变量的平均值（先求和，再除以n），存入D。

若计算中有余数，余数将被舍弃；

当n的值不在1~256的范围时，会计算出错。

指令示例

(D10+D11+D12+D13)/4=D20
假如D10=K5,D11=K5,D12=K15,D13=K52;则D20=K19.余数1被舍去

3.5.5.4 LIMIT

设置输入数值的上限值/下限值然后输出的指令。

LIMIT — 上下限限位控制

16位指令	LIMIT 连续执行/LIMITP 脉冲执行			
32位指令	DLIMIT 连续执行/DLIMITP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	下限值	下限限位值(最小输出界限值)	-	INT/DINT
S2	上限值	上限限位值(最大输出界限值)	-	INT/DINT
S3	输入值	需要通过上下限限位控制的输入值	-	INT/DINT
D	输出值	保存已经过上下限限位控制的输出值的软元件起始编号	-	INT/DINT

表3–67 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S3	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令

  通过在[S1]、[S2]中设定上下限值，使输出值[D]在一定范围内依据输入值[S3]进行输出。

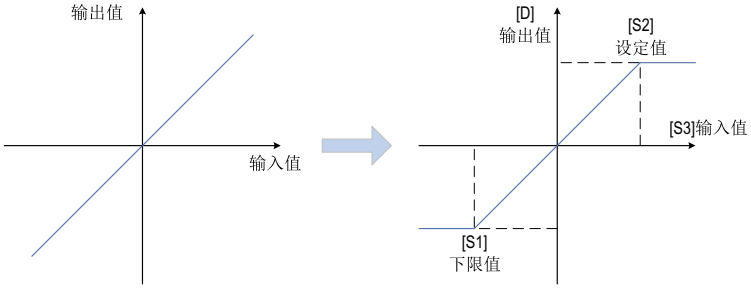
  当[S1]＞[S3]时，[S1]→[D]；

  当[S2]＜[S3]时，[S2]→[D]；

  当[S1]≤[S3]≤[S2]时，[S3]→[D]。

  如果仅控制上限的话，请在下限[S1]中设定16位有符号最小值，即-32768。

  如果仅控制下限的话，请在上限[S2]中设定16位有符号最大值，即32767。



- 32位指令

  通过在[S1+1，S1]、[S2+1，S2]中设定上下限值，使输出值[D+1，D]在一定范围内依据输入值[S3+1，S3]进行输出。

  当[S1+1，S1]＞[S3+1，S3]时，[S1+1，S1]→[D+1，D]；

  当[S2+1，S2]＜[S3+1，S3]时，[S2+1，S2]→[D+1，D]；

  当[S1+1，S1]≤[S3+1，S3]≤[S2+1，S2]时，[S3+1，S3]→[D+1，D]。

  如果仅控制上限的话，请在下限[S1+1，S1]中设定32位有符号最小值，即-2,147,483,648。

  如果仅控制下限的话，请在上限[S2+1，S2]中设定32位有符号最大值，即2,147,483,647。

以下情况会报故障：

在16位指令和32位指令中，下限值>上限值时会报错误。

指令示例

3.5.5.5 BZAND

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令。

BZAND — 死区控制

16位指令	BZAND 连续执行/BZANDP 脉冲执行			
32位指令	DBZAND 连续执行/DBZANDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	下限值	死区(无输出区域)的下限值	-	INT/DINT
S2	上限值	死区(无输出区域)的上限值	-	INT/DINT
S3	输入值	要通过死区控制的输入值	-	INT/DINT
D	输出值	保存经过死区控制的输出值的软元件编号	-	INT/DINT

表3–68 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
S3	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令

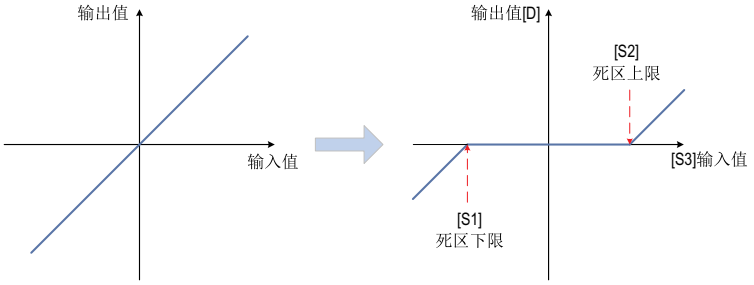
  通过在[S1]、[S2]中设定死区范围，使输入值[S3]在死区范围外输出至[D]软元件中。

  输出值如下所示被控制。

  当[S1]＞[S3]时，[S3]-[S1]→[D]；

  当[S2]＜[S3]时，[S3]-[S2]→[D]；

  当[S1]≤[S3]≤[S2]时，0→[D]。



- 32位指令

  通过在[S1+1，S1]、[S2+1,S2]中设定死区范围，使输入值[S3+1，S3]在死区范围外输出至[D+1，D]软元件中。

  当[S1+1，S1]＞[S3+1，S3]时，[S3+1，S3]-[S1+1，S1]→[D+1，D]；

  当[S2+1，S2]＜[S3+1，S3]时，[S3+1，S3]-[S2+1，S2]→[D+1，D]；

  当[S1+1，S1]≤[S3+1，S3]≤[S2+1，S2]时，0→[D+1，D]。

指令在使用过程中，数据的溢出情况符合循环处理，即最大值加1则变最小值，最小值减1则变最大值。

以下情况会报故障：

在16位指令和32位指令中，下限值>上限值时会报错误。

指令示例

3.5.5.6 ZONE

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令

ZONE — 区域控制

16位指令	ZONE 连续执行/ZONEP 脉冲执行			
32位指令	DZONE 连续执行/DZONEP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	负偏差	加在输入值上的负偏差值(可以正数、负数、0)	-	INT/DINT
S2	正偏差	加在输入值上的正偏差值(可以正数、负数、0)	-	INT/DINT
S3	输入值	要通过区域控制的输入值	-	INT/DINT
D	输出值	保存已通过区域控制的输出值的软元件起始编号	-	INT/DINT

表3–69 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
S3	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令

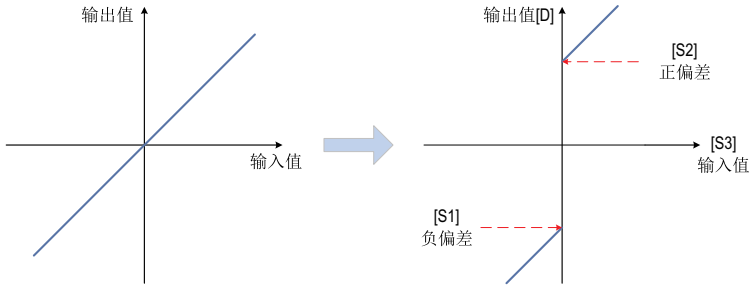
  依据输入值[S3]的符号确定，加上[S2]或[S1]，执行结果保存至[D]元件中。

  当[S3]＜0时，[S3]+[S1]→[D]；

  当[S3]＞0时，[S3]+[S2]→[D]；

  当[S3]=0时，0→[D]。

  如下所示执行。



- 32位指令
  依据输入值[S3+1，S3]的符号确定，加上[S2+1，S2]或 [S1+1，S1]，执行结果保存至[D+1，D]元件中。

指令示例

3.5.5.7 SCL

根据指定的数据表格，对输入值执行定坐标后输出的指令

SCL — 定坐标(不同点坐标数据)

16位指令	SCL 连续执行/SCLP 脉冲执行			
32位指令	DSCL 连续执行/DSCLP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	输入值	执行定坐标的输入值或是保存输入值的软元件编号	-	INT/DINT
S2	表格数据	定坐标用的转换表格软元件的起始编号	1-256	INT/DINT，数组*不定
D	输出值	保存被定坐标控制的输出值的软元件编号	-	INT/DINT

表3–70 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令
  依据[S2]中表格确定的图形，找到输入值[S1]对应的输出值[D]，若输出值不是整数值时，小数第1位四舍五入输出。

  指令实现如下所示：

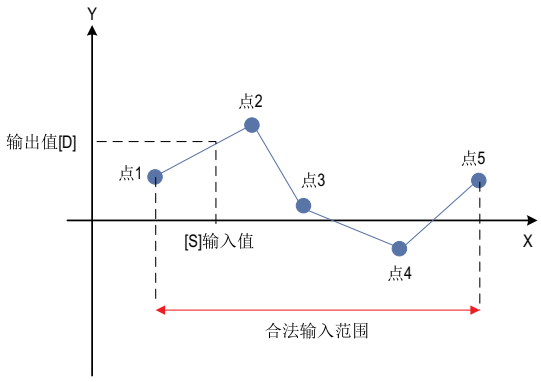


表3–71 16位指令的[S2]数据组织形式

设定项目		设定数据表格的软元件分配
坐标点数假设为5		[S2]
点1	X坐标	[S2+1]
	Y坐标	[S2+2]
点2	X坐标	[S2+3]
	Y坐标	[S2+4]
点3	X坐标	[S2+5]
	Y坐标	[S2+6]
点4	X坐标	[S2+7]
	Y坐标	[S2+8]
点5	X坐标	[S2+9]
	Y坐标	[S2+10]

- 32位指令

  依据[S2+1，S2]中表格确定的图形，找到输入值[S1+1，S1]对应的输出值[D+1，D]，若输出值不是整数值时，小数第1位四舍五入输出。

表3–72 32位指令的[S2]数据组织形式

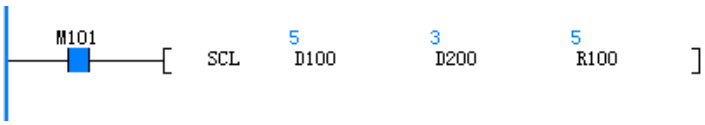
设定项目		设定数据表格的软元件分配
坐标点数假设为5		[S2+1,S]
点1	X坐标	[S2+3,S2+2]
	Y坐标	[S2+5,S2+4]
点2	X坐标	[S2+7,S2+6]
	Y坐标	[S2+9,S2+8]
点3	X坐标	[S2+11,S2+10]
	Y坐标	[S2+13,S2+12]
点4	X坐标	[S2+15,S2+14]
	Y坐标	[S2+17,S2+16]
点5	X坐标	[S2+19,S2+18]
	Y坐标	[S2+21,S2+20]

以下情况会报故障：

表格数据的x坐标未按照升序进行排列，报错误。

[S1]在表格数据设定的范围之外时，报错误。

指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	5
D200	16位整数	十进制	3
D201	16位整数	十进制	0
D202	16位整数	十进制	0
D203	16位整数	十进制	10
D204	16位整数	十进制	10
D205	16位整数	十进制	20
D206	16位整数	十进制	0
	16位整数	十进制	
R100	16位整数	十进制	5
	16位整数	十进制	

3.5.5.8 SCL2

根据指定的数据表格，对输入值执行定坐标后输出的指令。

SCL2 — 定坐标2(X/Y坐标数据)

16位指令	SCL2 连续执行/SCL2P 脉冲执行			
32位指令	DSCL2 连续执行/DSCL2P 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	输入值	执行定坐标的输入值或是保存输入值的软元件编号	-	INT/DINT
S2	表格数据	定坐标用的转换表格软元件的起始编号	1-256	INT/DINT，数组*不定
D	输出值	保存被定坐标控制的输出值的软元件编号	-	INT/DINT

表3–73 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令
  依据[S2]中表格确定的图形，找到输入值[S1]对应的输出值[D]，若输出值不是整数值时，小数第1位四舍五入输出。

  指令实现如下所示：

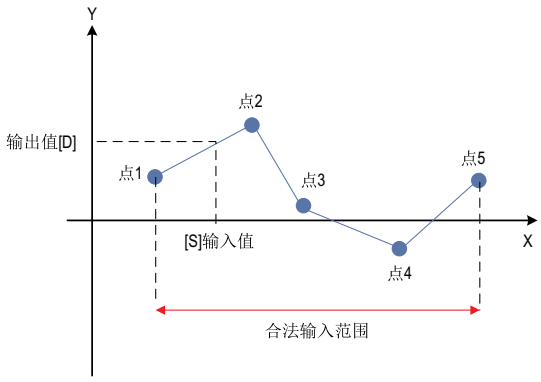


表3–74 16位指令的[S2]数据组织形式

设定项目		设定数据表格的软元件分配
坐标点数假设为5		[S2]
X坐标	点1	[S2+1]
	点2	[S2+2]
	点3	[S2+3]
	点4	[S2+4]
	点5	[S2+5]
Y坐标	点1	[S2+6]
	点2	[S2+7]
	点3	[S2+8]
	点4	[S2+9]
	点5	[S2+10]

- 32位指令

  依据[S2+1，S2]中表格确定的图形，找到输入值[S1+1，S1]对应的输出值[D+1，D]，若输出值不是整数值时，小数第1位四舍五入输出。

表3–75 32位指令的[S2+1、S2]数据组织形式

设定项目		设定数据表格的软元件分配
坐标点数假设为5		[S2,S2+1]
X坐标	点1	[S2+3,S2+2]
	点2	[S2+5,S2+4]
	点3	[S2+7,S2+6]
	点4	[S2+9,S2+8]
	点5	[S2+11,S2+10]
Y坐标	点1	[S2+13,S2+12]
	点2	[S2+15,S2+14]
	点3	[S2+17,S2+16]
	点4	[S2+19,S2+18]
	点5	[S2+21,S2+20]

以下情况会报故障：

表格数据的x坐标未按照升序进行排列，报错误。

[S1]在表格数据设定的范围之外时，报错误。

指令示例

3.5.6 指数运算

3.5.6.1 指令列表

指数运算指令涵盖以下指令条目

指令类别	名称	功能
指数运算	EXP	二进制浮点数指数运算
	LOGE	二进制浮点数自然对数运算
	LOG	二进制浮点数以10为低的对数运算
	ESQR	二进制浮点开方运算
	SQR	二进制数据开方运算
	POW	浮点数权值指令

3.5.6.2 EXP

进行以e(2.71828)为底的二进制浮点数数据的指数运算。

EXP — 浮点数指数运算

16位指令	—			
32位指令	DEXP 连续执行/DEXPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求二进制浮点数指数的2进制浮点数变量	-	REAL
D	运算结果	指数运算后计算结果的存储单元	-	REAL

表3–76 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

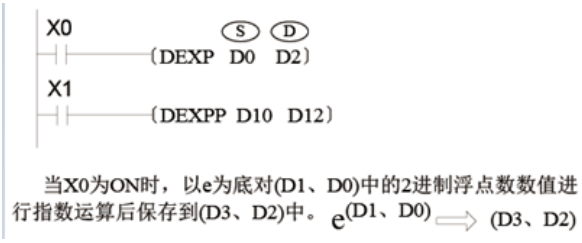
该指令是进行以e(2.71828)为底的二进制浮点数数据的指数运算。其中：

- S为待求二进制浮点数指数的2进制浮点数变量；
- D为指数运算后计算结果的存储单元。

说明

当运算结果不在 [2^{-126} ≤运算结果＜2^{128}]时将发生运算出错。

指令示例



3.5.6.3 LOG

进行以10为底的二进制浮点数数据的常用对数运算。

LOG — 浮点数常用对数运算

16位指令	—			
32位指令	DLOG 连续执行/DLOGP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求二进制浮点数常用对数的2进制浮点数变量	-	REAL
D	运算结果	常用对数运算后计算结果的存储单元	-	REAL

表3–77 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

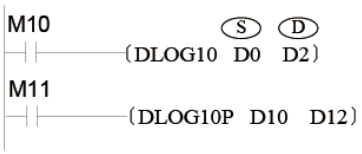
该指令是进行以10为底的二进制浮点数数据的常用对数运算。其中：

- S为待求二进制浮点数常用对数的2进制浮点数变量；
- D为对数运算后计算结果的存储单元。

说明

S中的值只能为正数，当S的内容值为0或者负数时将发生运算出错。

指令示例



当M10为ON时，以10为底对(D1、D0)中的2进制浮点数数值进行常用对数运算后保存到(D3、D2)中。

$\log_{10}$(D1、D0) ⟹ (D3、D2)

3.5.6.4 LOGE

进行以e(2.71828)为底的二进制浮点数数据的自然对数运算。

LOGE — 浮点数自然对数运算

16位指令	—			
32位指令	DLOGE 连续执行/DLOGEP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求二进制浮点数自然对数的2进制浮点数变量	-	REAL
D	运算结果	自然对数运算后计算结果的存储单元	-	REAL

表3–78 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

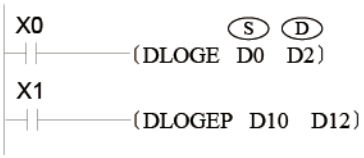
该指令是进行以e(2.71828)为底的二进制浮点数数据的自然对数运算。其中：

- S为待求二进制浮点数对数的2进制浮点数变量；
- D为对数运算后计算结果的存储单元。

说明

S中的值只能为正数，当S的内容值为0或者负数时将发生运算出错。

指令示例



当X0为ON时，以e为底对(D1、D0)中的2进制浮点数数值进行自然对数运算后保存到(D3、D2)中。

$\log_{e}$(D1、D0) ⟹ (D3、D2)

从自然对数向常用对数的转换（用0.4342945分割常用对数的值）公式如下：

$$10^{x}=e^{\frac{x}{0.4342945}}$$

3.5.6.5 ESQR

进行二进制浮点的开平方运算，即求二进制浮点数的平方根。

ESQR — 浮点开方指令

16位指令	—			
32位指令	DESQR 连续执行/DESQRP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	待求平方根的二进制浮点数变量	-	REAL
D	运算结果	二进制浮点平方根的存储单元	-	REAL

表3-79 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

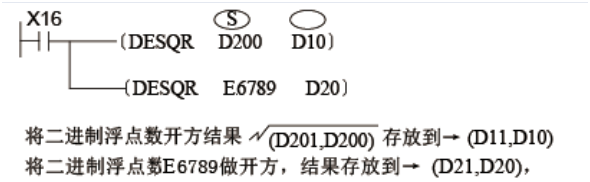
功能和指令说明

该指令是进行二进制浮点的开平方运算，即求二进制浮点数的平方根。

若计算结果为零，则0标志（M8020）会置位。

S只有正数有效，如果是负数则计算出错。

指令示例



3.5.6.6 SQR

整数的开方运算。

SQR — 开方运算指令

16位指令	SQR 连续执行/SQRP 脉冲执行			
32位指令	DSQR 连续执行/DSQRP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	将开方的数据或数据存储字软元件地址	–	INT/DINT
D	运算结果	开方结果数据存储字软元件地址	–	INT/DINT

表3-80 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

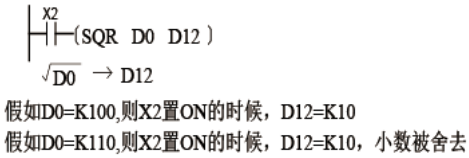
将S按BIN值开平方运算，结果存入D。

只能指定S为正数，如S为负数则运算错误标志M8067会置ON，指令不被执行；

运算结果D只取整数。舍去小数点，有小数点被舍去时借位标志M8021置ON；

运算结果是0时，零位标志M8020置ON。

指令示例



3.5.6.7 POW

将二进制浮点数据[S1+1,S1]及[S2+1,S2]以次幂数相乘后存放于[D+1,D]中。

POW — 浮点数权值指令

16位指令	—			
32位指令	DPOW 连续执行/DPOWP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	底数	底数软元件起始地址，取值非0。	-	REAL
S2	次幂	次幂数软元件起始地址	-	REAL
D	结果	运算结果软元件起始地址	-	REAL

表3-81 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

运算时均以浮点数格式执行，故[S1]、[S2]需转换为浮点数值。

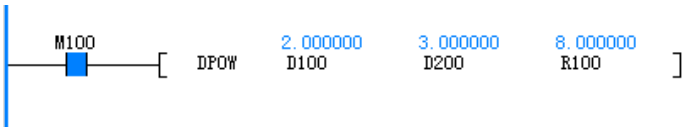
1）若运算结果的绝对值大于可表示的最大浮点值，则进位标志M8022=ON。

2）若运算结果的绝对值小于可表示的最小浮点值，则借位标志M8021=ON。

3）若运算结果为0，则零标志M8020=ON。

指令示例

已知[S1]=2，[S2]=3， 则[D]=2^3=8。



3.6 数据处理指令

3.6.1 数据转换

3.6.1.1 指令列表

数据转换指令涵盖以下指令条目：

指令类别	名称	功能
数据转换	INT	二进制浮点→BIN整数的转换
	BCD	二进制数据转换BCD数据
	BIN	BCD数据转换二进制数据
	FLT	二进制数据→二进制浮点数转换
	EBCD	二进制浮点→十进制浮点转换
	EBIN	十进制浮点→二进制浮点转换
	DABIN	10进制ASCII→BIN的转换
	BINDA	BIN→10进制ASCII的转换
	WTOB	字节单位的数据分离
	BITW	位元件赋值给字元件
	BTOW	字节单位的数据结合
	WBIT	字元件赋值给位元件
	WTODW	16位字元件转换成32位字元件
	DWTOW	32位字元件转换成16位字元件
	UNI	16位数据的4位结合
	DIS	16位数据的4位分离
	ASCI	HEX→ASCII转换
	HEX	ASCII→HEX转换

3.6.1.2 INT

进行二进制浮点的取整运算，丢弃小数部分，将二进制结果存于D中。

INT — 浮点→整数转换指令

16位指令	INT 连续执行/INTP 脉冲执行
32位指令	DINT 连续执行/DINTP 脉冲执行

操作数	名称	描述	范围	数据类型
S	数据源	待取整变换的二进制浮点数变量	-	REAL，固定为32位
D	运算结果	变换后BIN整数结果的存储单元	-	INT/DINT

表3–82 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

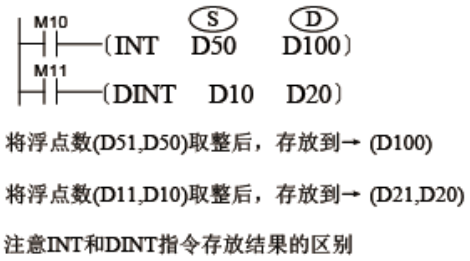
该指令是进行二进制浮点的取整运算，丢弃小数部分，将二进制结果存于D中。

- S=0时，M8020会置位。
- ｜S｜≤1时，借位标志（M8021）会置位。

若运算结果若超出下列范围时（溢位），则进位标志（M8022）会置位。

- 16位指令：-32,768~32,767
- 32位指令：-2,147,483,648~2，147，483，647

指令示例



3.6.1.3 BCD

二进制数据转换BCD数据。

BCD — 二进制数据转BCD指令

16位指令	BCD 连续执行/BCDP 脉冲执行			
32位指令	DBCD 连续执行/DBCDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	二进制码数据或数据存储字软元件地址	–	INT/DINT
D	转换结果	BCD码数据存储字软元件地址	–	INT/DINT

表3–83 软元件列表

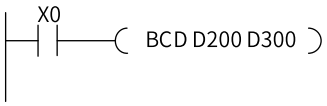
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

需要触点驱动，有2个操作变量，将S（BIN）的值进行BCD变换后存入D中。该指令常用于将数据显示前的数据格式处理。

- 使用16bit指令时，范围是0-9999，当转换结果超过9999时会出错；
- 使用32bit指令时，范围是0-99999999，当转换结果超过99999999时会出错。

指令示例



将D200的BIN值转换成BCD值后，将结果存放于D300中。
若D200=H000E（十六进制）=K14（十进制），则变换后D300的值为10100（BIN）

3.6.1.4 BIN

BCD数据转换二进制数据。

BIN— BCD转二进制数据指令

16位指令	BIN 连续执行/BINP 脉冲执行			
32位指令	DBIN 连续执行/DBINP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	BCD码数据或数据存储字软元件地址	–	INT/DINT
D	转换结果	二进制码数据存储字软元件地址	–	INT/DINT

表3–84 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

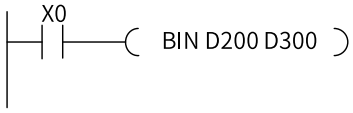
功能和指令说明

需要触点驱动，有2个操作变量，将S(BCD)的值进行BIN变换后存入D中。该指令常用于将外部端口读入数据（如编码盘设置）处理成能直接用于运算的BIN格式。

S（BCD）的有效范围，16bit：0~9999；32bit：0~99，999，999

S的数据内容不是BCD值（以Hex表示有任一位数不在0~9的范围内）时将会产生运算错误。

指令示例



将D200的BCD值转换成BIN值后，
将结果存放于D300中。

3.6.1.5 FLT

把二进制整数转化为二进制浮点数。

FLT — 二进制整数至浮点数转换指令

16位指令	FLT 连续执行/FLTP 脉冲执行			
32位指令	DFLT 连续执行/DFLTP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	整数	要转化的二进制数或数据存储字软元件地址	–	INT/DINT
D	浮点数	浮点数数据存储字软元件地址	–	REAL，固定为32位

表3–85 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√[1]	√	-	-
D	-	-	-	√	√	√[1]	-	-	-

说明

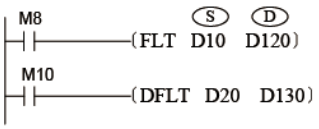
[1]支持指针变量元件，不支持C/T元件。

功能和指令说明

将整数S转换为浮点数，结果存入D和D＋1单元。

这个指令的逆变换指令是INT（将2进浮点数值变换成BIN整数）

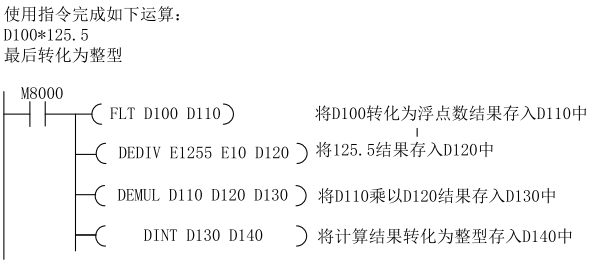
指令示例一



当M8=ON时，将16bit数D10中(16位BIN整数)转换为二进制浮点数后，存放到（D121,D120）

当M10=ON时，将32bit数（D21,D20）中(32位BIN整数)转换为二进制浮点数后，存放到（D131,D130）

指令示例二



3.6.1.6 EBCD

EBCD — 浮点→十进制浮点转换指令

16位指令	—			
32位指令	DEBCD 连续执行/DEBCDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	二进制浮点变量	–	REAL
D	运算结果	十进制浮点数结果的存放单元	–	DINT

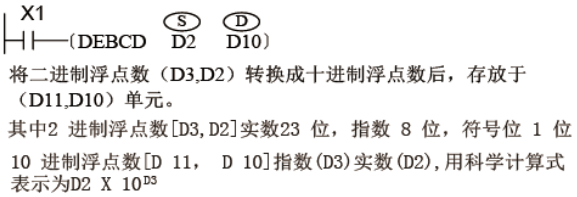
表3–86 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是进行二进制浮点数转换为十进制浮点的运算。

指令示例



PLC内部浮点数据计算均为二进制形式，转换为十进制方便监控。

3.6.1.7 EBIN

EBIN — 十进制浮点→浮点转换指令

16位指令	—			
32位指令	DEBIN 连续执行/DEBINP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	十进制浮点变量	–	DINT
D	结果	二进制浮点数结果的存放单元	–	REAL

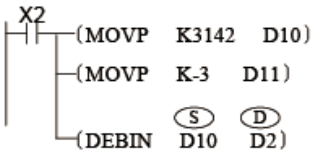
表3-87 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是进行十进制浮点数转换为二进制浮点的运算。

指令示例



将十进制浮点数3.142（先放于D11，D10）转换成二进制浮点数后，存放于（D3，D2）单元。

3.6.1.8 DABIN

将以10进制数字的ASCII码(30H～39H)形式现实的数据转换成BIN数据的指令。

DABIN – 10进制ASCII→BIN的转换

16位指令	DABIN 连续执行/DABINP 脉冲执行			
32位指令	DDABIN 连续执行/DDABINP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	输入值	保存要转换成BIN值的数据(ASCII码)的软元件起始编号	-	INT/DINT，数组*3
D	输出值	保存转换结果的软元件编号	-	INT/DINT

表3-88 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

16位指令

将存储于[S]~[S+2]的10进制ASCII码（30H~39H）数据转换成16位数据（BIN），存储于[D]中。

[S]～[S+2]的数值范围为-32768～32767。

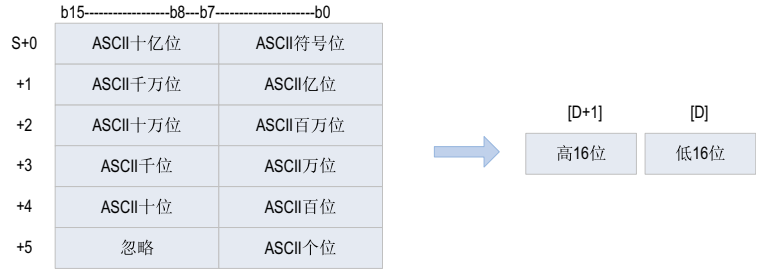
要转换的数据为正时设定符号位(最低字节)为“20H(空格)”，为负时设定“2DH(-)”。

各位数的ASCII码为30H～39H的范围。

各位数的ASCII码为“20H(空格)”、“00H(NULL)”时，作为“30H”处理。

32位指令

将存储于[S]~[S+5]的10进制ASCII码（30H~39H）数据转换成32位数据（BIN），存储于[D+1、D]中。



[S]～[S]+5的数值范围为-2,147,483,648～2,147,483,647，忽略[S+5]的高字节。

各位数的ASCII码为“30H”～“39H”的范围。

各位数的ASCII码为“20H(空格)”、“00H(NULL)”时，作为“30H”处理。

报错

以下一些情况下会发生运算错误。

- 符号位为“20H(空格)”或者“2DH(-)”以外的值，报错误。
- 数据位的ASCII码为“30H”～“39H”、“20H(空格)”、“00H(NULL)”以外的值，报错误。
- 转换数据超出了16位或32位有符号数范围，报错误。
- 16位指令[S+2]或32位指令[S+5]超相应的软元件范围，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x202D
D101	16位整数	十六进制	0x3220
D102	16位整数	十六进制	0x3637
	16位整数	十进制	
R100	16位整数	十进制	-276

3.6.1.9 BINDA

将BIN数据转换成ASCII码(30H～39H)的指令。

BINDA — BIN→10进制ASCII的转换

16位指令	BINDA 连续执行/BINDAP 脉冲执行			
32位指令	DBINDA 连续执行/DBINDAP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	输入值	保存要转换成ASCII码的BIN值的软元件编号	-	INT/DINT，数组*4
D	输出值	保存转换结果的软元件编号	-	INT/DINT

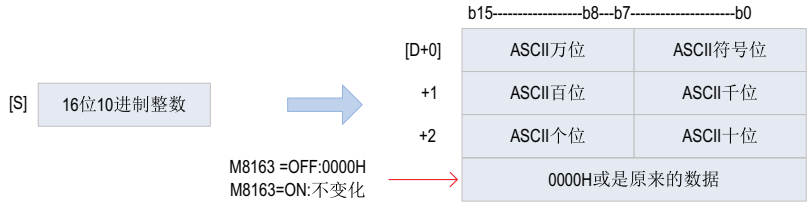
表3-89 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

16位指令

将[S]的16位数据(BIN)按照10进制的各个位数转换成ASCII码(30H～39H)，然后保存到[D]开始的软元件中。



16位数据 [S]的数值范围为-32768～32767。

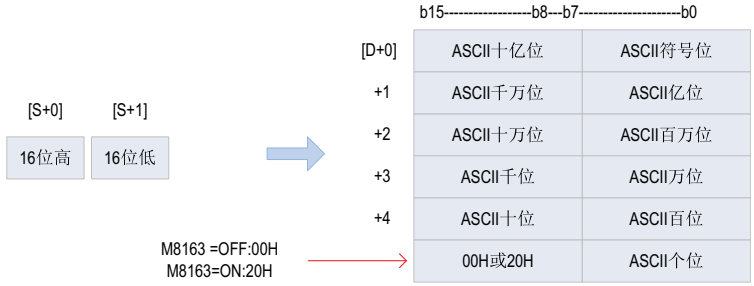
运算结果如下：
16位数据为正时，符号位保存“20H(空格)”，为负时保存“2DH(-)”。

有效位数的左侧的0时保存“20H(空格)”。

根据M8163的ON/OFF状态确定[D+3]的值。

32位指令

将的32位数据(BIN)按照10进制的各个位数转换成ASCII码(30H～39H)，然后保存到[D]开始的软元件中。



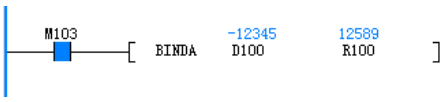
32位数据(BIN)[S+1、S]的数值范围为-2,147,483,648~2,147,483,647。

运算结果如下：
32位数据为正时，符号位保存“20H(空格)”，为负时保存“2DH(-)”。

有效位数的左侧的0时保存“20H(空格)”。

根据M8163的ON/OFF状态确定[D+3]的值。

指令示例



3.6.1.10 WBIT

字元件赋值给位元件组合的指令。

WBIT — 字元件赋值给位元件指令

16位指令	WBIT 连续执行/WBITP 脉冲执行			
32位指令	DWBIT 连续执行/DWBITP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	待赋值给位元件的组合值	-	INT/DINT
D	位元件	位元件起始元件	-	INT/DINT，数组*n
n	位元件个数	位元件数量	1-16/32	INT/DINT

表3–90 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	√[1]	√	√	-	-	-	-	-	-
n	-	-	-	√	√	√	√	-	-

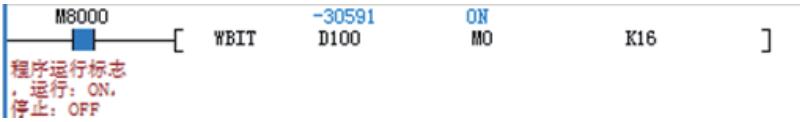
说明

[1]不支持X元件；

功能和指令说明

将S里的二进制数转化为位状态赋值到D起始的n个bit位中。

指令示例



	元件名称	数据类型	显示格式	当前值
1	M0	BOOL	二进制	ON
2	M1	BOOL	二进制	OFF
3	M2	BOOL	二进制	OFF
4	M3	BOOL	二进制	OFF
5	M4	BOOL	二进制	OFF
6	M5	BOOL	二进制	OFF
7	M6	BOOL	二进制	OFF
8	M7	BOOL	二进制	ON
9	M8	BOOL	二进制	OFF
10	M9	BOOL	二进制	OFF
11	M10	BOOL	二进制	OFF
12	M11	BOOL	二进制	ON
13	M12	BOOL	二进制	OFF
14	M13	BOOL	二进制	OFF
15	M14	BOOL	二进制	OFF
16	M15	BOOL	二进制	ON
17	D100	INT	二进制	2#1000100010000001

3.6.1.11 UNI

将连续的16位数据的低4位结合在一起的指令

UNI – 16位数据的4位结合

16位指令	UNI 连续执行/UNIP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	保存要结合的数据的软元件起始编号	-	INT
D	结果	保存已结合的数据的软元件编号	-	INT
n	结合数	结合数(0～4，n=0时不处理)	0,1-4	INT

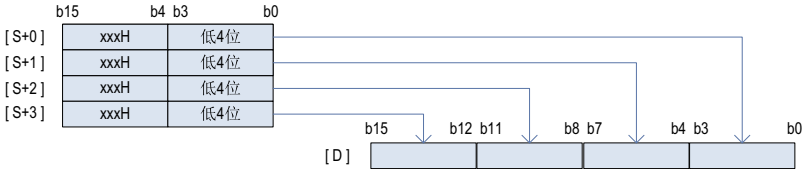
表3–91 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将S起始的n个16位数据的低4位结合成16位数据保存到D中。

n的取值范围1~4。n=0时，指令不执行。n取值1~3时，剩余高位0填充。



以下一些情况下会发生运算错误：

- S中设定的软元件超范围时，报错误；
- n设定超范围时，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1111
D101	16位整数	十六进制	0x2222
D102	16位整数	十六进制	0x3333
D104	16位整数	十六进制	0x0
	16位整数	十六进制	
D120	16位整数	十六进制	0x321
	16位整数	十六进制	

3.6.1.12 DWTOW

将32位字元件的值赋值给16位字元件的指令。

DWTOW – 32位字元件转换16位字元件指令

16位指令	—			
32位指令	WTODW 连续执行/WTODWP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	双字元件	双字元件起始元件	-	数组*n
D	字元件	字元件起始元件	-	数组*n
n	元件个数	元件数量	0，1-256	INT/DINT

表3–92 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

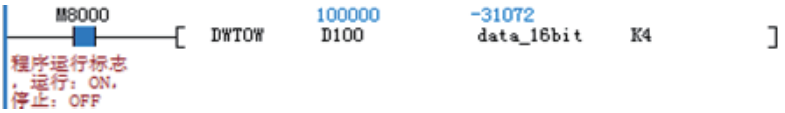
功能和指令说明

将S起始的n个32位数据转换成16位数据存放到D起始的n个16位寄存器中，该指令用于将32位数据进行转换，转换之后的数据可在16位指令中使用。

说明

使用该指令时，当数值大于16位上限时，转换之后只保留低位。

指令示例



	元件名称	数据类型	显示格式	当前值
1	D100	DINT	十进制	100000
2	D102	DINT	十进制	-10000
3	D104	DINT	十进制	20000
4	D106	DINT	十进制	30000
5	data_16bit[0]	INT	十进制	-31072
6	data_16bit[1]	INT	十进制	-10000
7	data_16bit[2]	INT	十进制	20000
8	data_16bit[3]	INT	十进制	30000

3.6.1.13 DIS

将16位数据以4位为单位分离的指令。

DIS – 16位数据的4位分离

16位指令	DIS 连续执行/DISP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	保存要分离的数据的软元件起始编号	-	INT

D	结果	保存已分离的数据的软元件编号	-	INT
n	分离数	分离数(0～4，n=0时不处理)	0,1-4	INT

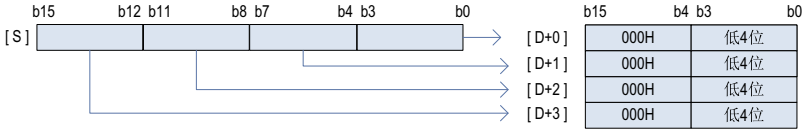
表3–93 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将S的16位数据以4位为单位进行分离后存储于D开始的软元件低4位中，高12位以0填充。

n的取值范围为1～4。n=0时，不执行指令的处理。

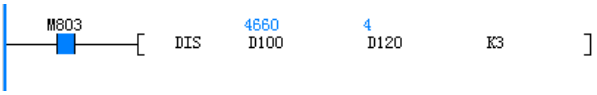


以下一些情况下会发生运算错误：

- D中设定的软元件超范围时，报错误；
- n设定超范围时，报错误。

指令示例

将D100的16位数据以4位为单位进行分离，存储于D120开始的3个连续D元件内。



3.6.1.14 BTOW

将连续的16位/32位数据的低8位(低字节)结合在一起的指令。

BTOW — 字节单位的数据结合

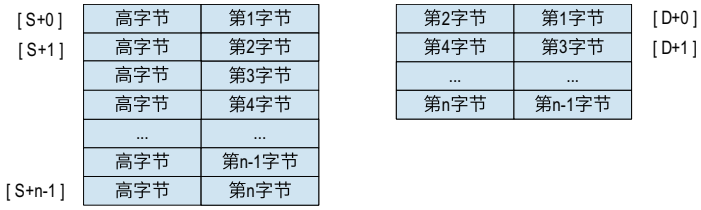
16位指令	BTOW 连续执行/BTOWP 脉冲执行			
32位指令	BTODW 连续执行/BTODWP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	保存要按照字节单位进行结合的数据的软元件起始编号	-	INT，数组*n
D	结果	保存已经按照字节单位结合的结果的软元件起始编号	-	INT/DINT，数组*n/2
n	结合数	要结合的字节数据个数(n≥0，且n=0时不处理)	0，1-256	INT/DINT

表3-94 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	✓	✓	-	-	-	-
D	-	-	-	✓	✓	-	-	-	-
n	-	-	-	✓	✓	✓	✓	-	-

功能和指令说明

将[S]开始的n个16位数据的低8位结合在一起后的16位/32位数据保存到[D]开始的软元件中，[S]的高8位被忽略。



当[S]、[D]开始的软元件超出其软元件范围时，会报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x5566
D103	16位整数	十六进制	0x7788
D104	16位整数	十六进制	0x99AA
D105	16位整数	十六进制	0xBBCC
D120	16位整数	十六进制	0x4422
D121	16位整数	十六进制	0x8866
D122	16位整数	十六进制	0xCCAA
D123	16位整数	十六进制	0x0

3.6.1.15 WTOB

将连续的16位/32位数据按照字节(8位)单位进行分离的指令。

WTOB — 字节单位的数据分离

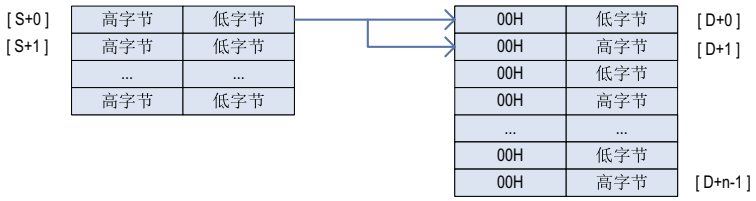
16位指令	WTOB 连续执行/WTOBP 脉冲执行			
32位指令	DWTOB 连续执行/DWTOBP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	保存要按照字节单位进行分离的数据的软元件起始编号	-	INT，数组*n/2
D	结果	保存已经按照字节单位分离的结果的软元件起始编号	-	INT/DINT，数组*n
n	分离数	要分离的字节数据个数(n≥0，且n=0时不处理)	0，1-256	INT INT/DINT

表3-95 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将[S]开始的软元件中保存的16位/32位数据以字节为单位保存到以[D]开始的n点软元件的低8位中，高8位保存00H。



当[S]、[D]开始的软元件超出其软元件范围时，会报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x5566
D120	16位整数	十六进制	0x22
D121	16位整数	十六进制	0x11
D122	16位整数	十六进制	0x44
D123	16位整数	十六进制	0x33
D124	16位整数	十六进制	0x66
D125	16位整数	十六进制	0x55
	16位整数	十进制	

3.6.1.16 BITW

位元件组合赋值给字元件的指令。

BITW — 位元件赋值给字元件指令

16位指令	BITW 连续执行/BITWP 脉冲执行			
32位指令	BITDW 连续执行/BITDWP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	位元件	位元件起始元件	-	INT/DINT，数组*n
D	目标数据	位元件组合值	-	INT/DINT
n	位元件个数	位元件数量	1-16/32	INT/DINT

表3-96 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	-	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将S起始的n个bit位转换成二进制数传送到D中。

指令示例

	元件名称	数据类型	显示格式	当前值
1	M0	BOOL	二进制	ON
2	M1	BOOL	二进制	OFF
3	M2	BOOL	二进制	OFF
4	M3	BOOL	二进制	OFF
5	M4	BOOL	二进制	OFF
6	M5	BOOL	二进制	OFF
7	M6	BOOL	二进制	OFF
8	M7	BOOL	二进制	ON
9	M8	BOOL	二进制	OFF
10	M9	BOOL	二进制	OFF
11	M10	BOOL	二进制	OFF
12	M11	BOOL	二进制	ON
13	M12	BOOL	二进制	OFF
14	M13	BOOL	二进制	OFF
15	M14	BOOL	二进制	OFF
16	M15	BOOL	二进制	ON
17	D100	INT	二进制	2#1000100010000001

3.6.1.17 WTODW

将16位字元件的值赋值给32位字元件的指令。

WTODW — 16位字元件转换32位字元件指令

16位指令	—			
32位指令	WTODW 连续执行/WTODWP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	字元件	字元件起始元件	-	数组*n
D	双字元件	双字元件起始元件	-	数组*n
n	元件个数	元件数量	0，1-256	INT/DINT

表3-97 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将S起始的n个16位数据转换成32位数据存放到D起始的n个32位寄存器中，该指令用于将16位数据进行转换，转换之后的数据可在32位指令中使用。

指令示例

	元件名称		数据类型	显示格式	当前值
1	...	D100	INT	十进制	10000
2	...	D101	INT	十进制	20000
3	...	D102	INT	十进制	30000
4	...	D103	INT	十进制	-20000
5	...	data_32bit[0]	DINT	十进制	10000
6	...	data_32bit[1]	DINT	十进制	20000
7	...	data_32bit[2]	DINT	十进制	30000
8	...	data_32bit[3]	DINT	十进制	-20000

3.6.1.18 ASCI

将S的值转换成ASCII码后，存储到D为起始地址的变量中。

ASCI — HEX→ASCII转换指令

16位指令	ASCI 连续执行/ASCIP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	数据源	待转换的变量地址或常数数值	–	INT，数组*n
D	转换结果	转换后ASCII码的存放起始地址	–	INT，数组*n
n	转换字符数	转换的字符位数，n的范围为1~256	1-256	INT

表3-98 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

该指令是将S的值转换成ASCII码后，存储到D为起始地址的变量中。其中：

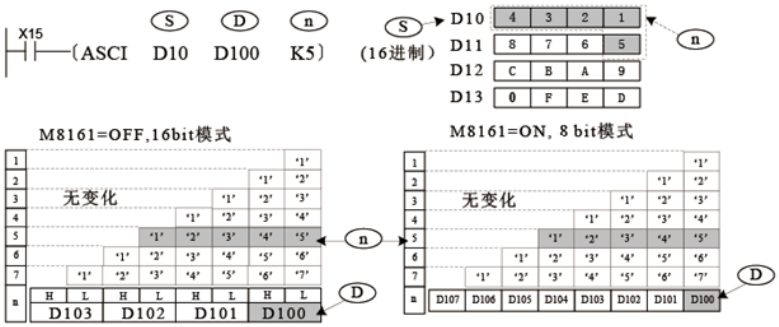
- S为待转换的变量地址或常数数值；
- D为转换后ASCII码的存放起始地址；
- n为转换的字符位数（范围为：1~256）。

说明

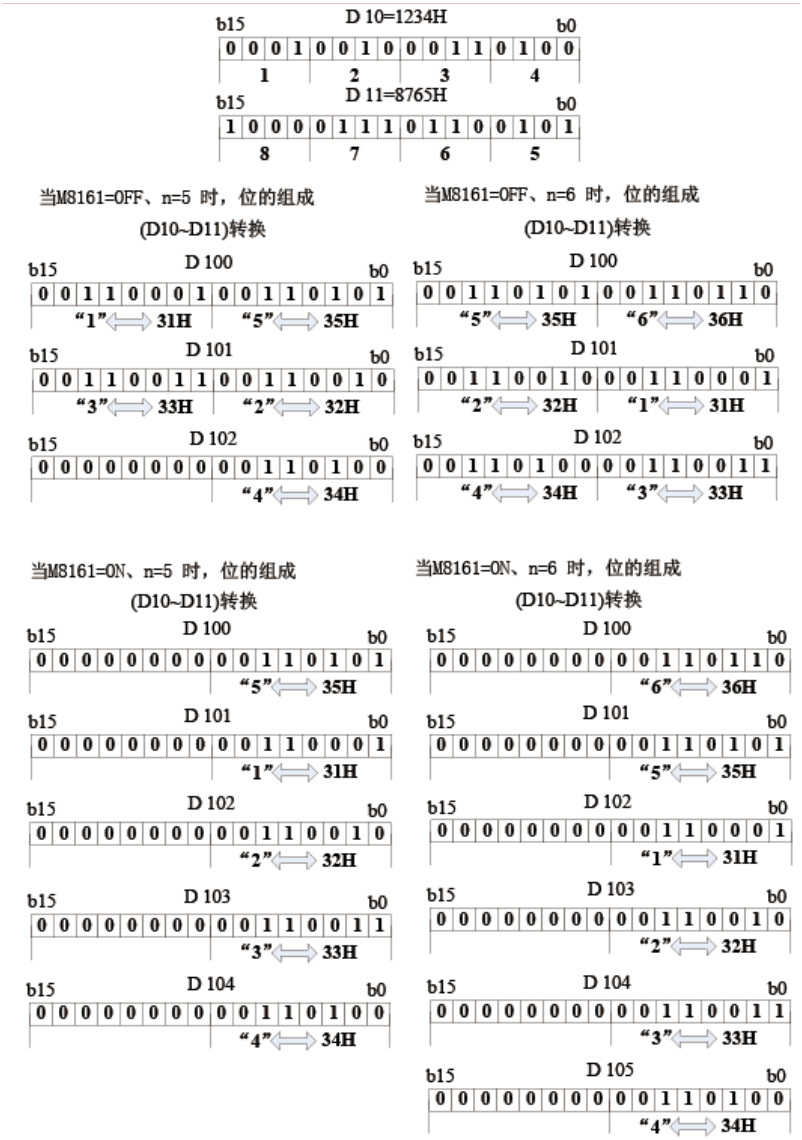
编程时请悉知，HEX/ASCI/CCD等指令共用M8161模式标志。

ASCI数值转换遵照ASCII与HEX进制数值对照表，如：ASCII‘0’对应HEX‘H30’；ASCII‘F’对应HEX‘H46’等。关于HEX和ASCII的对照关系请参考“*第416页“4.1 ASCII码转换表”**附录 ASCII码转换表*”后面。

指令示例



其中，M8161标志决定了计算结果存放目的变量的宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节分别存储；当M8161=ON时，为8bit模式，只有变量的低字节存储结果，因此实际使用变量区域的长度增加。



3.6.1.19 HEX

将S起始变量的值转换成HEX码后，存储到D为起始地址的变量中，转换的字符数、存储模式可以设定。

HEX — ASCII→HEX转换指令

16位指令	HEX 连续执行/HEXP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	数据源	待转换的变量地址或常数数值，若为寄存器变量，以32bit变量宽度（即4个ASCII字符）为单位进行转换分隔	–	INT，数组*n
D	转换结果	转换后HEX码的存放起始地址，占用的变量空间与S2有关	–	INT，数组*n
n	转换字符数	转换的字符位数	1-256	INT

表3-99 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

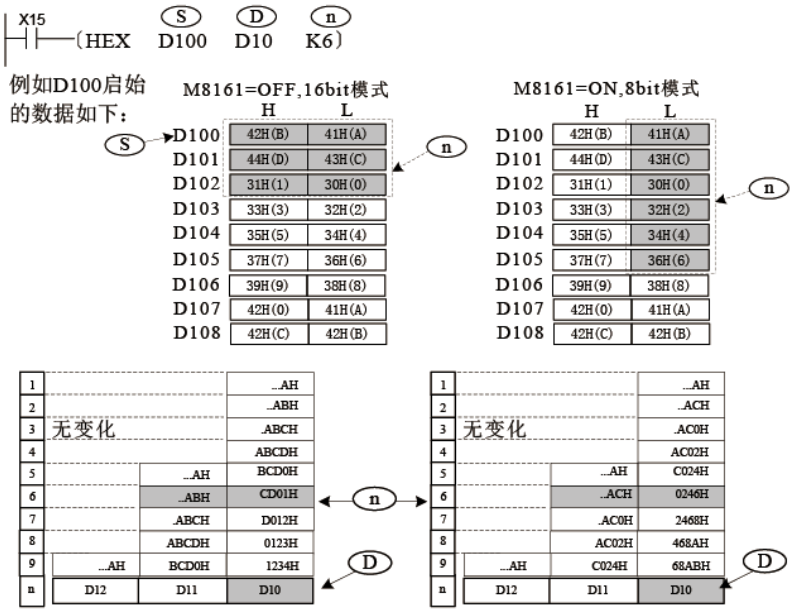
该指令是将S起始变量的值转换成HEX码后，存储到D为起始地址的变量中，转换的字符数、存储模式可以设定。其中：

- S为待转换的变量地址或常数数值，若为寄存器变量，以32bit变量宽度（即4个ASCII字符）为单位进行转换分隔；
- D为转换后HEX码的存放起始地址，占用的变量空间与n有关；
- n为转换的字符位数（范围为：1~256）。

说明

- 编程时请悉知，HEX/ASCI/CCD等指令共用M8161模式标志。
- S数据区的源数据必须为ASCII码字符，否则转换出错；
- 若输出的数据为BCD格式，HEX转换后，需要进行BCD－BIN转换，才是正确的数值。

指令示例



其中，M8161标志决定了变量宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节都参与运算；当M8161=ON时，为8bit模式，只有变量的低字节参与运算，高字节的内容丢弃，因此实际使用变量区域S的长度增加。

当M8161=OFF、n=5 时，位的组成
用了D100~D102(高低字节)转换

D 10	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0
		B			C				D				0			
D 11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
														A		

当M8161=OFF、n=6 时，位的组成
用了D100~D102(高低字节)转换

D 10	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
		C			D				0				1			
D 11	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1
											A			B		

当M8161=ON、n=5 时，位的组成
用了D100~D104(低字节)转换

D 10	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	
		C			0				2				4			
D 11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
														A		

当M8161=ON、n=6 时，位的组成
用了D100~D105(低字节)转换

D 10	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0
		0			2				4				6			
D 11	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0
											A			C		

3.6.2 数据传送与比较

3.6.2.1 指令列表

数据传送与比较指令涵盖以下指令条目：

指令类别	名称	功能
数据传送与比较	MOV	赋值传送
	EMOV	二进制浮点传送
	SMOV	移位传送
	BMOV	数据成批传送
	FMOV	数据一对多传送
	CML	数据取反传送
	CMP	数据比较
	ECMP	浮点比较指令
	ZCP	区域比较
	EZCP	浮点区域比较指令

3.6.2.2 MOV

将源址S中的数据复制到终址。

MOV — 传送

16位指令	MOV 连续执行/MOVP 脉冲执行			
32位指令	DMOV 连续执行/DMOVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	进行传送的数据或数据存储字软元件地址	–	INT/DINT
D	数据复制目的地	目的地数据存储字软元件地址	–	INT/DINT

表3–100 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	

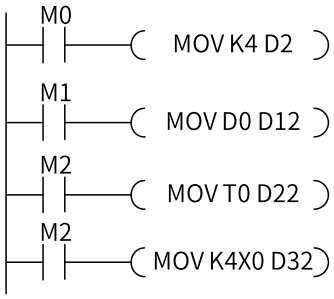
功能和指令说明

需要触点驱动，有2个操作变量，将S的值复制到D中。

当为32bit指令（DMOV）时，S和D都会使用相邻高地址的变量单元参与运算。

例如语句：〔DMOV D1 D5〕的操作结果是：D1→D5；D2→D6。

指令示例



指令列表

```
LD    M0
MOV   K4  D2
LD    M1
MOV   D0  D12
LD    M2
MOV   T0  D22
LD    M3
MOV   K4X0  D32
```

当X0=ON时，将K4复制到D2中，当X0由ON变为OFF时，D2保存K4的内容不变，除非用户程序再次修改D2的值。
或者PLC由STOP切换到RUN后重新上电时，D2的值才会变为0。
停电保持寄存器上电或者由停止到运行保持原来的值不变。

3.6.2.3 EMOV

进行二进制浮点数数据的传送。需要触点驱动，当指令执行时，将S的2进制浮点数数据值复制到D中。

EMOV — 浮点赋值指令

16位指令	—			
32位指令	DEMOV 连续执行/DEMOVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	二进制浮点数数据的传送源	–	REAL
D	传送目的地	保存二进制浮点数数据的存放单元	–	REAL

表3–101 软元件列表

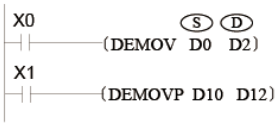
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	√	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是进行二进制浮点数数据的传送。需要触点驱动，当指令执行时，将S的2进制浮点数数据值复制到D中。其中：

- S为二进制浮点数数据的传送源；
- D为保存二进制浮点数数据的存放单元。

指令示例



假设(D1、D0)的2进制浮点数数值为12.345，当X0为ON时，(D3、D2)的2进制浮点数数值变为12.345。当M0由ON⟹ OFF时，(D3、D2)保持12.345的内容不变，除非用户程序再次将(D3、D2)的值修改。或者PLC由STOP⟹RUN和PLC重新上电，(D3、D2)的值才会变化，停电保持寄存器上电或者由停止到运行保持原来的值不变。

3.6.2.4 BMOV

当驱动条件成立时，将以S为首址的n个寄存器的数据一一对应传送到以D为首址的n个寄存器中。

BMOV — 数据批量传送

16位指令	BMOV 连续执行/BMOVP 脉冲执行			
32位指令	DBMOV 连续执行/DBMOVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据来源首址	进行批传送的数据存储字软元件首地址	–	INT/DINT
D	传送目标首址	传送目的地的数据存储字软元件首地址	–	INT/DINT
n	数据长度	批传送的字软元件的点数	1-512	INT/DINT

表3–102 软元件列表

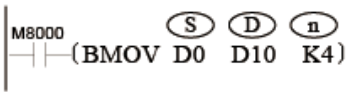
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

需要触点驱动，有3个操作变量，将由S指定起始地址的n个变量值复制到由D指定起始地址的n个单元中。

其中n的取值范围是1～512。

指令示例



完成的操作是：
D0 → D10
D1 → D11
D2 → D12
D3 → D13

3.6.2.5 SMOV

将S中以m1数位为起始的共m2数位的数位数据移动到终址D中以n数位为起始的共m2数位中。

SMOV — 移位传送

16位指令	SMOV 连续执行/SMOVP 脉冲执行			
32位指令	DSMOV 连续执行/DSMOVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	数据源	进行数位传送的数据存储字软元件地址	–	INT/DINT
m1	传送起始数位	S中要移动的起始数位的位置	1-4/1-8	INT/DINT
m2	传送数位个数	S中要移动的数位的个数	1-n1	INT/DINT
D	目的地装置	传送的目的地数据存储字软元件地址	–	INT/DINT
n	目的地起始数位	移入D中的起始数位的位置	n2-4/n2-8	INT/DINT

表3–103 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
m1	-	-	-	√	√	√	√	-	-
m2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

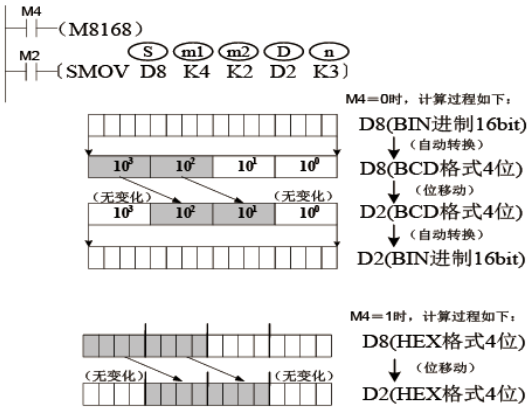
功能和指令说明

需要触点驱动，最多有5个操作变量，其中：

- S为待复制的数据源变量，当M8168为OFF时是BCD模式（十进制的位），M操作数的范围是0000~9999/00000000~99999999，不能为负数。当M8168为ON时是BIN模式，S操作数可以是负数。
- m1为数据源传送的起始位号，（1～4/1～8）范围；
- m2为数据源传送的位数，（1～m1）范围；
- D为数据源传送的目的变量；
- n为数据源传送的目的变量的起始位，（m2～4/m2～8）范围。

数据位的传送过程与特殊标志M8168的状态有关，当M8168为OFF时是BCD模式（十进制的位），当M8168为ON时是BIN模式，在BIN模式下以4个位作为一个单位作传送（十六进制的位）。

指令示例



假设D8=K1234，D2=K5678，则当M8168为OFF时（BCD模式），将M2置ON，则D2的值变为K5128；

当M8168为ON时（BIN模式），此时D8=H04D2=K1234，D2=H162E=K5678，将M2置ON，则D2=H104E=K4174。

3.6.2.6 FMOV

当驱动条件成立时，将S中的数据传送到以D为首址的n个寄存器中。

FMOV — 多点传送

16位指令	FMOV 连续执行/FMOVP 脉冲执行			
32位指令	DFMOV 连续执行/DFMOVP 脉冲执			
操作数	名称	描述	范围	数据类型
S	数据源	进行一对多传送的数据或数据存储字软元件	–	INT/DINT
D	数据传送目的地首址	传送目的地的数据存储字软元件首地址	–	INT/DINT
n	目标个数	多点传送的字软元件的点数	1-512	INT/DINT

表3–104 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

需要触点驱动，有3个操作变量，将由S的数据复制到由D指定起始地址的n个单元中。

其中n的取值范围是1~512。

FMOV是16位的多点传送指令，DFMOV是32位多点传送。

指令示例

当M8置ON时完成的操作是：
k100 → D100
k100 → D101
k100 → D102
k100 → D103

3.6.2.7 CML

将源址S按位取反后传送至终址D。

CML — 取反传送

16位指令	CML 连续执行/CMLP 脉冲执行			
32位指令	DCML 连续执行/DCMLP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	取反数据源	进行数据取反的数据或数据存储字软元件地址	–	INT/DDINT
D	传送目的地	数据取反后传送的数据存储字软元件地址	–	INT/DDINT

表3–105 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

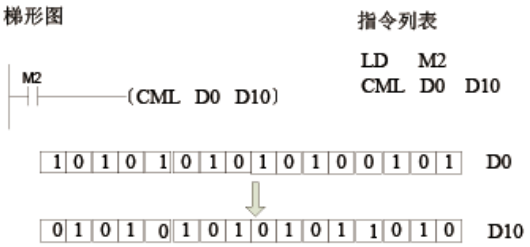
需要触点驱动，有2个操作变量，将S的BIN值逐位取反后复制到D中。

当D的位数不足16bit时，将S取反后按低位对齐传送到D变量中；

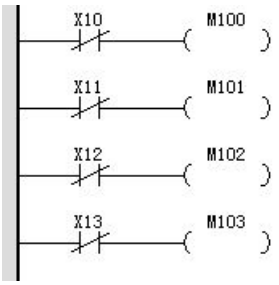
当为32bit指令（DCML）时，S和D都会使用相邻高地址的变量单元参与运算。

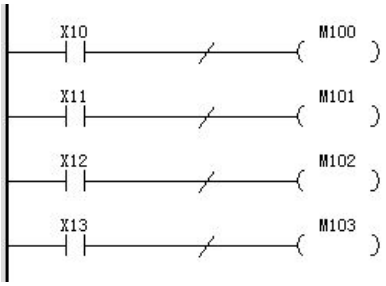
例如语句：〔DCML D1 D5〕的操作结果是：/D1→D5；/D2→D6。

指令示例一



指令示例二





3.6.2.8 CMP

CMP — 比较指令

16位指令	CMP 连续执行/CMPP 脉冲执行			
32位指令	DCMP 连续执行/DCMPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	比较值1	比较值1数据或数据存储字软元件地址	-	INT/DINT
S2	比较值2	比较值2数据或数据存储字软元件地址	-	INT/DINT
D	比较结果	比较结果ON/OFF位首址，占用连续3个位	-	BOOL，数组*3

表3–106 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	√[1]	√	√	-	-	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令完成对两个操作变量的大小作比较，将比较结果输出给指定的位变量，操作数均按有符号数进行代数比较操作。

当驱动条件成立时，比较 S1与S2的大小，并根据比较结果（S1>S2，S1=S2，S1<S2）将终址位元件D,D+1，D+2其中一个置为ON。

其中D会占用3个连续地址的位变量。

指令示例

当X0=ON时，M0~M2其中之一会ON。
X0由ON变为OFF时，不执行CMP指令，M0~M2仍保持
X0=OFF之前的状态。
若要清除M0~M2的比较结果，可用RST或ZRST指令。
若要得到≥、≤、≠的结果时，可将M0~M2串并联来实现。

3.6.2.9 ECMP

ECMP — 浮点比较指令

16位指令	—			
32位指令	DECMP 连续执行/DECMPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	比较数1	待比较的二进制浮点数1	-	REAL
S2	比较数2	待比较的二进制浮点数2	-	REAL
D	比较结果	比较结果的存放单元，共占用3个（位）变量单元	-	BOOL，数组*3

表3–107 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
D	√[1]	√	√	-	-	-	-	-	-

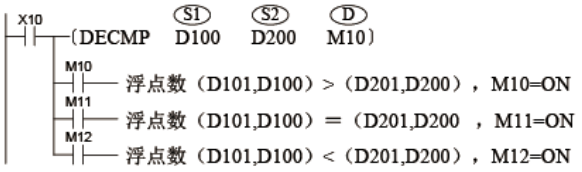
说明

[1]不支持X元件。

功能和指令说明

该指令是进行2个浮点数变量的比较，将比较的结果输出到启始的3个变量中。

指令示例



当X10=ON时，M10~M12其中之一会ON。
X10由ON变OFF 时，不执行DECP指令，M10~M12仍保持X10
=OFF之前的状态，要清除M10~M12的比较结果可用RST或者ZRST
对M10~M12进行清除。
若需要得到≥、≤、≠的结果时，可将M10~M12
串并联即可取得。

3.6.2.10 ZCP

当驱动条件成立时，根据源址S所处的区间（S<S1，S1≤S≤S2，S＞S2）将终址位元件D,D+1，D+2,其中一个置为ON。

ZCP — 区域比较

16位指令	ZCP 连续执行/ZCPP 脉冲执行			
32位指令	DZCP 连续执行/DZCPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	区间比较下限值	数据或数据存储字软元件地址	-	INT/ DINT
S2	区间比较上限值	数据或数据存储字软元件地址	-	INT/ DINT
S	比较变量	数据或数据存储字软元件地址	-	INT/ DINT
D	比较结果	比较结果ON/OFF位首址，占用连续3个位	-	BOOL，数组*3

表3–108 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	√	-	-
S	-	-	-	√	√	√	√	-	-
D	√[1]	√	√	-	-	-	-	-	

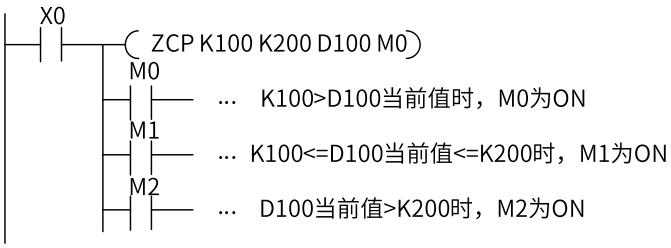
说明

[1]不支持X元件。

功能和指令说明

需要触点驱动，有4个操作变量。当控制能流有效时，按有符号数进行代数比较操作，以S1、S2为区间，将s的值位于该区间的位置作为结果，存入D为起始地址的3个连续位变量中。

指令示例



当X0=ON时，M3~M5其中之一会ON。
X0由ON变为OFF时，不执行ZCP指令，M3~M5仍保持X0=OFF之前的状态。
若要清除M3~M5的比较结果，可用RST或ZRST指令。

3.6.2.11 EZCP

进行二进制浮点数变量的区间比较，将比较的结果输出到D启始的3个变量中。

EZCP — 浮点区域比较指令

16位指令	—			
32位指令	DEZCP 连续执行/DEZCPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	区间比较下限	二进制浮点变量区间的下限	-	REAL
S2	区间比较上限	二进制浮点变量区间的上限	-	REAL
S	比较数	待比较的二进制浮点变量	-	REAL
D	比较结果	比较结果的存放单元，共占用3个（位）变量单元	-	BOOL，数组*3

表3–109 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	√	-
S2	-	-	-	√	√	√	-	√	-
S	-	-	-	√	√	√	-	√	-
D	√[1]	√	√	-	-	-	-	-	

说明

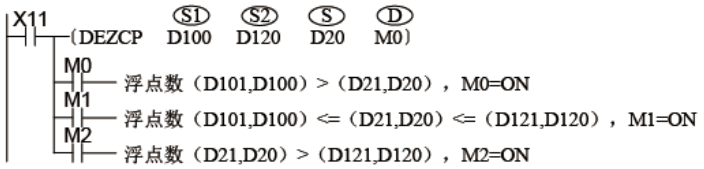
[1]不支持X元件。

功能和指令说明

该指令是进行二进制浮点数变量的区间比较，将比较的结果输出到D启始的3个变量中。其中：

- S1为二进制浮点变量区间的下限；
- S2为二进制浮点变量区间的上限；
- S待比较的二进制浮点变量；
- D为比较结果的存放单元，共占用3个（位）变量单元。

指令示例



当X11=ON时，M0~M2其中之一会ON。
当X11由ON变OFF时，不执行DEZCP指令，M0/M1/M2保持X11=OFF以前的状态不变。要清除M0~M2的比较结果可用RST或者ZRST对M0~M2进行清除。

3.6.3 表格操作

3.6.3.1 指令列表

表格操作指令涵盖以下指令条目：

指令类别	名称	功能
表格操作	SORT	数据排序
	SORT2	数据排序2
	SER	数据查找
	FDEL	数据表的数据删除
	FINS	数据表的数据插入
	POP	后入数据的读取

3.6.3.2 SORT

当驱动条件成立时，对以S为首址的m1行m2列的数据表格的第n列数据进行升序排列，排列结果存储到以D为首址的数据表格中。

SORT — 数据排序指令

16位指令	SORT 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	表1首址	表格1数据存储字软元件首址【占用m1Xm2点】	–	INT，数组*n1*n2
n1	表格行数	表格行数或数据存储字软元件地址	1-32	INT
n2	表格列数	表格列数或数据存储字软元件地址	1-8	INT
D	表2首址	表格2数据存储字软元件首址【占用m1Xm2点】	–	INT，数组*n1*n2
n	排序列数	指定排序的列数【1~m2】	1-n2	INT

表3–110 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
n1	-	-	-	-	-	-	√	-	-

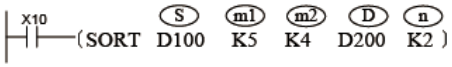
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
n2	-	-	-	-	-	-	√	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	-	-	-	√	-	-

功能和指令说明

该指令是将m1行×m2列的数组（由S、m1、m2描述），以第n列参数排序后，存放于由D单元起始的变量区域。其中：

- S为第1行（或称第1条记录）的首个变量的起始单元；
- m1数组的行数，或称记录数；
- m2数组的列数，或称每条记录的栏目数；
- D为排序后存放的起始单元，占用随后的变量单元数目与排序前的数组变量数目相同；
- n为以排序为依据的数组列号，n的值在1～m2范围内。

指令示例



当X10＝ON时，开始排序运算，指令执行完毕，M8029被置ON；

若需要再次排序，需将X10＝OFF一次。

上述指令的等效表格及其数据举例：

行号＼列号	1 学号	2 语文	3 数学	4 物理
1	D100 1	D105 85	D110 78	D115 83
2	D101 2	D106 82	D111 91	D116 81
3	D102 3	D107 77	D112 89	D117 88
4	D103 4	D108 90	D113 81	D118 75
5	D104 5	D109 87	D114 95	D119 77

按指令要求n＝K2排序后的表格数据结果：

行号＼列号	1 学号	2 语文	3 数学	4 物理
1	D200 3	D205 77	D210 89	D215 88
2	D201 2	D206 82	D211 91	D216 81
3	D202 1	D207 85	D212 78	D217 83
4	D203 5	D208 87	D213 95	D218 77
5	D204 4	D209 90	D214 81	D219 75

若指令中n＝K4，则排序后的表格数据结果如下：

行号＼列号	1 学号	2 语文	3 数学	4 物理
1	D200 4	D205 90	D210 81	D215 75
2	D201 5	D206 87	D211 95	D216 77
3	D202 2	D207 82	D212 91	D217 81
4	D203 1	D208 85	D213 78	D218 83
5	D204 3	D209 77	D214 89	D219 88

3.6.3.3 SORT2

以行为单位，对指定的列进行升序/降序排序。该指令数据按行方向进行连续的存储，便于增加行数据。

SORT2 — 数据排序2

16位指令	SORT2 连续执行			
32位指令	DSORT2 连续执行			
操作数	名称	描述	范围	数据类型
S	源地址	保存数据表格的软元件起始编号[占用m1×m2点]	-	INT/DINT，数组*n1*n2
m1	行数	数据(行)数[1～32]	1-32	INT/DINT
m2	列数	数据(列)数[1～6]	1-8	INT/DINT
D	目的地址	保存运算结果的软元件起始编号[占用m1×m2点]	-	INT/DINT,数组*n1*n2
n	目标列	作为排序标准的群数据(列)的列编号[1～m2]	1-n2	INT/DINT

表3–111 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
m1	-	-	-	-	-	-	√	-	-
m2	-	-	-	-	-	-	√	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	-	-	-	√	-	-

功能和指令说明

针对[S]开始的(m1×m2)点的数据表格(排序前)，以n列的群数据为基准，以行为单位，将数据行进行升序或是降序的排列，然后保存到从[D]开始的(m1×m2)点的数据表格(排序后)中。

通过M8165的ON/OFF状态来设定排序，M8165置ON，降序排列，反之，升序排列。

指令能流有效时，开始进行数据的排列，经过m1个扫描之后，排序完成，指令完成标志位M8029置ON。

以下为3x3数据排序过程：

- 排序前

项目	第1列	第2列	第3列
第1行	S	S+1	S+2
	1	2	8
第2行	S+3	S+4	S+5
	2	6	7
第3行	S+6	S+7	S+8
	3	4	3

- 按照第2列进行排序后(升序)

项目	第1列	第2列	第3列
第1行	D	D+1	D+2
	1	2	8
第2行	D+3	D+4	D+5
	3	4	3
第3行	D+6	D+7	D+8
	2	6	7

说明

- 指令执行过程中，不可更改操作数。
- 再次执行指令，请将能流OFF→ON一次。
- 动作过程中，请勿使操作数和数据的内容变化。
- S和D不可部分重叠，只可以完全重叠或者是完全错开。
- 32位指令使用与16位指令一致，操作数占用两个16位软元件。

以下一些情况下会发生运算错误。

1）S或D中设定的软元件超范围时，报错误；

2）m1、m2或n设定超范围时，报错误。

指令示例

对从D100起始的第2列数据按照升序排序，结果存储于R100开始的地址。

指令示例

对从D100起始的第2列数据按照升序排序，结果存储于R100开始的地址。

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	1
D101	16位整数	十进制	2
D102	16位整数	十进制	8
D103	16位整数	十进制	2
D104	16位整数	十进制	6
D105	16位整数	十进制	7
D106	16位整数	十进制	3
D107	16位整数	十进制	4
D108	16位整数	十进制	3
R100	16位整数	十进制	1
R101	16位整数	十进制	2
R102	16位整数	十进制	8
R103	16位整数	十进制	3
R104	16位整数	十进制	4
R105	16位整数	十进制	3
R106	16位整数	十进制	2
R107	16位整数	十进制	6
R108	16位整数	十进制	7
	16位整数	十进制	
M8029	BOOL	二进制	ON
M8165	BOOL	二进制	OFF

3.6.3.4 SER

当驱动条件成立时，从以源址S1为起始的k个数据中，检索出符合条件S2的数据的地址，存放于以D为首址的5个寄存器中。

SER — 数据查找指令

16位指令	SER 连续执行/SERP 脉冲执行			
32位指令	DSER 连续执行/DSERP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	检索数据首址	进行检索的数据的首地址，检索连续K个寄存器	–	INT/DINT，数组*n
S2	对比数据	进行对比的数据或数据存储字软元件地址	–	INT/DINT
D	检索结果首址	检索结果存储字软元件地址	–	INT/DINT，数组*5
n	检索数据个数	要检索的数据个数	1-256	INT/DINT

表3–112 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	√		
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

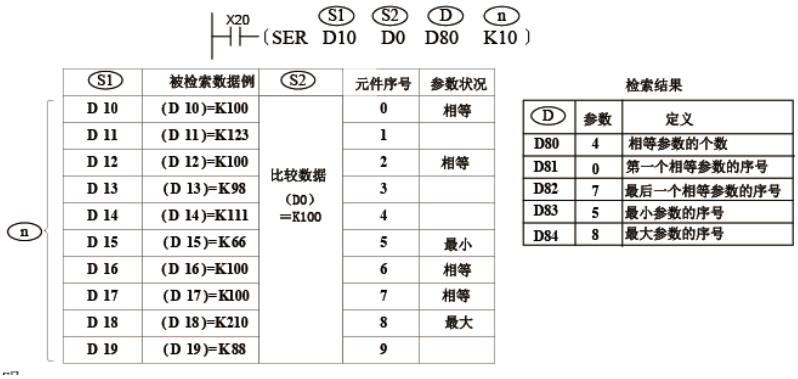
该指令用于从一组数据中，查找相同数据的单元、同时对最大值、最小值的检索。其中：

- S1为数据组的的起始地址；
- S2为待检索的数据；
- D为检索结果存放区的起始地址；
- n为被检索数据区的长度， n=1~256。

当使用32bit指令时，S1、S2、D均指向32bit变量，n也按32bit变量宽度进行计算。

当驱动条件成立时，从以源址S1为起始的k个数据中，检索出符合条件S2的数据的地址，存放于以D为首址的5个寄存器中。

指令示例



使用说明：

当指令能流X20为ON时，方才进行比较。比较的方法为有符号数的代数比较方法进行，例如－8<2；

当最小值、最大值有多个时，分别显示序号最大的元件；

存储检索结果的单元占用D开始的5个连续单元。若不存在相等数据时，上例中的D80～D82均为0。

3.6.3.5 FDEL

删除数据表格中任意数据的指令。

FDEL — 数据表的数据删除

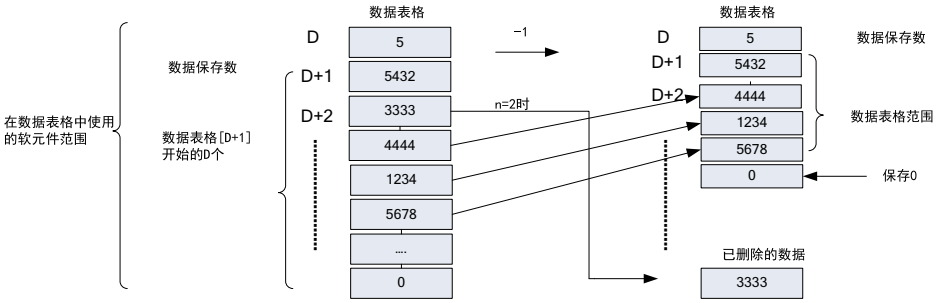
16位指令	FDEL 连续执行/FDELP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	被删数据	保存被删除的数据的软元件编号	-	INT
D	数据表格信息	数据表格的起始软元件编号 D：数据保存数 D+1：数据表格的起始位置	-	INT，数组*(D+1)
n	被删位置	要删除的数据的表格位置	1-256	INT

表3–113 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

删除从[D+1]开始的数据表格的第n个数据，将删除的数据保存到[S]中，同时[D+1]中第n+1个开始的数据逐个向前靠拢，且数据保存数[D]减1。



以下一些情况会报故障：

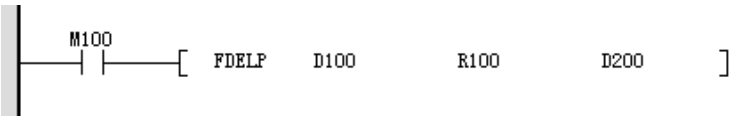
1）数据保存数超出相应软元件范围时，报错误；

2）被删位置n＞数据保存数[D]时，报错误；

3）n设定值≤0时，报错误；

4）数据保存数设定值≤0时，报错误。

指令示例



指令执行前

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	0
R100	16位整数	十进制	5
R101	16位整数	十进制	1111
R102	16位整数	十进制	2222
R103	16位整数	十进制	3333
R104	16位整数	十进制	4444
R105	16位整数	十进制	5555
R106	16位整数	十进制	0
D200	16位整数	十进制	3

指令执行后

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3333
R100	16位整数	十进制	4
R101	16位整数	十进制	1111
R102	16位整数	十进制	2222
R103	16位整数	十进制	4444
R104	16位整数	十进制	5555
R105	16位整数	十进制	0
R106	16位整数	十进制	0
D200	16位整数	十进制	3

3.6.3.6 FINS

在数据表格中的任意位置处插入数据的指令。

FINS — 数据表的数据插入

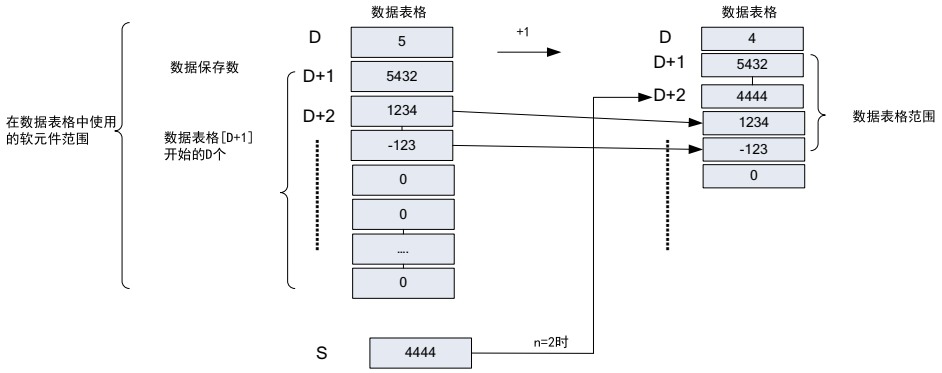
16位指令	FINS 连续执行/FINSP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	被插数据	保存插入数据的软元件编号	-	INT
D	数据表格信息	数据表格的起始软元件编号 D：数据保存数 D+1：数据表格的起始位置	-	INT，数组 *(D+2)
n	被插位置	插入数据的表格位置	1-256	INT

表3–114 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将[S]数据插入从[D+1]开始的数据表格的第n个中，同时该数据表先前的第n个数据逐个后移，且数据保存数[D]加1。



以下一些情况会报故障：

1）数据保存数超出相应软元件范围时，报错误；

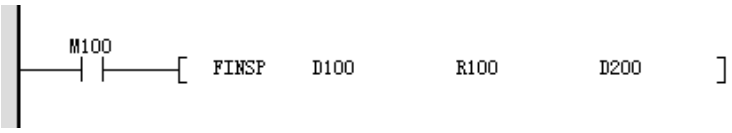
2）插入后数据表格超出相应软元件范围时，报错误；

3）被插位置n＞数据保存数[D]时，报错误；

4）n设定值≤0时，报错误；

5）数据保存数设定值≤0时，报错误。

指令示例



指令执行前

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3333
R100	16位整数	十进制	4
R101	16位整数	十进制	1111
R102	16位整数	十进制	2222
R103	16位整数	十进制	4444
R104	16位整数	十进制	5555
R105	16位整数	十进制	0
R106	16位整数	十进制	0
D200	16位整数	十进制	3

指令执行后

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3333
R100	16位整数	十进制	5
R101	16位整数	十进制	1111
R102	16位整数	十进制	2222
R103	16位整数	十进制	3333
R104	16位整数	十进制	4444
R105	16位整数	十进制	5555
R106	16位整数	十进制	0
D200	16位整数	十进制	3

3.6.3.7 POP

该指令用于读出使用先入后出控制用的移位写入指令(SFWR)，写入最后的数据。

POP — 后入数据的读取

16位指令	POP 连续执行/POPP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	被读数据	保存先入数据(包含指针数据)的起始软元件编号 S：指针数据(被保存的数据个数) S+1：数据区域	-	INT，数组*n
D	保存结果	保存后出的数据的软元件编号	-	INT
n	数据点数	被保存的数据的点数 (由于包含了指针数据，所以请设置为+1后的值，2≤n≤512)	2-512	INT

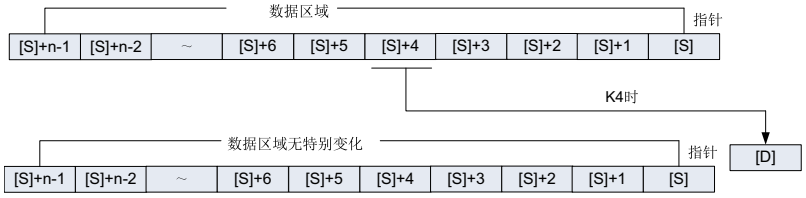
表3–115 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

对于[S~S+n-1]的字软元件，每次执行该指令，读出以S起始地址和[S]中值为偏移指针(指针数据)的元件中值，保存至[D]中，指针数据[S]自减1。n的值可以指定2~512。

地址	内容
S	指针数据（被保存的数据个数）
[S]+1	数据区域 （使用移位写入指令（SFWR）被先入的数据）
[S]+2	
[S]+3	
~	
[S]+n-3	
[S]+n-2	
[S]+n-1	



[S]指针当前值为0时，零位标志位M8020为ON，指令不执行。

此时，请先使用比较指令确认的当前值是否1≤[S]≤(n-1)，然后执行这个指令。

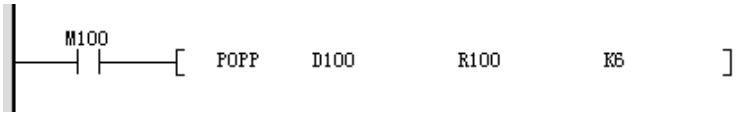
[S]指针当前值为1时，[S]中被写入0，零位标志位M8020置ON。

以下一些情况会报故障：

[S]＞n-1时，报错误。

[S]＜0时，报错误。

指令示例



指令执行前

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	4
D101	16位整数	十进制	1111
D102	16位整数	十进制	2222
D103	16位整数	十进制	3333
D104	16位整数	十进制	4444
D105	16位整数	十进制	5555
D106	16位整数	十进制	6666
D107	16位整数	十进制	0
	16位整数	十进制	
R100	16位整数	十进制	0

指令执行后

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3
D101	16位整数	十进制	1111
D102	16位整数	十进制	2222
D103	16位整数	十进制	3333
D104	16位整数	十进制	4444
D105	16位整数	十进制	5555
D106	16位整数	十进制	6666
D107	16位整数	十进制	0
	16位整数	十进制	
R100	16位整数	十进制	4444

3.6.4 数据移位

3.6.4.1 指令列表

数据移位指令涵盖以下指令条目：

指令类别	名称	功能
数据移位	ROR	循环右移
	ROL	循环左移
	RCR	带进位的循环右移
	RCL	带进位的循环左移
	SFTR	位右移
	SFTL	位左移
	WSFR	字右移
	WSFL	字左移
	SFWR	先进先出的数据写入
	SFRD	先进先出的数据读出
	SFR	16位数据n位右移(带进位)
	SFL	16位数据n位左移(带进位)

3.6.4.2 ROR

当驱动条件成立时，D中数据向右移动K位，移出D的低位数据循环进入D的高位。

ROR — 数据循环右移

16位指令	ROR 连续执行/RORP 脉冲执行			
32位指令	DROR 连续执行/DRORP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	将循环的装置	数据存储字软元件地址	–	INT/DINT
n	单次移动位数	有效范围: 1≤ n ≤16 (16 位), 1≤ n≤32 (32 位)	1-16/32	INT/DINT

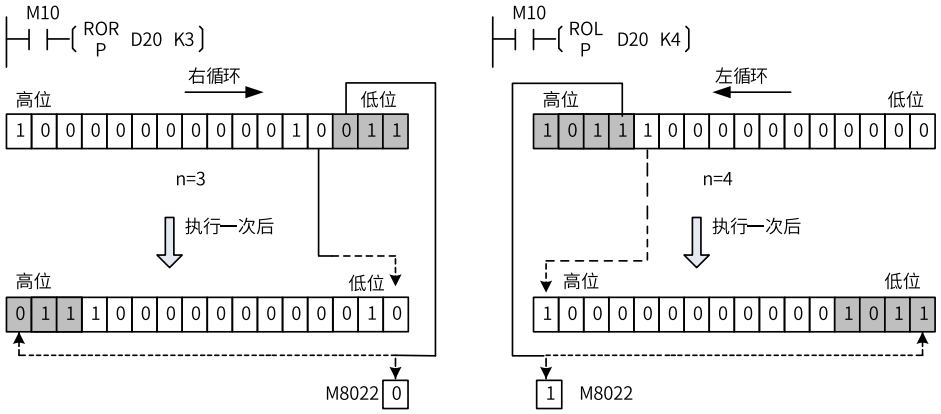
表3–116 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将D的内容连同进位标志M8022循环左移n位。本指令一般使用脉冲执行型指令。当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元。

指令示例



3.6.4.3 ROL

当驱动条件成立时，D中数据向左移动K位，移出D的高位数据循环进入D的低位。

ROL — 数据循环左移

16位指令	ROL 连续执行/ROLP 脉冲执行			
32位指令	DROL 连续执行/DROLP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	将循环的装置	数据存储字软元件地址	–	INT/DINT
n	单次移动位数	有效范围: 1≤ n ≤16 (16 位), 1≤ n ≤32 (32 位)	1-16/32	INT/DINT

表3–117 软元件列表

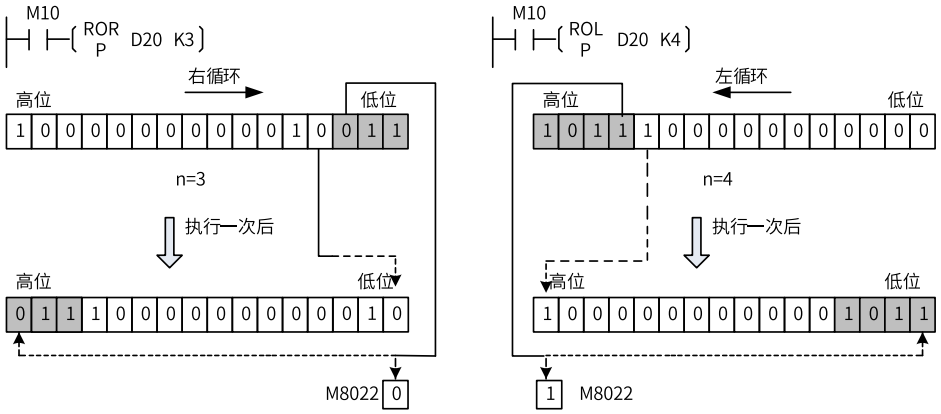
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将D的内容循环左移n位。本指令一般使用脉冲执行型指令。当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元：

循环移动的最终位被存入进位标志中。

指令示例



3.6.4.4 RCR

当驱动条件成立时，D中数据带进位（M8022）向右移动k位，移出的低位带进位（M8022）循环进入D的高位。

RCR — 数据带进位循环右移

16位指令	RCR 连续执行/RCRP 脉冲执行			
32位指令	DRCR 连续执行/DRCRP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	将循环的装置	数据存储字软元件地址	–	INT/DINT
n	单次移动位数	有效范围：1≤ n ≤16 (16 位), 1≤ n≤32 (32 位)	1-16/32	INT/DINT

表3–118 软元件列表

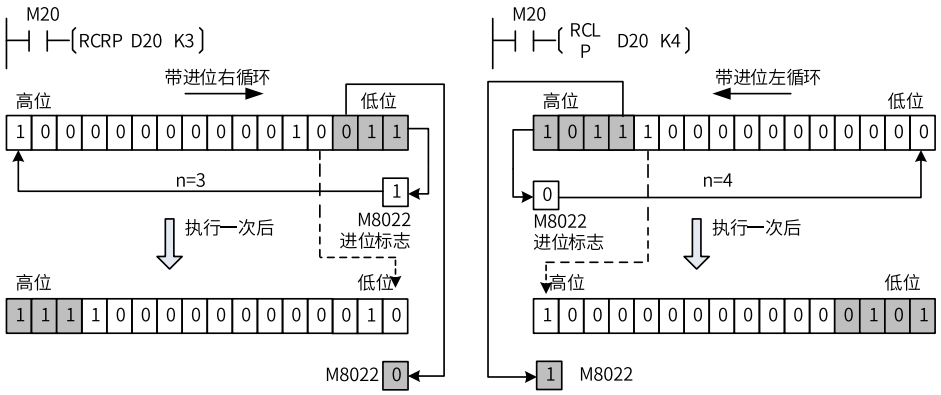
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将D的内容连同进位标志M8022循环右移n位。

本指令一般使用脉冲执行型指令。当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元。

指令示例



3.6.4.5 RCL

当驱动条件成立时，D中数据带进位（M8022）向左移动k位，移出的高位带进位（M8022）循环进入D的低位。

RCL — 数据带进位循环左移

16位指令	RCL 连续执行/RCLP 脉冲执行			
32位指令	DRCL 连续执行/DRCLP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	将循环的装置	数据存储字软元件地址	–	INT/DINT
n	单次移动位数	有效范围: 1≤ n ≤16 (16 位), 1≤ n ≤32 (32 位)	1-16/32	INT/DINT

表3–119 软元件列表

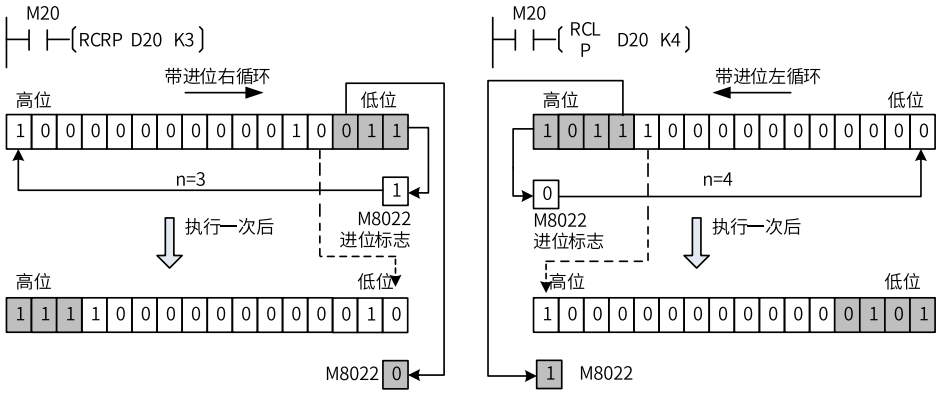
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将D的内容连同进位标志M8022循环左移n位。本指令一般使用脉冲执行型指令。

当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元。

指令示例



3.6.4.6 SFTR

当驱动条件成立时，将以D为首址的长度为K1的位元件组合向右移动K2位，高位由以S为首址的长度为K2的位元件组合移入，被移出的K2个低位被舍弃，位元件组合S保持原值不变。

SFTR — 多个位变量右移

16位指令	SFTR 连续执行/SFTRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	位元件首址	移动位元件起始地址	–	BOOL，数组*n2
D	移入位首址	被移入位元件起始地址	–	BOOL，数组*n1
n1	移入位个数	被移入位元件个数	1-256	INT
n2	位元件个数	移动位元件个数	1-n1	INT

表3–120 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	-	-	-	-
D	√[1]	√	√	-	-	-	-	-	-
n1	-	-	-	√	√	√	√	-	-
n2	-	-	-	√	√	√	√	-	-

说明

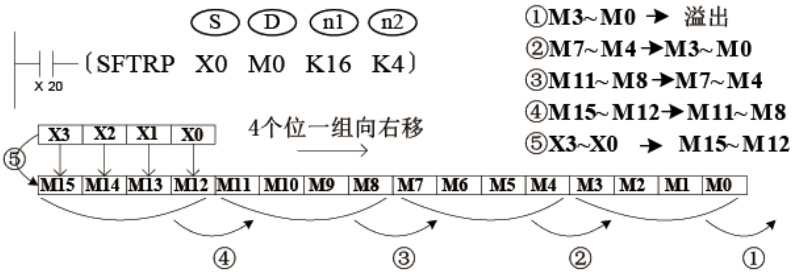
[1]不支持X元件。

功能和指令说明

对于位变量，将S地址起始的n2位变量与D地址起始的n1变量，按向右方向移动n2位后，将结果保存在D中。

本指令一般使用脉冲执行型指令。

指令示例



3.6.4.7 SFTL

当驱动条件成立时，将以D为首址的长度为K1的位元件组合向左移动K2位，低位由以S为首址的长度为K2的位元件组合移入，被移出的K2个高位被舍弃，位元件组合S保持原值不变。

SFTL — 多个位变量左移

16位指令	SFTL 连续执行/SFTLP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	位元件首址	移动位元件起始地址	–	BOOL，数组*n2
D	移入位首址	被移入位元件起始地址	–	BOOL，数组*n1
n1	移入位个数	被移入位元件个数	1-256	INT
n2	位元件个数	移动位元件个数	1-n1	INT

表3–121 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	√	√	√	-	-	-	-	-	-
D	√[1]	√	√	-	-	-	-	-	-
n1	-	-	-	√	√	√	√	-	-
n2	-	-	-	√	√	√	√	-	-

说明

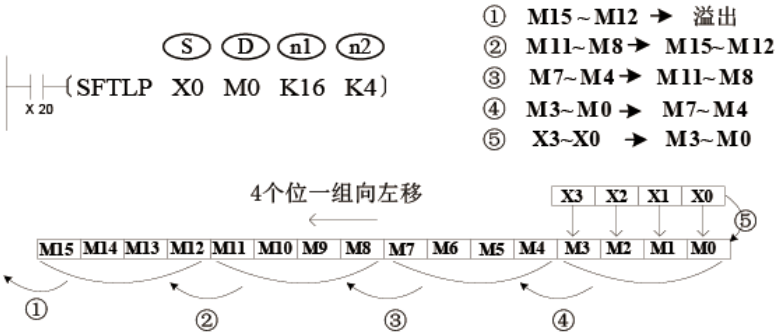
[1]不支持X元件。

功能和指令说明

对于位变量，将s地址起始的n2位变量与D地址起始的n1变量，按向左方向移动n2位后，将结果保存在D中。

本指令一般使用脉冲执行型指令。

指令示例



3.6.4.8 WSFR

当驱动条件成立时，将以D为首址的长度为K1的字元件组合向右移动K2位，高字由以S为首址的长度为K2的字元件组合移入，被移出的K2个低字被舍弃，字元件组合S保持原值不变。

WSFR — 多个字变量右移

16位指令	WSFR 连续执行/WSFRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	字元件首址	移动字元件起始地址	–	INT，数组*n2
D	移入字首址	被移入字元件起始地址	–	INT，数组*n1
n1	移入字个数	被移入字元件个数	1-256	INT
n2	字元件个数	移动字元件个数	1-n1	INT

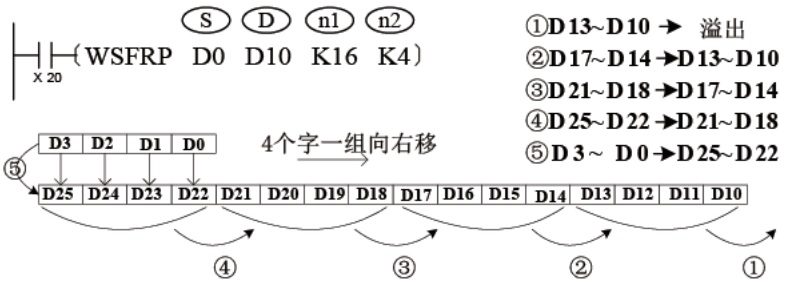
表3–122 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n1	-	-	-	√	√	√	√	-	-
n2	-	-	-	√	√	√	√	-	-

功能和指令说明

以字为单位，将S地址起始的n2字变量与D地址起始的n1字变量，按向右方向移动n2个字。本指令一般使用脉冲执行型指令。

指令示例



3.6.4.9 WSFL

当驱动条件成立时，将以D为首址的长度为K1的字元件组合向左移动K2位，低字由以S为首址的长度为K2的字元件组合移入，被移出的K2个高字被舍弃，字元件组合S保持原值不变。

WSFL — 多个字变量左移

16位指令	WSFL 连续执行/WSFLP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	字元件首址	移动字元件起始地址	–	INT，数组*n2
D	移入字首址	被移入字元件起始地址	–	INT，数组*n1
n1	移入字个数	被移入字元件个数	1-256	INT
n2	字元件个数	移动字元件个数	1-n1	INT

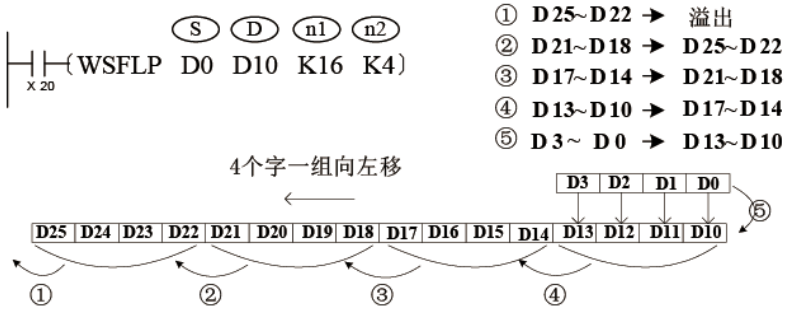
表3–123 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n1	-	-	-	√	√	√	√	-	-
n2	-	-	-	√	√	√	√	-	-

功能和指令说明

以字为单位，将S地址起始的n2字变量与D地址起始的n1字变量，按向左方向移动n2个字。本指令一般使用脉冲执行型指令。

指令示例



3.6.4.10 SFWR

当驱动条件成立时，在长度为n的数据寄存器中向以D+1开始的数据寄存器中写入S中所存储的当前值，每写入一个数据到数据库中，指针D就自动加1。

SFWR — 数据“先进先出”写入

16位指令	SFWR 连续执行/SFWRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	数据源	要写入的数据或数据存储字元件地址	–	INT
D	数据区首址	数据区数据存储字元件首址	–	INT，数组*n
n	数据区长度	数据区长度，包含指针	2-512	INT

表3–124 软元件列表

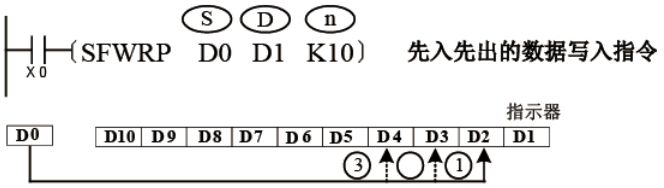
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将S的值写入由D地址起始，个数为n的“先进先出”队列中，以第一个编号装置作为指针，当指令执行时，指针内容值先加1，之后S所指定的装置其内容值会写入先入先出D数据串列中由指针所指定的位置。

本指令一般使用脉冲执行型指令。

指令示例



当X0＝1时，D0的内容被存入D2，D1的内容变为1。当X0再次从OFF→ON时，D0的内容被存入D3，D1的内容变为2，以此类推。若D1的内容超过n-1，则指令不处理，而进位标志M8022会置1。

3.6.4.11 SFRD

当驱动条件成立时，在长度为n的数据寄存器中，把以S+1开始的寄存器的数据依次传送到D寄存器中。

SFRD — 数据“先进先出”读出

16位指令	SFRD 连续执行/SFRDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	数据区首址	数据区数据存储字元件首址	-	INT，数组*n
D	读出数据	读出的数据的存放地址	-	INT
n	数据区长度	数据区的长度	2-512	INT

表3–125 软元件列表

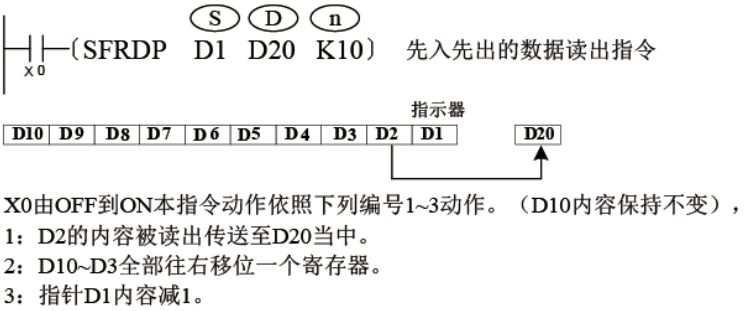
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

从“先进先出”队列S的首项读出到D中，然后将队列S逐字右移1个字，将队列指针递减。，以第一个编号装置作为指针，当指令执行时，指针内容值先减1，之后S所指定的装置其内容值会写入先入先出D数据串列中由指针所指定的位置。若指针已经为0，则指令不处理前述操作，而0标志M8020会置1。

本指令一般使用脉冲执行型指令。

指令示例



3.6.4.12 SFR

使字软元件中的数据向右移动n位的指令。

SFR — 数据n位右移(带进位)

16位指令	SFR 连续执行/SFRP 脉冲执行			
32位指令	DSFR 连续执行/DSFRP 脉冲执行			
操作数	名称	描述	范围	数据类型
D	待移动字	保存要移动的数据的软元件编号	-	INT/DINT
n	移动次数	移动的次数 (16位：0≤n≤15，32位：0≤n≤31)	0-15/31	INT/DINT

表3–126 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

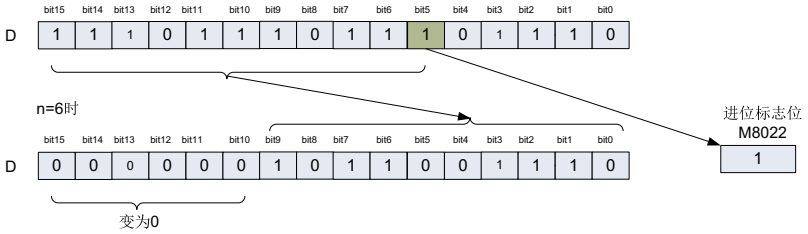
16位指令：

n指定0～15的数字。当n≥16时，按照n%16进行移位，即按照余数进行移位。例如n=20时，20%16=4，所以右移4位。

32位指令：

n指定0～31的数字。当n≥32时，按照n%32进行移位，即按照余数进行移位。例如n=40时，40%32=8，所以右移8位。

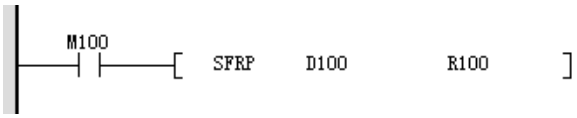
将[D]中数据的bn-1位的1/0状态写入进位标志位M8022中，[D]中最高位开始的n位用0进行填充。



报错

移动次数n<0时，报错误。

指令示例



指令执行前

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0xAAAA
	16位整数	十进制	
R100	16位整数	十六进制	0x8
	16位整数	十进制	
M8022	BOOL	二进制	OFF
	16位整数	十进制	

指令执行后

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0xAA
	16位整数	十进制	
R100	16位整数	十六进制	0x8
	16位整数	十进制	
M8022	BOOL	二进制	ON
	16位整数	十进制	

3.6.4.13 SFL

使字软元件中的数据向左移动n位的指令。

SFL — 数据n位左移(带进位)

16位指令	SFL 连续执行/SFLP 脉冲执行			
32位指令	DSFL 连续执行/DSFLP 脉冲执行			
操作数	名称	描述	范围	数据类型

D	待移动字	保存要移动的数据的软元件编号	-	INT/DINT
n	移动次数	移动的次数 （16位：0≤n≤15，32位：0≤n≤31）	0-15/31	INT/DINT

表3–127 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

16位指令：

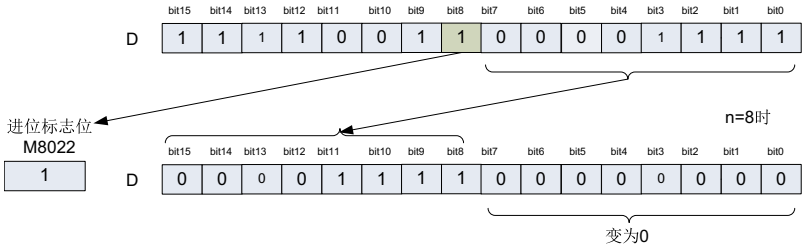
n指定0～15的数字。当n≥16时，按照n%16进行移位，即按照余数进行移位。例如n=20时，20%16=4，所以左移4位。

32位指令：

n指定0～31的数字。当n≥32时，按照n%32进行移位，即按照余数进行移位。例如n=40时，40%32=8，所以左移8位。

将[D]中数据的bn位的1/0状态写入进位标志位M8022中。

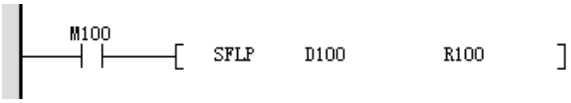
[D]中最低位开始的n位用0进行填充。



报错

移动次数n<0时，报错误。

指令示例



指令执行前

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x5555
	16位整数	十进制	
R100	16位整数	十六进制	0x8
	16位整数	十进制	
M8022	BOOL	二进制	OFF
	16位整数	十进制	

指令执行后

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x5500
	16位整数	十进制	
R100	16位整数	十六进制	0x8
	16位整数	十进制	
M8022	BOOL	二进制	ON

3.6.5 其他数据处理

3.6.5.1 指令列表

其他数据处理指令涵盖以下指令条目：

指令类别	名称	功能
其他数据处理	SWAP	高低字节交换
	BON	ON位判断
	SUM	统计ON位总数
	RAND	产生有范围限制的随机数据
	XCH	数据交换
	ABS	绝对值指令
	EABS	浮点绝对值指令
	PID	PID运算指令
	CCD	校验码
	CRC	CRC校验码计算
	LRC	LRC校验码计算
	RAMP	斜坡指令

3.6.5.2 SWAP

将指定变量S的高低字节的值进行互相交换。

SWAP — 高低字节交换指令

16位指令	SWAP 连续执行/SWAPP 脉冲执行			
32位指令	DSWAP 连续执行/DSWAPP 脉冲执行			
操作数	名称	描述	范围	数据类型
S/D	操作数	欲执行上/下字节互换的数据存储单元	-	INT/DINT

表3–128 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S/D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是将指定变量S的高低字节的值进行互相交换。

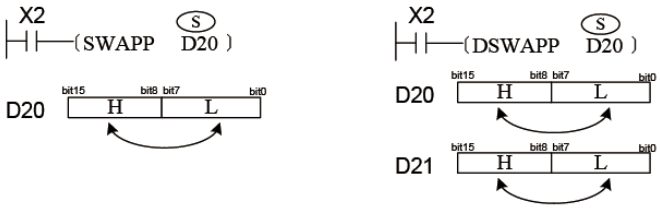
16 位指令时，高8 位与低8 位的值进行互相交换。

32 位指令时，两个寄存器的高8 位与低8 位的值各自进行互相交换

说明

此指令一般使用脉冲执行型指令，否则若采用连续执行指令，则程序每扫描一次，就会进行一次交换。

指令示例



左图中将D20 的高8 位与低8 位的值进行互相交换
右图中将D20 的高8 位与低8 位的值进行互相交换，
D21 的高8 位与低8 位的值进行互相交换，

3.6.5.3 BON

当驱动条件成立时，源址S中二进制数据的第K位状态，控制D状态。

BON — ON位数判断

16位指令	BON 连续执行/BONP 脉冲执行			
32位指令	DBON 连续执行/DBONP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	数据或数据存储字软元件地址	–	INT/DINT
D	被控位	被控位元件	–	BOOL
n	指定位	源址S中被指定的位0≤ n ≤15 (16 位)，0≤ n ≤31 (32 位)	0-15/31	INT/DINT

表3–129 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	√[1]	√	√	-	-	-	-	-	-
n	-	-	-	√	√	√	√	-	-

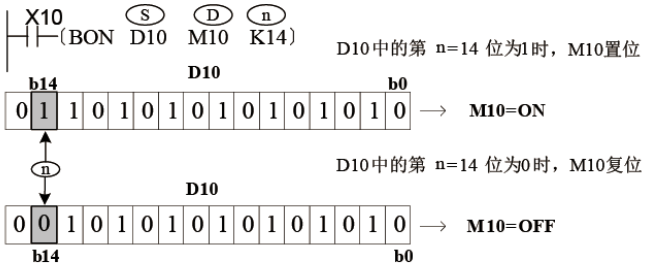
说明

[1]不支持X元件。

功能和指令说明

判断S的第n位的状态，结果存入D。

指令示例



X10由ON变成Off 时，M10 仍保持之前的状态。

3.6.5.4 SUM

当驱动条件成立时，对源址S中表示的二进制数据中为“1”的个数进行统计，统计结果存放到D。

SUM — 统计ON位数指令

16位指令	SUM 连续执行/SUMP 脉冲执行			
32位指令	DSUM 连续执行/DSUMP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	被统计数	被统计数或数据存储软元件地址	–	INT/DINT
D	统计结果	数据存储软元件地址	–	INT/DINT

表3–130 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

计算S的BIN进制值中为1的位个数，存入D。

使用D SUM和D SUM P指令的情况下，（S+1，S）的32位中的1的个数写入D，同时D+1全部为0。

若S中的位全部为零，则零标志位M8020会置ON。

指令示例

3.6.5.5 RAND

产生设定范围内的随机数

RAND — 产生有范围限制的随机数据

16位指令	—			
32位指令	RAND 连续执行/RANDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	随机数下限	随机数的下限		DINT
S2	随机数上限	随机数的上限		DINT
S3	随机数种子	随机数计算种子，作为输入，生成的随机数随种子变化		DINT
D	产生随机数	产生的随机数值		DINT

表3–131 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

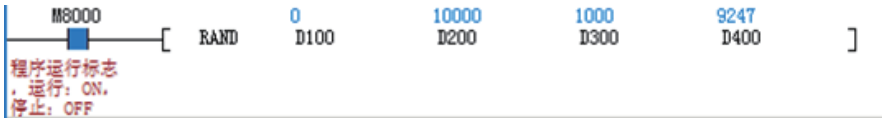
功能和指令说明

产生设定范围内的随机数。随机数由随机数种子产生，范围已确定的情况下，随机数种子与生成的随机数一一对应，即随机数种子变化，生成的随机数随之变化。

指令参数说明：

- 下限：随机数的范围下限。
- 上限：随机数的范围上限。
- 随机数种子：用于产生随机数的输入，不受上下限限制。
- 随机数：产生的随机数结果，该指令产生介于范围下限和范围上限之间的随机数。如果范围S1>S2，会报错。

指令示例



	元件名称	数据类型	显示格式	当前值
1	D100	DINT	十进制	0
2	D200	DINT	十进制	10000
3	D300	DINT	十进制	1000
4	D400	DINT	十进制	9247
5				

3.6.5.6 XCH

当驱动条件成立时，将S和D中的数据互换。

XCH — 交换指令

16位指令	XCH 连续执行/XCHP 脉冲执行			
32位指令	DXCH 连续执行/DXCHP 脉冲执行			
操作数	名称	描述	范围	数据类型

S	数据1	进行数据交换的数据存储字软元件1	–	INT/DINT
D	数据2	进行数据交换的数据存储字软元件2	–	INT/DINT

表3–132 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

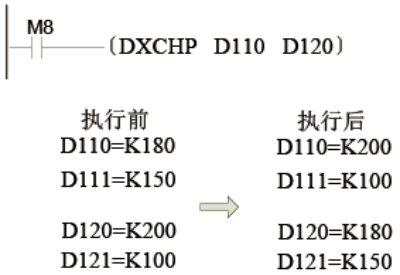
功能和指令说明

需要触点驱动，有2个操作变量，用于将S和D的值彼此交换。

指令示例

示例一

示例二



3.6.5.7 ABS

整数的绝对值计算。

ABS — 绝对值指令

16位指令	ABS 连续执行/ABSP 脉冲执行			
32位指令	DABS 连续执行/DABSP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	求取绝对值的源数据	–	INT16/DINT32
D	绝对值	得到的绝对值	–	INT16/DINT32

表3–133 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

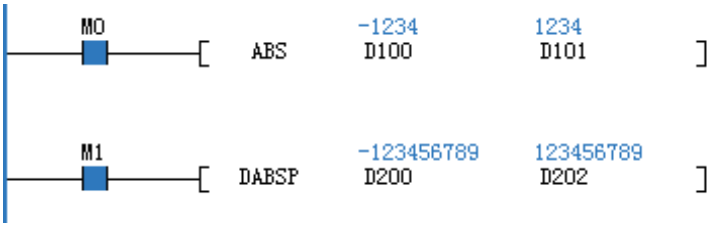
功能和指令说明

需要触点驱动，有2个操作变量，将整数S的值绝对值赋值给D。

当为32bit指令（DABS）时，S和D都会使用相邻高16bit地址的变量单元参与运算。

S：待计算绝对值的整数。

D：绝对值计算结果。



3.6.5.8 EABS

浮点数的绝对值计算。

EABS — 浮点绝对值指令

16位指令	—			
32位指令	DEABS 连续执行/DEABSP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	求取绝对值的源数据	–	REAL32
D	绝对值	得到的绝对值	–	REAL32

表3–134 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-

功能和指令说明

该指令是计算单精度浮点数的绝对值。

S：待计算绝对值的浮点数。

D：绝对值计算结果。

3.6.5.9 PID

完成PID运算，用于闭环控制系统参数的控制。

PID — PID运算指令

16位指令	PID 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	目标值	PID设定目标值	–	INT16
S2	反馈值	实测的反馈值	–	INT16
S3	运算参数	存放运算结果的起始单元	–	INT16，数组*32
D	输出值	PID输出值存放单元	–	INT16，数组*2

表3–135 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
S3	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

待续

3.6.5.10 CCD

当驱动条件成立时，将以S为首址的K个数据进行求和校验，结果存放到D；异或校验，结果存放到D+1。

CCD — 校验码指令

16位指令	CCD 连续执行/CCDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	数据源	待求校验和运算的变量启始地址，使用后续地址的变量单元	–	INT，数组*n
D	运算结果	D用于存放校验和；D＋1单元用于存放逐项异或逻辑运算结果	–	INT，数组*2
n	校验字节数	校验的变量字节数	1-256	INT

表3–136 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	-	√	-	-

功能和指令说明

该指令是对S启始的n个变量进行两种校验和运算，将直接加法的求和运算结果存于D；将逐个异或逻辑运算的结果存于D＋1单元中。本指令用于作通信时，为了确保数据传输时的正确性所做的字符串总和检查（SumCheck）。其中：

“累加和”就是将指定的n个变量直接相加的计算结果。

“异或”逻辑运算的方法则是：

- 将参与运算的变量都换算为二进制数；
- 先统计每个变量的bit0为1的个数，若为偶数个，则异或结果的bit0为0；若为奇数个，则异或结果的bit0为1；
- 再统计每个变量的bit1为1的个数，若为偶数个，则异或结果的bit1为0；若为奇数个，则异或结果的bit1为1。

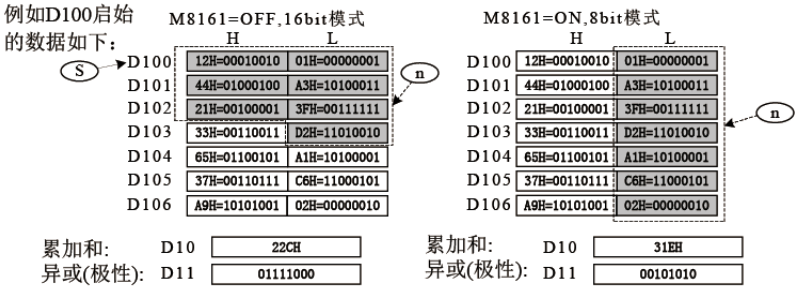
依此类推，逐个计算bit2～bit7，所得的二进制数换算为HEX数值即为异或结果（或称极性值）。

说明

编程时请悉知，HEX/ASCI/CCD等指令共用M8161模式标志。

指令示例

M8161标志决定了变量宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节都参与运算；当M8161=ON时，为8bit模式，只有变量的低字节参与运算，高字节的内容被丢弃，因此实际使用变量区域的长度增加，见下例图；



3.6.5.11 CRC

CRC（循环冗余校验）是通信中常使用的一种校验，用于计算CRC校验码。

CRC — 累加和校验码指令

16位指令	CRC 连续执行/CRCP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	校验码运算起始元件（RTU模式）	–	INT，数组*n
n	数据个数	操作数范围（K1~K256）	1-256	INT
D	结果	存放运算结果的起始元件	–	INT，数组*2

表3–137 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	-	√	-	-

功能和指令说明

LRC 校验码：由通信地址到数据内容结束加起来的值取2 的补码即为校验码。

16位转换模式：当M8161置OFF时，以16bit为单位，提取以[S]软元件开始的高8位和低8位数据共n点参与CRC运算，结果存储于[D]元件的高8位和低8位中。

8位转换模式：当M8161置ON时，以8bit为单位，提取以[S]软元件开始的低8位数据共n点参与CRC运算，结果的低8位存储于[D]软元件，高8位存储于[D+1]。

报错

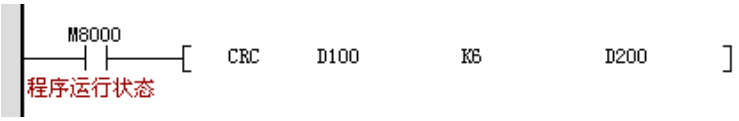
以下情况会报错误：

出错标志位M8067置ON，同时出错代码保存于D8067。

n超范围时，报6706号错误。

指令示例

M8161 置ON，8位转换模式，取D100-D105元件的低8位进行CRC运算的结果存储于D200和D201的低8位。



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1
D101	16位整数	十六进制	0x2
D102	16位整数	十六进制	0x4
D103	16位整数	十六进制	0x20
D104	16位整数	十六进制	0x0
D105	16位整数	十六进制	0x12
	16位整数	十进制	
M8161	BOOL	二进制	ON
	16位整数	十进制	
D200	16位整数	十六进制	0xF8
D201	16位整数	十六进制	0xFD

M8161 置OFF，16位转换模式，取D100-D105元件的低8位进行CRC运算的结果存储于D200高8位和低8位。

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1
D101	16位整数	十六进制	0x2
D102	16位整数	十六进制	0x4
D103	16位整数	十六进制	0x20
D104	16位整数	十六进制	0x0
D105	16位整数	十六进制	0x12
	16位整数	十进制	
M8161	BOOL	二进制	OFF
	16位整数	十进制	
D200	16位整数	十六进制	0xB202
D201	16位整数	十六进制	0x0

3.6.5.12 LRC

ASCII 模式LRC校验码运算。

LRC — ASCII 模式LRC校验码运算

16位指令	LRC 连续执行/LRCP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	验码运算起始软元件编号(ASCII模式)		INT，数组*n
n	操作个数	操作数范围 K1~K256，且为偶数	1-256	INT
D	结果	运算结果存放寄存器		INT，数组*2

表3–138 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	-	√	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

LRC 校验码：由通信地址到数据内容结束加起来的值取2 的补码即为校验码。

例如： 01 H + 03 H + 21 H + 02 H + 00 H + 02 H=29 H，然后取2 的补码=D7H（即ASCII码为44H与37H）。

16位转换模式：当M8161置OFF时，以16bit为单位，提取以[S]软元件开始的高8位和低8位数据共n点参与LRC运算，结果存储于[D]元件的高8位和低8位中。

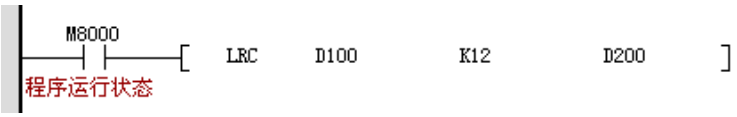
8位转换模式：当M8161置ON时，以8bit为单位。提取以[S]软元件开始的低8位数据共n点参与LRC运算，结果的低8位存储于[D]软元件，高8位存储于[D+1]。

报错

以下一些情况会报错误：

- 超范围时，报错误。
- 为奇数时，报错误。

指令示例



1. 16位模式（M8161 = OFF）

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x3130
D101	16位整数	十六进制	0x3330
D102	16位整数	十六进制	0x3132
D103	16位整数	十六进制	0x3230
D104	16位整数	十六进制	0x3030
D105	16位整数	十六进制	0x3230
D200	16位整数	十六进制	0x3744
M8161	BOOL	二进制	OFF
	16位整数	十进制	

2. 8位模式（M8161 = ON）

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x30
D101	16位整数	十六进制	0x31
D102	16位整数	十六进制	0x30
D103	16位整数	十六进制	0x33
D104	16位整数	十六进制	0x32
D105	16位整数	十六进制	0x31
D106	16位整数	十六进制	0x30
D107	16位整数	十六进制	0x32
D108	16位整数	十六进制	0x30
D109	16位整数	十六进制	0x30
D110	16位整数	十六进制	0x30
D111	16位整数	十六进制	0x32
D112	16位整数	十六进制	0x0
D200	16位整数	十六进制	0x44
D201	16位整数	十六进制	0x37
M8161	BOOL	二进制	ON
	16位整数	十六进制	

3.6.5.13 RAMP

当驱动条件成立时，在S3个扫描周期后，D的值由S1线性变化到S2。

RAMP — 斜坡信号指令

16位指令	RAMP 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	起始值	斜坡起始值数据存储字软元件地址	–	INT
S2	终止值	斜坡终止址数据存储字软元件地址	–	INT
D	当前值	斜坡当前值数据存储字软元件地址	–	INT，数组*2
n	周期数	完成斜坡变化的扫描周期数	1-32767	INT

表3–139 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	-	-	-	√	-	-

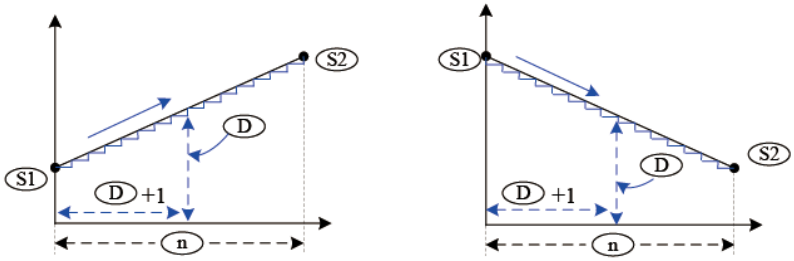
功能和指令说明

该指令的功能是在给定的两个数据中间，在指定的时间区间，进行线性插值，按扫描执行的时间依次输出过程值，直到区间末端的终点值为止。其中：

- S1为斜坡信号的起始值单元；
- S2为斜坡信号的终点值单元；
- D为线性插值信号的过程值存放单元，而插值次数的计时器存放在D＋1单元；
- n为完成插补过程的程序扫描执行次数（范围：1~32767）。由于插值输出是在正常主循环中进行的，为了保证插值输出的线性，需要将程序执行设置为固定扫描方式。

插值计算按整型数计算，丢弃了计算小数。每次插值运算完毕，M8029置ON。

指令功能如下图：



3.7 矩阵指令

3.7.1 矩阵运算

3.7.1.1 指令列表

矩阵运算指令涵盖以下指令条目：

指令类别	名称	功能
矩阵运算	BK+	数据块加法运算
	BK-	数据块减法运算
	MAND	矩阵与运算
	MOR	矩阵或运算
	MXOR	矩阵异或运算
	MXNR	矩阵同或运算
	MINV	矩阵反相运算

3.7.1.2 BK+

数据块的BIN加法运算的指令。

BK+ — 数据块加法运算

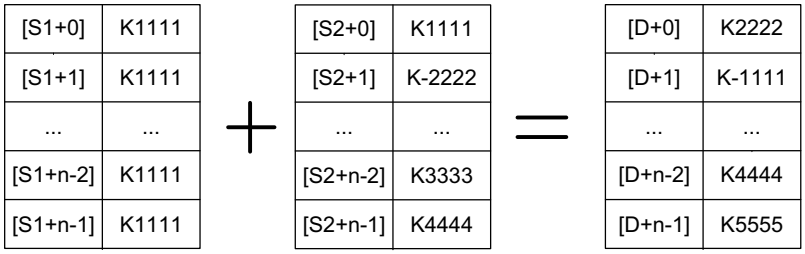
16位指令	BK+ 连续执行/BK+P 脉冲执行			
32位指令	DBK+ 连续执行/DBK+P 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	源地址	保存执行加法运算的数据的软元件起始编号		INT/DINT，数组*n
S2	源地址	执行加法运算的常数，或是保存执行加法运算的数据的软元件起始编号		INT/DINT，数组*n
D	目的地址	保存运算结果的软元件起始编号		INT/DINT，数组*n
n	操作个数	数据的个数	1-256	INT/DINT

表3–140 软元件列表

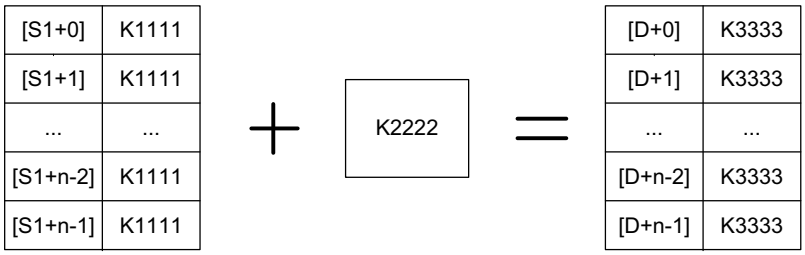
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将[S1]开始的n个（16/32位）数据和[S2]开始的n个(16/32位)数据进行加法运算，运算结果存储于[D]开始的ｎ个（16/32位）单元中。



可以在[S2]中直接设定（16/32位）有符号常数。



若从[S1]、[S2]、[D]开始的软元件超出了该类元件的范围，则会报错误且指令不执行。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	1111
D101	16位整数	十进制	1111
D102	16位整数	十进制	1111
D103	16位整数	十进制	1111
D104	16位整数	十进制	1111
D105	16位整数	十进制	0
D200	16位整数	十进制	2222
D201	16位整数	十进制	2222
D202	16位整数	十进制	2222
D203	16位整数	十进制	2222
D204	16位整数	十进制	2222
D205	16位整数	十进制	0
R100	16位整数	十进制	3333
R101	16位整数	十进制	3333
R102	16位整数	十进制	3333
R103	16位整数	十进制	3333
R104	16位整数	十进制	3333
R105	16位整数	十进制	0

3.7.1.3 BK-

数据块的BIN减法运算的指令。

BK- — 数据块减法运算

16位指令	BK- 连续执行/BK-P 脉冲执行			
32位指令	DBK- 连续执行/DBK-P 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	源地址	保存减法运算数据的软元件起始编号		INT/DINT，数组*n
S2	源地址	要进行减法运算的常数，或是保存进行减法运算的数据的软元件起始编号		INT/DINT，数组*n
D	目的地址	保存运算结果的软元件起始编号		INT/DINT，数组*n
n	操作个数	数据的个数	1-256	INT/DINT

表3–141 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	√	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

将[S1]开始的n个（16/32位）数据和[S2]开始的n个(16/32位)数据进行减法运算，运算结果存储于[D]开始的n个（16/32位）单元中。

[S1+0]	K1111
[S1+1]	K1111
...	...
[S1+n-2]	K1111
[S1+n-1]	K1111

—

[S2+0]	K1111
[S2+1]	K-2222
...	...
[S2+n-2]	K3333
[S2+n-1]	K4444

=

[D+0]	K0
[D+1]	K3333
...	...
[D+n-2]	K-2222
[D+n-1]	K-3333

可以在[S2]中直接设定（16/32位）有符号常数。

[S1+0]	K1111
[S1+1]	K1111
...	...
[S1+n-2]	K1111
[S1+n-1]	K1111

— K-2222 =

[D+0]	K3333
[D+1]	K3333
...	...
[D+n-2]	K3333
[D+n-1]	K3333

若从[S1]、[S2]、[D]开始的软元件超出了该类元件的范围，则会报错误且指令不执行。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3333
D101	16位整数	十进制	3333
D102	16位整数	十进制	3333
D103	16位整数	十进制	3333
D104	16位整数	十进制	3333
D105	16位整数	十进制	0
D200	16位整数	十进制	2222
D201	16位整数	十进制	2222
D202	16位整数	十进制	2222
D203	16位整数	十进制	2222
D204	16位整数	十进制	2222
D205	16位整数	十进制	0
R100	16位整数	十进制	1111
R101	16位整数	十进制	1111
R102	16位整数	十进制	1111
R103	16位整数	十进制	1111
R104	16位整数	十进制	1111
R105	16位整数	十进制	0

3.7.1.4 MAND

对矩阵进行与运算，并将运算结果存储在D元件中。

MAND — 矩阵与运算

16位指令	MAND 连续执行/MANDP 脉冲执行			
32位指令	DMAND 连续执行/DMANDP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	矩阵1	参与运算操作数元件1	-	INT/DINT，数组*n
S2	矩阵2	参与运算操作数元件2	-	INT/DINT，数组*n

D	运算结果	运算结果保存元件起始元件	-	INT/DINT，数组*n
n	组数	需要参与计算的组数，取值范围1~256	1-256	INT/DINT

表3–142 软元件列表

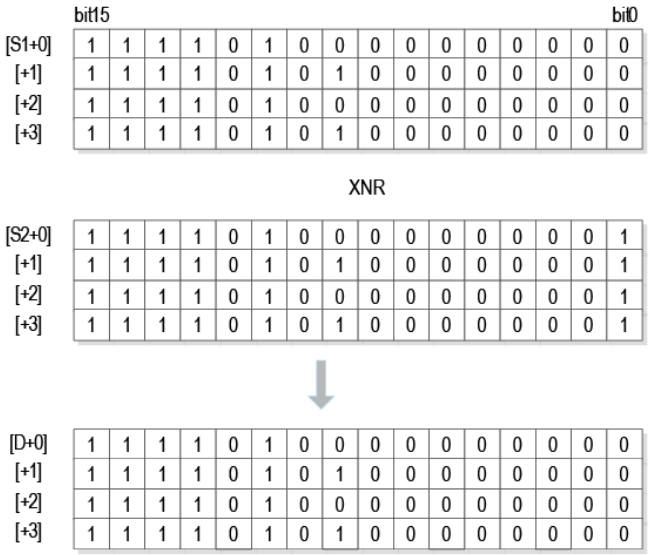
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

软元件[S1]开始的n字节数据与软元件[S2]开始的n字节数据进行按位与（AND）运算，结果存储于[D]开始的软元件中。

矩阵的与（AND）运算的规则为两bit均为1时结果为1，否则为0。

假如n=4，矩阵与运算如下所示：



指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0xFFFF
D101	16位整数	十六进制	0xFFFF
D102	16位整数	十六进制	0xFFFF
	16位整数	十进制	
D200	16位整数	十六进制	0xFF
D201	16位整数	十六进制	0xFF00
D202	16位整数	十六进制	0xAAAA
	16位整数	十进制	
R100	16位整数	十六进制	0xFF
R101	16位整数	十六进制	0xFF00
R102	16位整数	十六进制	0xAAAA

3.7.1.5 MOR

对矩阵进行或运算，并将运算结果存储在D元件中。

MOR — 矩阵或运算

16位指令	MOR连续执行/MORP 脉冲执行			
32位指令	DMOR 连续执行/DMORP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	矩阵1	参与运算操作数元件1	-	INT/DDINT，数组*n
S2	矩阵2	参与运算操作数元件2	-	INT/DDINT，数组*n
D	运算结果	运算结果保存元件起始元件	-	INT/DDINT，数组*n
n	组数	需要参与计算的组数，取值范围1~256	1-256	INT/DDINT

表3–143 软元件列表

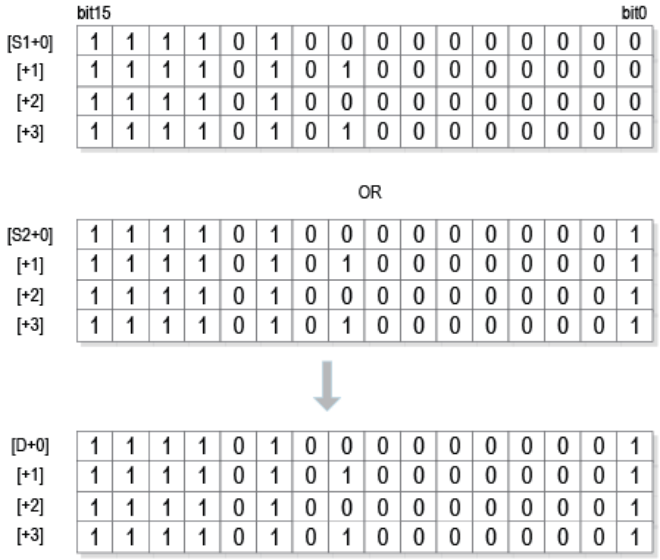
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

软元件[S1]开始的n字节数据与软元件[S2]开始的n字节数据进行按位或（OR）运算，结果存储于[D]开始的软元件中。

矩阵的或（OR）运算的规则为任意一bit均为1时结果为1，否则为0。

假如n=4，矩阵与运算如下所示：



指令示例

3.7.1.6 MXNR

对矩阵进行同或运算，并将运算结果存储在D元件中。

MXNR — 矩阵用或运算

16位指令	MXNR 连续执行/MXNRP 脉冲执行			
32位指令	DMXNR 连续执行/DMXNRP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	矩阵1	参与运算操作数元件1	-	INT/DINT，数组*n
S2	矩阵2	参与运算操作数元件2	-	INT/DINT，数组*n
D	运算结果	运算结果保存元件起始元件	-	INT/DINT，数组*n
n	组数	需要参与计算的组数，取值范围1~256	1-256	INT/DINT

表3–144 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

软元件[S1]开始的n字节数据与软元件[S2]开始的n字节数据进行按位异或（XNR）运算，结果存储于[D]开始的软元件中。

矩阵的异或（XNR）运算的规则为两bit不同时结果为1，否则为0。

假如n=4，矩阵与运算如下所示：

	bit15															bit0
[S1+0]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+1]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
[+2]	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
[+3]	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0

OR(按位同或)

[S2+0]	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1
[+1]	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
[+2]	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1
[+3]	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1

[D+0]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+1]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+2]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0
[+3]	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x0
D101	16位整数	十六进制	0x0
D102	16位整数	十六进制	0x0
	16位整数	十进制	
D200	16位整数	十六进制	0x0
D201	16位整数	十六进制	0x0
D202	16位整数	十六进制	0xAAAA
	16位整数	十进制	
R100	16位整数	十六进制	0xFFFF
R101	16位整数	十六进制	0xFFFF
R102	16位整数	十六进制	0x5555

3.7.1.7 MXOR

对矩阵进行异或运算，并将运算结果存储在D元件中。

MXOR — 矩阵异或运算

16位指令	MXOR 连续执行/MXORP 脉冲执行			
32位指令	DMXOR 连续执行/DMXORP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	矩阵1	参与运算操作数元件1	-	INT/DINT，数组*n
S2	矩阵2	参与运算操作数元件2	-	INT/DINT，数组*n
D	运算结果	运算结果保存元件起始元件	-	INT/DINT，数组*n
n	组数	需要参与计算的组数	1-256	INT/DINT

表3-145 软元件列表

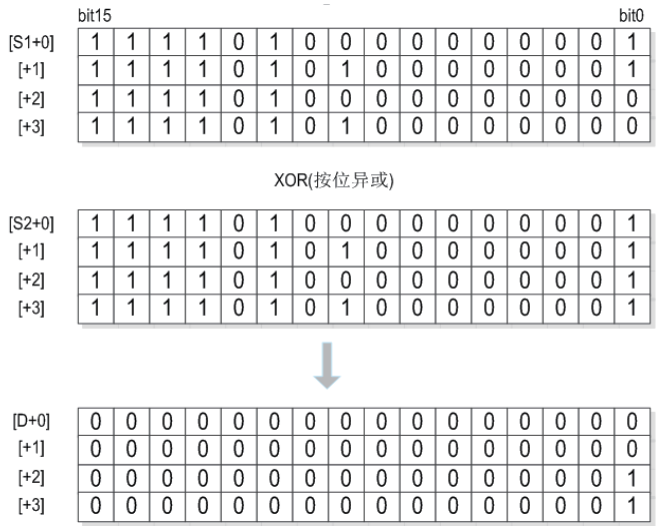
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	-	-	-
S2	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

软元件[S1]开始的n字节数据与软元件[S2]开始的n字节数据进行按位异或（XOR）运算，结果存储于[D]开始的软元件中。

矩阵的异或（XOR）运算的规则为两bit不同时结果为1，否则为0。

假如n=4，矩阵与运算如下所示：



指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x0
D101	16位整数	十六进制	0x0
D102	16位整数	十六进制	0x5555
	16位整数	十进制	
D200	16位整数	十六进制	0x0
D201	16位整数	十六进制	0x0
D202	16位整数	十六进制	0xAAAA
	16位整数	十进制	
R100	16位整数	十六进制	0x0
R101	16位整数	十六进制	0x0
R102	16位整数	十六进制	0xFFFF

3.7.1.8 MINV

对指定矩阵中的所有位取反。

MINV — 矩阵反相运算

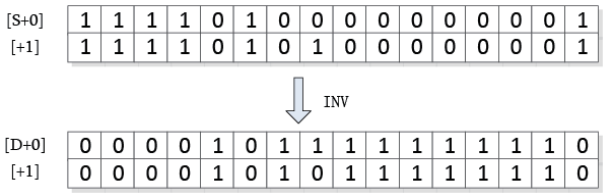
16位指令	MINV 连续执行/MINVP 脉冲执行			
32位指令	DMINV 连续执行/DMINVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	矩阵	参与运算操作数元件	-	INT/DINT，数组*n
D	运算结果	运算结果保存元件起始元件	-	INT/DINT，数组*n
n	组数	需要参与计算的组数，取值范围1~256	1-256	INT/DINT

表3–146 软元件列表

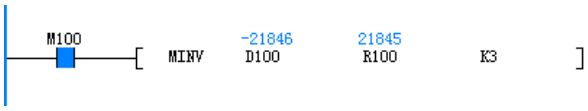
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	√	-	-	-
D	-	-	-	√	√	√	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

软元件[S]开始的n字节数据进行按位取反的运算，结果存储于[D]开始的软元件中。



指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0xAAAA
D101	16位整数	十六进制	0xAAAA
D102	16位整数	十六进制	0xAAAA
	16位整数	十进制	
R100	16位整数	十六进制	0x5555
R101	16位整数	十六进制	0x5555
R102	16位整数	十六进制	0x5555

3.7.2 矩阵比较

3.7.2.1 指令列表

矩阵比较指令涵盖以下指令条目：

指令类别	名称	功能
矩阵比较	BKCMP=	矩阵等于比较(S1＝S2)
	BKCMP>	矩阵大于比较(S1＞S2)
	BKCMP<	矩阵小于比较(S1＜S2)
	BKCMP<>	矩阵不等于比较(S1≠S2)
	BKCMP<=	矩阵小于等于比较(S1≤S2)
	BKCMP>=	矩阵大于等于比较(S1≥S2)

3.7.2.2 BKCMP#

按照各个指令的比较条件来比较数据块。

BKCMP= — 矩阵等于比较 (S1＝S2)

BKCMP> — 矩阵大于比较 (S1＞S2)

BKCMP< — 矩阵小于比较 (S1＜S2)

BKCMP<> — 矩阵不等于比较 (S1≠S2)

BKCMP<= — 矩阵小于等于比较 (S1≤S2)

BKCMP>= — 矩阵大于等于比较 (S1≥S2)

16位指令	BKCMP= 连续执行/BKCMP=P 脉冲执行			
32位指令	DBKCMP= 连续执行/DBKCMP=P 脉冲执行			
16位指令	BKCMP> 连续执行/BKCMP>P 脉冲执行			
32位指令	DBKCMP> 连续执行/DBKCMP>P 脉冲执行			
16位指令	BKCMP< 连续执行/BKCMP<P 脉冲执行			
32位指令	DBKCMP< 连续执行/DBKCMP<P 脉冲执行			
16位指令	BKCMP<> 连续执行/BKCMP<>P 脉冲执行			
32位指令	DBKCMP<> 连续执行/DBKCMP<>P 脉冲执行			
16位指令	BKCMP>= 连续执行/BKCMP>=P 脉冲执行			
32位指令	DBKCMP>= 连续执行/DBKCMP>=P 脉冲执行			
16位指令	BKCMP<= 连续执行/BKCMP<=P 脉冲执行			
32位指令	DBKCMP<= 连续执行/DBKCMP<=P 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	比较值	比较值或是保存比较值的软元件编号	-	INT/DINT，数组*n
S2	被比较值	保存比较源数据的软元件起始编号	-	INT/DINT，数组*n
D	目的地址	保存比较结果的软元件起始编号	-	BOOL，数组*n
n	操作个数	数据的个数	1-256	INT/DINT

表3–147 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	-	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	√[1]	√	√	-	-	-	-	-	-
n	-	-	-	√	√	√	√	-	-

说明

- # 为=、>、<、<>、<=、>=之一；
- [1]不支持X元件。

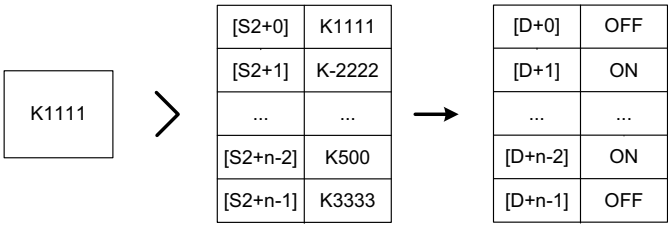
功能和指令说明

将[S1]开始的n个（16/32位）数据和[S2]开始的n个(16/32位)数据进行比较，运算结果存储于[D]开始的 n 个（16/32位）单元中。

以BKCMP> 为例子进行说明：

可以在[S1]中直接设定（16/32位）有符号常数。

以BKCMP> 为例子进行说明：



从[D]开始的n个比较结果全为ON时，M8333置ON。

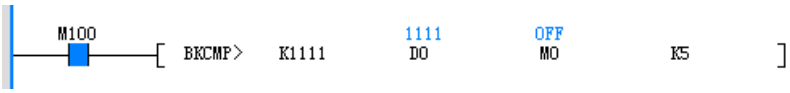
以下情况会报故障，指令不执行：

1）从[S1]、[S2]、[D]开始的软元件超出了该类元件的范围，报错误。

2）16位指令使用了32位变量，报错误。

3）32位变量请用32位运算(DBKCMP=、DBKCMP>、DBKCMP<等)指令进行比较。

指令示例



元件名称	数据类型	显示格式	当前值
D0	16位整数	十进制	1111
D1	16位整数	十进制	-2222
D2	16位整数	十进制	0
D3	16位整数	十进制	500
D4	16位整数	十进制	3333
M100	BOOL	二进制	ON
	16位整数	十进制	
M0	BOOL	二进制	OFF
M1	BOOL	二进制	ON
M2	BOOL	二进制	ON
M3	BOOL	二进制	ON
M4	BOOL	二进制	OFF

3.8 字符串指令

3.8.1 指令列表

字符串指令涵盖以下指令条目：

指令类别	名称	功能
字符串指令	STR	整数→字符串的转换
	STRMOV	字符串直接赋值指令
	VAL	字符串→整数的转换
	ESTR	二进制浮点数→字符串的转换
	EVAL	字符串→二进制浮点数的转换
	$ADD	字符串的组合
	LEN	检出字符串的长度
	INSTR	字符串的检索
	RIGHT	从字符串右侧开始取出
	LEFT	从字符串左侧开始取出
	MIDR	从字符串中任意取出
	MIDW	字符串中任意替换
	$MOV	字符串的传送

3.8.2 STR

将整数转换成字符串(ASCII码)的指令。

STR — 整数→字符串的转换

16位指令	STR 连续执行/STRP 脉冲执行			
32位指令	DSTR 连续执行/DSTRP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	转换数值	保存要转换的整数的软元件编号	-	INT/DINT，数组*2
S2	转换的数据	保存要转换数值的位数的软元件起始编号	-	INT/DINT
D	输出	保存已转换的字符串的软元件起始编号	-	INT/DINT，数组*不定

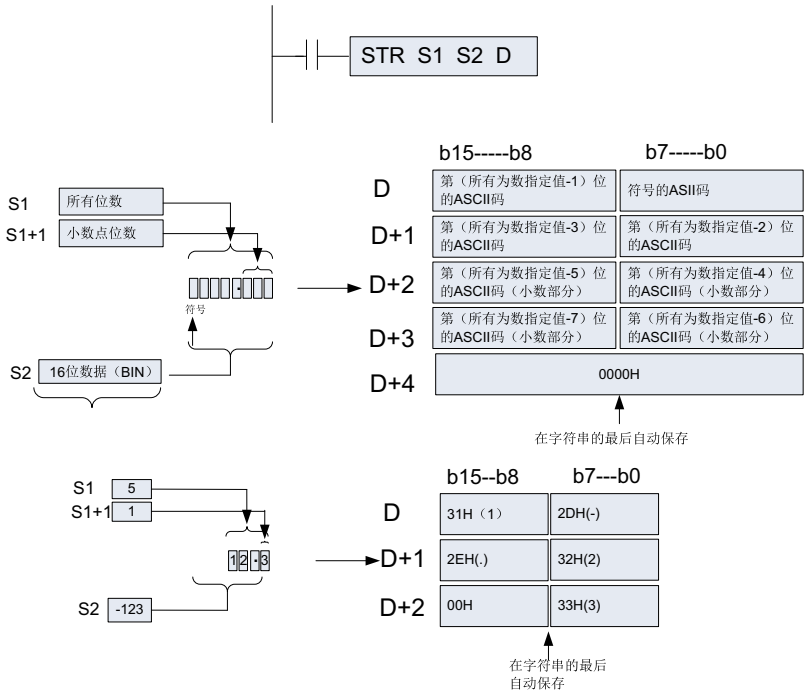
表3-148 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	-	-	-	-
S2	-	-	-	✓	✓	-	-	-	-
D	-	-	-	✓	✓	-	-	-	-

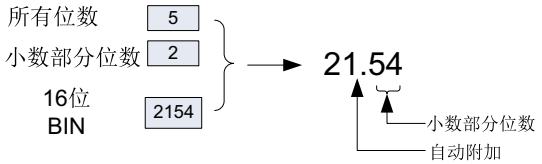
功能和指令说明

1. 16位运算(STR/STRP)

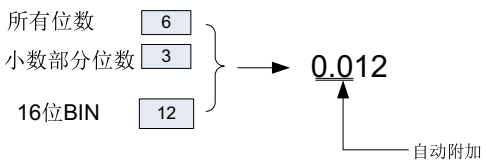
将[S2]的16位数据(BIN)，在所有位数[S1]、小数点部分位数[S1+1]指定的位置中加上小数点后，转换成字符串，保存到[D]开始的软元件中。



- 在2～8位数的范围内设定所有位数[S1]。
- 在0～5位数的范围内设定小数部分位数[S1+1]。但是，请设定为小数部分位数≦(所有位数-3)。
- 要转换的16位数据(BIN) 的值在-32768～32767的范围内。转换后的字符串数据会如下所示保存到[D]开始的软元件编号中。
- 在符号中，16位数据(BIN) [S2]为正时保存“空格(20H)”，为负时保存“-(2DH)”。
- 在小数部分位数[S1+1]中设定为 “0”以外的数字时，会自动在小数部分位数+1位数的位置上加上小数点“. (2EH)”。小数部分位数[S1+1]为“0”时，不附加小数点。

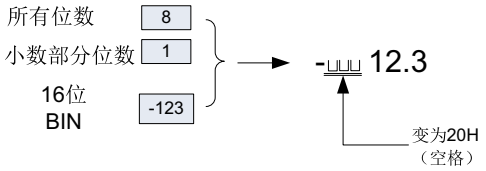


与[S2]的16位数据(BIN)的位数相比， [S1+1]的小数部分位数较多时，会自动向右对齐，在左边附加“0 (30H)”后进行转换。



如果除去小数点和符号以外，所有位数[S1]的位数多于[S2]的16位数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。

此外，[S2]的16位数据(BIN)的位数较多时，会发生错误：

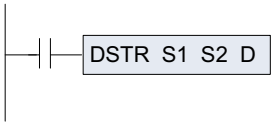


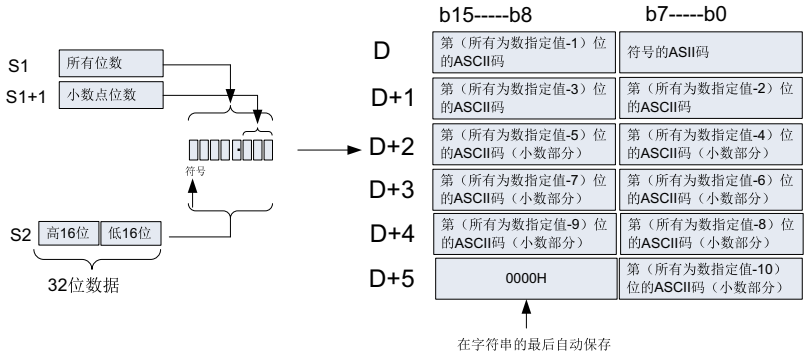
在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。

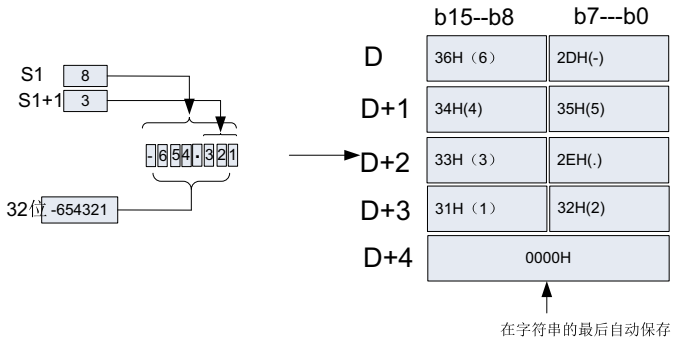
总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8位)中保存“00H”。

2. 32位运算(DSTR/DSTRP)

将[S2+1，S2]的32位数据(BIN)，在所有位数[S1]、小数点部分位数[S1+1]指定的位置中加上小数点后，转换成字符串，保存到[D]开始的软元件中。

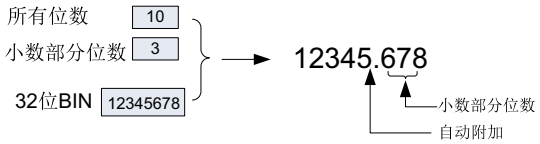




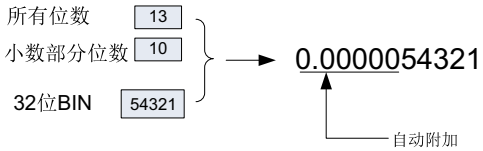


- 在2～13位数的范围内设定所有位数[S1]。
- 在0～10位数的范围内设定小数部分位数[S1+1]。但是，请设定为小数部分位数≦(所有位数-3)。
- 要转换的32位数据(BIN) 的值在 -2,147,483,648～2,147,483,647 的范围内。转换后的字符串数据会如下所示保存到[D]开始的软元件编号中。

- 在符号中，32位数据(BIN) [S2]为正时保存“空格(20H)”，为负时保存“-(2DH)”。
- 在小数部分位数[S1+1]中设定为“0”以外的数字时，会自动在小数部分位数+1位数的位置上加上小数点“. (2EH)”。小数部分位数[S1+1]为“0”时，不附加小数点。

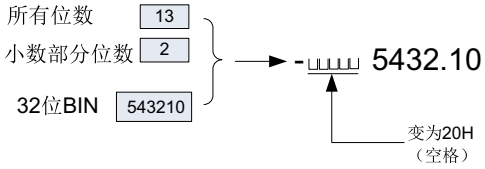


与[S2]的16位数据(BIN)的位数相比， [S1+1]的小数部分位数较多时，会自动向右对齐，在左边附加“0 (30H)”后进行转换。



如果除去小数点和符号以外，所有位数[S1]的位数多于[S2]的32位数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。

此外，[S2]的32位数据(BIN)的位数较多时，会发生错误：



在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。

总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8位)中保存“00H”。

报错

以下情况会发生运算错误：

- 所有位数[S1]超出如下所示的范围时。

指令运算	设定范围
16位运算	2~8
32位运算	2~13

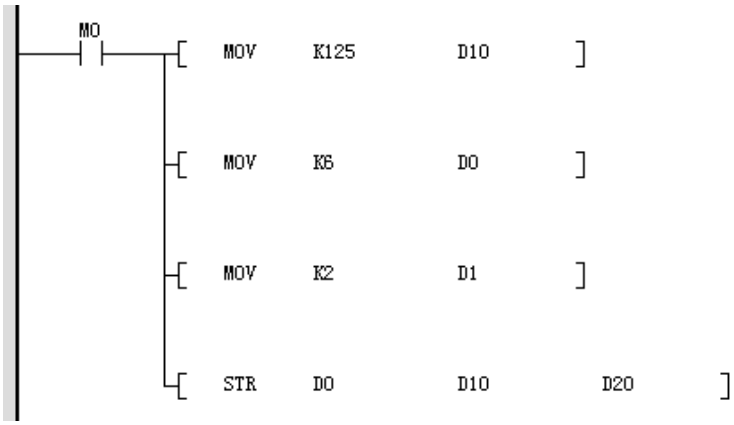
- 所有位数[S1+1]超出如下所示的范围时。

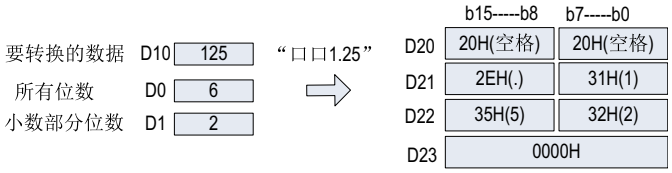
指令运算	设定范围
16位运算	0~5
32位运算	0~10

- 所有位数[S1]和小数部分位数[S1+1]的关系不符合如下所示的范围时。
  - (所有位数-3)≧小数部分位数。
  - 所有[S1+]符号、小数点的位数比[S2]的BIN数据位数少。
  - 保存字符串[D]的以后的软元件超出了相应的软元件范围。

指令示例

当M0为ON时，根据D0、D1的位数指定，将D10中保存的BIN数据(16位)转换成字符串，然后保存到D20～D23中的程序。





3.8.3 STRMOV

用于为字符串直接赋值的指令。

STRMOV — 字符串赋值指令

16位指令	STRMOV 连续执行/STRMOVP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	字符串数据	要赋值存储的字符串数据	1-127	“字符串”
D	存储寄存器	目标存储寄存器	-	INT，数组*strlen

表3–149 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	-	-	字符串[1]
D	-	-	-	√	√	-	-	-	-

说明

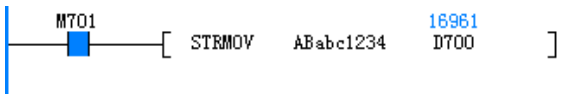
[1]仅支持直接输入“字符串”常数。

功能和指令说明

支持字符串直接输入的指令，可以借助该指令，实现通讯中字符串数据的收发。

字符串数据按照顺序依次存储。

指令示例



	元件名称	数据类型	显示格式	当前值
1	D700	16位整数	十六进制	0x4241
2	D701	16位整数	十六进制	0x6261
3	D702	16位整数	十六进制	0x3163
4	D703	16位整数	十六进制	0x3332
5	D704	16位整数	十六进制	0x34
6	D705	16位整数	十进制	0
7	D706	16位整数	十进制	0

3.8.4 VAL

将字符串(ASCII码)转换成整数的指令。

VAL — 字符串转换整数

16位指令	VAL 连续执行/VALP 脉冲执行			
32位指令	DVAL 连续执行/DVALP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	转换数值	保存要转换数值的字符串软元件起始编号	-	INT/DINT，数组*不定
D1	转换的数据	保存要转换的BIN数据的软元件编号	-	INT/DINT，数组*2
D2	输出	保存已转换的字符串的软元件起始编号	-	INT/DINT

表3–150 软元件列表

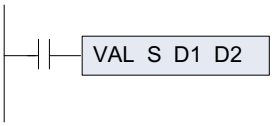
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D1	-	-	-	√	√	-	-	-	-
D2	-	-	-	√	√	-	-	-	-

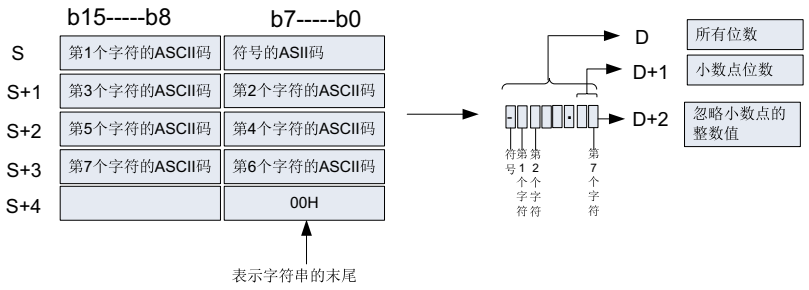
功能和指令说明

1. 16位运算(VAL/VALP)

将[S]开始的软元件中保存的字符串转换成16 位数据(BIN)，然后将所有位数保存到[D1]中、将小数部分位数保存到[D+1]中、将BIN数据保存到中[D2]。

从字符串转换成BIN时，以字节为单位将[S]开始到保存“00H”的软元件编号为止的数据，作为字符串进行处理。





- 要转换的字符串数据

  字符串的字符数数值范围（忽略小数点）

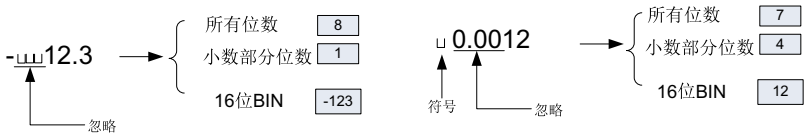
项目	数值范围
所有字符数	2~8字符
小数部分的字符数	0~5个字符
忽略小数点的数值范围	-32768~32767例如 “123.45—>12345”

可在要转换的字符中使用的字符种类

项目	字符类别
正的数值	空格（20H）
负的数值	—（2DH）
小数点	“.（2EH）”
数字	“0（30H）”~“9（39H）”

- [D1] 中保存所有位数。所有位数，就是所有的字符数(包括数字、符号、小数点)。
- [D1+1]中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。
- [D2]中忽略小数点，将字符串转换成16位的数据(BIN)。

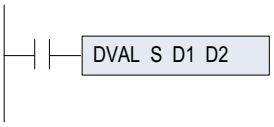
但是，在开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略，而转换成16位数据(BIN)。

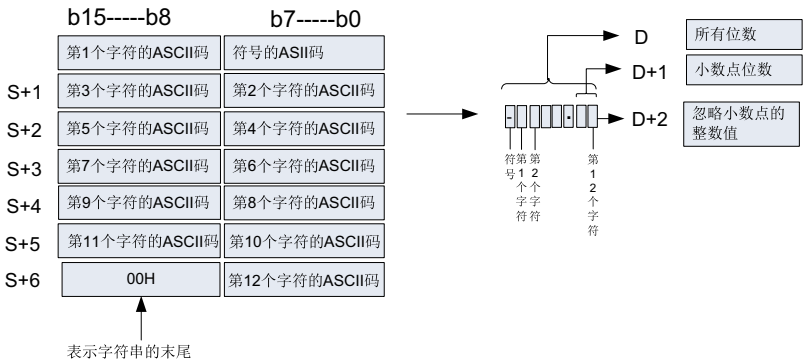


2. 32位运算(DVAL/DVALP)

将[S]开始的软元件中保存的字符串转换成32 位数据(BIN)，然后将所有位数保存到[D1]中、将小数部分位数保存到[D+1]中、将BIN数据保存到中[D2+1，D2]。

从字符串转换成BIN时，以字节为单位将[S]开始到保存“00H”的软元件编号为止的数据，作为字符串进行处理。





- 要转换的字符串数据

  字符串的字符数数值范围（忽略小数点）

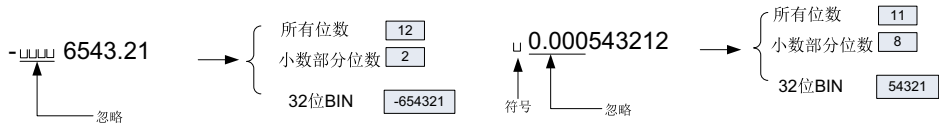
项目	数值范围
所有字符数	2~8字符
小数部分的字符数	0~10个字符
忽略小数点的数值范围	-2,147,483,64 8～2,147,483,647例如 “123.45—>12345”

可在要转换的字符中使用的字符种类

项目	字符类别
正的数值	空格（20H）
负的数值	—（2DH）
小数点	“.（2EH）”
数字	“0（30H）”~“9（39H）”

- [D1] 中保存所有位数。所有位数，就是所有的字符数(包括数字、符号、小数点)。
- [D1+1]中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。
- [D2+1，D2]中忽略小数点，将字符串转换成32位的数据(BIN)。

但是，在开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略，而转换成32位数据(BIN)。



说明

- 符号数据“空格(20H)”或是“-(2DH)”，必须保存在第1个字节(在[S]中设定的起始软元件的低8位)里。
- 此外，从[S]的第2个字节开始到字符串末尾00H的ASCII码数据，只能保存“0(30H)”～“9(39H)”、“空格(20H)”以及小数点“. (2EH)”。在第2个字节后保存“-(2DH)”，会发生运算错误。

报错

以下情况会发生运算错误：

- 要转换的字符串[S]超出如下所示的范围。

指令运算	设定范围
16位运算	2~8
32位运算	2~13

- 要转换的字符串[S]的小数部分超出如下所示的范围。

指令运算	设定范围
16位运算	0~5
32位运算	0~10

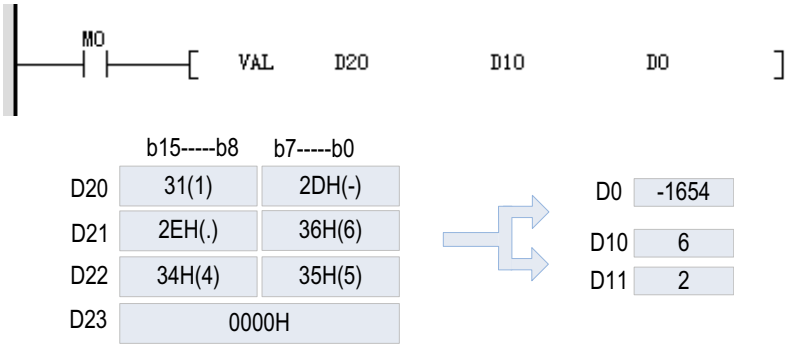
- 要转换的字符串（[S]以后）的所有字符数，以及小数部分的字符数之间的关系不符合如下所示的范围：
  - (所有位数-3) ≧ 小数部分位数。
  - 在符号中设定了“空格(20H)”、“-(2DH)”以外的ASCII码。
  - 各数字的位数中设定了“0(30H)”～“9(39H)”，以及小数点“.(2EH)”以外的ASCII码。
  - 要转换的字符串([S]以后)中设定了多个小数点“.(2EH)”。
- 转换后的BIN数据超出如下所示的范围。

指令运算	设定范围
16位运算	-32768~32767
32位运算	-2,147,483,648～2,147,483,647

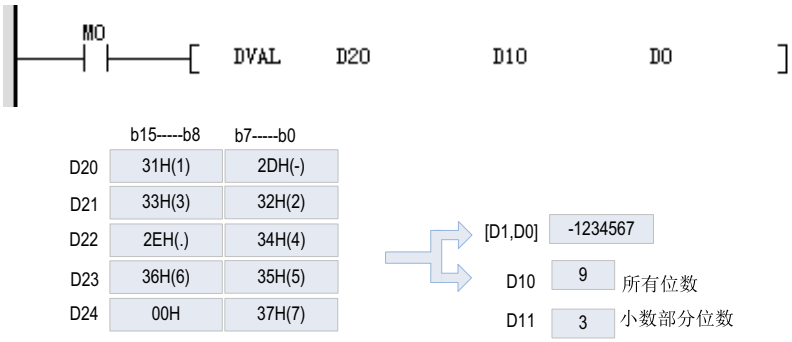
- [S]从开始到相应软元件的最后软元件编号之间不存在“00H”。

指令示例

- 当M0为ON后，将D20～D22中保存的字符串数据视为整数值转换成BIN值，然后保存到D0中的程序。



- 当M0为ON后，将D20～D24中保存的字符串数据视为整数值转换成BIN值，然后保存到[D1，D0]中的程序。



3.8.5 ESTR

将2进制浮点数(实数)转成指定位数的字符串(ASCII码)。

ESTR — 浮点数转换字符串

16位指令	—			
32位指令	DESTR 连续执行/DESTRP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	操作数	要转换的2进制浮点数数据的软元件起始编号	–	REAL
S2	起始编号	保存要转换数值的显示指定的软元件起始编号	–	DINT，数组*3
D	结果	保存已转换的字符串的软元件起始编号	–	DINT，数组*不定

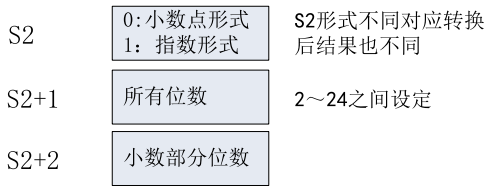
表3–151 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

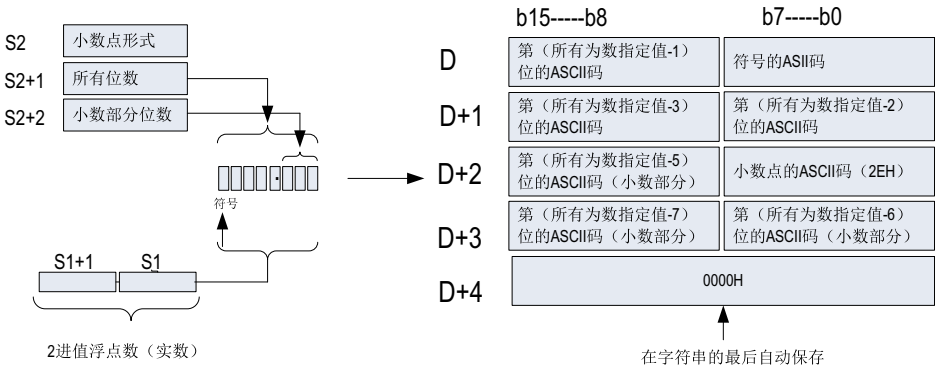
功能和指令说明

32位运算 (DESTR)

根据[S2，S2+1，S2+2]中指定的内容，将[S+1，S]中的内容转成字符串，并将结果保存到D开始的软元件中。



- 小数点形式

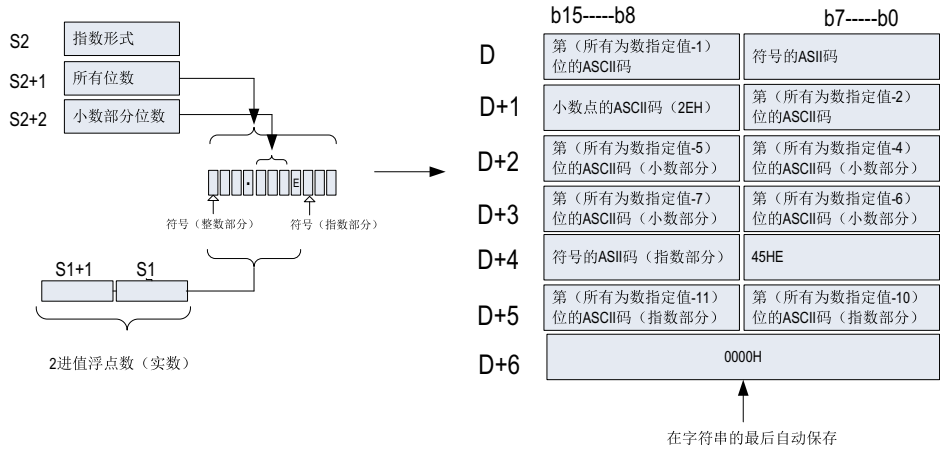


在[S2+1]中可以指定的所有位数如下（最大24位）：

- 小数部分的位数为0时，所有位数 ≥ 2。
- 小数部分的位数不为0时，所有位数 ≥（小数部分位数+3）。

在[S2+2]中可以指定的小数部分为0~7位数，但请设定为小数部分位数≤（所有位数-3）。

- 指数形式



在[S2+1]中可以指定的所有位数如下（最大24位）：

- 小数部分的位数为0时，所有位数 ≥ 6。
- 小数部分的位数不为0时，所有位数 ≥（小数部分位数+7）。

在[S2+2]中可以指定的小数部分为0~7位数，但请设定为小数部分位数≤（所有位数-3）。

报错

以下情况会出现运算错误，错误标志位(M8067)置ON，错误代码保存在D8067中。

- [S1]不在下列范围中时。(错误代码: K6706)
- [S2]中指定的形式为0、1以外时。(错误代码: K6706)
- [S2+1]中指定的所有位数指定不符合下面的范围时。(错误代码: K6706)
  小数点形式

  小数部分的位数为“0”时 ……所有位数≧2

  小数部分的位数不为“0”时……所有位数≧(小数部分位数+3)

  指数形式

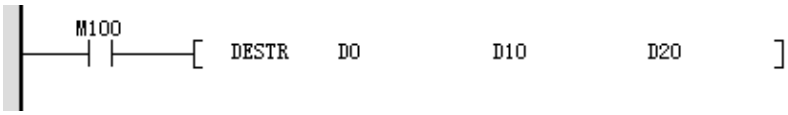
  小数部分的位数为“0”时 ……所有位数≧6

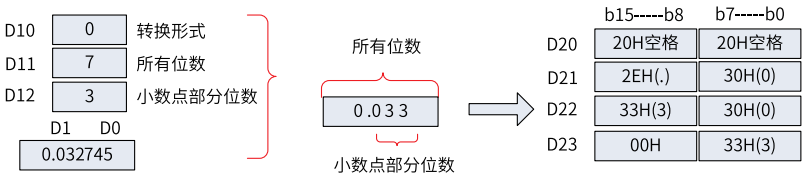
  小数部分的位数不为“0”时……所有位数≧(小数部分位数+7)
- [S+2]中指定的小数部分的位数指定不符合下面范围时。(错误代码: K6706)
  小数点形式时: 小数点位数≦(所有位数-3)

  指数形式时： 小数点位数≦(所有位数-7)
- 当保存指定字符串的软元件[D]的范围，超出相应软元件的范围时。(错误代码: K6705)
- 转换结果超出已指定的所有位数时。(错误代码: K6705)

指令示例

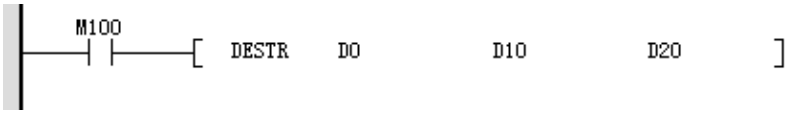
- M100为ON时，根据D10～D12中指定的内容（小数点形式），将D0、D1的内容(2进制浮点数数据)做转换，并且保存在D20以后的软元件中的程序。

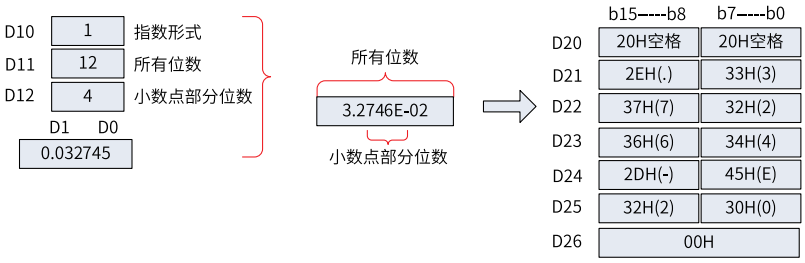




元件名称	数据类型	显示格式	当前值
D0	浮点数	十进制	0.032745
D10	16位整数	十进制	0
D11	16位整数	十进制	7
D12	16位整数	十进制	3
M100	BOOL	二进制	ON
D20	16位整数	十六进制	0x2020
D21	16位整数	十六进制	0x2E30
D22	16位整数	十六进制	0x3330
D23	16位整数	十六进制	0x33

- M100为ON时，根据D10～D12中指定的内容（指数形式），将D0、D1的内容(2进制浮点数数据)做转换，并且保存在D20以后的软元件中的程序。





元件名称	数据类型	显示格式	当前值
D0	浮点数	十进制	0.032745
D10	16位整数	十进制	1
D11	16位整数	十进制	12
D12	16位整数	十进制	4
M100	BOOL	二进制	ON
D20	16位整数	十六进制	0x2020
D21	16位整数	十六进制	0x2E33
D22	16位整数	十六进制	0x3732
D23	16位整数	十六进制	0x3534
D24	16位整数	十六进制	0x2D45
D25	16位整数	十六进制	0x3230
D26	16位整数	十六进制	0x0

3.8.6 EVAL

将字符串 (ASCII码) 转成2进制浮点数数据的指令

EVAL — 字符串转换浮点数

16位指令	—			
32位指令	DEVAL 连续执行/DEVALP 脉冲执行			
操作数	名称	描述	范围	数据类型

S	操作数	保存要转换成2进制浮点数数据的字符串数据的软元件的起始编号	–	DINT，数组*不定
D	结果	保存已转换的2进制浮点数数据的软元件的起始编号	–	REAL

表3–152 软元件列表

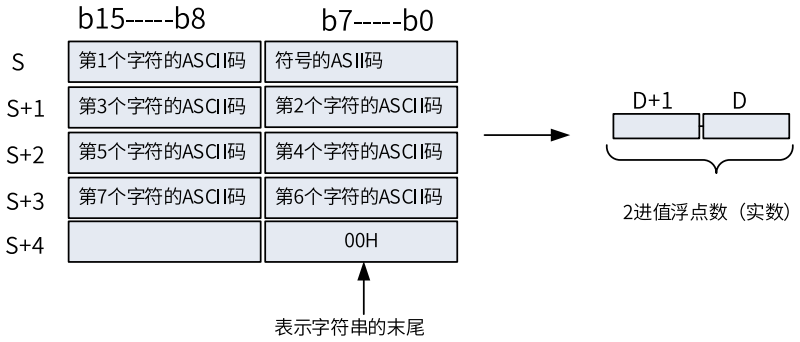
操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

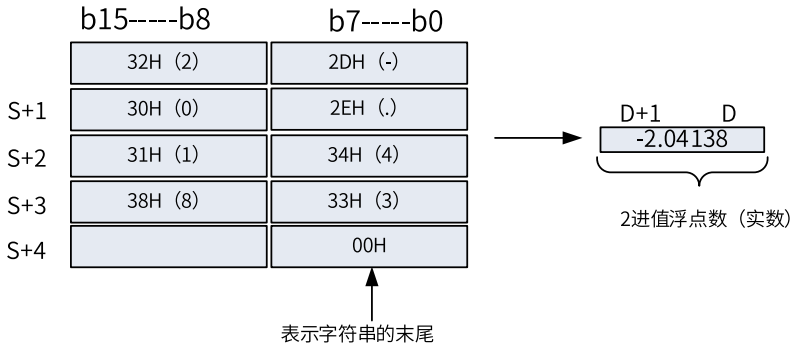
32位运算（DEVAL）

将[S] 开始的软元件中保存的字符串转换成2进制浮点数数据后，保存到 [D+1，D]中。

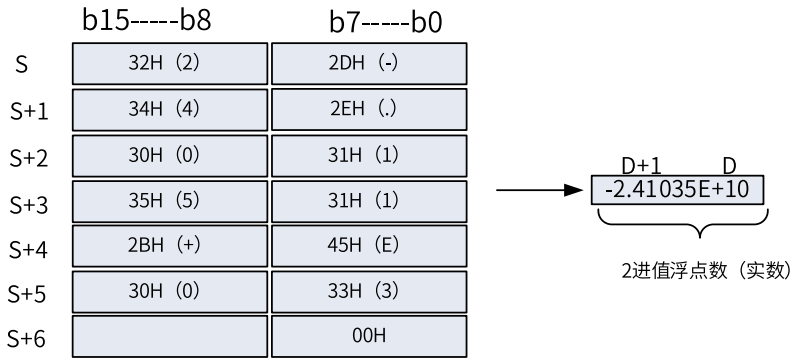
指定的字符串无论是小数点形式，还是指数形式，都可以转换成2进制浮点数数据。



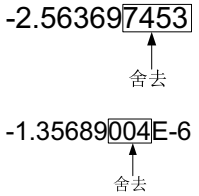
- 小数点形式



- 指数形式



[S]中指定要转换成2进制浮点数的字符串，去除了符号、小数点、指数部分仍然有7位数以上时，舍去第7位数以后的数；



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

指数形式，在指数部分的符号内指定"2BH"(+)，省略符号后作为正的指数转换。将指数部分的符号指定为"2DH"(-)则作为负的指数转换。

[S]指定的字符串中，在最初的“0”以外的数值之间如果存在“20H”(空格)或是“30H”(0)时，会忽略“20H”、“30H”而进行转换。

字符串最大可以设定到24个字符。“20H”(空格)、“30H”(0)也作为一个字符来计算。

相关软元件

软元件	名称	条件	动作
M8020	零位	转换结果真的为零（尾数部分为“0”时）	零位标志位(M8020)为ON。
M8021	借位	转换结果的绝对值＜2^{-126}	D的值小于32位实数的绝对值最小值2^{-126}部分被舍去，借位标志位（M8021）为ON。
M8022	进位	转换结果的绝对值≥2^{128}	D的值大于32位实数的绝对值最小值2^{128}部分被舍去，借位标志位（M8022）为ON。

报错

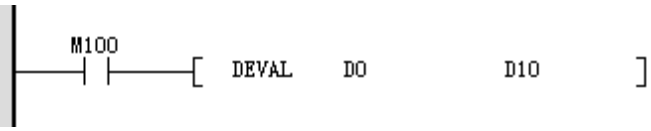
以下情况会发生运算错误：

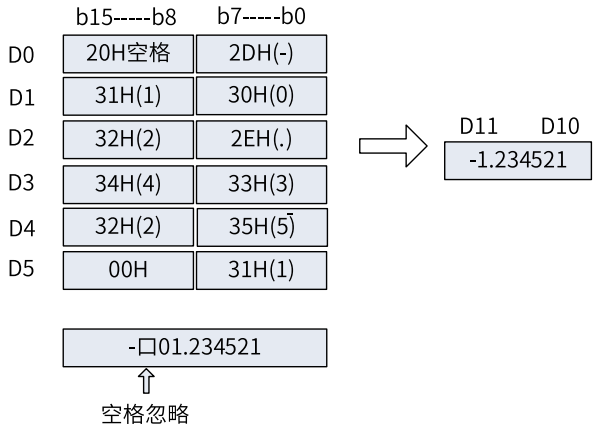
- 整数部分、小数部分中存在“30H”(0)～“39H”(9)以外的字符。
- [S]指定的字符串中存在2个以上的“2EH”(.)。
- 指数部分中有“45H”(E)、“2BH”(+)、“2DH”(-)以外的字符或是有多个指数部分。
- [s]开始的相应软元件范围内没有“00H”。

- [s]以后的字符数为0或是超出了24个字符。

指令示例

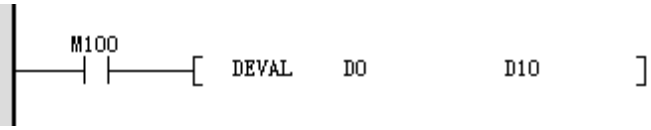
- 当M101为ON时，将D0开始的软元件中保存的字符串转换成2进制浮点数（小数点形式），并保存到D10、D11中的程序。

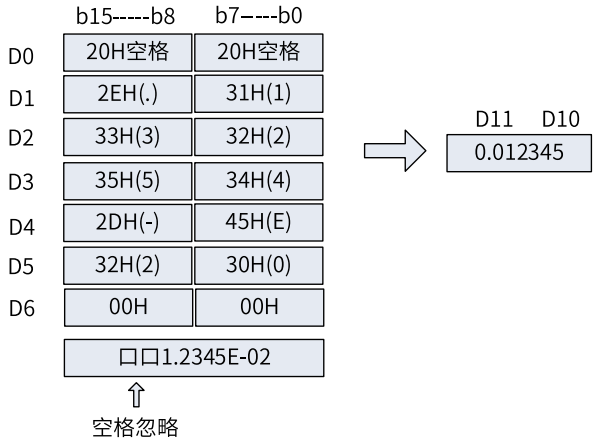




元件名称	数据类型	显示格式	当前值
D10	浮点数	十进制	-1.234521
M100	BOOL	二进制	ON
D0	16位整数	十六进制	0x202D
D1	16位整数	十六进制	0x3130
D2	16位整数	十六进制	0x322E
D3	16位整数	十六进制	0x3433
D4	16位整数	十六进制	0x3235
D5	16位整数	十六进制	0x31

- 当M100为ON时，将D0开始的软元件中保存的字符串转换成2进制浮点数（指数形式），并保存到D10、D11中的程序。





元件名称	数据类型	显示格式	当前值
D10	浮点数	十进制	0.012345
M100	BOOL	二进制	ON
D0	16位整数	十六进制	0x2020
D1	16位整数	十六进制	0x2E31
D2	16位整数	十六进制	0x3332
D3	16位整数	十六进制	0x3534
D4	16位整数	十六进制	0x2D45
D5	16位整数	十六进制	0x3230
D6	16位整数	十六进制	0x0

3.8.7 $ADD

连接字符串与字符串的指令。

$ADD — 字符串的组合

16位指令	$ADD 连续执行/$ADDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	被连接值	保存连接源数据(字符串)的软元件起始编号，或是被直接指定的字符串	-	INT，数组*不定
S2	连接值	保存要连接的数据(字符串)的软元件起始编号，字符串或是被直接指定的字符串	-	INT，数组*不定
D	连接后结果	保存连接后的数据(字符串)的软元件起始编号	-	INT，数组*不定

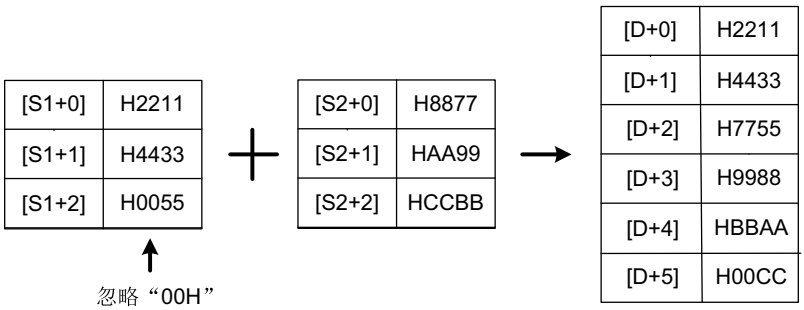
表3–153 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

将[S2]开始字符串数据连接到[S1]开始的字符串数据的结尾处，然后将新的字符串保存至[D]。

[S1]和[S2]中的字符数据以字节为单位，结束于第一个00H字节。



字符串的结合，就是指忽略中表示指定字符串末尾的“00H”，然后把最后字符接上指定的字符串。此外，执行字符串的结合后会自动将“00H”附加在最后。

连接后的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”；

连接后的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”；

[S1]、[S2]中任意一个的值从“00H”开始时（字符数为0时），在[D]中保存“0000H”。

报错

以下情况会报错误：

- 在对应[S1]和[S2]的操作数合法范围内，未找到“00H”，报错误。
- 在存储合并之后的字符串超出了[D]的操作数范围，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x2211
D101	16位整数	十六进制	0x4433
D102	16位整数	十六进制	0x55
	16位整数	十进制	
D200	16位整数	十六进制	0x8877
D201	16位整数	十六进制	0xAA99
D202	16位整数	十六进制	0xCCBB
D203	16位整数	十六进制	0xDD
	16位整数	十进制	
R100	16位整数	十六进制	0x2211
R101	16位整数	十六进制	0x4433
R102	16位整数	十六进制	0x7755
R103	16位整数	十六进制	0x9988
R104	16位整数	十六进制	0xBBAA
R105	16位整数	十六进制	0xDDCC
R106	16位整数	十进制	0

3.8.8 LEN

检测出指定字符串的字符数(字节数)的指令。

LEN — 检测字符串的长度

16位指令	LEN 连续执行/LENP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	检测数据	保存要检测出字符数的字符串的软元件起始编号	-	INT，数组*不定
D	检测结果	保存已检测出的字符串的长度(字节数)的软元件编号	-	INT，数组*不定

表3–154 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

计算出从[S]开始的字符串的字符数，将其保存到[D]中。字符串以字节为单位，从[S]开始至第一个“00H”为止。

以下情况会报错误：

1）在对应的[S]合法范围内未检出“00H”，报错误。

2）检出的字符数大于32767时，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x55
D103	16位整数	十六进制	0x0
	16位整数	十进制	

3.8.9 INSTR

从指定的字符串中检索指定字符串的指令。

INSTR — 字符串的检索

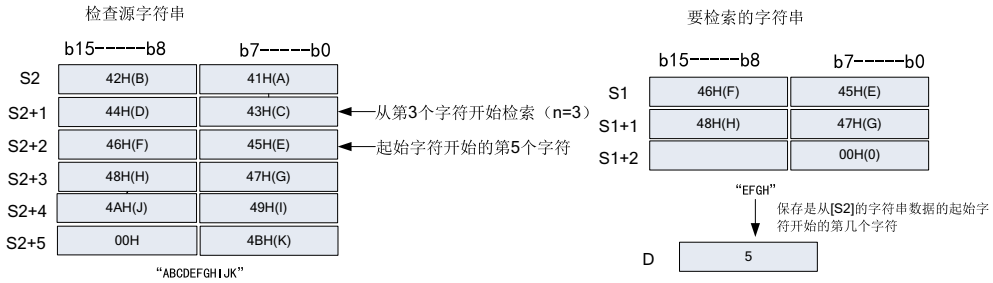
16位指令	INSTR 连续执行/INSTRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	源数据	保存要检索的字符串的软元件起始编号	-	INT，数组*不定
S2	检索源	保存检索源字符串的软元件起始编号	-	INT，数组*不定
D	检索结果	保存检索结果的软元件起始编号	-	INT
n	检索位置	开始检索的位置	1-32767	INT

表3–155 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

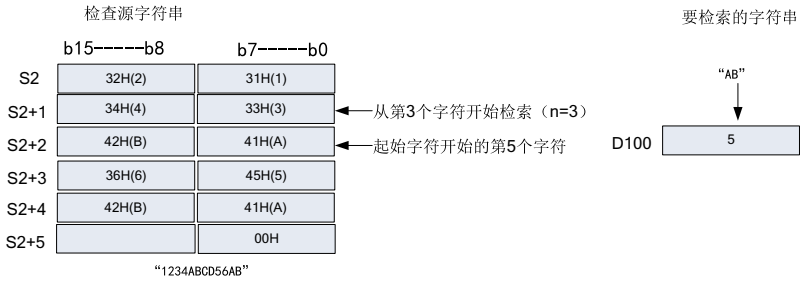
功能和指令说明

从检索源数据[S2]起始的软元件中左侧（字符串的开头）起第n个字符开始，检索与[S1]起始的软元件中相同的字符串，将检索到的字符串的起始位置信息（左测起第几个字符位置信息）保存至[D]中。



若[S2]中未检出与[S1]一致的字符串，[D]中结果保存0。

若开始检索位置n为负数或0时，指令不执行。



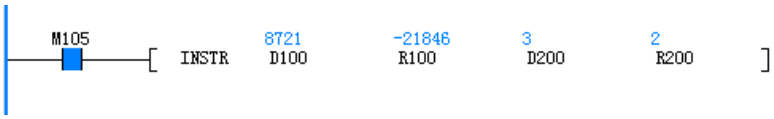
报错

以下情况会报错误：

1）检索起始位置n超过了[S2]中的字符数时，报错误。

2）[S1]或者[S2]开始的相应软元件的软元件范围内未检出“00H”时，报错误。

指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x2211
D101	16位整数	十六进制	0x4433
D102	16位整数	十六进制	0x0
	16位整数	十六进制	
D200	16位整数	十六进制	0x3
R200	16位整数	十六进制	0x2
	16位整数	十六进制	
R100	16位整数	十六进制	0xAAAA
R101	16位整数	十六进制	0x2211
R102	16位整数	十六进制	0x4433
R103	16位整数	十六进制	0xAAAA
R104	16位整数	十六进制	0xAAAA
R105	16位整数	十六进制	0x0

3.8.10 RIGHT

从指定字符串的右侧取出指定字符数的指令。

RIGHT — 从字符串右侧开始取出

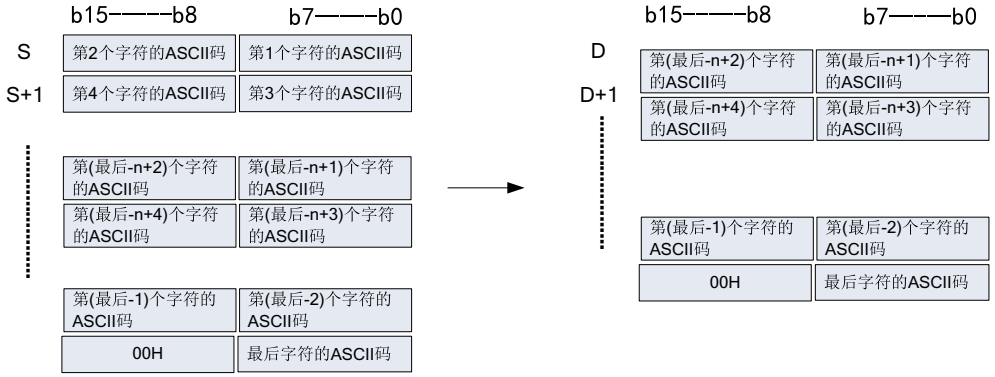
16位指令	RIGHT 连续执行/RIGHTP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	保存字符串的软元件起始编号	-	INT，数组*不定
D	取出结果	保存被取出的字符串的软元件起始编号	-	INT，数组*不定
n	取出个数	要取出的字符数	1-32767	INT

表3–156 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

从[S]开始的字符串的右侧开始取出n个字符保存至[D]开始的软元件中。



此外，取出的字符串进行保存时，会自动添加“00H”。

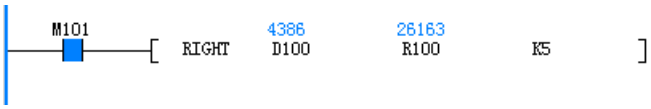
- 要取出的字符数为奇数时，会在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时，会自动在保存最后字符的软元件的下一个软元件内保存“0000H”。
- 要取出的字节数为0时，[D]保存“0000H”。

报错

以下情况会报错误：

- [S]开始的软元件合法范围内未检出“00H”，由于越界查找，报错误。
- [D]的合法范围内，无法完全保存取出的n个字符，报错误。
- n设定过大，超过了[S]中指定的字符数，报错误。
- n为负值时，报错误。

指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x5566
D103	16位整数	十六进制	0x7788
D104	16位整数	十六进制	0x0
	16位整数	十进制	
R100	16位整数	十六进制	0x6633
R101	16位整数	十六进制	0x8855
R102	16位整数	十六进制	0x77
R103	16位整数	十六进制	0x0

3.8.11 LEFT

从指定的字符串的左侧取出指定字符数的字符的指令。

LEFT — 从字符串左侧开始取出

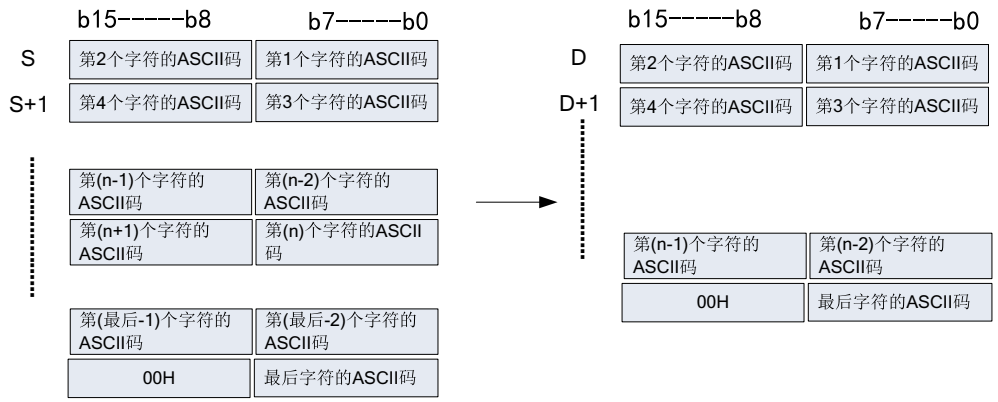
16位指令	LEFT 连续执行/LEFTP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	源数据	保存字符串的软元件起始编号	-	INT，数组*不定
D	取出结果	保存被取出的字符串的软元件起始编号	-	INT，数组*不定
n	取出个数	要取出的字符数	1-32767	INT

表3–157 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
n	-	-	-	√	√	√	√	-	-

功能和指令说明

从[S]开始的字符串的左侧（字符串的开头）开始取出n个字符保存至[D]开始的软元件中。



此外，取出的字符串进行保存时，会自动添加“00H”。

- 要取出的字符数为奇数时，会在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时，会自动在保存最后字符的软元件的下一个软元件内保存“0000H”。
- 要取出的字节数为0时，[D]保存“0000H”。

报错

以下情况会报错误：

- [S]开始的软元件合法范围内未检出“00H”，由于越界查找，报错误。
- [D]的合法范围内，无法完全保存取出的n个字符，报错误。
- n设定过大，超过了[S]中指定的字符数，报错误。
- n为负值时，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x5566
D103	16位整数	十六进制	0x7788
D104	16位整数	十六进制	0x0
	16位整数	十进制	
R100	16位整数	十六进制	0x1122
R101	16位整数	十六进制	0x3344
R102	16位整数	十六进制	0x5566
R103	16位整数	十六进制	0x0

3.8.12 MIDW

用指定字符串中任意位置的字符串来替换指定字符串的指令。

MIDW — 字符串中任意替换

16位指令	MIDW 连续执行/MIDWP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	源数据	保存要替换的字符串的软元件起始编号	-	INT，数组*不定

D	替换结果	保存替换后的字符串的软元件起始编号	-	INT，数组*不定
S2	替换位置信息	指定要替换的字符的起始位置以及字符数的软元件起始编号 S2： 被替换的字符串的起始字符位置 S2+1： 要替换的字符数	-	INT，数组*2

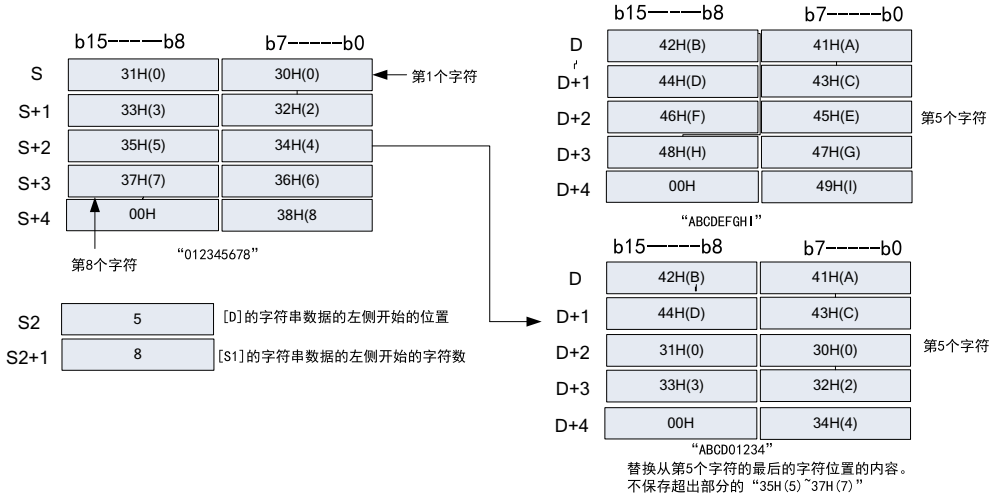
表3–158 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-

功能和指令说明

将[S1]左侧（字符串的开头）开始的[S2+1]个字符替换到[D]开始的软元件中，替换起始位置从[D]字符串的第[S2]个字符开始。

- [S1]中指定的字符串，就是指从[S1]开始搜索至第一个“00H”为止的数据。
- [S2+1]中设定的被替换的字符数为0，指令不执行。
- [S2+1]中设定的字符数为-1时，[S1]中字符串数据全部被替换到指定的[D]开始的软元件中。
- [S2+1]中设定的被替换的字符数，如果超出了[D]中最后的字符时，则替换到[D]最后字符位置为止，超出的部分字符不保存。

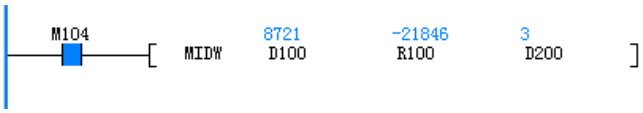


报错

以下情况会报错误：

- [S1]或者[D]开始的软元件合法范围内未检出“00H”，由于越界查找，报错误；
- [S2]值设定过大，超过了[D]中指定的字符数，报错误；
- [S2]设定值n为负值时，报错误；
- [S2+1]设定值为-2以下，报错误；
- [S2+1]设定值超出了[S1]中字符数，报错误。

指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x2211
D101	16位整数	十六进制	0x4433
D102	16位整数	十六进制	0x6655
D103	16位整数	十六进制	0x0
D200	16位整数	十六进制	0x3
D201	16位整数	十六进制	0x4
D202	16位整数	十六进制	0x0
R100	16位整数	十六进制	0xAAAA
R101	16位整数	十六进制	0x2211
R102	16位整数	十六进制	0x4433
R103	16位整数	十六进制	0xAAAA
R104	16位整数	十六进制	0xAAAA
R105	16位整数	十六进制	0x0

3.8.13 MIDR

取出指定字符串中任意位置字符串的指令。

MIDR — 从字符串中任意取出

16位指令	MIDR 连续执行/MIDRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	源数据	保存字符串的软元件起始编号	-	INT，数组*不定
D	取出结果	保存被取出的字符串的软元件起始编号	-	INT，数组*不定
S2	取出位置信息	指定要取出的字符的起始位置以及字符数的软元件起始编号 S2： 起始字符位置 S2+1： 字符数	-	INT，数组*2

表3–159 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-

功能和指令说明

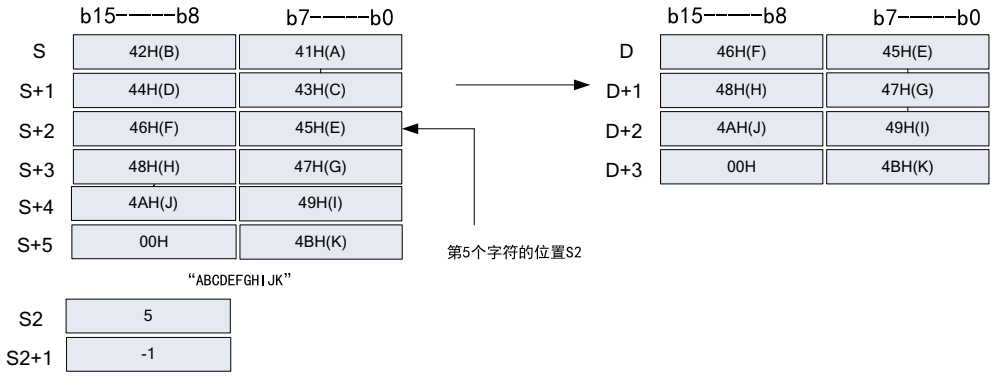
从[S1]软元件开始保存的字符串左侧（字符串的开头）起第[S2]个字符开始，取出[S2+1]个字符数，保存至[D]开始的软元件中。

- 取出的字符数[S2+1]为奇数时，会在保存最后字符的软元件的高字节中保存“00H”。
- 取出的字符数[S2+1]为偶数时，会自动在保存最后字符的软元件的下一个软元件内保存“0000H”。

[S1]中指定的字符串，就是指从[S1]开始搜索至第一个“00H”为止的数据。

[S2+1]中设定值为0时，指令不执行。

[S2+1]中设定值为-1时，从[S2]指定的字符开始至[S1]最终字符的数据全部保存至[D]开始的软元件中。



报错

以下情况会报错误：

- [S1]开始的软元件合法范围内未检出“00H”，由于越界查找，报错误；
- [S2]值设定过大，超过了[D]中指定的字符数，报错误；
- [D]的合法范围内，无法完全保存取出的[S2+1]个字符，报错误；
- [S2]设定值n为负值时，报错误；
- [S2+1]设定值为-2以下，报错误；
- [S2+1]设定值超出了[S1]中字符数，报错误。

指令示例

元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x1122
D101	16位整数	十六进制	0x3344
D102	16位整数	十六进制	0x5566
D103	16位整数	十六进制	0x7788
D104	16位整数	十六进制	0x0
D200	16位整数	十进制	3
D201	16位整数	十进制	4
R100	16位整数	十六进制	0x3344
R101	16位整数	十六进制	0x5566
R102	16位整数	十六进制	0x0

3.8.14 $MOV

传送字符串数据的指令。

$MOV — 字符串的传送

16位指令	$MOV 连续执行/$MOVP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型

S	源地址	传送源中被直接指定的字符串(最大32个字符)，或是保存字符串的软元件起始编号	-	INT，数组*不定
D	目的地址	保存传送字符串的软元件起始编号	-	INT，数组*不定

表3–160 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

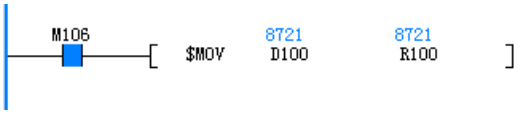
把[S]起始的字符串数据拷贝至[D]起始的软元件中。字符串从[S]起始，至第一个“00H”为止，一次传送完成，且传送过程中字符串结束符“00H”或“0000H”也一并传送。

报错

以下情况会报错误：

- [S]开始的软元件合法范围内未检出00H，由于越界查找，报错误。
- [D]在合法的地址范围之内，无法保存下所有的字符串，报错误。

指令示例



元件名称	数据类型	显示格式	当前值
D100	16位整数	十六进制	0x2211
D101	16位整数	十六进制	0x4433
D102	16位整数	十六进制	0x6655
D103	16位整数	十六进制	0x0
R100	16位整数	十六进制	0x2211
R101	16位整数	十六进制	0x4433
R102	16位整数	十六进制	0x6655
R103	16位整数	十六进制	0x0

3.9 时钟指令

3.9.1 指令列表

时钟指令涵盖以下指令条目：

指令类别	名称	功能
时钟指令	TCMP	时钟数据比较
	TZCP	时钟数据区间比较
	TADD	时钟数据加法运算
	TSUB	时钟数据减法运算
	HTOS	时，分，秒数据的秒转换
	STOH	秒对时，分，秒数据的转换
	TRD	时钟数据读取
	TWR	时钟数据写入
	HOUR	计时表

3.9.2 TCMP

将指定的时、分、秒数值，与内部实时时钟进行比较，输出比较结果。

TCMP — 时钟数据比较指令

16位指令	TCMP 连续执行/TCMPP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	时	指定比较用时间的“时”，范围是0～23	-	INT
S2	分	指定比较用时间的“分”，范围是0～59	-	INT
S3	秒	指定比较用时间的“秒”，范围是0～59	-	INT
S	PLC时间数据起始	实时时钟的时间寄存器的启始地址，通常为时钟读取TRD或MOV指令读取后的存放单元	-	INT，数组*3
D	比较结果	比较结果的存放变量启始地址，占用后续共3个变量单元	-	BOOL，数组*3

表3–161 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
S3	-	-	-	√	√	-	√	-	-
S	-	-	-	√	√	-	-	-	-
D	√[1]	√	√	-	-	-	-	-	-

说明

[1]不支持X元件。

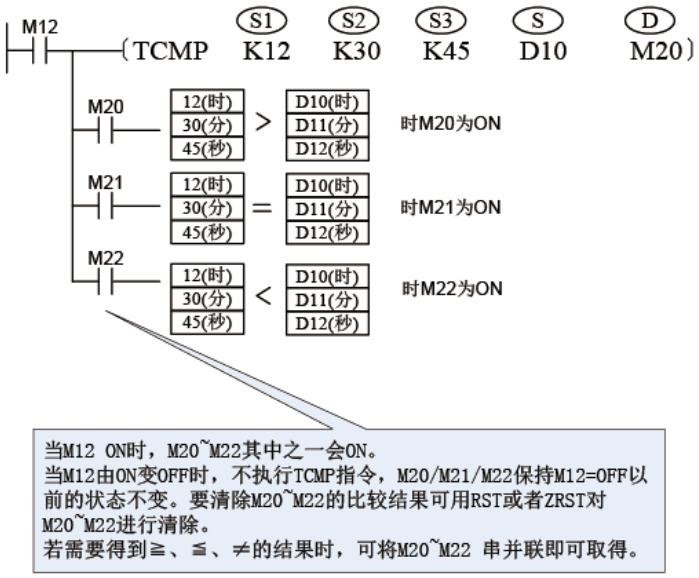
功能和指令说明

该指令是将指定的时、分、秒数值，与内部实时时钟进行比较，输出比较结果。其中：

- S1为指定比较用时间的“时”，范围是0～23：

- S2为指定比较用时间的“分”，范围是0～59；
- S3为指定比较用时间的“秒”，范围是0～59；
- S为实时时钟的时间寄存器的启始地址，通常为时钟读取TRD或MOV指令读取后的存放单元；
- D为比较结果的存放变量启始地址，占用后续共3个变量单元。

指令示例



3.9.3 TZCP

比较结果的存放变量启始地址，占用后续共3个变量单元。

TZCP — 时钟数据区域比较指令

16位指令	TZCP 连续执行/TZCPP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	比较下限	设定的时间比较下限，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
S2	比较上限	设定的时间比较上限，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
S	PLC时间数据起始	实时时钟的时间寄存器的启始地址，通常为时钟读取TRD或MOV指令读取后的存放单元	-	INT，数组*3
D	比较结果	比较结果的存放变量启始地址，占用后续共3个变量单元	-	BOOL，数组*3

表3–162 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	√	-	-
S2	-	-	-	√	√	-	√	-	-
S3	-	-	-	√	√	-	√	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	√[1]	√	√	-	-	-	-	-	-

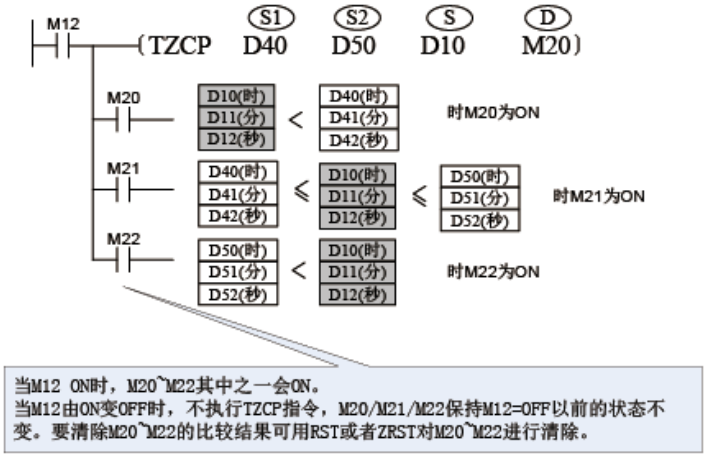
说明

[1]不支持X元件。

功能和指令说明

该指令是将内置实时时钟数据与指定的两组时/分/秒预设值进行区间比较，输出比较结果。

指令示例



3.9.4 TADD

将2组时钟数据的时/分/秒对应相加，结果保存于指定的变量中。

TADD — 时钟数据加法运算指令

16位指令	TADD 连续执行/TADDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	时间被加数	时间被加数，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
S2	时间加数	时间加数，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
D	时间和	时间相加和，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3

表3–163 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是将2组时钟数据的时/分/秒对应相加，结果保存于指定的变量中。其中：

若计算结果超过24小时，进位标志M8022置1，实际显示的时间会减去24：00：00的数值；

若计算结果为00：00：00，零标志M8020置1。

指令示例

完成的操作如下：

如果加法运算结果超过24小时，则进位标志M8022置On。

3.9.5 TSUB

将2组时钟数据的时/分/秒对应相减，结果保存于指定的变量中。

TSUB — 时钟数据减法运算指令

16位指令	TSUB 连续执行/TSUBP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	时间被减数	时间被减数，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
S2	时间减数	减数，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3
D	时间差	时间相减差，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据	-	INT，数组*3

表3–164 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	-	-	-	-
S2	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是将2组时钟数据的时/分/秒对应相减，结果保存于指定的变量中。

若计算结果为负，借位标志M8021置1，实际显示的时间会加上24：00：00的数值；

若计算结果00：00：00，零标志M8020置1。

指令示例

完成的操作如下：

如果减法运算结果为负数时，则借位标志M8021置On。

3.9.6 HTOS

将时分秒时间转换为秒的指令。

HTOS — 小时/分/秒数据转换为秒

16位指令	HTOS 连续执行/HTOSP 脉冲执行			
32位指令	DHTOS 连续执行/DHTOSP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	保存转换前的时间(时刻)数据(时、分、秒)的软元件的起始编号	-	INT，数组*3（固定为3个寄存器）
D	结果	保存转换后的时间(时刻)数据(秒)的软元件编号	-	INT/DINT

表3–165 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令
  将[S，S+1，S+2]的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到D中。
  - 小时的范围：0-9
  - 分钟的范围：0-59
  - 秒钟的范围：0-59
- 32位指令
  将[S，S+1，S+2]的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到[D，D+1]中。
  - 小时的范围：0-32767
  - 分钟的范围：0-59
  - 秒钟的范围：0-59

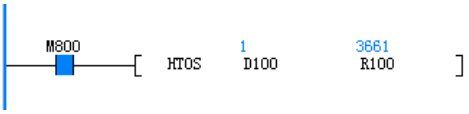
报错

以下情况指令报错且不执行：

- 16位指令和32位指令操作数超范围，报错；
- 16位指令转换结果大于32767，报错；
- S、S+1、S+2超过设定范围时，报错。

指令示例

将存储于D100、D101、D102里面的时、分、秒数据转换成秒数据存储于R100中。



元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	1
D101	16位整数	十进制	1
D102	16位整数	十进制	1
	16位整数	十进制	
R100	16位整数	十进制	3661
	16位整数	十进制	

3.9.7 STOH

将秒单位的时间(时刻)数据转换成[时、分、秒]单位数据的指令。

STOH — 秒转换为小时/分/秒数据

16位指令	STOH 连续执行/STOHP 脉冲执行			
32位指令	DSTOH 连续执行/DSTOHP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源数据	保存转换前的时间(时刻)数据(秒)的软元件编号	-	INT/DINT
D	结果	保存转换后的时间(时刻)数据(时、分、秒)的软元件起始编号	-	INT 数组*3（固定为3个寄存器）

表3–166 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D	-	-	-	√	√	-	-	-	-

功能和指令说明

- 16位指令
  将[S]的秒转换成时、分、秒数据，转换结果存储于[D，D+1，D+2](时、分、秒)中。

  [S]的取值范围：0～32767。

- 32位指令
  将[S，S+1]的秒转换成时、分、秒数据，转换结果存储于[D，D+1，D+2](时、分、秒)中。

  [S，S+1]的取值范围：0～117964799。

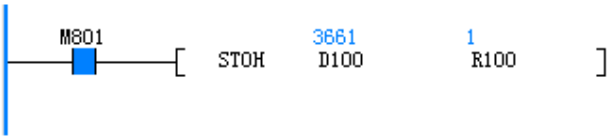
报错

以下情况会报错且指令不执行：

- 16位指令和32位指令操作数超范围，报错；
- 16位指令和32位指令转换秒数超范围，报错。

指令示例

将D100里面的秒数转换成时、分、秒分别存储于R100、R101、R102。



元件名称	数据类型	显示格式	当前值
D100	16位整数	十进制	3661
	16位整数	十进制	
R100	16位整数	十进制	1
R101	16位整数	十进制	1
R102	16位整数	十进制	1
	16位整数	十进制	

3.9.8 TRD

读取PLC内置的实时时钟的年/月/日/时/分/秒/星期，将该7个数据保存于指定的寄存器中。

TRD — 时钟数据读取指令

16位指令	TRD 连续执行/TRDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
D	时间存储首址	时间的启始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：年、月、日、时、分、秒、星期等数据	-	INT，数组*7

表3–167 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是读取PLC内置的实时时钟的年/月/日/时/分/秒/星期，将该7个数据保存于指定的寄存器中。

推荐使用脉冲型指令TRDP。

其中，D为保存读取时间的启始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：年、月、日、时、分、秒、星期等数据。

指令示例

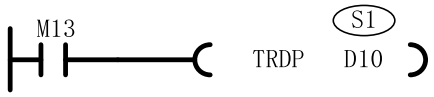


表3–168 转换过程

项目	→	目标值
年(2000~2099)	→	D0
月(1~12)	→	D1
日(1~31)	→	D2
时(0~23)	→	D3
分(0~59)	→	D4
秒(0~59)	→	D5
星期[0(日)~6]	→	D6

说明

一般情况下，如需使用可编程控制器的时钟，需先用TRD指令将时钟读出来放到D寄存器方可使用。

3.9.9 TWR

该指令将指定时钟数据S (含年/月/日/时/分/秒/星期)的7个数据写入PLC的内置实时时钟数据。

TWR — 时钟数据写入指令

16位指令	TWR 连续执行/TWRP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S	时间写入数据首址	为保存读取时间的启始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：年、月、日、时、分、秒、星期等数据	-	INT，数组*7

表3–169 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-

功能和指令说明

该指令是将指定时钟数据S（含年/月/日/时/分/秒/星期）的7个数据写入PLC内置的实时时钟数据里。

推荐使用脉冲型指令TWRP。

其中，S为保存读取时间的启始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：年、月、日、时、分、秒、星期等数据。

指令示例

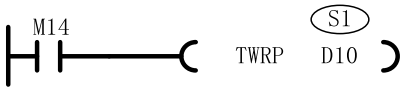


表3–170 转换过程

数据源	→	项目
D0	→	年(2000~2099)
D1	→	月(1~12)
D2	→	日(1~31)
D3	→	时(0~23)
D4	→	分(0~59)
D5	→	秒(0~59)
D6	→	星期[0(日)~6]

说明

写入时钟时候，会将7个数据全部写入。因此预先设值的时候不能缺少某个变量，比如未设置星期，则默认是0，表示星期天；如果未设置月，则默认是0，但PLC会判定该月变量错误，导致此次修改时钟无效。

3.9.10 HOUR

记录驱动条件满足的累加时间，当达到设定的时间后，令指定输出有效。

HOUR — 计时表指令

16位指令	HOUR 连续执行			
32位指令	DHOUR 连续执行			
操作数	名称	描述	范围	数据类型
S	设定时间	设定时间，单位为“小时”，当累加时间达到该设定值后，令输出有效	-	INT/DINT
D1	计时时间	累计时间启始单元	-	INT/DINT，数组*2
D2	计时到达标志	计时到达告警输出变量单元，当计时到达设定值后，该指定单元状态有效	-	BOOL

表3–171 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	√	√	-	-	-	-
D1	-	-	-	√	√	-	-	-	-
D2	√[1]	√	√	-	-	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令是记录驱动条件满足的累加时间，当达到设定的时间后，令指定输出有效。其中：

- S为设定时间，单位为“小时”，当累加时间达到该设定值后，令输出有效；
- D1累计时间启始单元；
- D2计时到达告警输出变量单元，当计时到达设定值后，该指定单元状态有效。
- 16bit时：D1设置范围K0~K32,767，单位：小时。D1+1为未满1个小时的现在时间值，设置范围K0~K3599，单位：秒。此时的D1共占用2个单元。
- 32bit时：D1+1、D1设置范围K0~K2147483647，单位：小时。D1+3、D1+2为未满1个小时的现在时间值，设置范围K0~K3599，单位：秒。此时的D1共占用4个单元。

指令D1计时没有负数，若D1制定为非停电保持的寄存器区域，则在PLC由STOP到RUN或者在掉电的时候会将D1的值清零。若需要在PLC掉电的情况下仍能保持当前值数据，请将D1指定为停电保持区域的寄存器。

指令示例

当M200＝ON时，累计该状态的持续时间，将时记录在D300中，将不满1小时的秒记录在D301中，当D300累计时间达到2000小时后，Y10输出状态为ON。

计时条件满足时，到达S指定数值后，累计计时仍继续进行，读数会继续增大；现在时间值D300到达最大数值32,767小时、D301达到3，599秒时会停止计时测量，要重新计时须将现在时间值D300、301清除为0。

3.10 高速计数器指令

3.10.1 指令列表

高速计数器指令涵盖以下指令条目：

指令类别	名称	功能
高速计数器	HC_Counter	高速计数器使能
	HC_Preset	高速计数器预置值
	HC_TouchProbe	探针
	HC_Compare	高速计数器比较
	HC_ArrayCompare	高速计数器数组比较
	HC_StepCompare	高速计数器等间距离比较

3.10.2 HC_Preset

根据触发信号将计数器值设置为预置值。

HC_Preset — 高速计数器预置值

图形块

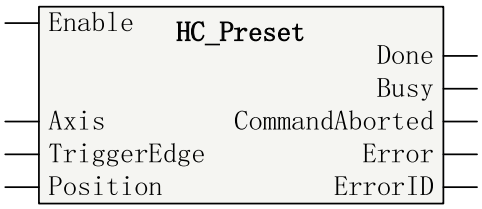


表3–172 指令列表格式

16位指令	—					
32位指令	HC_ Preset连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	TriggerEdge	触发边沿 0：指令上升沿触发 1：外部X上升沿触发 2：外部X下降沿触发 3：外部X上升沿或下降沿触发	是	0	0-3	INT16
S3	Position	预置位置，单位：Unit	否	-	-	REAL32

D1	Done	完成	是	FALSE	TRUE FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE FALSE	BOOL
D3	CommandAborted	执行中断	是	FALSE	TRUE FALSE	BOOL
D4	Error	错误	是	FALSE	TRUE FALSE	BOOL
D5	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看*第253页“3.10.8 指令错误代码”**指令错误代码*

表3–173 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	✓	-
D1	✓[1]	-	✓	-	✓	-	-	-	-
D2	✓[1]	-	✓	-	✓	-	-	-	-
D3	✓[1]	-	✓	-	✓	-	-	-	-
D4	✓[1]	-	✓	-	✓	-	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

使用HC_Preset指令，根据预置条件，实现对计数器轴位置赋值。

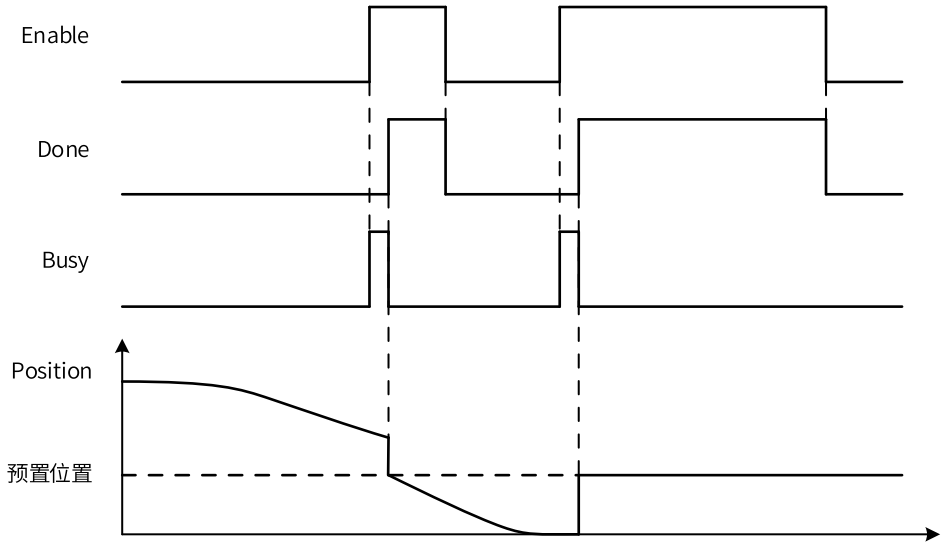
预置条件TriggerType可选择指令上升沿触发或外部X输入触发。

项目	设置	定义
TriggerType	0	指令能流上升沿触发
	1	外部X上升沿触发
	2	外部X下降沿触发
	3	外部X上升沿或下降沿触发

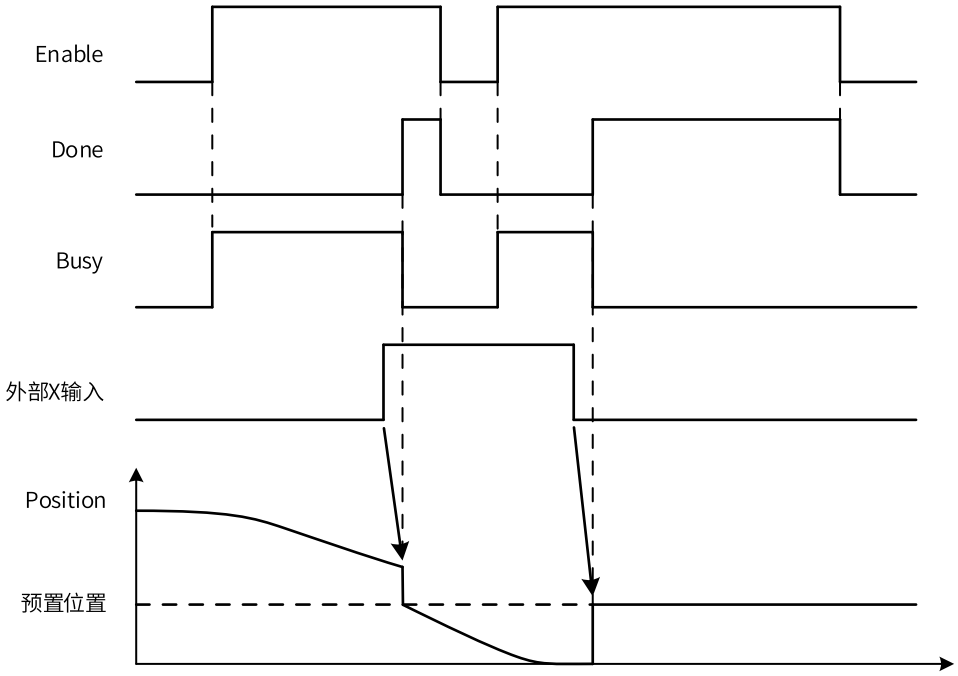
预置条件选择外部X输入触发时，需要在计数器参数设置勾选预置功能，选择输入端子和触发条件，输入端子可任意设置选择X0~X7，触发条件可选择上升沿或下降沿。

时序图

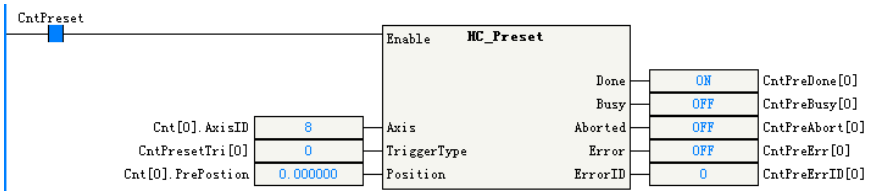
- 指令上升沿触发（TriggerEdge=0），指令时序图如下图所示。



- 外部X触发，以上升沿或下降沿触发（TriggerEdge=3）为例，指令时序图如下图所示。



指令示例



3.10.3 HC_Counter

控制高速计数器计数开始或停止。

HC_Counter — 高速计数器使能

图形块

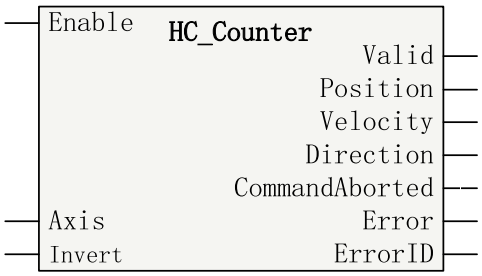


表3–174 指令列表格式

16位指令	—					
32位指令	HC_ Counter连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	Invert	计数反向*1	是	0	0-1	INT16
D1	Valid	有效状态，进入可计数状态时为ON	是	FALSE	TRUE FALSE	BOOL
D2	Position	当前位置，单位：Unit	是	0	-	REAL32
D3	Velocity	当前速度，单位：Unit/s	是	0	-	REAL32
D4	Direction	计数方向	是	FALSE	TRUE FALSE	BOOL
D5	CommandAborted	执行中断	是	FALSE	TRUE FALSE	BOOL
D6	Error	错误	是	FALSE	TRUE FALSE	BOOL
D7	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看*第253页“3.10.8 指令错误代码”**指令错误代码*

表3–175 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	-	✓	-	✓	-	-	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D2	-	-	-	✓	✓	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	-	✓	-	✓	-	-	-	-
D5	✓[1]	-	✓	-	✓	-	-	-	-
D6	✓[1]	-	✓	-	✓	-	-	-	-
D7	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

使用HC_Counter指令，可对计数器轴的位置计数和速度测量。

计数器轴位置值根据模式设置，在计数器轴模式的范围内变化，位置单位为Unit。

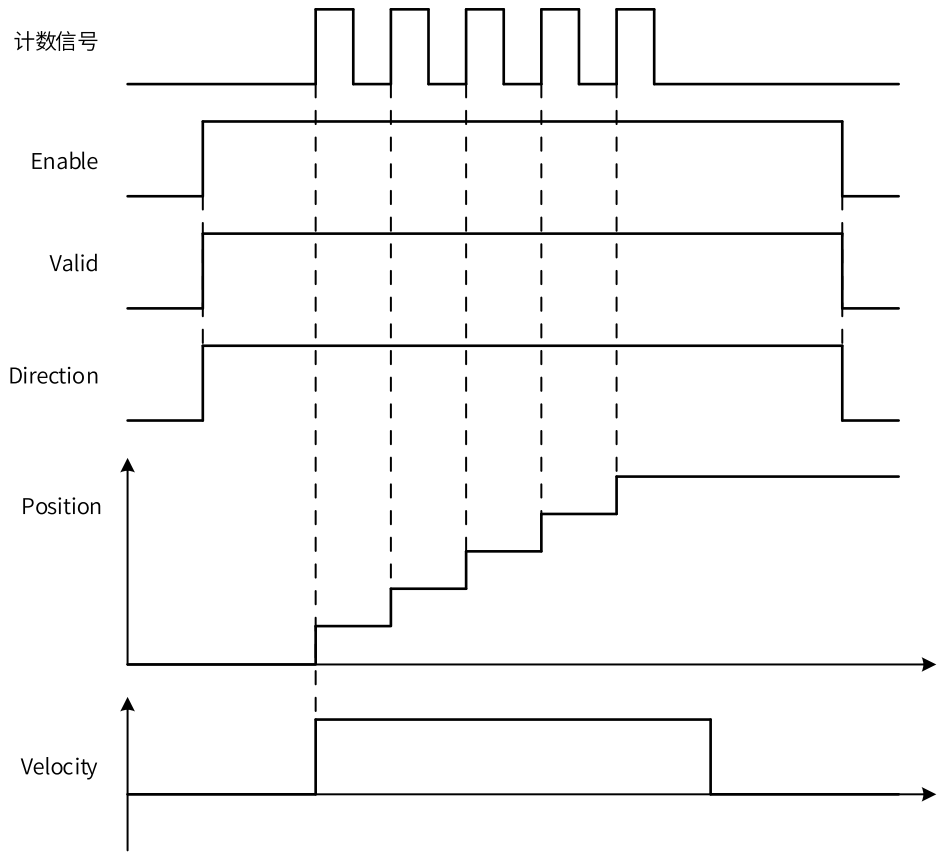
Invert(计数反向) 参数

通过Invert参数设置计数器的计数方向，不同计数模式的计数方向选择定义如下表所示。更改Invert的设置后，需要重新使能功能块指令生效。

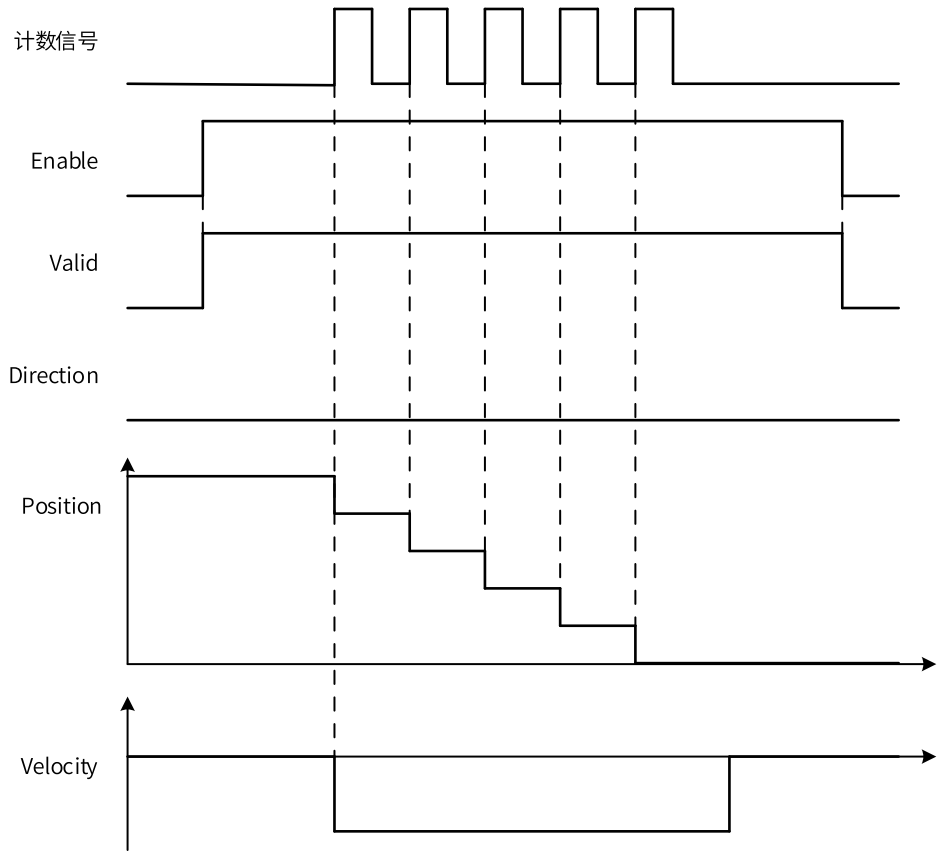
Invert	A/B相	脉冲+方向	CW/CCW	单相计数
0	A相超前B相增计数 B相超前A相减计数	方向信号低电平减计数 方向信号高电平增计数	A相增计数 B相减计数	增计数
1	A相超前B相减计数 B相超前A相增计数	方向信号低电平增计数 方向信号高电平减计数	A相减计数 B相增计数	减计数

时序图

- 以“脉冲+方式”模式为例，方向信号=ON/Invert=0或方向信号=OFF/Invert=1时计数器增计数，如下图所示：

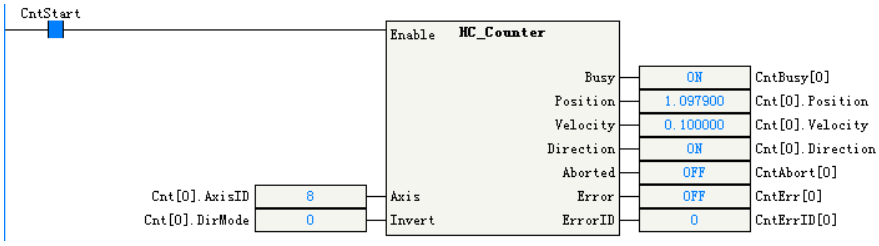


- 以“脉冲+方式”模式为例，方向信号=ON/Invert=1或方向信号=OFF/Invert=0时，计数器增计数，如下图所示。



指令示例

计数器轴速度为当前实时速度，速度单位为Unit/s。计数器轴可测量的最小速度为1s时间内计数器1个脉冲对应的速度，如计数器1个脉冲对应0.01Unit，则可测量的最小速度为0.01Unit/s。



3.10.4 HC_TouchProbe

根据触发信号的发生记录计数器值。

HC_TouchProbe — 高速计数器探针

图形块

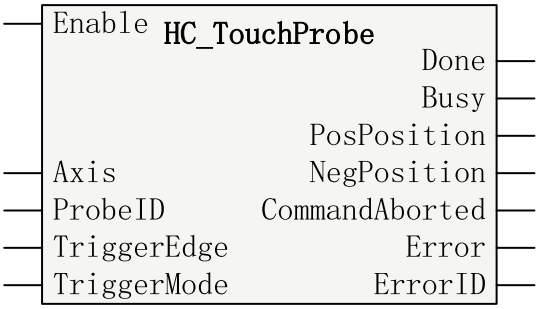


表3–176 指令列表格式

16位指令	—					
32位指令	HC_ TouchProbe连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	ProbeID	探针ID 0：探针1 1：探针2	否	0	0-1	INT16
S3	TriggerEdge	触发边沿 1：外部X上升沿触发 2：外部X下降沿触发 3：外部X上升沿与下降沿触发	是	1	1-3	INT16
S4	TriggerMode	触发模式 0：单次触发 1：连续触发	是	0	0-1	INT16
D1	Done	完成	是	FALSE	TRUE FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE FALSE	BOOL

D3	PosPosition	上升沿锁存位置，单位：Unit	是	0	-	REAL32
D4	NegPosition	下降沒锁存位置，单位：Unit	是	0	-	REAL32
D5	CommandAborted	执行中断	是	FALSE	TRUE FALSE	BOOL
D6	Error	错误	是	FALSE	TRUE FALSE	BOOL
D7	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看第253页“3.10.8 指令错误代码”指令错误代码

表3–177 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	-	✓	-	✓	-	-	-	-
D2	✓[1]	-	✓	-	✓	-	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-
D5	✓[1]	-	✓	-	✓	-	-	-	-
D6	✓[1]	-	✓	-	✓	-	-	-	-
D7	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

使用HC_TouchProbe功能块指令，可在外部输入触发条件有效时，锁存计数器轴位置值。

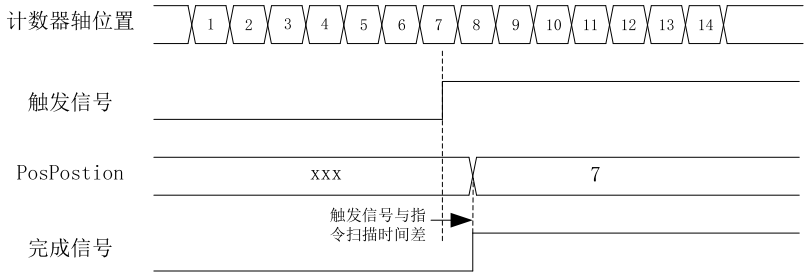
每个计数器轴支持2路探针，使用时，需要在计数器参数设置勾选对应的探针功能，选择输入端子和触发条件，输入端子可任意设置选择X1~X7。

参数TriggerEdge设置探针触发边沿。上升沿触发位置锁存在输出参数PosPosition中，下降沿触发位置锁存在输出参数NegPosition中。

项目	设置	定义
TriggerEdge	1	外部X上升沿触发
	2	外部X下降沿触发
	3	外部X上升沿或下降沿触发

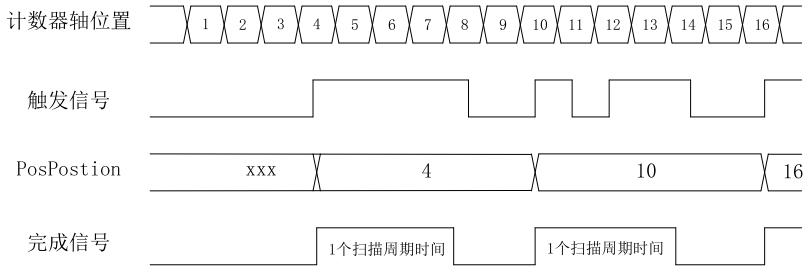
指令中TriggerMode参数可设置单次触发和连续触发模式。

- 使用单次触发模式，功能块指令能流有效，外部输入触发条件有效时，锁存1次计数器轴位置，输出完成信号。探针位置根据触发边沿实时锁存计数器轴位置，不受程序执行影响。程序指令执行时，受扫描周期影响，程序扫描执行到锁存指令时，将锁存位置更新到指令输出参数中。



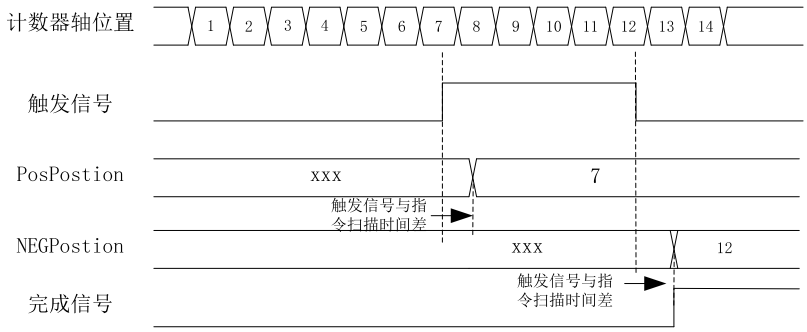
上升沿单次触发模式

- 使用连续触发模式，功能块指令能流有效，外部输入触发条件有效时，锁存计数器轴位置，输出完成信号，完成信号有效时间1个扫描周期。完成信号变为OFF后，外部输入触发条件有效，会继续锁存计数器轴位置，并输出有效时间为1个扫描周期的完成信号。在完成信号有效的1个扫描周期时间内，若外部输入触发条件有效，此时不会锁存计数器轴的位置。

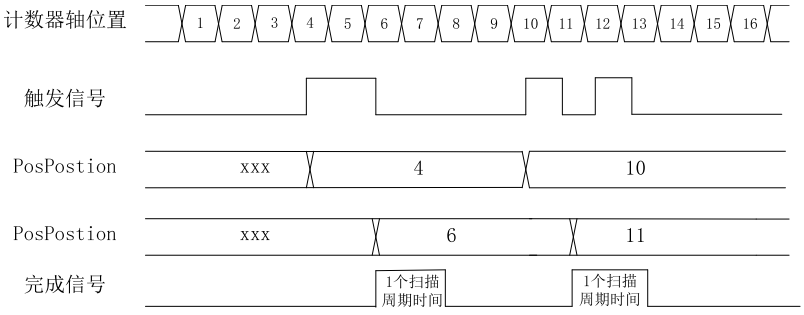


上升沿连续触发模式

- 使用双边沿触发模式时，当上升沿和下降沿都触发完成锁存后，输出完成后信号。单次触发模式时，完成信号持续到指令完成；连续触发模式时，完成信号有效时间1个扫描周期，完成信号有效的1个扫描周期内，不响应触发锁存信号。



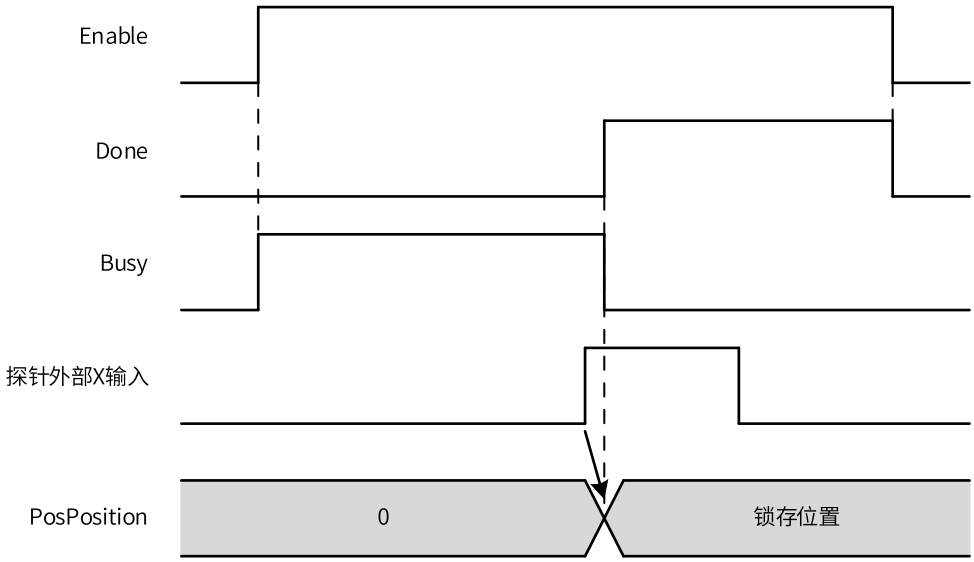
上升下降沿单次触发模式



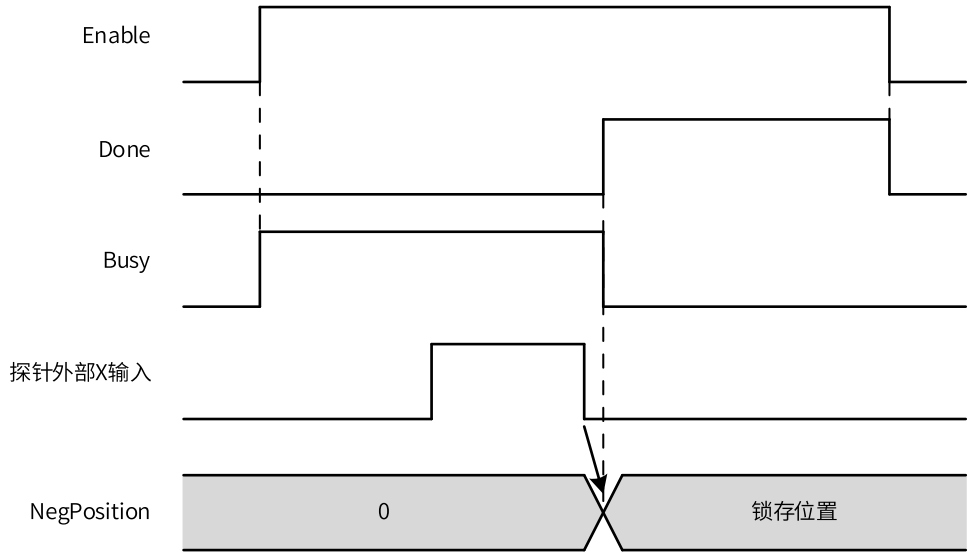
上升下降沿连续触发模式

时序图

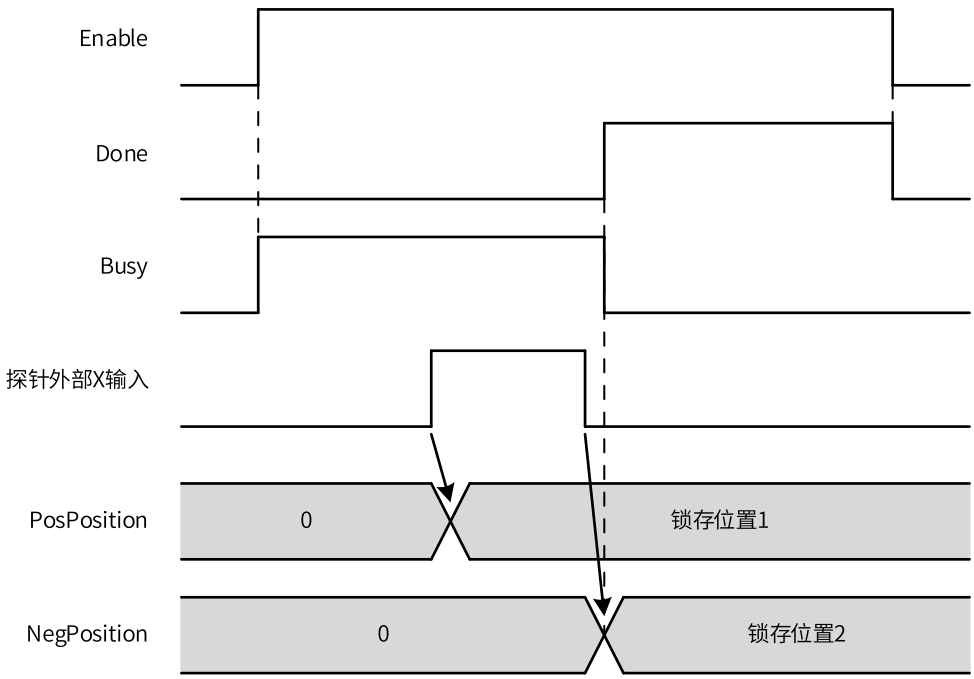
- 外部X上升沿触发（TriggerEdge=1），单次触发模式（TriggerMode=0），指令时序图如下图所示。



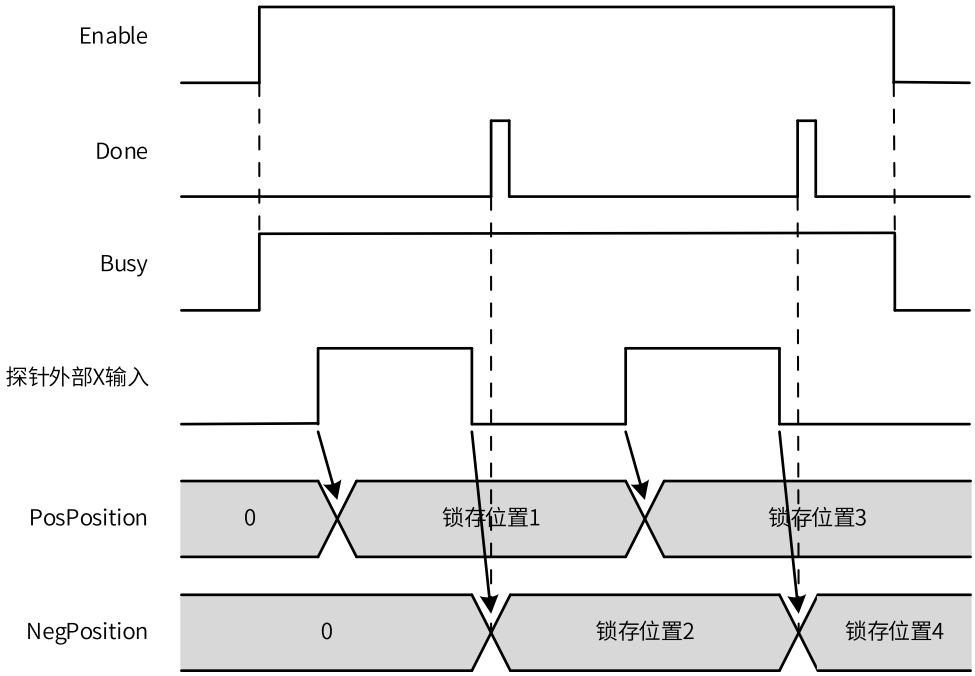
- 外部X下降沿触发（TriggerEdge=2），单次触发模式（TriggerMode=0），指令时序图如下图所示。



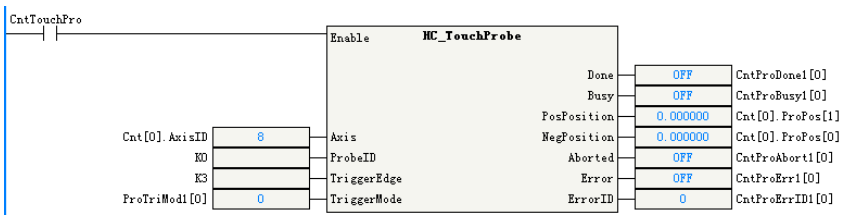
- 外部X上升沿与下降沿触发（TriggerEdge=3），单次触发模式（TriggerMode=0），指令时序图如下图所示。



- 外部X上升沿与下降沿触发（TriggerEdge=3），连续触发模式（TriggerMode=1），指令时序图如下图所示。



指令示例



3.10.5 HC_Compare

探测计数器计数到达指定值。

HC_Compare — 高速计数器比较

图形块

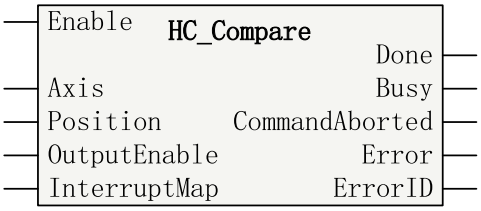


表3–178 指令列表格式

16位指令	—					
32位指令	HC_ Compare连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	Position	比较位置，单位：Unit	否	-	-	REAL32
S3	OutputEnable	硬件输出使能 0：不使用硬件输出 1：使用硬件输出	是	0	0-1	INT16
S4	InterruptMap	比较值与计数值相匹配时产生中断，并关联中断 0：不产生中断 1：关联比较中断1 … 16：关联比较中断16	是	0	0-16	INT16
D1	Done	完成	是	FALSE	TRUE FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE FALSE	BOOL
D3	CommandAborted	执行中断	是	FALSE	TRUE FALSE	BOOL
D4	Error	错误	是	FALSE	TRUE FALSE	BOOL
D5	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看*第253页“3.10.8 指令错误代码”**指令错误代码*

表3–179 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	✓	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	-	✓	-	✓	-	-	-	-
D2	✓[1]	-	✓	-	✓	-	-	-	-
D3	✓[1]	-	✓	-	✓	-	-	-	-
D4	✓[1]	-	✓	-	✓	-	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

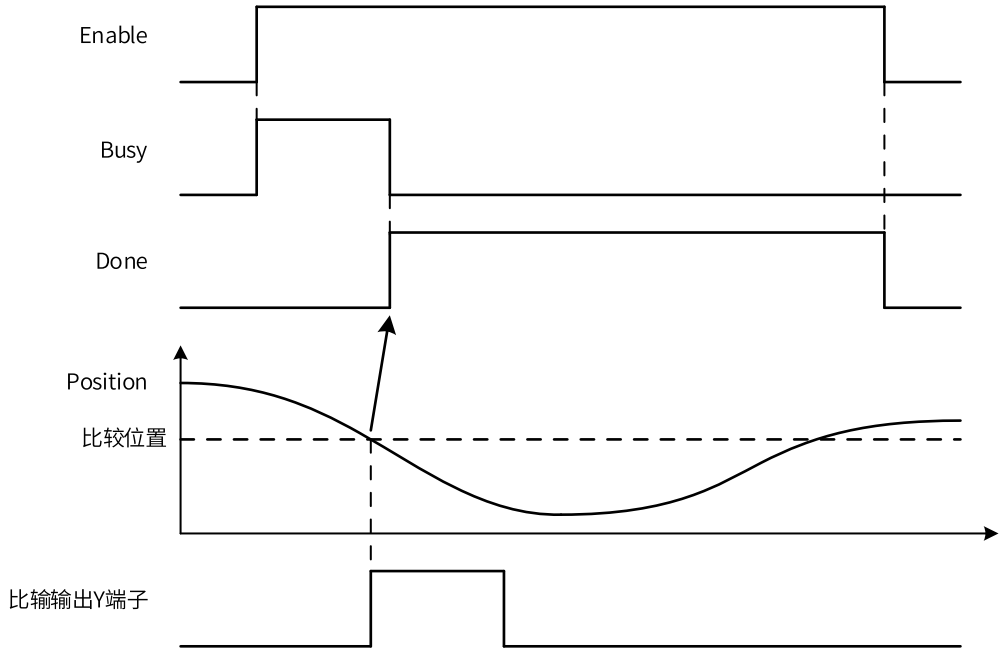
[1]不支持X元件。

功能和指令说明

使用HC_Compare可实现计数器轴的单个位置比较。

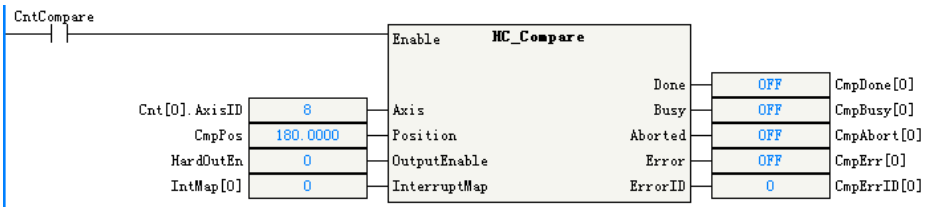
时序图

使用硬件输出（OutputEnable=1），指令时序图如下图所示。



指令示例

实现计数器轴与单个位置比较，指令能流有效时，计数器轴位置达到比较位置后，输出完成信号。



3.10.6 HC_ArrayCompare

按次序连续探测计数器计数到达指定数组序列。

HC_ArrayCompare — 高速计数器数组比较

图形块

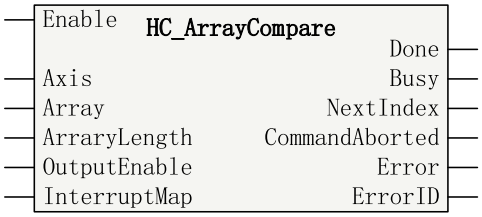


表3–180 指令列表格式

16位指令	—					
32位指令	HC_ ArrayCompare连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	Array	比较位置数组，单位：Unit	否	-	-	REAL32
S3	ArrayLength	数组长度	否	-	0-100	INT16
S4	OutputEnable	硬件输出使能 0：不使用硬件输出 1：使用硬件输出	是	0	0-1	INT16
S5	InterruptMap	比较值与计数值相匹配时产生中断，并关联中断 0：不产生中断 1：关联比较中断1 … 16：关联比较中断16	是	0	0-16	INT16
D1	Done	完成	是	FALSE	TRUE FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE FALSE	BOOL
D3	NextIndex	下一个比较值索引	是	0	0-100	INT16
D4	Aborted	执行中断	是	FALSE	TRUE FALSE	BOOL

D5	Error	错误	是	FALSE	TRUE FALSE	BOOL
D6	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看第253页“3.10.8 指令错误代码”指令错误代码

表3–181 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
S5	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	-	✓	-	✓	-	-	-	-
D2	✓[1]	-	✓	-	✓	-	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	-	✓	-	✓	-	-	-	-
D5	✓[1]	-	✓	-	✓	-	-	-	-
D6	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

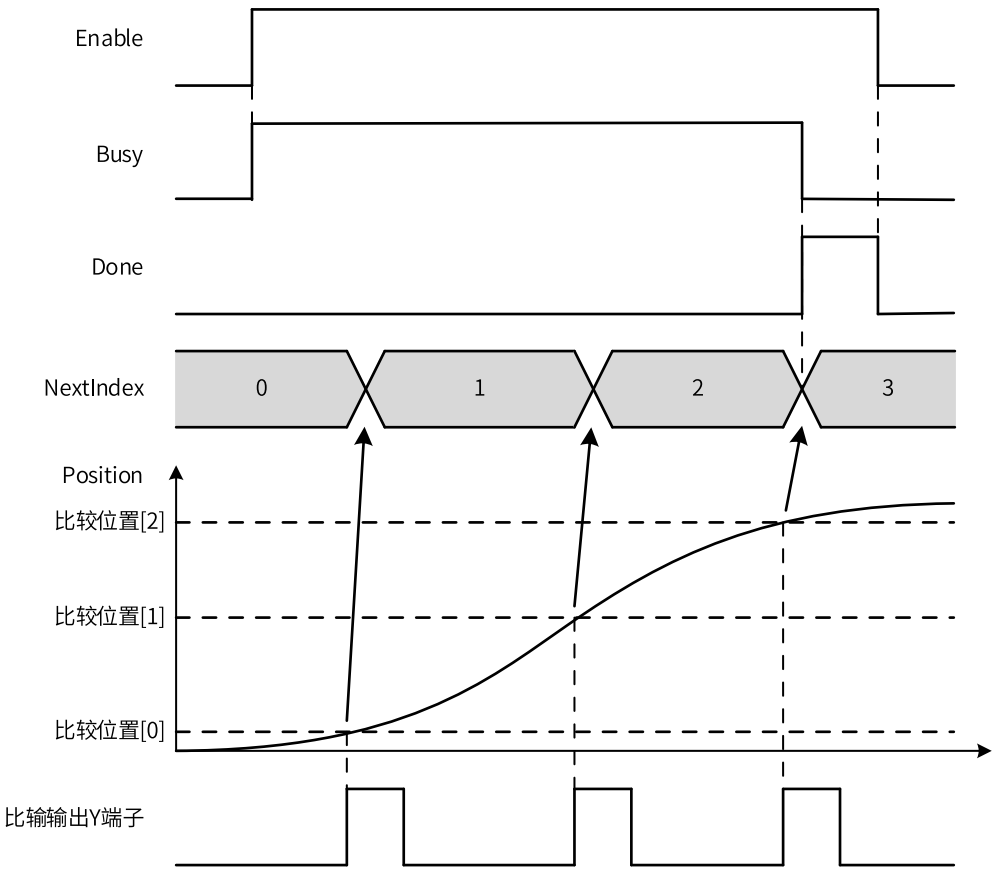
使用HC_ArrayCompare可实现计数器轴的多位置连续比较。

指令能流有效时，计数器轴位置与数组第1个位置开始比较，比较相等后，与数组下一个位置值比较。指令中ArrayLength设定数组长度，所有数组长度设定的数组位置比较完成后，完成信号持续输出，完成多位置连续比较。

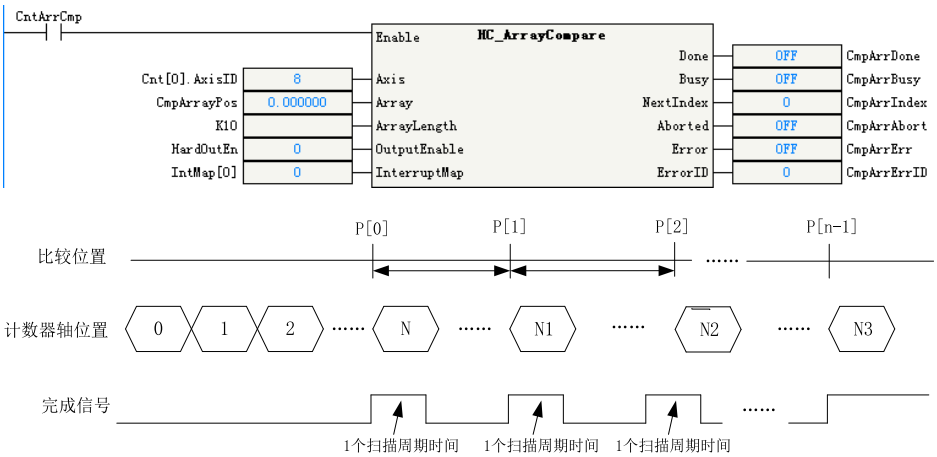
输出参数NextIndex指示下一个比较点标号，即已完成比较相等的个数。

时序图

比较3个位置（ArrayLength=3），使用硬件输出（OutputEnable=1），指令时序图如下图所示。



指令示例



3.10.7 HC_StepCompare

按次序连续探测计数器计数到达指定范围与间距的连续序值。

HC_StepCompare — 高速计数器等间距离比较

图形块

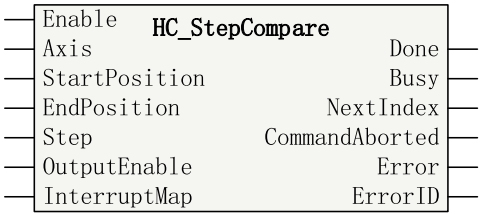


表3-182 指令列表格式

16位指令	—					
32位指令	HC_StepCompare连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴号，指定操作的本地编码器轴	否	-	0-32767	INT16
S2	StartPosition	起始位置，单位：Unit	否	-	-	REAL32
S3	EndPosition	结束位置，单位：Unit	否	-	-	REAL32
S4	Step	间距，单位：Unit	否	-	正数	REAL32
S5	OutputEnable	硬件输出使能 0：不使用硬件输出 1：使用硬件输出	是	0	0-1	INT16
S6	InterruptMap	比较值与计数值相匹配时产生中断，并关联中断 0：不产生中断 1：关联比较中断1 … 16：关联比较中断16	是	0	0-16	INT16
D1	Done	0：未完成 1：完成	是	FALSE	TRUE FALSE	BOOL
D2	Busy	执行中	是	FLASE	TRUE FALSE	BOOL
D3	NextIndex	下一个比较值索引	是	0	0-100	INT16
D4	CommandAborted	执行中断	是	FALSE	TRUE FALSE	BOOL
D5	Error	错误	是	FALSE	TRUE FALSE	BOOL
D6	ErrorID	错误码	是	0	*1	INT16

说明

*1：请查看*第253页“3.10.8 指令错误代码”**指令错误代码*

表3–183 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	✓	-
S3	-	-	-	✓	✓	✓	✓	✓	-
S4	-	-	-	✓	✓	✓	✓	✓	-
S5	-	-	-	✓	✓	✓	✓	-	-
S6	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	-	✓	-	✓	-	-	-	-
D2	✓[1]	-	✓	-	✓	-	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	-	✓	-	✓	-	-	-	-
D5	✓[1]	-	✓	-	✓	-	-	-	-
D6	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

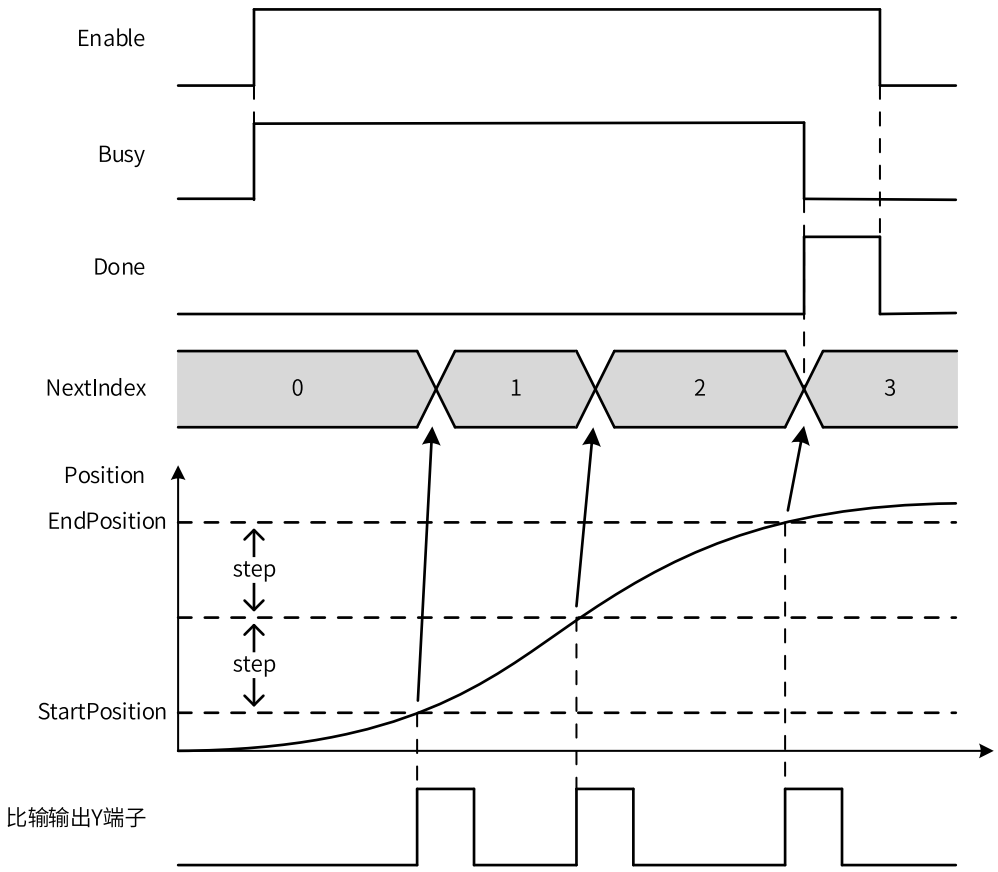
使用HC_StepCompare可以实现计数器轴与等间距连续位置比较。

指令能流有效时，计数器轴位置与StartPosition位置开始比较，比较相等后，比较位置增加或减小Step间距后继续比较。等间距比较完最后一个比较位置后，完成信号持续输出。

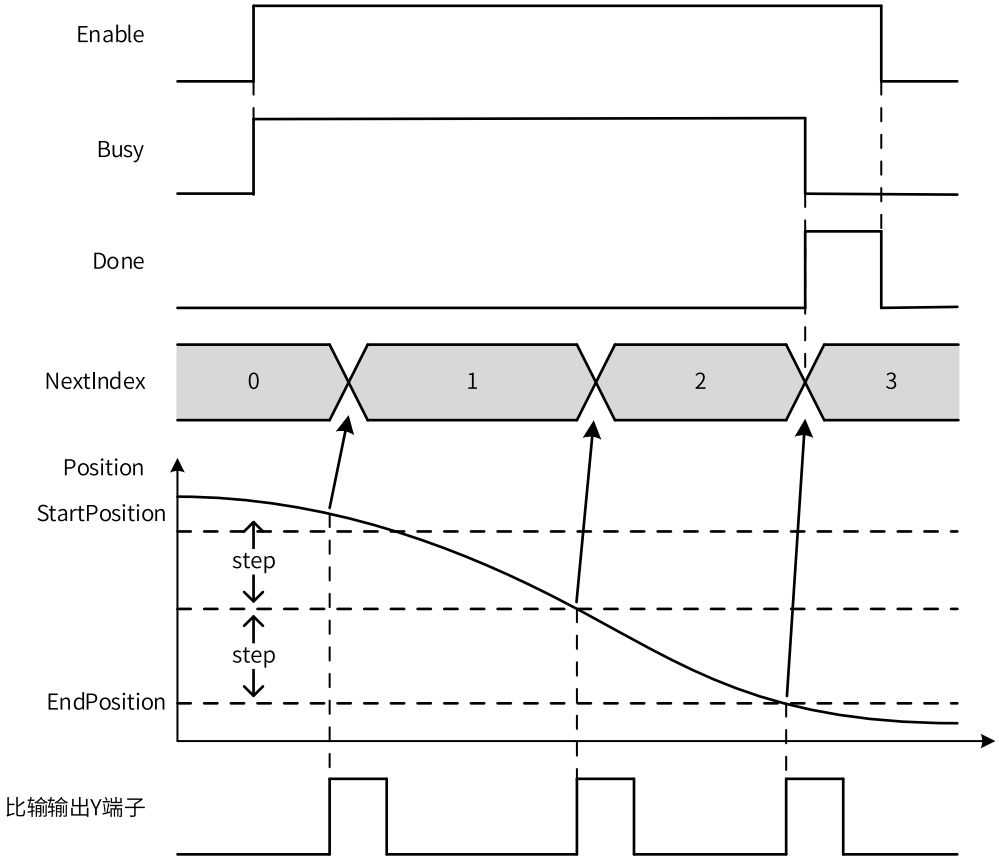
输出参数NextIndex指示下一个比较点标号，标号从0开始计算，即0表示第1个比较点。因此，该标号正好与已经完成的比较点个数相等。

时序图

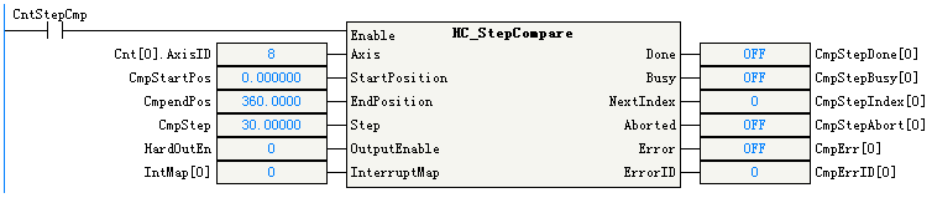
- 使用硬件输出（OutputEnable=1），StartPosition < EndPosition，指令时序图如下图所示。



- 使用硬件输出（OutputEnable=1），StartPosition>EndPosition，指令时序图如下图所示。



指令示例



3.10.8 指令错误代码

高速计数器功能块的指令错误代码如下表所示：

表3-184 指令错误代码

错误代码	说明
0	无错误
100	无效轴号/非本地编码器轴
200	反向参数输入错误
201	触发模式参数输入错误
202	触发边沿参数输入错误，无效触发边沿或未配置X输入
300	探针ID参数输入错误
301	输出使能参数输入错误,无效输出使能或未配置Y输出
400	中断映射参数输入错误
500	预置位置参数输入错误
501	比较位置参数输入错误
502	起始位置参数输入错误
503	结束位置参数输入错误
504	间距参数输入错误
600	数组长度参数输入错误或等间距比较超过100个位置。
1000	计数超下限
1001	计数超上限

3.11 定时器指令

3.11.1 定时器指令参数

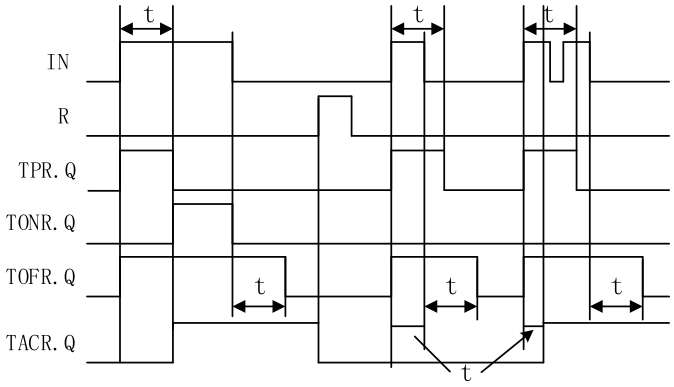
H5U定时器分为脉冲定时器（TPR）、接通延时定时器（TONR）、关断延时定时器（TOFR）和时间累加定时器。

定时器的时间基准为1ms，执行定时器指令时更新定时器计数值和状态，程序中最大支持4096条定时器指令。这4种定时器指令参数相同，参数如下：

表3-185 定时器指令参数

名称	定义	数据类型	说明
IN	指令执行输入	/	启动输入
PT	输入变量	DINT	延时时间
R	输入变量	BOOL	复位输入
Q	输出变量	BOOL	定时器输出
ET	输出变量	DINT	当前定时时间

定时器时序动作



3.11.2 指令列表

定时器指令涵盖以下指令条目：

表3-186 定时器指令列表

指令类别	名称	功能
定时器指令	TPR	脉冲定时器
	TONR	接通延时定时器
	TOFR	关断延时定时器
	TACR	时间累加定时器

3.11.3 TPR

TPR — 脉冲定时器

16位指令	—			
32位指令	TPR 连续执行			
操作数	名称	描述	范围	数据类型
S1	PT	预设定时，单位ms	-	DINT
S2	R	复位*1	-	BOOL
D1	Q	输出结果*1	-	BOOL
D2	ET	经过时间*1，单位ms	-	DINT

说明

*1：允许不设置指令参数，若未设置，使用默认值或无输出。

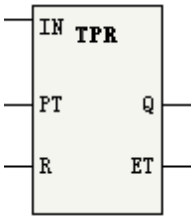
表3–187 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	✓	✓	✓	-	-	✓	-	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和参数说明



定时器指令IN输入能流从OFF变为ON时，定时器启动计时，输出Q变为ON，此时无论IN输入能流如何变化，在PT参数指定的时间内，Q保持为ON。定时时间到达PT参数指定的时间后，Q变为OFF。

在定时器计时运行期间，ET输出当前的计时时间。定时器计时时间到达PT参数指定的时间后，若IN输入能流为ON，ET值保持；若IN输入能流为OFF，则ET值为0。

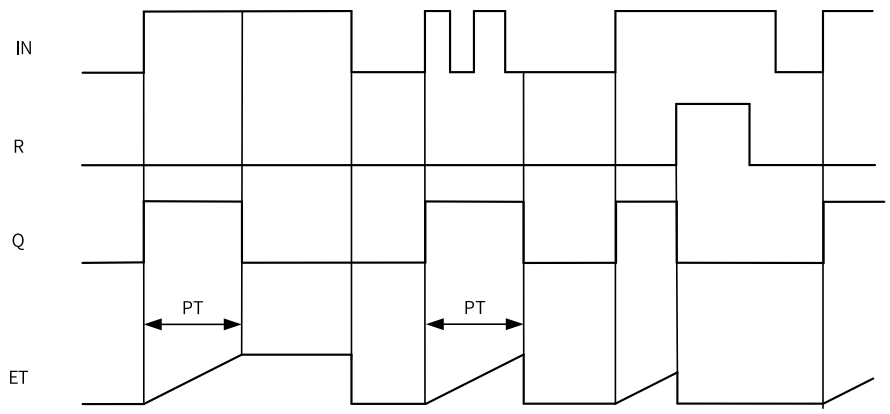
定时器计时过程中，如果复位输入R从OFF变为ON，TPR定时器定时时间复位为0，输出Q变为OFF。复位输入R变为OFF后，如果IN输入能流有效，即可恢复定时器计时。

参数说明：

PT：设定值范围0-2147483647ms（最大约24天）；若PT设定值小于等于0，按照0定时。

时序图

参数IN、R、Q、ET时序图如下：



说明

输出参数“ET” 和“Q” 在执行本指令时更新。因此“Q”的状态变化不是定时器启动后的经过时间等于“PT” 的时刻，而是定时器启动后的经过时间到达“PT” 后，首次执行本指令的时刻。即，输出参数会发生最大1个扫描周期的延迟。

3.11.4 TONR

TONR — 接通延时定时器

16位指令	—			
32位指令	TONR 连续执行			
操作数	名称	描述	范围	数据类型
S1	PT	预设定时，单位ms	-	DINT
S2	R	复位*1	-	BOOL
D1	Q	输出结果*1	-	BOOL
D2	ET	经过时间*1，单位ms	-	

说明

*1：允许不设置指令参数，若未设置，使用默认值或无输出。

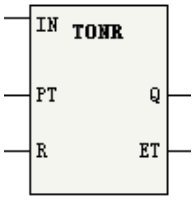
表3–188 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	√	√	√	-	-	√	-	-	-
D1	√[1]	√	√	-	-	√	-	-	-
D2	-	-	-	√	√	√	-	-	-

说明

[1]不支持X元件。

功能和指令说明



定时器指令IN输入能流从OFF变为ON时，定时器启动计时，输出Q为OFF。在IN输入能流保持为ON期间，定时器的运行时间是PT参数指定的时间，定时时间达到PT参数指定的时间后，Q变为ON。在定时过程中或定时完成后，IN输入能流变为OFF，定时结束，Q变为OFF。

IN输入能流为ON时，在定时器计时运行期间，ET输出当前的计时时间，定时器计时时间到达PT参数指定的时间后，ET值保持；若IN输入能流为OFF，则ET值为0。

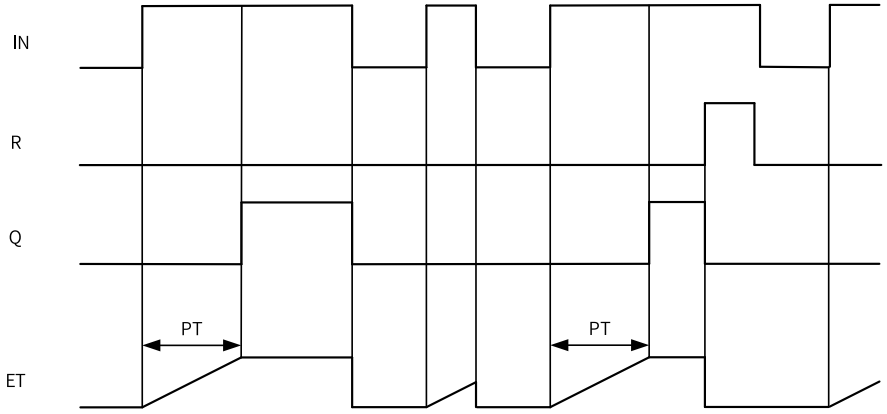
定时器计时过程中，如果复位输入R从OFF变为ON，TONR定时器定时时间复位为0，输出Q变为OFF。复位输入R变为OFF后，要恢复定时器计数，需将IN输入能流重新从OFF变为ON。

参数说明：

PT：设定值范围0-2147483647ms（最大约24天）；若PT设定值小于等于0，按照0定时。

时序图

参数IN、R、Q、ET时序图如下：



说明

输出参数“ET” 和“Q” 在执行本指令时更新。因此“Q”的状态变化不是定时器启动后的经过时间等于“PT” 的时刻，而是定时器启动后的经过时间到达“PT” 后，首次执行本指令的时刻。即，输出参数会发生最大1 个扫描周期的延迟。

3.11.5 TOFR

TOFR — 关断延时定时器

16位指令	—			
32位指令	TOFR 连续执行			
操作数	名称	描述	范围	数据类型
S1	PT	预设定时，单位ms	-	DINT
S2	R	复位*1	-	BOOL

D1	Q	输出结果*1	-	BOOL
D2	ET	经过时间*1，单位ms	-	DINT

说明

*1：允许不设置指令参数，若未设置，使用默认值或无输出。

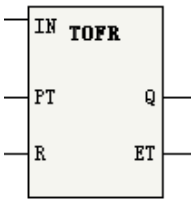
表3–189 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	√	√	√	-	-	√	-	-	-
D1	√[1]	√	√	-	-	√	-	-	-
D2	-	-	-	√	√	√	-	-	-

说明

[1]不支持X元件。

功能和指令说明



定时器指令IN输入能流从OFF变为ON时，定时器启动计时，输出Q为ON，IN输入能流从ON变为OFF时，在IN保持为OFF期间，定时器的运行时间是PT参数指定的时间，定时器定时时间达到PT参数指定的时间后，Q变为OFF。

IN输入能流为ON时，ET输出值为0，IN从ON变为OFF时，在定时器计时运行期间，ET输出当前的计时时间，定时器计时时间到达PT参数指定的时间后，ET值保持。

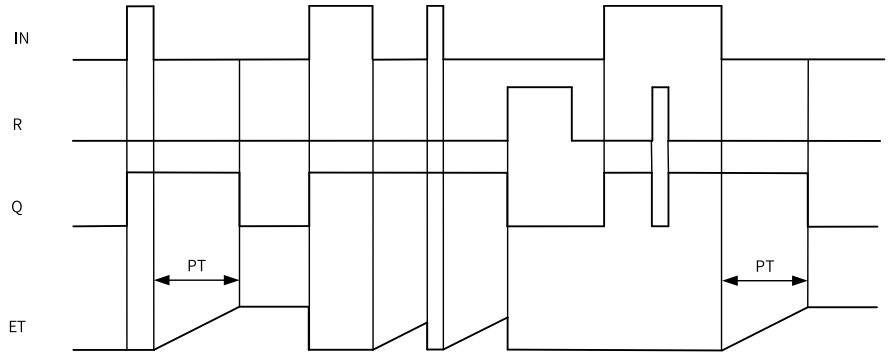
IN输入能流为ON时，如果复位输入R从OFF变为ON，输出Q变为OFF，如果R恢复为OFF，输出Q恢复为ON。IN输入能流从ON变为OFF时，TOFR定时器在定时过程中或定时完成后，如果复位输入R从OFF变为ON，输出Q变为OFF，ET复位为0。复位输入R变为OFF后，要恢复定时器计数，需将IN输入能流重新从ON变为OFF。

参数说明：

PT：设定值范围0-2147483647ms（最大约24天）；若PT设定值小于等于0，按照0定时。

时序图

参数IN、R、Q、ET时序图如下：



说明

输出参数“ET” 和“Q” 在执行本指令时更新。因此“Q”的状态变化不是定时器启动后的经过时间等于“PT” 的时刻，而是定时器启动后的经过时间到达“PT” 后，首次执行本指令的时刻。即，输出参数会发生最大1 个扫描周期的延迟。

3.11.6 TACR

TACR — 时间累加定时器

16位指令	—			
32位指令	TACR 连续执行			
操作数	名称	描述	范围	数据类型
S1	PT	预设定时，单位ms	-	DINT
S2	R	复位*1	-	BOOL
D1	Q	输出结果*1	-	BOOL
D2	ET	经过时间*1，单位ms	-	DINT

说明

*1：允许不设置指令参数，若未设置，使用默认值或无输出。

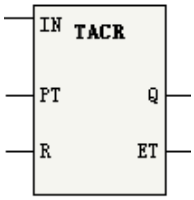
表3–190 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	√	√	√	-	-	√	-	-	-
D1	√[1]	√	√	-	-	√	-	-	-
D2	-	-	-	√	√	√	-	-	-

说明

[1]不支持X元件。

功能和指令说明



定时器指令IN输入能流为ON时，若定时器计时值未达到PT参数所指定的时间，定时器持续计数，输出Q为OFF，定时器定时时间达到PT参数指定的时间后，Q变为ON。在IN为ON且定时器计时期间，若IN变为OFF，定时器计时保持不变，IN重新变为ON后，定时器从当前保持值开始计数，达到PT参数指定的时间后，Q变为ON。

IN输入能流为ON时，ET输出当前计时值，计时时间到达PT参数指定的时间后，ET值保持。IN输入能流为OFF时，ET保持不变。

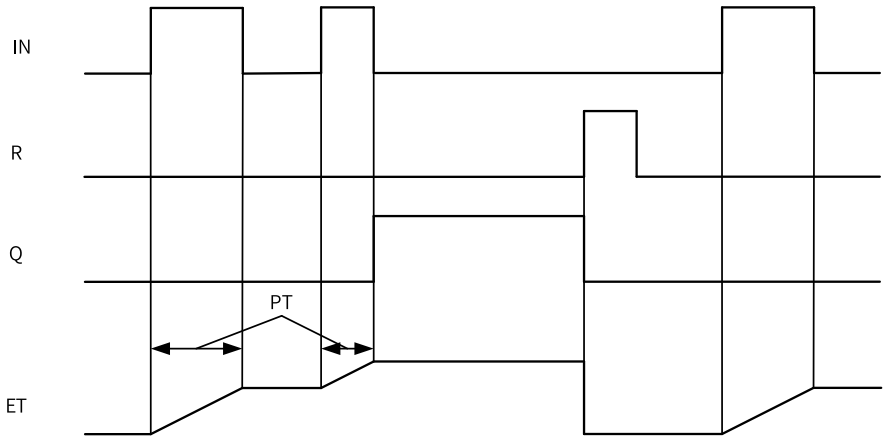
定时器在定时过程中或定时完成后，如果复位输入R从OFF变为ON，输出Q变为OFF，ET复位为0。复位输入R变为OFF后，要恢复定时器计数，需将IN输入能流重新从ON变为OFF。

参数说明：

PT：设定值范围0-2147483647ms（最大约24天）；若PT设定值小于等于0，按照0定时。

时序图

参数IN、R、Q、ET时序图如下：



说明

输出参数“ET” 和“Q” 在执行本指令时更新。因此“Q”的状态变化不是定时器启动后的经过时间等于“PT” 的时刻，而是定时器启动后的经过时间到达“PT” 后，首次执行本指令的时刻。即，输出参数会发生最大1 个扫描周期的延迟。

3.12 指针指令

3.12.1 指令列表

指针指令涵盖以下指令条目：

指令类别	名称	功能
指针指令	PTGET	指针变量赋值指令
	PTINC	指针变量地址增1指令
	PTDEC	针变量地址减1指令
	PTADD	指针变量地址加法指令
	PTSUB	指针变量地址减法指令
	PTMOV	指针变量相互赋值指令
	PTLD>	指针变量触点比较指令
	PTLD>=	指针变量触点比较指令
	PTLD<=	指针变量触点比较指令
	PTLD=	指针变量触点比较指令
	PTLD<>	指针变量触点比较指令
	PTAND>	指针变量触点比较指令
	PTAND>=	指针变量触点比较指令
	PTAND<	指针变量触点比较指令
	PTAND<=	指针变量触点比较指令
	PTAND=	指针变量触点比较指令
	PTAND<>	指针变量触点比较指令
	PTOR>	指针变量触点比较指令
	PTOR>=	指针变量触点比较指令
	PTOR<	指针变量触点比较指令
	PTOR<=	指针变量触点比较指令
	PTOR=	指针变量触点比较指令
	PTOR<>	指针变量触点比较指令

3.12.2 PTGET

PTGET — 指针变量赋值指令

16位指令	PTGET（位） 连续执行/PTGETP 脉冲执行			
32位指令	—			
16位指令	PTGET（字） 连续执行/PTGETP 脉冲执行			
32位指令	PTGET（双字） 连续执行/PTGETP 脉冲执行			
操作数	名称	描述	范围	数据类型
S1	指针元件		-	DINT
S2	目标变量	指针变量指向的目标起始地址	-	BOOL、字、双字

表3–191 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	√[1]	-	-	-
S2	√	√	√	√	√	-	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- PTGET指令可以获取位、字、双字元件或变量的地址；
- POINTER指针变量的使用，要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

PTGET　PT5 D10 ; 使PT5指针变量指向D10元件。

3.12.3 PTINC

PTINC — 指针变量地址增1指令

16位指令	—			
32位指令	PTINC 连续执行/PTINCP 脉冲执行			
操作数	名称	描述	范围	数据类型
S/D	指针元件	-	-	DINT

表3–192 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S/D	-	-	-	-	-	√[1]	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 递增递减是以POINTER指针变量指向的变量为单位，指向当前变量的下一个元件，即位元件指向下一个位元件，字变量指向下一个字变量，双字变量指向下一个双字变量；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

1. PTGET　PT5 D10 ; 使PT5指针指向D10元件；
2. PTINC PT5 ; PT5指针指向指向下一个元件，即D11。

3.12.4 PTDEC

PTDEC — 指针变量地址减1指令

16位指令	—			
32位指令	PTDEC 连续执行/PTDECP 脉冲执行			
操作数	名称	描述	范围	数据类型
S/D	指针元件	-	-	DINT

表3–193 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S/D	-	-	-	-	-	√[1]	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 递增递减是以POINTER指针变量指向的变量为单位，指向当前变量的上一个元件，即位元件指向上一个位元件，字变量指向上一个字变量，双字变量指向上一个双字变量；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

1. PTGET PT5 D10 ; 使PT5指针指向D10元件；
2. PTDEC PT5，PT5指针指向指向上一个元件，即D9。

3.12.5 PTADD

PTADD — 指针变量地址加法指令

16位指令	PTADD 连续执行/PTADDP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	源指针	-	-	DINT
S2	偏移地址	-	0-32767	INT
D	目标指针	-	-	DINT

表3–194 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	√[1]	-	-	-
S2	-	-	-	√	√	-	√	-	-
D	-	-	-	-	-	√[1]	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 加法减法是以POINTER指针变量指向的变量为单位，指向当前变量的下n个元件，即位元件指向下n个位元件，字变量指向下n个字变量，双字变量指向下n个双字变量；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

1. PTGET　PT5　D10 ; 使PT5指针指向D10元件；
2. PTADD PT5 K4 PT5，PT5指针指向PT5加4个元件指针的位置，即D14；
3. PTADD PT5 K5 PT6，PT6指针指向PT5加5个元件指针的位置，即D15，同时PT5指针保持不变，仍指向D10。

3.12.6 PTSUB

PTSUB — 指针变量地址减法指令

16位指令	PTSUB 连续执行/PTSUBP 脉冲执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	源指针	-	-	DINT
S2	偏移地址	-	0-32767	INT
D	目标指针	-	-	DINT

表3–195 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	√[1]	-	-	-
S2	-	-	-	√	√	-	√	-	-
D	-	-	-	-	-	√[1]	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 加法减法是以POINTER指针变量指向的变量为单位，指向当前变量的下n个元件，即位元件指向下n个位元件，字变量指向下n个字变量，双字变量指向下n个双字变量；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

1. PTGET　PT5 D10 ; 使PT5指针指向D10元件；
2. PTSUB PT5 K4 PT5，PT5指针指向PT5减4个元件指针的位置，即D6；
3. PTSUB PT5 K5 PT6，PT6指针指向PT5减5个元件指针的位置，即D5，同时PT5指针保持不变，仍指向D10。

3.12.7 PTMOV

PTMOV — 指针变量相互赋值指令

16位指令	—			
32位指令	PTMOV 连续执行/PTMOVP 脉冲执行			
操作数	名称	描述	范围	数据类型
S	源指针	-	-	DINT
D	目标指针	-	-	DINT

表3–196 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	√[1]	-	-	-
D	-	-	-	-	-	√[1]	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 指针相互赋值，是指针变量的地址备份，即指向同一个地址；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；
- 电平执行，推荐使用脉冲型指令。

指令示例

1. PTGET　PT5 D10 ; 使PT5指针指向D10元件；
2. PTMOV PT5 PT6，PT6指针指向PT5指针的位置，即D10。

3.12.8 PT#

指针变量触点比较指令，包含PTLD、PTAND、PTOR指令，#为>、>=、<、<=、=、<>之一。

PT* — 指针变量触点比较指令

16位指令	—
32位指令	PTLD> 连续执行
16位指令	—
32位指令	PTLD>= 连续执行
16位指令	—
32位指令	PTLD< 连续执行
16位指令	—
32位指令	PTLD<= 连续执行
16位指令	—
32位指令	PTLD= 连续执行
16位指令	—
32位指令	PTLD<> 连续执行
16位指令	—
32位指令	PTAND> 连续执行
16位指令	—
32位指令	PTAND>= 连续执行
16位指令	—
32位指令	PTAND< 连续执行
16位指令	—
32位指令	PTAND<= 连续执行
16位指令	—
32位指令	PTAND= 连续执行
16位指令	—
32位指令	PTAND<> 连续执行
16位指令	—
32位指令	PTOR> 连续执行
16位指令	—
32位指令	PTOR>= 连续执行
16位指令	—
32位指令	PTOR< 连续执行
16位指令	—
32位指令	PTOR<= 连续执行
16位指令	—
32位指令	PTOR= 连续执行
16位指令	—

32位指令	PTOR<> 连续执行			
操作数	名称	描述	范围	数据类型
S1	当前指针	-	-	-
S2	比较对象	-	-	-

说明

PTLD*、PTAND*、PTOR*指令，输入方式统一为“PT*”，由后台自动生成对应指令。

表3–197 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	√[1]	-	-	-
S2	√	√	√	√	√	-	-	-	-

说明

[1]仅支持POINTER指针变量。

功能和指令说明

- 指针的触点比较，是比较指针指向的软元件与比较对象的关系，比如PT5指向；
- POINTER指针变量的使用：要先使用PTGET赋值，否则可能指向错误的位置，导致系统异常执行；

指令示例

1. PTGET　PT5　D10 ; 使PT5指针指向D10元件；
2. PT> PT5 D5，则输出能流为ON；PT> PT5 D20，则输出能流为OFF。

3.13 通信协议指令

3.13.1 指令列表

通讯协议指令涵盖以下指令条目：

指令类别	名称	功能
通讯协议指令	SerialSR	串口自由协议发送接收
	TCP_Listen	TCP监听指令
	TCP_Accept	TCP接受连接请求指令
	TCP_Connect	TCP发起连接请求指令
	TCP_Close	TCP关闭连接指令
	TCP_Send	TCP发送数据指令
	TCP_Receive	TCP接收数据指令
	UDP_Bind	UDP套接字绑定指令
	UDP_Send	UDP发送数据指令
	UDP_Receive	UDP接收数据指令
	ECT_ReadParameter_CoE	读取EtherCAT从站的SDO参数
	ECT_WriteParameter_CoE	写入EtherCAT从站的SDO参数

3.13.2 SerialSR

SerialSR — 串口自由协议发送接收

16位指令	SerialSR 连续执行			
32位指令	—			
操作数	名称	描述	范围	数据类型
S1	Port	端口号		INT
S2	SendBuf	发送数据区		INT
S3	SendSize	发送字节数		INT
S4	RcvBuf	接收缓冲区		INT
S5	RcvSize	接收字节数		INT
S6	Timeout	接收超时时间，单位ms*1		INT
D1	Status	指令运行状态*1		BOOL
D2	Sent	已发送数据大小*1		INT
D3	Received	已接收数据大小*1		INT
D4	Done	完成标志*1		INT

说明

*1：允许不设置指令参数，若未设置，使用默认值或无输出。

表3–198 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	√	√	√	√	-	-
S2	-	-	-	√	√	√	-	-	-
S3	-	-	-	√	√	√	√	-	-
S4	-	-	-	√	√	√	-	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S5	-	-	-	√	√	√	√	-	-
S6	-	-	-	√	√	√	√	-	-
D1	-	-	-	√	√	√	-	-	-
D2	-	-	-	√	√	√	-	-	-
D3	-	-	-	√	√	√	-	-	-
D4	√[1]	√	√	-	-	√	-	-	-

说明

[1]不支持X元件。

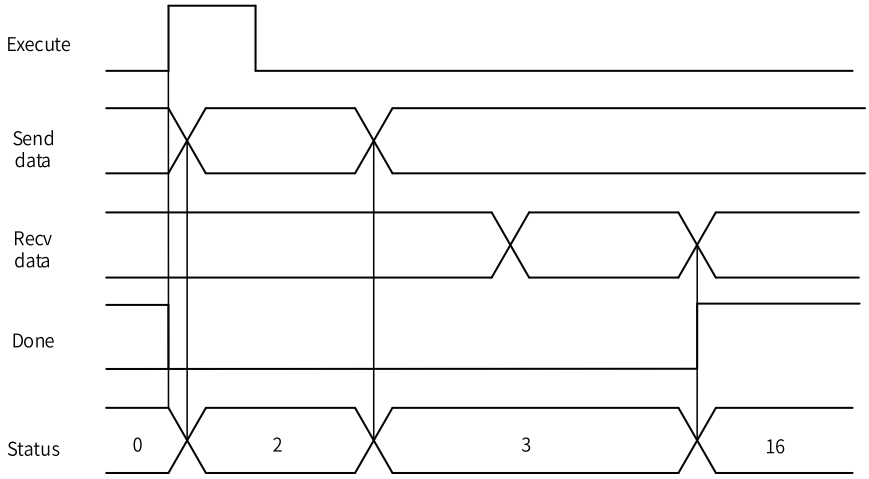
功能和指令说明

- 功能简介

  实现自由协议收发，指令触发后，通过指定端口发送指定长度的数据，发送完成后，接收指定长度的数据。发送接收过程中，更新对应的输出。

- 参数说明
  - S1 — 端口号，当前默认为0（因为当前只有一个串口，故不对该参数检查。保留该参数做后续扩展）；
  - S6 — 超时时间，如果设定的时间值不在范围内，会自动调整到允许的范围内；
  - D2 — 运行状态：0-空状态，1-预留，2-发送，3-接收；16-完成，32-发送异常，48-接收异常，64-其他异常；

时序图



说明

- 指令是上升沿触发执行；
- D2参数显示的是串口运行状态机，包括正常和异常状态值；指令的错误码不显示于D2参数中，而是按照标准指令的错误显示于错误表中；
- 超时时间指的是发送和接收的总时间；
- 本指令仅在端口可用时执行，且同一个端口同一个时间点内只能有一条本指令在执行。如果端口冲突、协议设定错误等，指令不会执行，相关错误码查看标准指令错误。

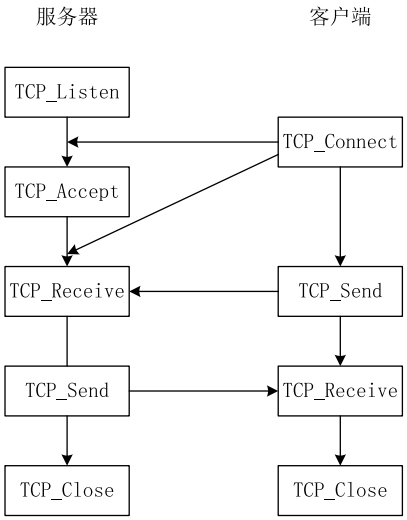
[1]不支持X元件。

3.13.3 面向连接的套接字TCP通信

传输控制协议（TCP, Transmission Control Protocol）, 一种面向连接的、可靠的、基于字节流的传输层通信协议。

互联网络与单个网络有很大的不同，因为互联网络的不同部分可能有截然不同的拓扑结构、带宽、延迟、数据包大小和其他参数。TCP的设计目标是能够动态地适应互联网络的这些特性，而且具备面对各种故障时的健壮性。

H5U提供了面向连接的套接字TCP通信接口，其流程如下图所示。



3.13.4 TCP_Listen

服务器必须等待客户端的连接请求，TCP_Listen指令用于实现等待功能。

TCP_Listen — TCP监听指令

图形块

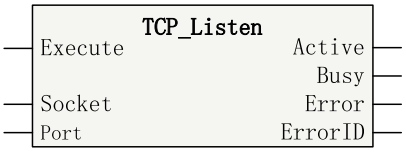


表3–199 指令表格式

16位指令	—					
32位指令	TCP_Listen 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	Port	端口，指定监听的本机端口*2	否	-	1-65535	DINT
D1	Active	有效状态	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	错误码	是	0	*3	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2不可以使用系统内部占用端口号(23、12939、12940)或Modbus-TCP服务器端口号(502)。
- *3请参见*第287页“3.13.17 套接字通信指令错误代码”*。

表3–200 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-

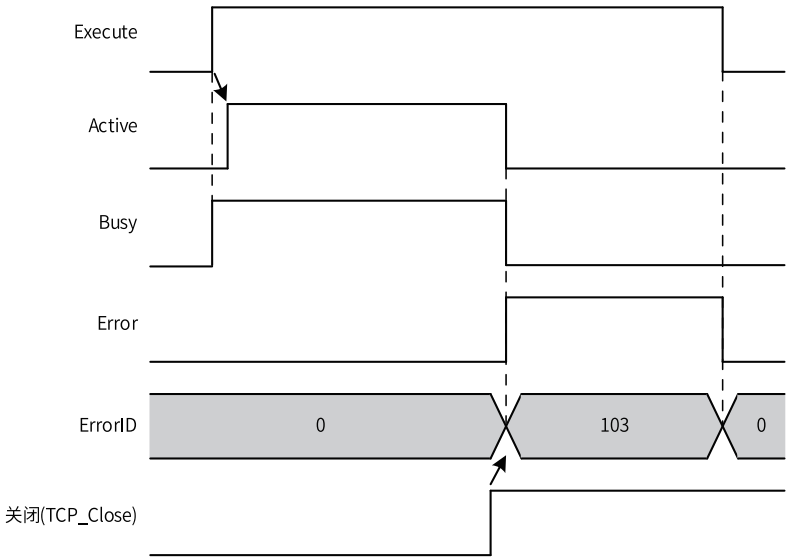
说明

[1]不支持X元件。

功能和指令说明

服务器必须等待客户端的连接请求，TCP_Listen指令用于指定监听的本地端口来等待客户端。当接收到客户端的连接请求后，需要通过TCP_Accept指令来建立与客户端间的通信。

时序图



3.13.5 TCP_Accept

处于监听状态的服务器在获得客户端的连接请求后，会将其放置在等待队列中。使用TCP_Accept指令接收客户端的连接请求。

TCP_Accept TCP接受连接请求指令

图形块

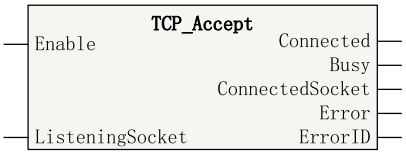


表3–201 指令表格式

16位指令	—					
32位指令	TCP_Accept 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	ListeningSocket	监听套接字*1	否	-	-	_sSocket
D1	Connected	连接状态	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	ConnectedSocket	连接套接字*1	否	-	-	_sSocket
D4	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	错误码	否	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2请参考*第287页“3.13.17 套接字通信指令错误代码”*套接字通信指令错误代码。

表3-202 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-

说明

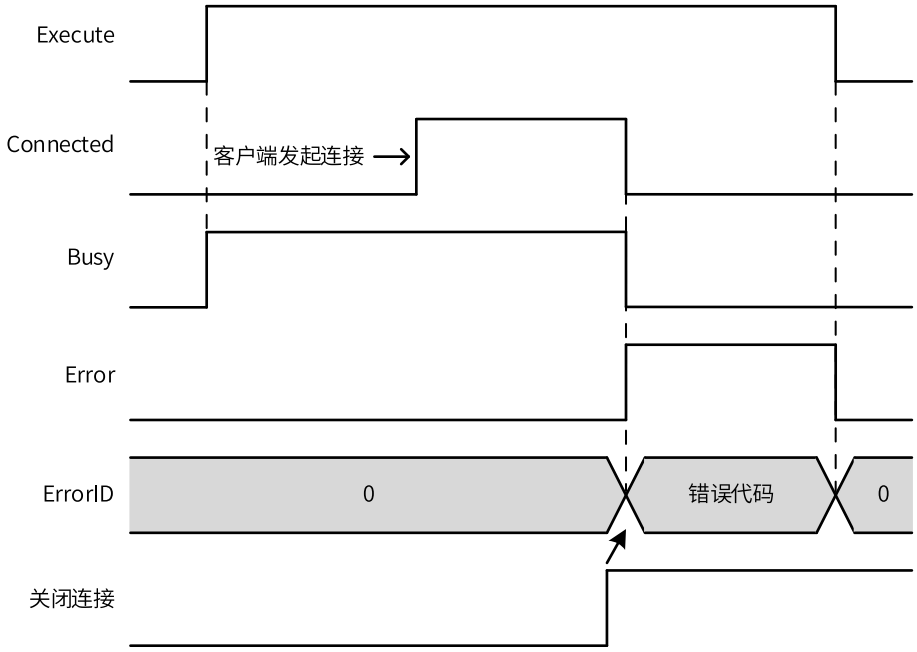
[1]不支持X元件。

功能和指令说明

处于监听状态的服务器当接收到客户端的连接请求后，需要通过TCP_Accept指令来建立与客户端间的通信。在成功建立通信后，就可以通过TCP_Send或TCP_Receive进行发送或接收数据。

通过多条TCP_Accept指令可以实现同一本地端口与多个客户端建立通信连接。

时序图



3.13.6 TCP_Connect

对于客户端而言，要与服务器进行通信，需要向服务器发出连接请求。TCP_Connect指令用于完成这项功能。

TCP_Connect — TCP发起连接请求指令

图形块

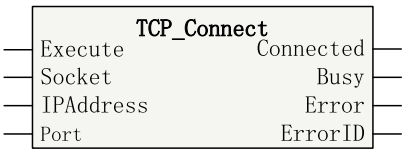


表3-203 指令表格式

16位指令	—					
32位指令	TCP_Connect 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	IPAddress	IP地址	否	-	-	DINT
S3	Port	端口	否	-	1-65535	DINT
D1	Connected	连接状态	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	错误码	是	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2 请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码*。

表3-204 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-

说明

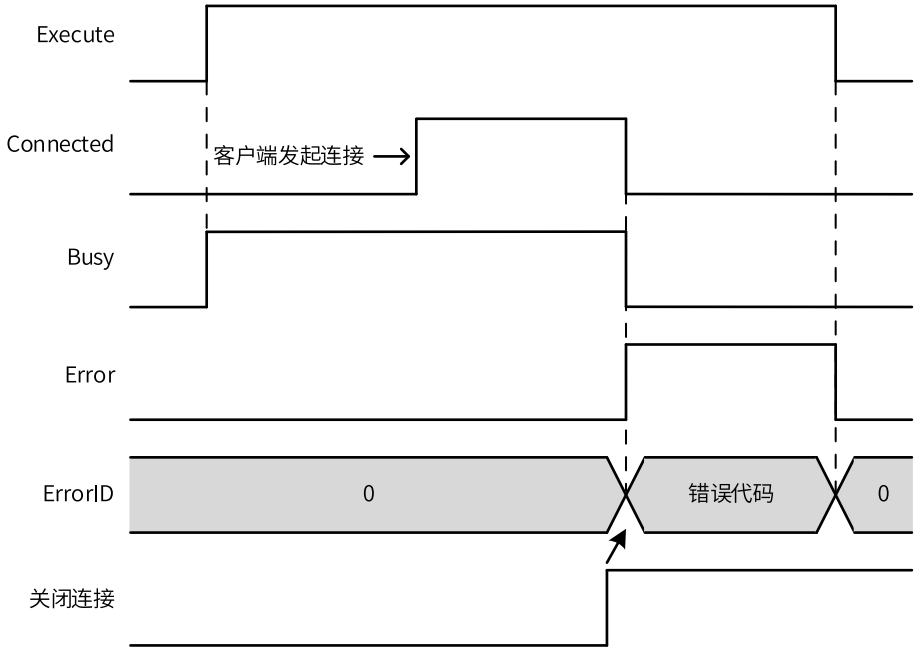
[1]不支持X元件。

功能和指令说明

当本机做为客户端，要与服务器进行通讯时，通过TCP_Connect指令去连接服务器指定的端口。在服务器接收连接后，就可以通过TCP_Send或TCP_Receive进行发送或接收数据。

TCP_Connect指令向服务器发出连接请求后，最多等待127秒，若服务器没有响应，则连接失败。

时序图



3.13.7 TCP_Close

使用TCP_Close指令可以在完成通信后关闭连接或监听。

TCP_Close — TCP关闭连接指令

图形块

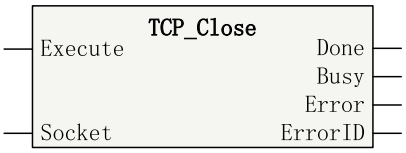


表3–205 指令表格式

16位指令	—					
32位指令	TCP_Close连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	错误码	是	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2 请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码*。

表3-206 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-

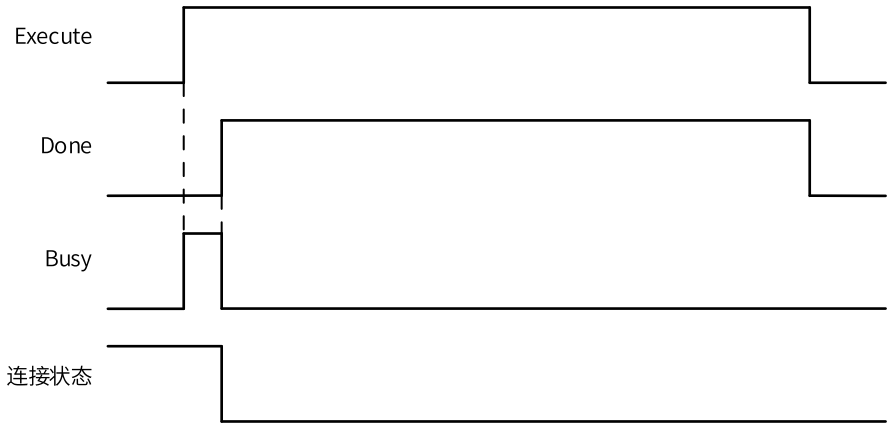
说明

[1]不支持X元件。

功能和指令说明

在完成通信后，可以使用TCP_Close指令来关闭连接、停止监听或终止正连接的套接字。

时序图



3.13.8 TCP_Send

当服务器与客户端之间成功建立连接后，可以通过TCP_Send指令向远程主机发送数据。

TCP_Send — TCP发送数据指令

图形块

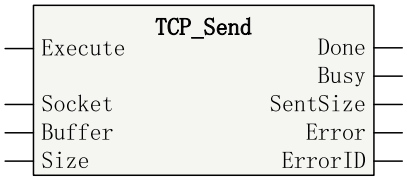


表3-207 指令表格式

16位指令	—
32位指令	TCP_Send连续执行

操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	Buffer	数据缓存	否	-	-	INT[]
S3	Size	数据大小	是	0	0-32767	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	SentSize	已发送数据大小	是	0	TRUE/FALSE	INT
D4	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	错误码	是	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2 请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码*。

表3–208 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

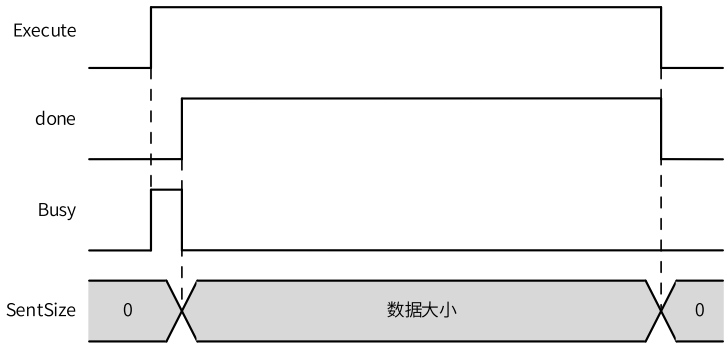
[1]不支持X元件。

功能和指令说明

服务器与客户端之间成功建立连接后，远程主机发送给本机消息数据将被保存在套接字缓存区中，通过TCP_Receive指令从指定套接字缓存区中获取已经收到的消息数据。

数据大小（size）参数必须小于或等于数据缓存（Buffer参数）的实际大小，否则存在数据访问越界的风险。

时序图



说明 当指令的数据帧还在发送中尚未发完，如果重新触发该条指令，则会因为时序异常，指令报错。为避免该问题，推荐使用指令的Done或Error状态的变化，触发下一次发送或触发下一条指令。

3.13.9 TCP_Receive

当服务器与客户端之间成功建立连接后，可以通过TCP_Receive指令从指定套接字中获取远程主机发送的消息数据。

TCP_Receive — TCP接收数据指令

图形块

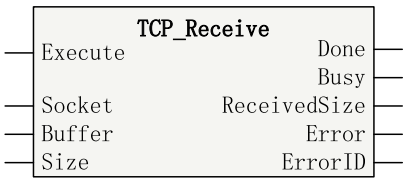


表3–209 指令表格式

16位指 令	—					
32位指令	TCP_Receive连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	Buffer	数据缓存	否	-	-	INT[]
S3	Size	数据大小	否	-	1-32767	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	ReceivedSize	接收到数据大小	是	0	0-32767	INT
D4	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	错误码	是	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码。*

表3–210 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

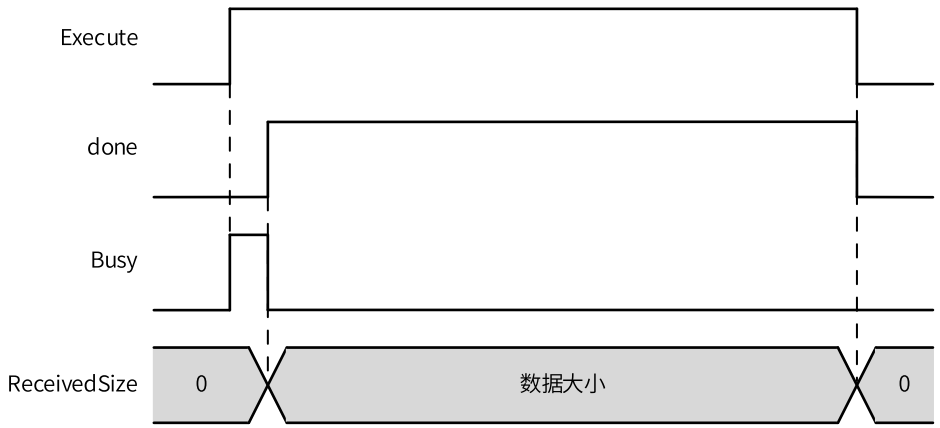
[1]不支持X元件。

功能和指令说明

数据大小（size）参数为0或不填时，将按字符串格式进行发送，即发送数据缓存（Buffer参数）中从第一个字节到结束符（不包括结束符）之间的数据，结束符的ASCII代码是0。

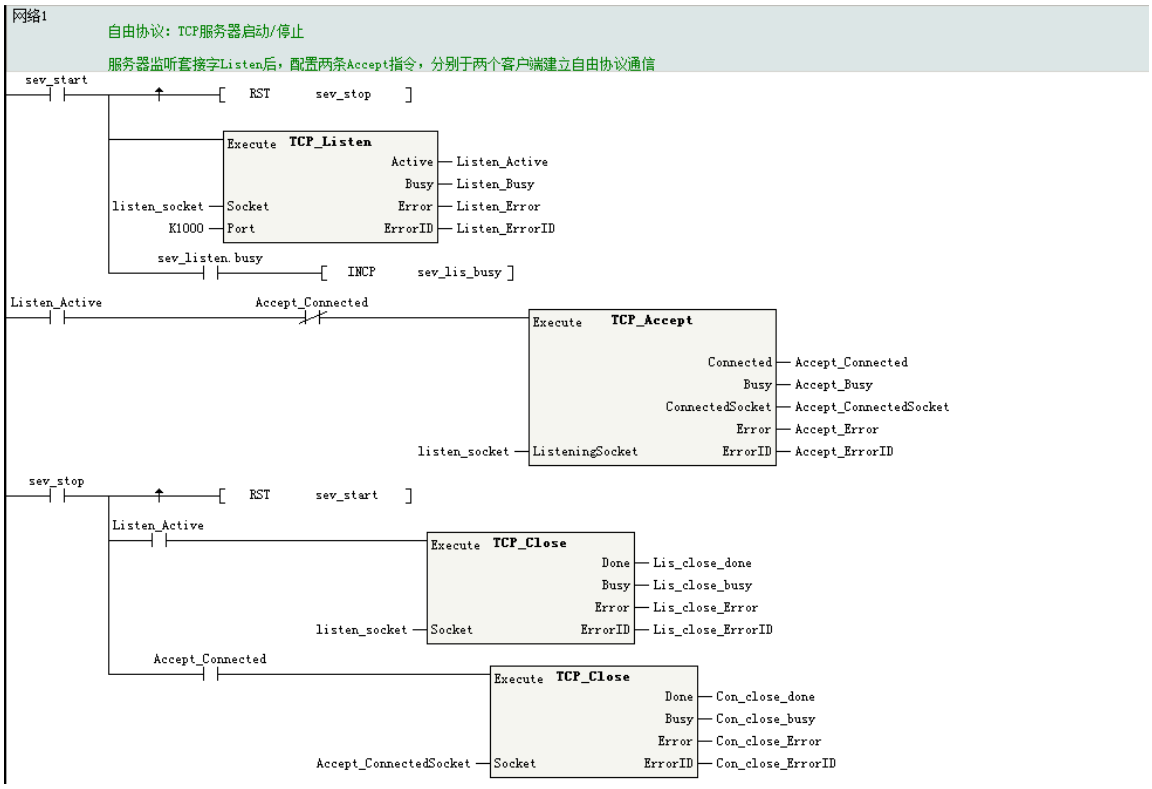
数据大小（size）参数必须小于或等于数据缓存（Buffer参数）的实际大小，否则存在数据访问越界的风险。

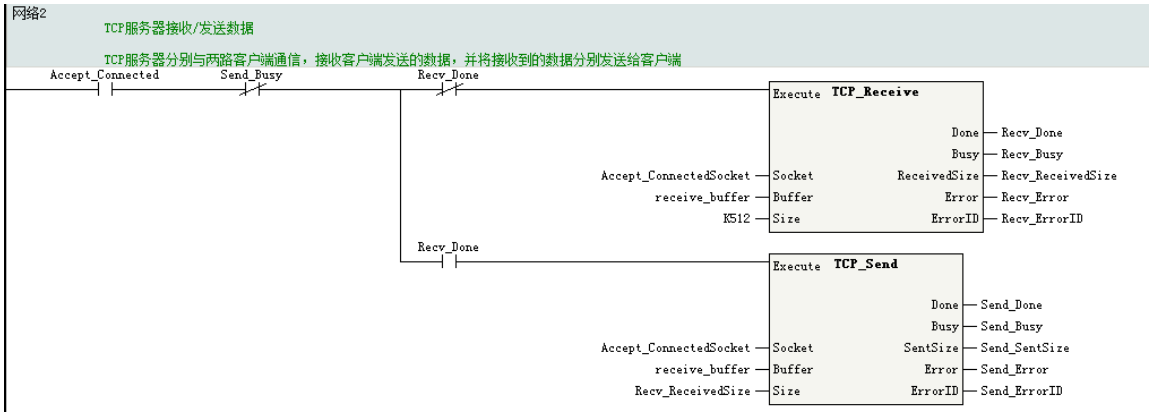
时序图



3.13.10 TCP服务器通信实例

实现监听TCP端口1000，并在接受客户端连接后，将从客户端接收到的数据发送回给客户端。
实例编程如下：

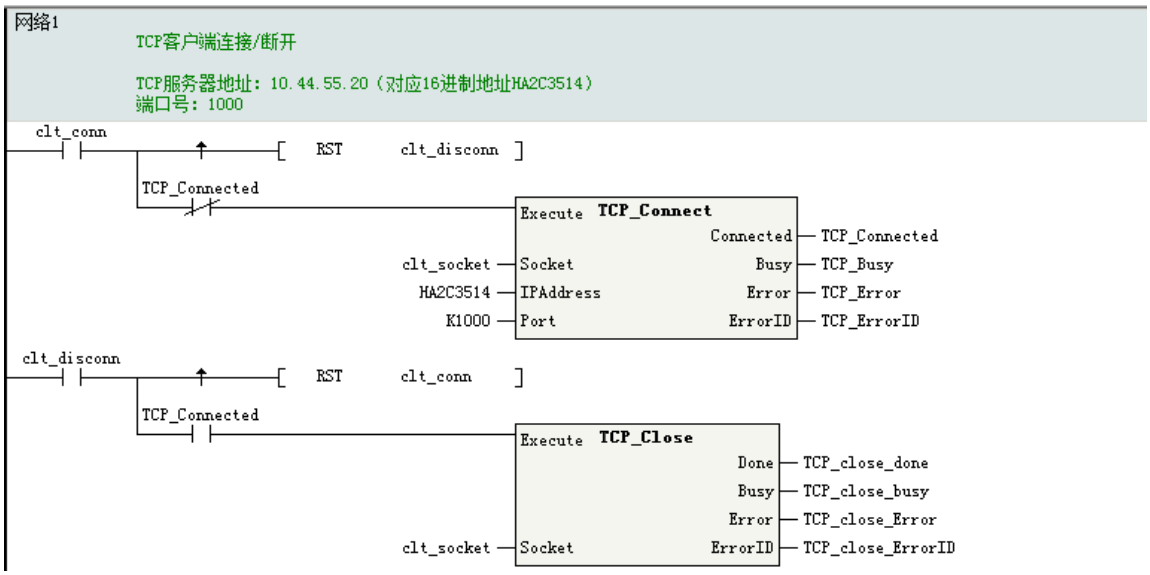


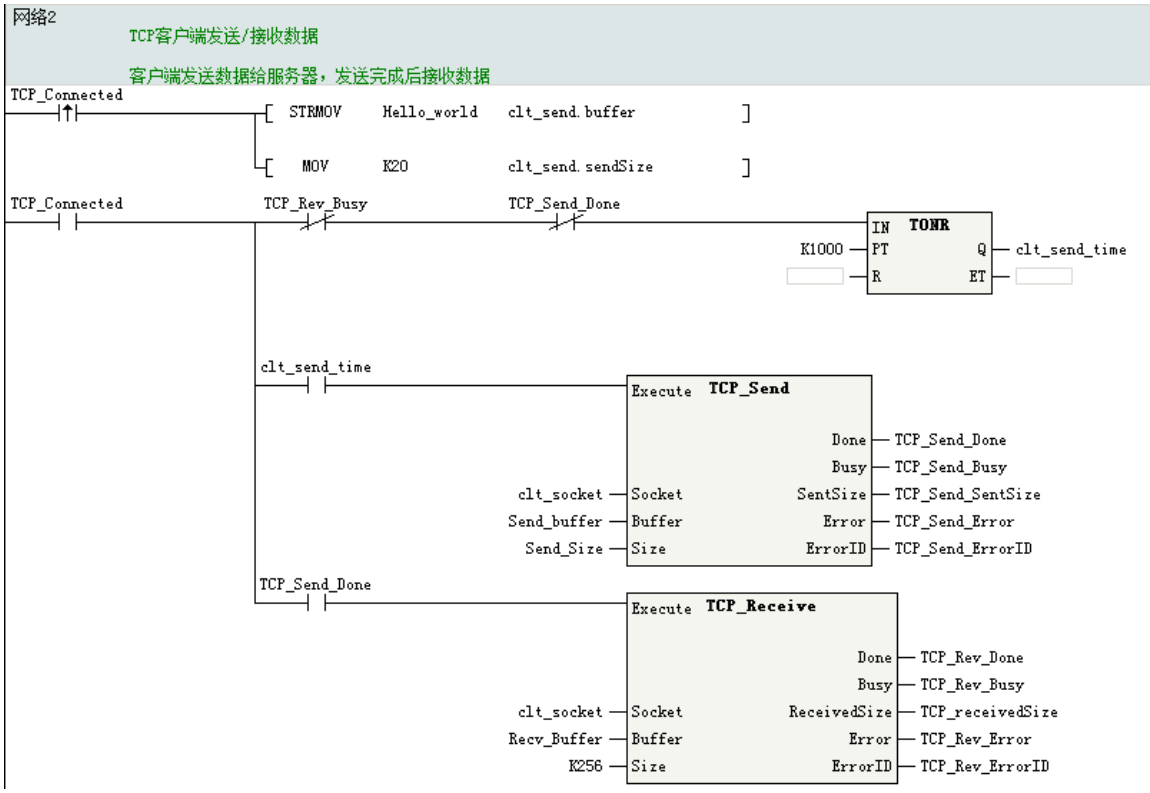


3.13.11 TCP客户端通信实例

实现连接10.44.53.20：1000，并在连接后，周期（1秒）向服务器发送"Hello!",收到服务器发送的任意数据后，停止发送。

实例编程如下：

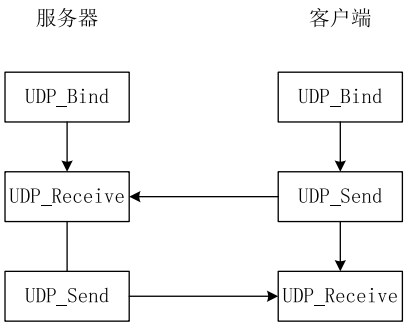




3.13.12 面向无连接的套接字UDP通信

用户数据报协议（UDP，User Datagram Protocol）一种无连接的传输层协，它主要用于不要求分组顺序到达的传输中，分组传输顺序的检查与排序由应用层完成，提供面向事务的简单不可靠信息传送服务。UDP报文没有可靠性保证、顺序保证和流量控制字段等，可靠性较差。但是正因为UDP协议的控制选项较少，在数据传输过程中延迟小、数据传输效率高，适合对可靠性要求不高的应用程序，或者可以保障可靠性的应用程序。

H5U的UDP通信连接套接字流程如下图所示：



3.13.13 UDP_Receive

UDP_Receive指令用于接收远程主机发过来的数据。

UDP_Receive — UDP接收数据指令

图形块

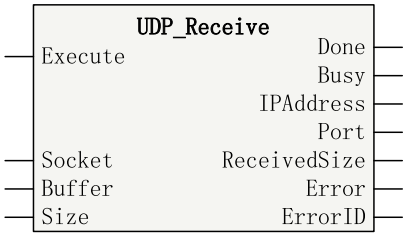


表3-211 指令表格式

16位指令	—					
32位指令	UDP_Receive连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	Buffer	数据缓存	否	-	-	INT[]
S3	Size	数据大小	否	-	1-32767	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	IPAddress	IP地址	是	0	-	DINT
D4	Port	端口	是	0	1-65535	DINT
D5	ReceivedSize	接收到数据大小	是	0	0-32767	INT
D6	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D7	ErrorID	错误码*2	是	0	-	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码。*

表3–212 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	✓	-	-
D4	-	-	-	✓	✓	✓	✓	-	-
D5	-	-	-	✓	✓	✓	-	-	-
D6	✓[1]	✓	✓	-	-	✓	-	-	-
D7	-	-	-	✓	✓	✓	-	-	-

说明

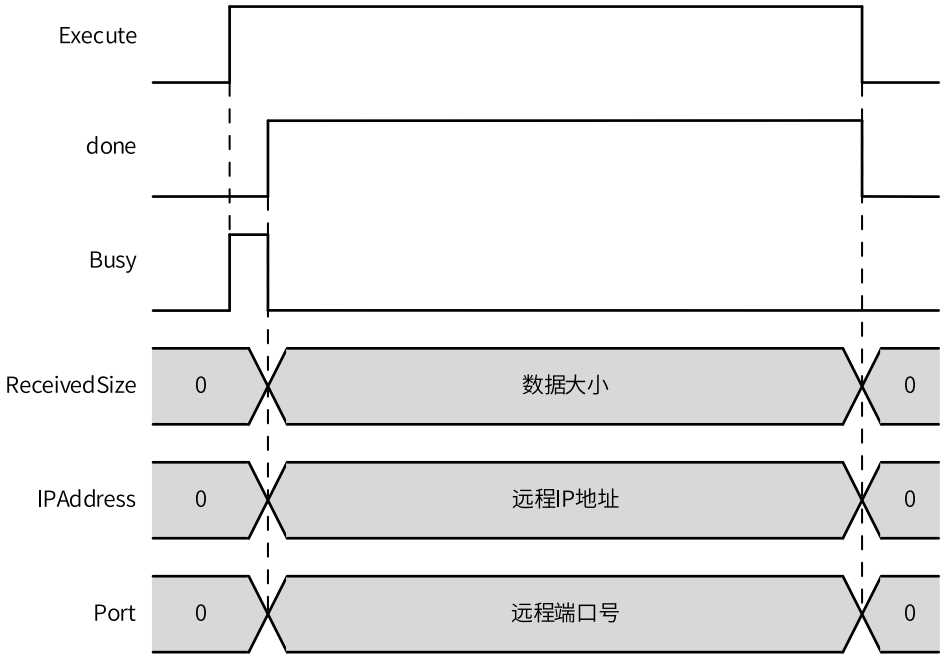
[1]不支持X元件。

功能和指令说明

远程UDP主机发送给本机消息数据将被保存在套接字缓存区中，通过UDP_Receive指令从指定套接字缓存区中获取已经收到的消息数据。

数据大小（size）参数必须小于或等于数据缓存（Buffer参数）的实际大小，否则存在数据访问越界的风险。

时序图



3.13.14 UDP_Bind

UDP发送或接收数据前，需要将套接字与本机的端口绑定，UDP_Bind指令用于将套接字与指定UDP端口相连。

UDP_Bind — UDP套接字绑定指令

图形块

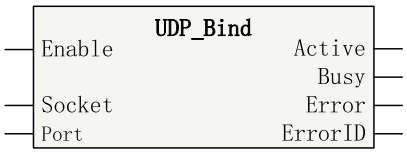


表3-213 指令表格式

16位指令	—					
32位指令	UDP_BIND连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	Port	端口*2	是	0	0-65535	DINT
D1	Active	有效状态	是	FASLE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FASLE	TRUE/FALSE	BOOL
D3	Error	错误	是	FASLE	TRUE/FALSE	BOOL
D4	ErrorID	错误码*3	是	0	-	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2不可以使用系统内部专用端口号(12939、12940)。
- *3请参考*第287页“3.13.17 套接字通信指令错误代码”**套接字通信指令错误代码*。

表3-214 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

UDP发送或接收数据前，需要将套接字与本机的端口绑定，UDP_Bind指令用于将套接字与指定UDP端口相连。当指定的端口号为0或为空时，UDP_Bind将自动随机分配一个端口号，分配的端口号将在套接字的LocalPort的指示。

时序图

3.13.15 UDP_Send

UDP_Send指令用于发送数据给指定的远程主机。
UDP_Send — UDP发送数据指令

图形块

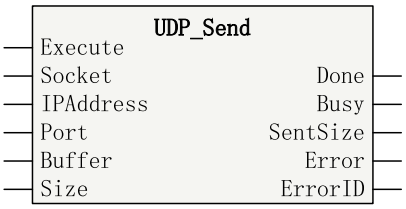


表3–215 指令表格式

16位指令	—					
32位指令	UDP_SEND连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Socket	套接字*1	否	-	-	_sSocket
S2	IPAddress	IP地址	否	-	-	DINT
S3	Port	端口	否	-	1-65535	DINT
S4	Buffer	数据缓存	否	-	-	INT[]
S5	Size	数据大小	是	0	0-32767	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	执行中	是	FALSE	TRUE/FALSE	BOOL
D3	SentSize	已发送数据大小	是	0	0-32767	INT
D4	Error	错误	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	错误码	是	0	*2	INT

说明

- *1 _sSocke数据类型对应的参数均为输入输出数据类型。
- *2请参考*第287页“3.13.17 套接字通信指令错误代码”*套接字通信指令错误代码。

表3–216 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	-	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
S5	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

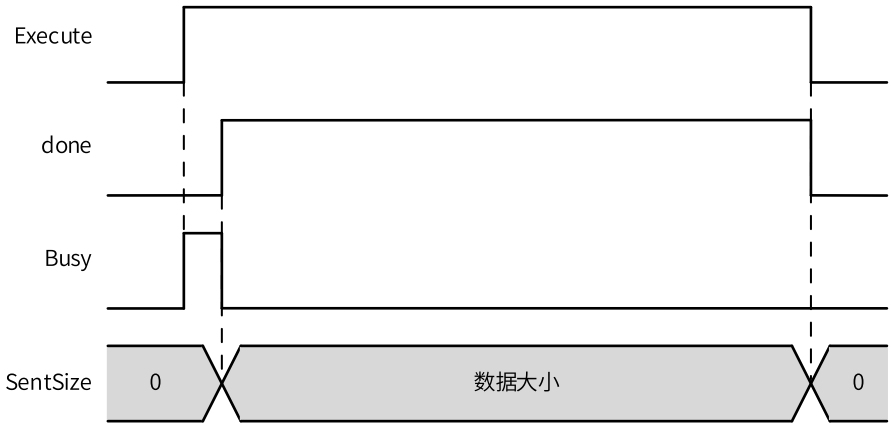
说明

[1]不支持X元件。

功能和指令说明

数据大小（size）参数为0或不填时，将按字符串格式进行发送，即发送数据缓存（Buffer参数）中从第一个字节到结束符（不包括结束符）之间的数据，结束符的ASCII代码是0。

时序图

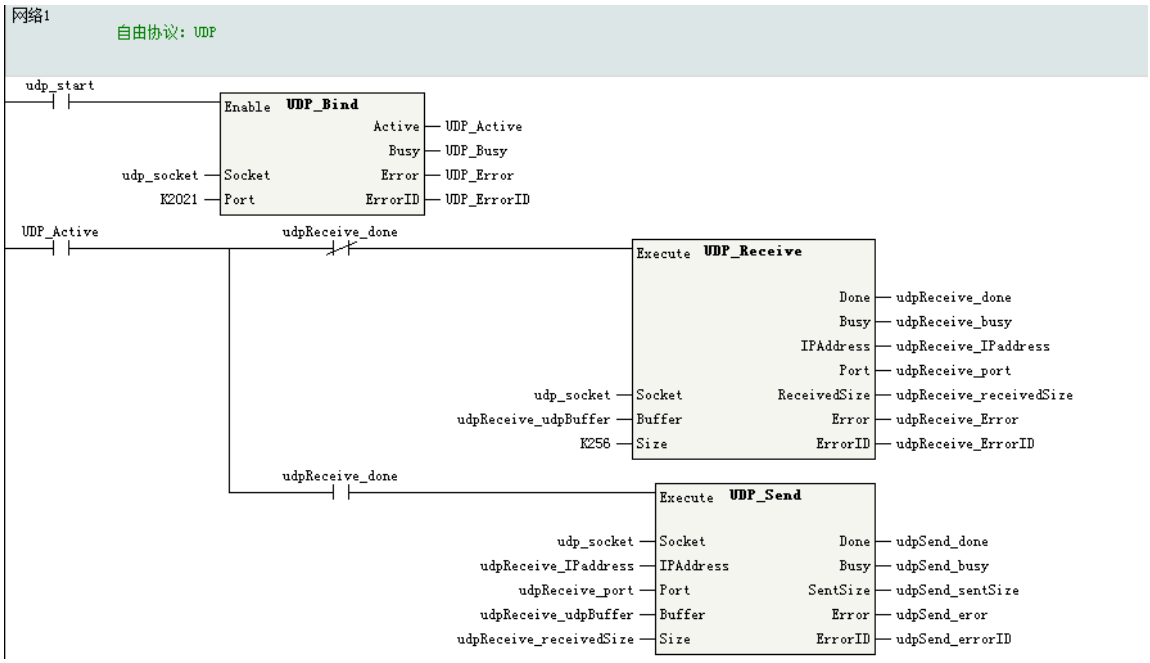


说明 当指令的数据帧还在发送中尚未发完，如果重新触发该条指令，则会因为时序异常，指令报错。为避免该问题，推荐使用指令的Done或Error状态的变化，触发下一次发送或触发下一条指令。

3.13.16 UDP通信实例

实现UDP绑定端口1000，并在接收远程主机发的数据后，将收到的数据回发给远程主机。

程序如下图所示：



3.13.17 套接字通信指令错误代码

套接字通信指令出错时，请查看下表获取错误代码定义。

表3–217 错误代码定义

错误代码	说明
1~8	系统错误
9	无效套接字
10~13	系统错误
14	无效数据缓存
15~76	系统错误
77	套接字状态错误
78~87	系统错误
88	在非套接字上进行套接字操作
89	需要目标地址与端口
90	报文太长
91	错误协议类型
92	协议不可用
93	不支持协议
94	不支持套接字类型
95	系统错误
96	不支持协议族
97	协议不支持地址群

错误代码	说明
98	地址与端口已被使用
99	无法分配请求的地址与端口
100	网络已关闭
101	网络不可达
102	网络由于复位断开连接
103	连接已经终止
104	连接被对方复位
105	没有可用的缓存空间
106	套接字已连接
107	套接字未连接
108	套接字关闭后不能再发送
109	系统错误
110	连接超时
111	连接被拒绝
112	主机已关闭
113	无法路由到主机
500	触发发送时序异常，之前的数据帧还在发送中

3.13.18 ECT_ReadParameter_CoE

ECT_ReadParameter_CoE — 读取从站的SDO参数

图形块

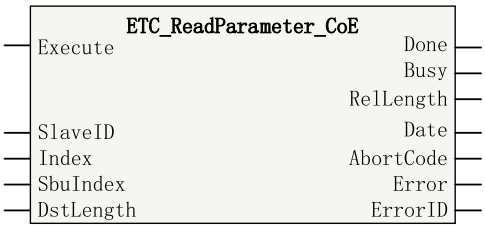


表3-218

16位指令	-			
32位指令	ECT_ReadParameter_CoE 连续执行			
操作数	名称	描述	范围	数据类型
S1	Slave	仅可以输入从站顺序地址	0-71	INT
S2	Index	索引	-	INT
S3	SubIndex	子索引	-	INT
S4	DstLength	目标长度	1-4	INT
D1	Done	轴完成标志	-	BOOL
D2	Busy	忙标志	-	BOOL
D3	RelLength	读到的实际长度，单位是字节	1-4	INT
D4	data	读到的数据	-	DINT

D5	AbortCode	读从站对象字点失败时的Abort Code码*1	-	DINT
D6	Error	指令故障标志	-	BOOL
D7	ErrorID	故障码*2	-	INT

说明

- *1请参见第页 “”
- *2请参见第页 “”

表3–219 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-
D6	✓[1]	✓	✓	-	-	✓	-	-	-
D7	-	-	-	✓	✓	✓	-	-	-

说明

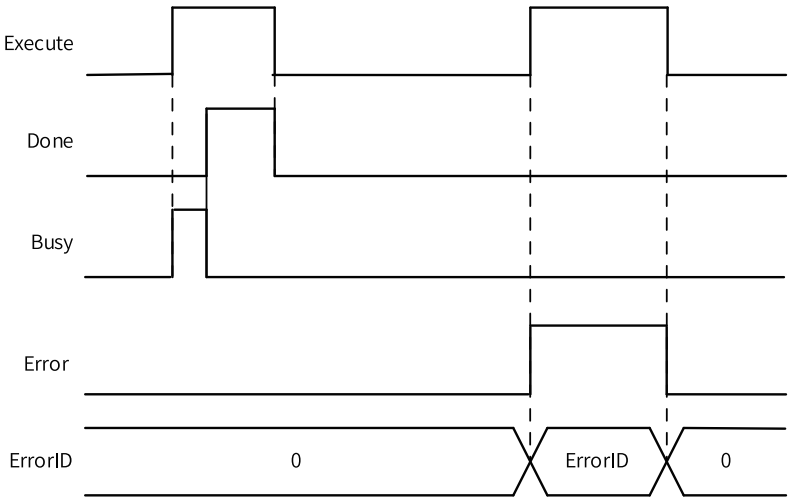
[1]不支持X元件。

功能和指令说明

本指令用于读取EtherCAT从站的对象字典，上升沿有效。

- SlaveID用于指定EtherCAT从站的组态地址。
- 在Execute的上升沿，指令锁存左侧的输入参数，并触发读取Index和SubIndex指定的对象字典。
- DstLength用于指定要读取的对象字典的长度，单位是字节。
- 当读取成功后，Done信号有效，Dtate用于显示读取到的值，RelLength用于显示实际读取到的对象字典的长度。当读取当失败时，Error输出有效，AbortCode和ErrorID配合确定读取失败的原因。
- 本指令中Data参数为DINT类型参数，占用4个字节的空间，当读取的对象字典为SINT或INT时，将读取到的结果放置在Data参数的低8位或低16位，然后将没有用到的高24位或高16位填充0。例如当读取到SINT类型和INT类型的-8时Data的实际存放的数据分别为0x000000f8和0x0000fff8。

时序图



3.13.19 ECT_WriteParameter_CoE

ECT_WriteParameter_CoE — 写入从站的SDO参数

图形块

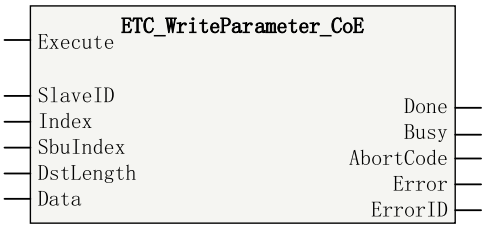


表3–220 指令表格式

16位指令	-			
32位指令	ECT_WriteParameter_CoE 连续执行			
操作数	名称	描述	范围	数据类型
S1	Slave	仅可以输入从站组态地址	0-71	INT
S2	Index	索引	-	INT
S3	SubIndex	子索引	-	INT
S4	DstLength	目标长度	1-4	INT
S5	Data	目标数据	-	DINT
D1	Done	轴完成标志	-	BOOL
D2	Busy	忙标志	-	BOOL
D3	AbortCode	读失败时的Abort Code*1	-	DINT
D4	Error	指令故障标志	-	BOOL
D5	ErrorID	故障码*2	-	INT

说明

- *1请参见第页 “”
- *2请参见第页 “”

表3–221 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
S5	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

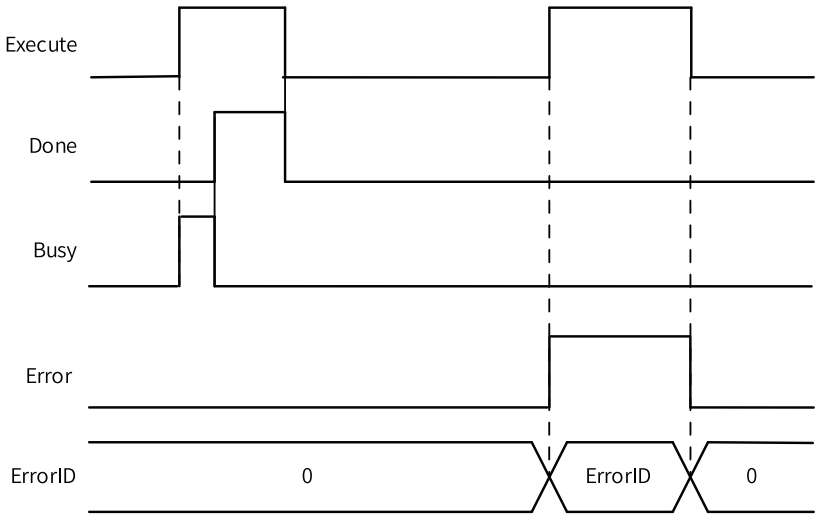
[1]不支持X元件。

功能和指令说明

本指令用于写EtherCAT从站的对象字典，上升沿有效。

- SlaveID用于指定EtherCAT从站的组态地址。
- 在Execute的上升沿，指令锁存左侧的输入参数，并将Data中的数据写入Index和SubIndex指定的对象字典。
- DstLength用于指定要写入的对象字典的长度，单位是字节。
- 当写入成功后，Done信号有效。当写入失败时，Error输出有效，AbortCode和ErrorID配合确定读取失败的原因。

时序图



3.14 运动控制轴指令

3.14.1 指令列表

运动控制轴指令涵盖以下指令条目：

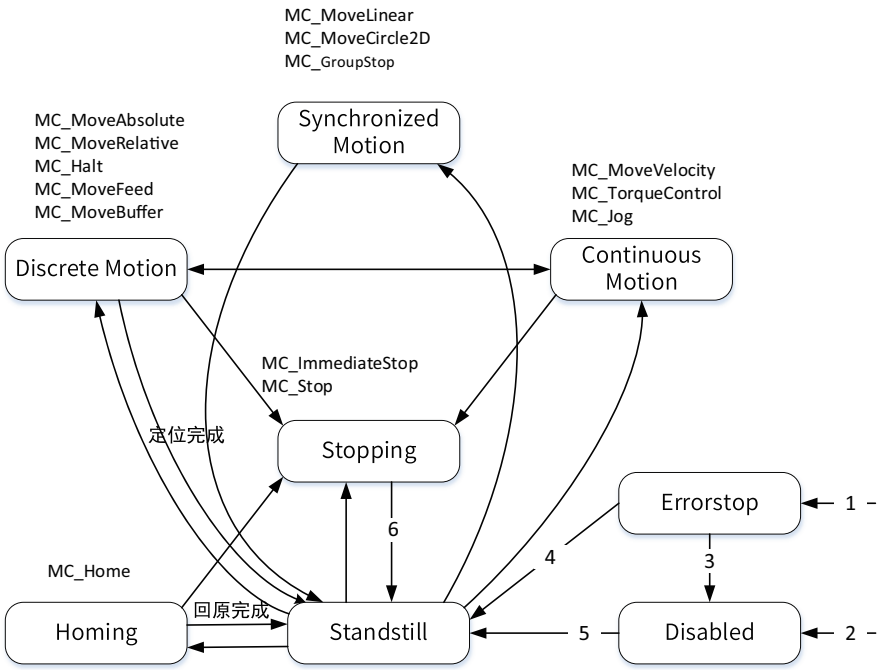
表3-222 运动控制轴指令

指令	名称
MC_Power	使能控制指令
MC_Reset	复位故障指令
MC_ReadStatus	读取轴状态指令
MC_ReadAxisError	读取轴错误指令
MC_ReadDigitalInput	读取数字量指令
MC_ReadActualPosition	读取实际位置指令
MC_ReadActualVelocity	读取实际速度指令
MC_ReadActualTorque	读取实际力矩指令
MC_SetPosition	设置定位指令
MC_TouchProbe	探针指令
MC_MoveRelative	相对定位指令
MC_MoveAbsolute	绝对定位指令
MC_MoveVelocity	速度指令
MC_Jog	点动运动指令
MC_TorqueControl	力矩控制指令
MC_Home	原点回归指令
MC_Stop	停止指令
MC_Halt	暂停指令
MC_ImmediateStop	急停指令

指令	名称
MC_MoveFeed	中断定长指令
MC_MoveBuffer	多段位置定位指令

3.14.2 MC轴状态机简介

H5U基于PLCOpen状态机对轴的状态和运动进行管理。



状态机详细描述如下：

表3-223 状态定义

状态	功能描述
Disabled	未使能状态
ErrorStop	故障停机状态
Standstill	使能状态
Homing	原点回归状态
Stopping	停止状态
Discrete Motion	离散运动
Continuous Motion	连续运行状态
Synchronized Motion	同步运行状态

表3-224 状态迁移条件

转换	转换条件
1	当轴的故障检测逻辑检测到故障时立即进入该状态
2	当轴无故障且MC_Power.Enable=FALSE时
3	当调用MC_Reset复位轴故障且MC_Power.Status=FASLE时
4	当调用MC_Reset复位轴故障且MC_Power.Status=TRUE时

转换	转换条件
5	当MC_Power.Enable=TRUE且MC_Power.Status=TRUE时
6	当MC_Stop(MC_ImmediateStop).Done=TRUE且MC_Stop(MC_ImmediateStop).Execute=TRUE时

指令参数精度

指令中目标位置和目标速度等浮点数采用的是单精度浮点型，因此在PLC程序中处理时指令的值要符合单精度浮点型的范围和精度，即其数值范围为-3.4E38～3.4E38，精度位7位有效数字，如果某个数的有效数字位数超过7位，超出的部分会自动四舍五入。

3.14.3 MC_Power

MC_Power — 使能控制指令

图形块

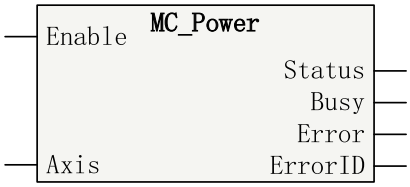


表3–225 指令表格式

16位指令	MC_Power 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Status	轴使能标志	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	故障码*1	是	0	-	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*章节“轴故障码”。

表3–226 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	
D2	✓[1]	✓	✓	-	-	✓	-	-	
D3	✓[1]	✓	✓	-	-	✓	-	-	
D4	-	-	-	✓	✓	✓	-	-	

说明

[1]不支持X元件。

功能和指令说明

MC_Power指令适用于EtherCAT总线轴和本地脉冲轴，用于设定轴的使能状态，高电平有效。

- 轴号的设定
  在Enable的上升沿指令锁存Axis指定的轴。

  如果采用轴号访问轴，当Enable=TRUE时修改Axis，指令报错，且原来控制的轴去使能。

  如果采用轴号访问轴，当Enable=FALSE时修改Axis，若Axis对应的轴原来处于使能状态，且本指令为一个PLC扫描周期内执行的最后一条Power指令，则Axis对应的轴将去使能，变成不可运行状态。

- 功能描述
  设置Enable为True以后，轴进入使能状态，指令的Status信号有效

  轴的PLCOpen状态机由Disabled状态进入StandStill状态。

  在使能之后可以执行如MC_MoveRelative等运动类指令。

  设置Enable为False以后，可解除轴的使能状态，中断运动类指令（如MC_MoveAbsolute）的执行。解除使能状态后，轴不接受动作指令，无法实现轴控制。但是，可以执行MC_Power、MC_Reset、MC_SetPosition等非运动指令。

  当轴由于故障进入errorstop状态后，重新使能MC_Power不能将轴切换到Standstill状态，必须先调用MC_Reset指令将轴的故障恢复。

- 多重调用
  当使用多条MC_Power指令时，以一个循环周期内最后一条执行的MC_Power指令的控制能流为准。

报错

当轴号不存在时指令报错。

当轴类型错误时指令报错。

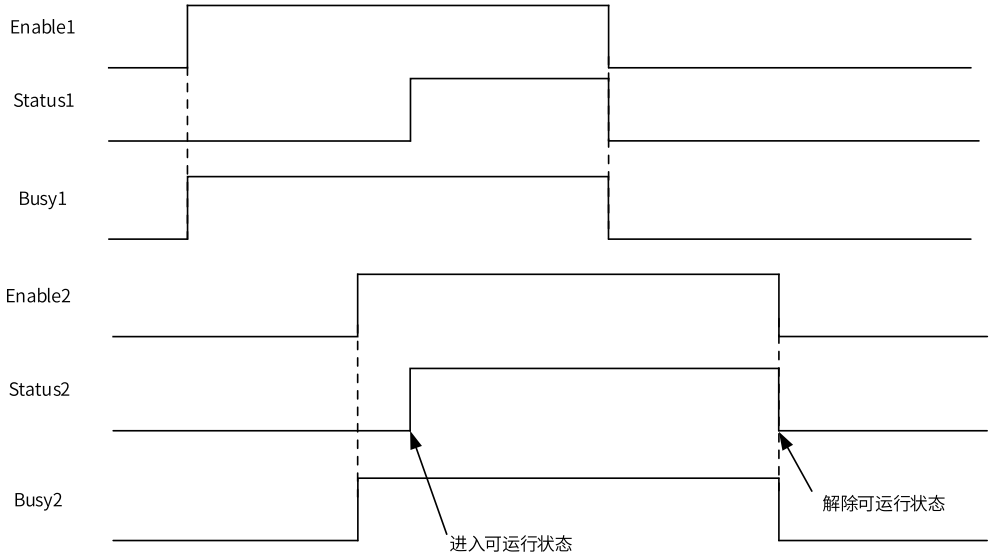
当轴初始化失败时指令报错。

轴的PDO中没有配置控制字、状态字、目标位置、当前位置四个参数时报错。

时序图

- 当使用一条MC_Power指令，轴正常使能时

- 当使用两条MC_Power指令，轴正常使能时



3.14.4 MC_Reset

MC_Reset — 复位故障指令

图形块

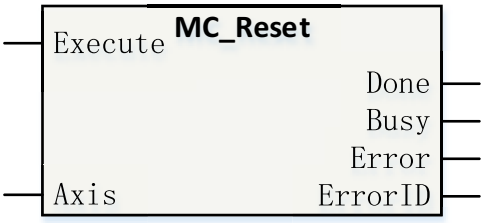


表3–227 指令表格式

16位指令	MC_Reset 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Done	复位完成标志	是	FALSE	TRUE/FALSE	BOOL

D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	故障码*1	是	0	-	INT

说明

*1请参见第370页“3.14.24 轴故障码”章节“轴故障码”。

表3–228 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	
D2	✓[1]	✓	✓	-	-	✓	-	-	
D3	✓[1]	✓	✓	-	-	✓	-	-	
D4	-	-	-	✓	✓	✓	-	-	

说明

[1]不支持X元件。

功能和指令说明

MC_Reset指令适用于EtherCAT总线轴和本地轴，用于复位轴的故障，上升沿触发。

在Execute信号有效的上升沿，指令尝试复位轴的故障，如果复位成功，则Done输出有效，否则Error信号有效，ErrorID将给出复位失败的原因。

复位成功后，如果驱动器处于使能状态则轴的PLCOpen状态机进入StandStill状态，如果驱动器没有使能则进入Disabled状态。

打断

本指令没有打断输出信号，在执行期间不可以被打断。

如果在一个扫描周期有两条复位指令，只要有一条复位指令有效，不论先后，程序将开始执行复位逻辑，如果复位成功，则被触发的指令Done信号输出有效。

报错

当轴号不存在时指令报错。

当轴类型错误时指令报错。

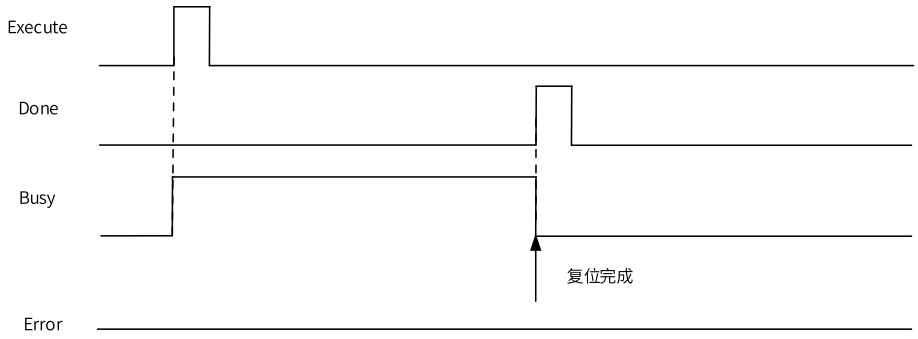
当轴初始化失败时指令报错。

在轴没有故障调用本指令时，指令报错。

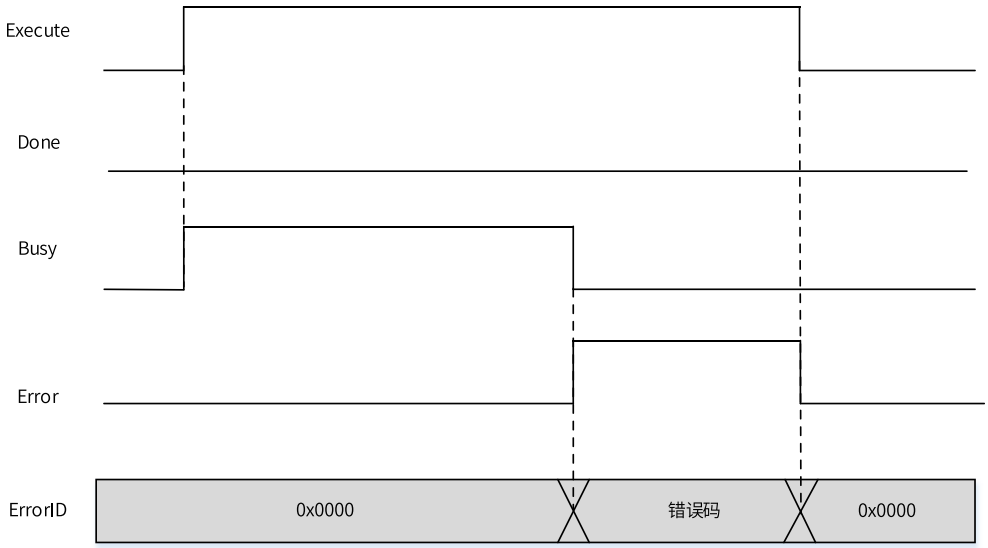
当轴自身无法被复位时指令报错。

时序图

- 在轴发生故障时调用MC_Reset指令且成功复位轴的故障



- 当驱动器发生不可复位的故障时



3.14.5 MC_ReadStatus

MC_ReadStatus — 读取轴状态指令

图形块

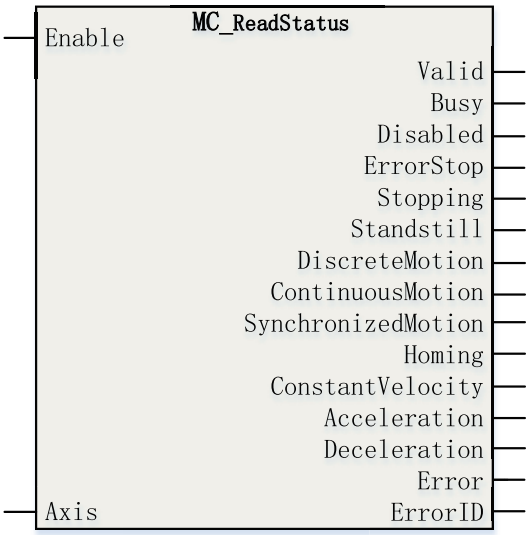


表3–229 指令列表格式

16位指令	MC_ReadStatus 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Disabled	PLCOpen状态机，去使能	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorStop	PLCOpen状态机，故障停机	是	FALSE	TRUE/FALSE	BOOL
D5	Stopping	PLCOpen状态机，停止运行	是	FALSE	TRUE/FALSE	BOOL
D6	Standstill	PLCOpen状态机，使能且非运行状态	是	FALSE	TRUE/FALSE	BOOL
D7	DiscreteMotion	PLCOpen状态机，点位运行模式	是	FALSE	TRUE/FALSE	BOOL
D8	ContinuousMotion	PLCOpen状态机，连续运行模式	是	FALSE	TRUE/FALSE	BOOL
D9	Synchronized Motion	PLCOpen状态机，同步运行模式	是	FALSE	TRUE/FALSE	BOOL
D10	Homing	PLCOpen状态机，原点回归模式	是	FALSE	TRUE/FALSE	BOOL
D11	ConstantVelocity	轴速度为0 轴正在做匀速运动 力矩模式下无效	是	FALSE	TRUE/FALSE	BOOL

D12	Accelerating	轴正在做加速度运动（速度的绝对值在增大） 力矩模式下无效	是	FALSE	TRUE/FALSE	BOOL
D13	Decelerating	轴正在做减速度运动（速度的绝对值在减小） 力矩模式下无效	是	FALSE	TRUE/FALSE	BOOL
D14	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D15	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”章节“轴故障码”。

表3–230 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	✓[1]	✓	✓	-	-	✓	-	-	-
D6	✓[1]	✓	✓	-	-	✓	-	-	-
D7	✓[1]	✓	✓	-	-	✓	-	-	-
D8	✓[1]	✓	✓	-	-	✓	-	-	-
D9	✓[1]	✓	✓	-	-	✓	-	-	-
D10	✓[1]	✓	✓	-	-	✓	-	-	-
D11	✓[1]	✓	✓	-	-	✓	-	-	-
D12	✓[1]	✓	✓	-	-	✓	-	-	-
D13	✓[1]	✓	✓	-	-	✓	-	-	-
D14	✓[1]	✓	✓	-	-	✓	-	-	
D15	-	-	-	✓	✓	✓	-	-	

说明

[1]不支持X元件。

功能和指令说明

在Enable=TRUE时，本指令将读取轴的PLCOpen状态机的状态和加减速状态。

在力矩模式下，ConstantVelocity、Acceleration和Deceleration三个参数一直为False。

EtherCAT任务的优先级大于PLC主任务，如果轴的状态在EtherCAT任务中只存在一个EtherCAT周期，则在PLC主任务中不能获取该状态。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴初始化失败时指令报错。

当轴类型错误时指令报错。

时序图

略

3.14.6 MC_ReadAxisError

MC_ReadAxisError — 读取轴错误指令

图形块

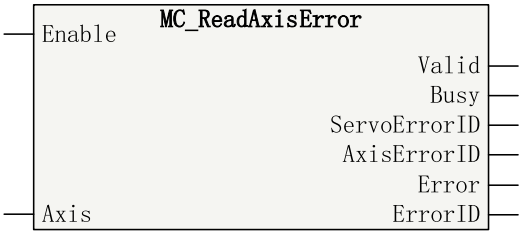


表3–231 指令列表格式

16位指令	MC_ReadAxisError 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ServoErrorID	如果PDO中配置了0x603F，则显示EtherCAT总线驱动器的0x603f的值，否则显示0	是	0	*2	INT
D4	AxisErrorID	轴故障代码，当Axis发生故障时显示故障码	是	0	*1	INT
D5	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D6	ErrorID	故障码	是	0	*1	INT

说明

- *1请参见第370页“3.14.24 轴故障码”章节“轴故障码”。
- *2如果是本地脉冲轴，请查看本地脉冲轴故障列表；如果是显示EtherCAT总线驱动器，请查看显示EtherCAT总线驱动器相关手册中的解释。

表3-232 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-
D5	✓[1]	✓	✓	-	-	✓	-	-	-
D6	-	-	-	✓	✓	✓	-	-	

说明

[1]不支持X元件。

功能和指令说明

本指令用于读取EtherCAT总线轴或者本地脉冲轴的故障。

当Enable=TRUE时，如果请求的轴存在且没有发生配置失败的错误，则Valid信号有效。AxisErrorID用于时时显示轴的故障码，如果没有故障则显示0，如果轴发生故障则显示故障码。如果总线驱动器的PDO中配置了0x603f，则ServoErrorID用于实时显示0x603F的值，否则显示0。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴初始化失败时指令报错。

当轴类型错误时指令报错。

时序图

略

3.14.7 MC_ReadDigitalInput

MC_ReadDigitalInput — 读取数字量输入指令

图形块

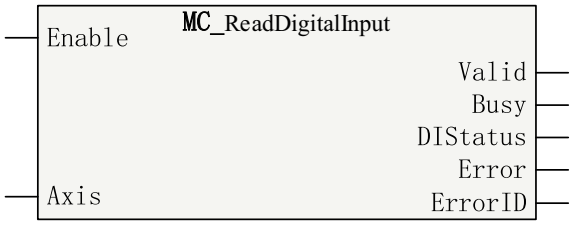


表3-233 指令列表格式

16位指令	—					
32位指令	MC_ReadDigitalInput 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	DIStatus	数字量输入端子的状态，符合402协议的标准格式定义如下： Bit0 - 反向限位信号；Bit1 - 正向限位信号 Bit2 - 原点信号；Bit3~31 - 自定义	是	0	-	DINT
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3-234 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令适用于EtherCAT总线轴和本地脉冲轴，不支持虚轴模式，用于读取轴的数字量输入端子的端子状态。

当Enable=TRUE时，如果请求的EtherCAT总线轴中PDO配置了0x60fd或本地脉冲轴的左右限位和原点信号不全为空，则Valid信号有效。

如果是EtherCAT总线轴，则DIStatus用于时时显示EtherCAT总线驱动器的数字量输入0x60fd，具体定义请查看相应的驱动器手册。

如果是本地脉冲轴，则用于显示限位和原点信号的输入状态，否则显示为0。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴初始化失败时指令报错。

当轴类型错误时指令报错。

当EtherCAT总线轴的PDO中没有配置0x60fd时指令报错。

时序图

略

3.14.8 MC_ReadActualPosition

MC_ReadActualPosition — 读取实际位置指令

图形块

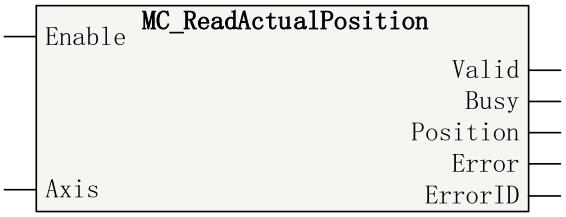


表3–235 指令列表格式

16位指令	—					
32位指令	MC_ReadActualPosition 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL

D3	Position	当前位置	是	0	-	REAL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–236 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	-	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令用于读取EtherCAT总线轴或者本地脉冲轴的反馈位置，高电平有效。

当Enable=TRUE时，如果EtherCAT总线轴中PDO配置了0x6064则Valid信号有效，Position显示轴的反馈位置。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴初始化失败时指令报错。

当轴类型错误时指令报错。

当EtherCAT总线轴的PDO中没有配置0x6064时指令报错。

时序图

略

3.14.9 MC_ReadActualTorque

MC_ReadActualTorque — 读取实际力矩指令

图形块

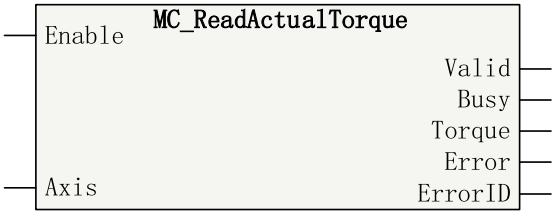


表3–237 指令列表格式

16位指令						
32位指令	MC_ReadActualTorque 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Torque	当前力矩（单位% 1）	是	0	正数/负数/0	REAL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–238 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	-	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令根据读取出EtherCAT总线轴的反馈力矩，高电平有效。本指令不支持虚轴模式。

当Enable=TRUE时，如果EtherCAT总线轴中PDO配置了0x6077则Valid信号有效，Torque显示轴的反馈力矩。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴类型错误时指令报错。

当轴初始化失败时指令报错。

当EtherCAT总线轴PDO没有配置0x6077时指令报错。

时序图

略

3.14.10 MC_ReadActualVelocity

MC_ReadActualVelocity — 读取实际速度指令

图形块

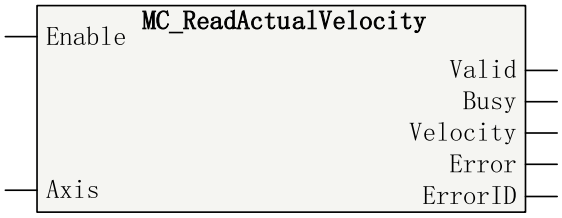


表3-239 指令列表格式

16位指令	—					
32位指令	MC_ReadActualVelocity 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Valid	有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Velocity	当前速度	是	0	-	REAL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–240 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	-	-	-	✓	✓	-	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令根据读取出的EtherCAT总线轴或者本地脉冲轴的反馈位置计算出实际速度，高电平有效。

当Enable=TRUE时，如果EtherCAT总线轴中PDO配置了0x6064则Valid信号有效，Velocity显示轴内部计算出的反馈速度。

打断

本指令没有打断标志，可以多条指令同时运行。

报错

当轴号不存在时指令报错。

当轴初始化失败时指令报错。

当轴类型错误时指令报错。

当EtherCAT总线轴的PDO中没有配置0x6064时指令报错。

时序图

略

3.14.11 MC_SetPosition

MC_SetPosition — 设置定位指令

图形块

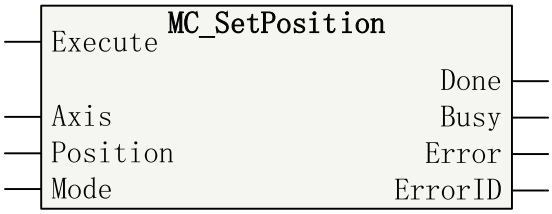


表3–241 指令列表格式

16位指令	—					
32位指令	MC_SetPosition 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Position	目标位置	否	-	正数/负数/0	REAL
S3	Mode	模式选择 0：绝对模式，将Position的值写入当前位置 1：相对模式，在当前位置的基础上增加Position对应的值	是	0	0-1	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D4	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–242 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
S3	-	-	-	✓	-	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

本指令用于设置EtherCAT总线轴或本地脉冲轴的当前位置，上升沿有效。

只有在轴的PLCOpen状态处于disabled、standstill和errorstop三种状态才可以调用本执行设置轴的当前位置，其他状态指令报错。

- 当Mode=0（绝对模式）时，在Execute的上升沿，指令将Position写入轴的当前位置。
- 当Mode=1（相对模式）时，在Execute的上升沿，指令在轴的当前位置基础上加上Position。

打断

本指令不支持打断，如果在一个扫描周期中同时有几条本指令，将执行首先有效的那条指令，且在指令Busy信号有效期间如果有其他SetPosition指令执行，其他指令报错。

报错

当轴号不存在时指令报错。

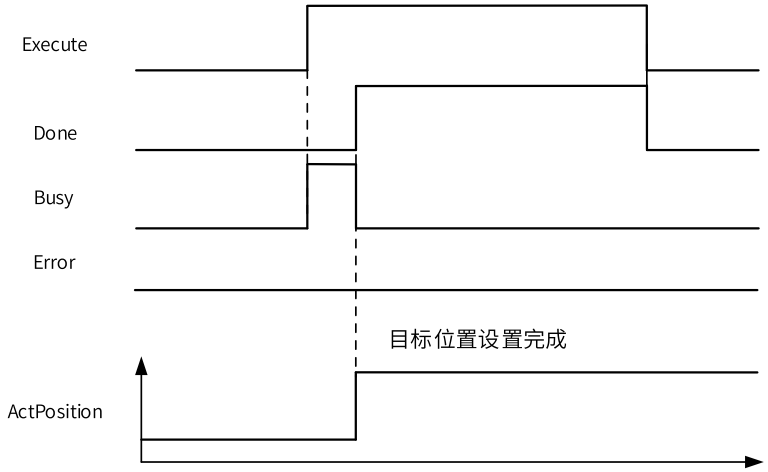
当轴类型错误时指令报错。

当轴初始化失败时指令报错。

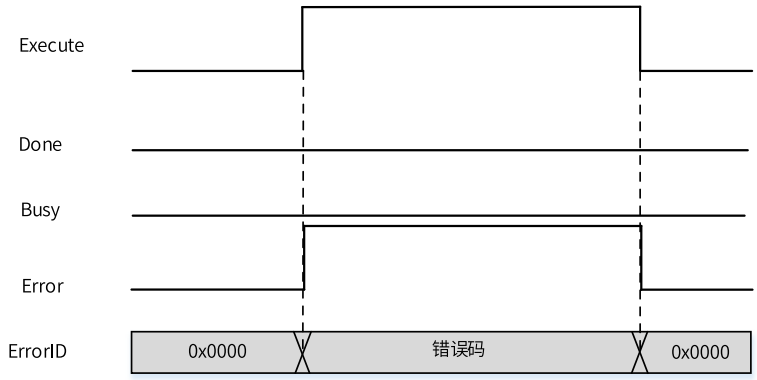
本指令只能在轴停止运行期间设置生效，其他状态则指令报错。

时序图

- 轴在standstill状态下执行本指令，相对模式。



- 轴在点动指令有效期间执行本指令。



3.14.12 MC_TouchProbe

MC_TouchProbe — 探针指令

图形块

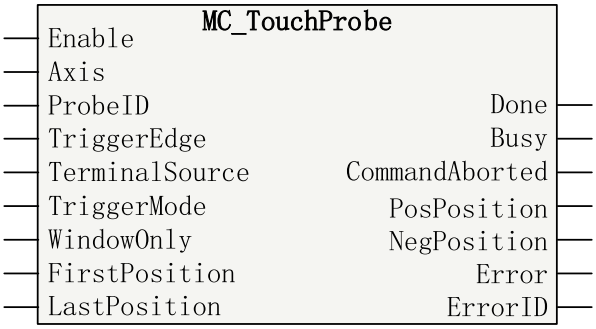


表3-243 指令列表格式

16位指令	—					
32位指令	MC_TouchProbe 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	ProbeID	探针ID 0：探针1 1：探针2	否	-	0-1	INT
S3	TriggerEdge	边沿类型 0：仅上升沿触发 1：仅下降沿触发 2：上升沿和下降沿都可以触发	否	-	0-2	INT
S4	TerminalSource	探针信号源(仅用于设置总线伺服驱动器) 0：DI端子 1：编码器Z信号	是	0	0-1	INT
S5	TriggerMode	触发类型 0：单次触发 1：连续触发	是	0	0-1	INT
S6	WindowOnly	使能探针窗口 0：禁止窗口功能，在所有位置范围内均检测探针信号。 1：使能窗口比较功能，只有当前位置大于等于FirstPosition且小于等于LastPosition时才检测探针信号。	是	FALSE	TRUE/FALSE	BOOL
S7	FirstPosition	探针窗口开始位置	是	0	正数/负数/0	REAL

S8	LastPosition	探针窗口结束位置	是	0	大于First Position	REAL
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	CmdAborted	打断	是	FALSE	TRUE/FALSE	BOOL
D4	PosPosition	上升沿锁存位置	是	0	正数/负数/0	REAL
D5	NegPosition	下降沿锁存位置	是	0	正数/负数/0	REAL
D6	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D7	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–244 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
S5	-	-	-	✓	✓	✓	✓	-	-
S6	✓	✓	✓	-	-	✓	-	-	-
S7	-	-	-	✓	✓	✓	-	✓	-
S8	-	-	-	✓	✓	✓	-	✓	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-
D6	-	-	-	✓	✓	-	-	-	-
D7	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

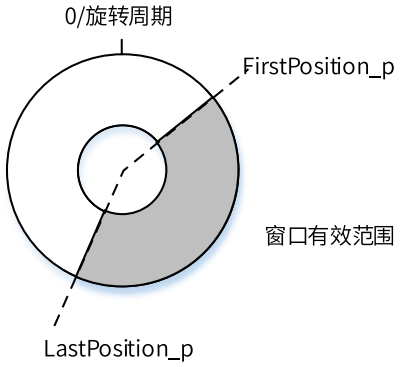
功能和指令说明

该指令用于实现EtherCAT总线轴或本地脉冲轴的探针功能，高电平有效。本指令不支持虚轴模式。

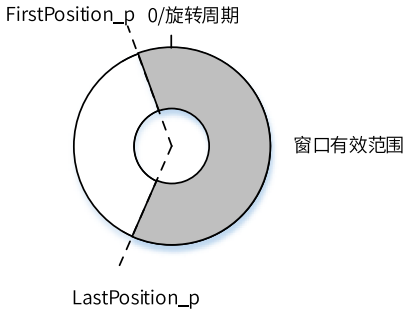
在EtherCAT总线轴模式下，驱动器需要配置探针功能(0x60b8)、探针状态(0x60b9)和锁存位置（0x60ba/0x60bb/0x60bc/0x60bd）。

在本地脉冲轴模式下，需要配置探针信号源。

- 在指令的上升沿，指令将锁存ProbeID、TriggerEdge等左侧的输入参数，其他状态更新参数无效。
- 在Enable=TRUE时，指令在检测到ProbeID指定的探针输入有效且满足探针检测条件时，功能块将锁存轴的当前位置。
- WindowOnly = FALSE，此时窗口检测功能无效。只要探针输入信号有效，就可以锁存探针信号有效时的轴的位置。
- WindowOnly = TRUE，此时窗口检测功能有效。
- 在线性模式下，只有轴的当前位置位于FirstPosition和LastPosition设定的区间内部时，指令才检测探针信号。
- 在环形模式下，指令首先将FirstPosition和LastPosition对循环周期求模，得到在一个循环周期内的区间位置FirstPosition_p和LastPosition_p。
  - 当FirstPosition_p < LastPosition_p时，窗口有效范围如图：



  - 当FirstPosition_p > LastPosition_p，有效范围如图：



    本指令可以单独检测探针信号的上升沿、下降沿或者同时检测上升沿和下降沿。

    在只检测上升沿（下降沿）时，指令将上升沿（下降沿）检测到的值写入PosPosition（NegPosition），此时完成一个检测周期将Done信号置位。

    如果同时检测上升沿和下降沿，则在指令的Enable有效之后，当指令检测到上升沿后立即将位置写入PosPosition，检测到下降沿后立即将位置写入NegPosition，此时才算作一个完整的检测周期输出Done信号，此时对上升沿和下降沿的输入顺序没有要求。

- 对于EtherCAT总线驱动器，本指令的输入TerminalSource可以用来设置端子类型为DI输入端子或者电机的Z信号（需驱动器支持）。如果驱动器不支持Z信号，本指令不会报错。

- 本指令可以单次触发或者连续触发。当选择单次触发时，Done信号输出有效时表示指令执行结束；当选择连续触发模式时， Done输出有效信号一个PLC扫描周期后重新复位，指令自动开始检测新的探针输入信号。

说明

窗口功能有效时，在窗口附近有可能发生丢失探针信号或者检测超出窗口范围的情况，举例如下：

- 在线性模式下，窗口范围是10-100，ethercat周期设置为8ms，速度是100，则每一个ethercat周期轴运动0.8。如果一个ethercat周期开始的时刻当前位置是9.9，则该ethercat周期不检测探针，到下一个ethercat周期，当前位置变成10.7，那么在10-10.7之间的探针信号将丢失。如果一个ethercat周期开始的时刻当前位置是99.9，则该ethercat周期检测探针，到下一个ethercat周期，当前位置变成100.7，那么在100-100.7之间可以响应探针信号。
- 在连续模式下如果探针信号的输入频率大于PLc扫描周期的频率，探针指令将丢失部分探针信号。

打断

探针指令支持检测探针1和探针2，如果程序中定义两条探针指令，两条指令的探针ID选择不同，则两条探针指令将独立工作。如果两条指令的探针ID相同，则后执行的探针指令将打断前一条探针指令。

中断定长指令也会用到探针信号，两条指令的调用规则如下：

- 如果程序中同时存在一条探针指令和一条Mc_MoveFeed指令，两条指令的探针ID选择不同，则两条指令将独立工作。
- 如果两条指令的探针ID相同，若先使能探针指令，在探针指令Busy信号有效期间使能Mc_MoveFeed指令，则探针指令被打断；若先使能Mc_MoveFeed指令，在Mc_MoveFeed指令的Busy信号有效InFeed信号无效期间触发探针指令，探针指令报错。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

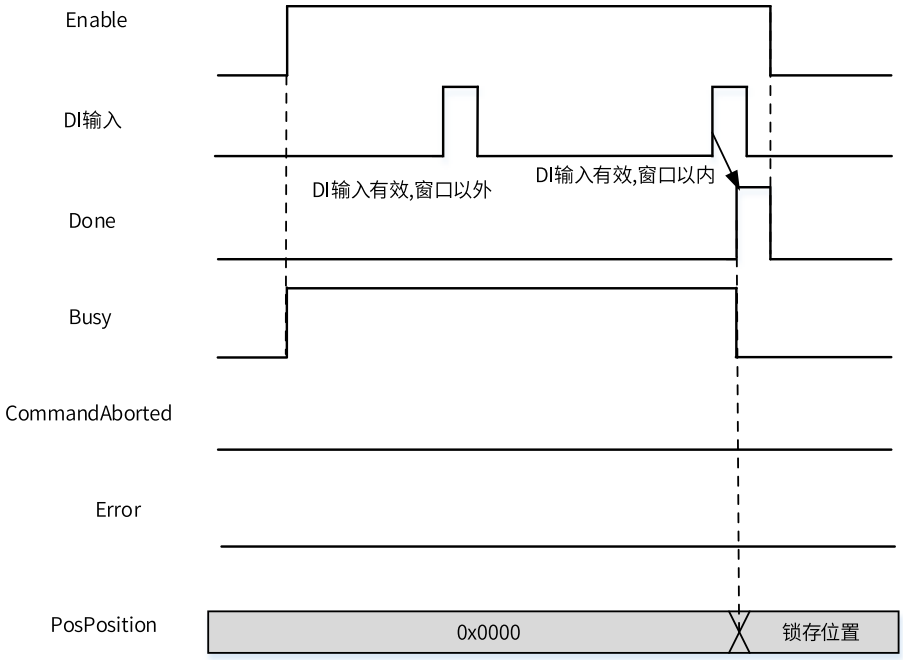
使能虚轴模式后报错9133。

如果没有配置相应的PDO指令报故障。

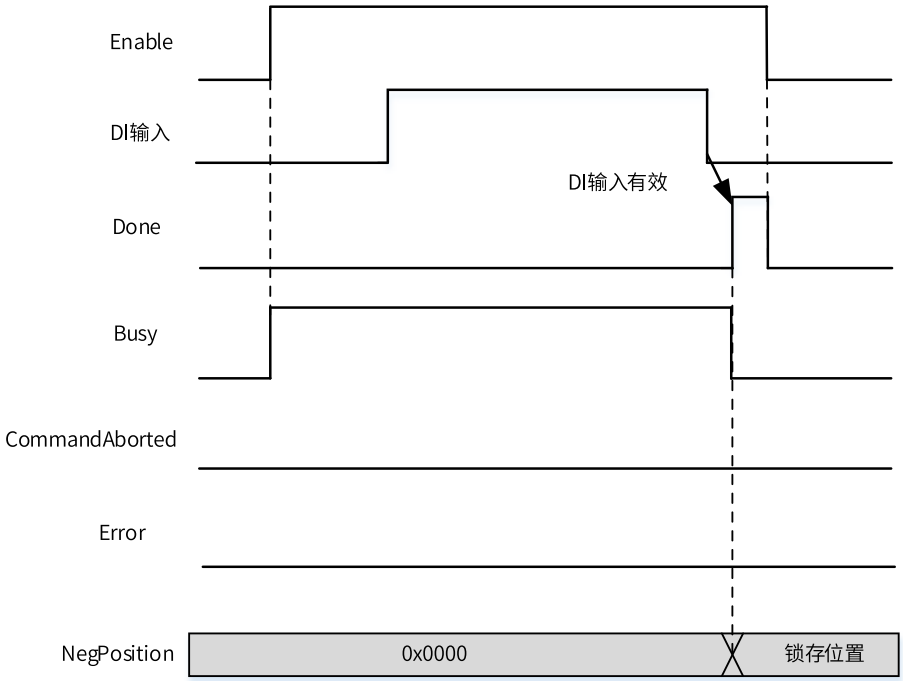
在Enable输入的上升如果指令左侧的参数超范围或不合理则指令报故障。

时序图

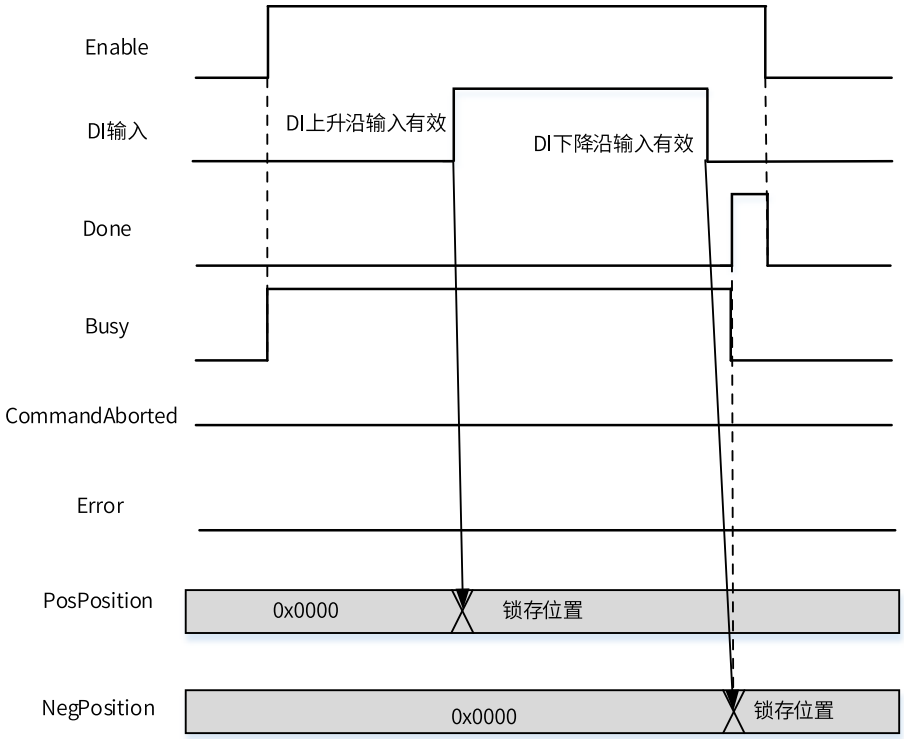
- 探针1上升沿有效，DI端子触发，单次触发，窗口功能有效。



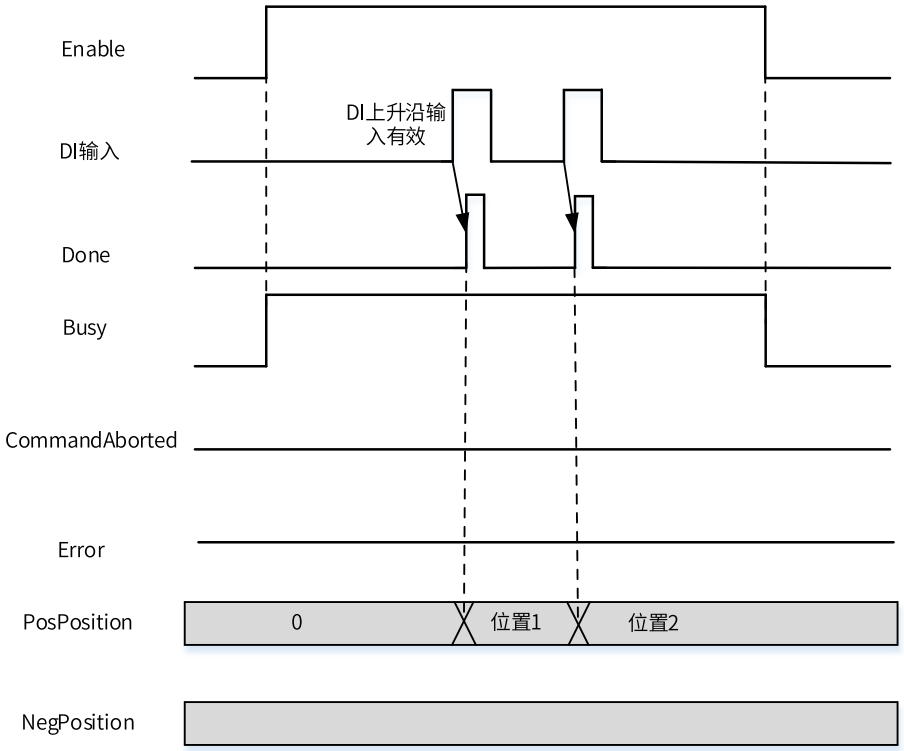
- 探针1下降沿有效，DI端子触发，单次触发，窗口功能无效。



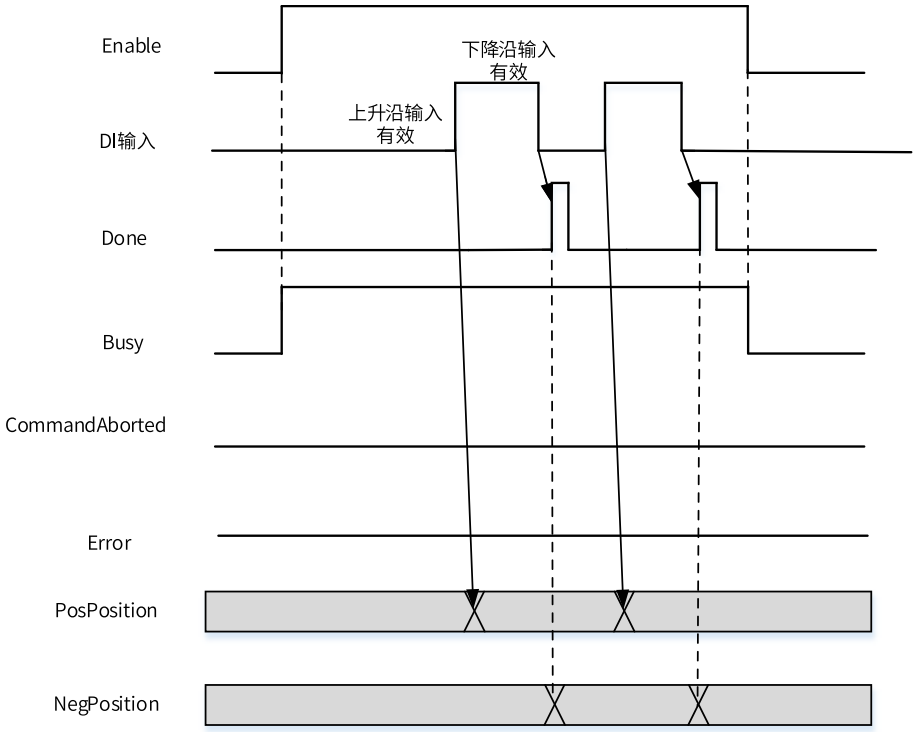
- 探针1上升沿和下降沿有效，DI端子触发，单次触发，窗口功能无效。



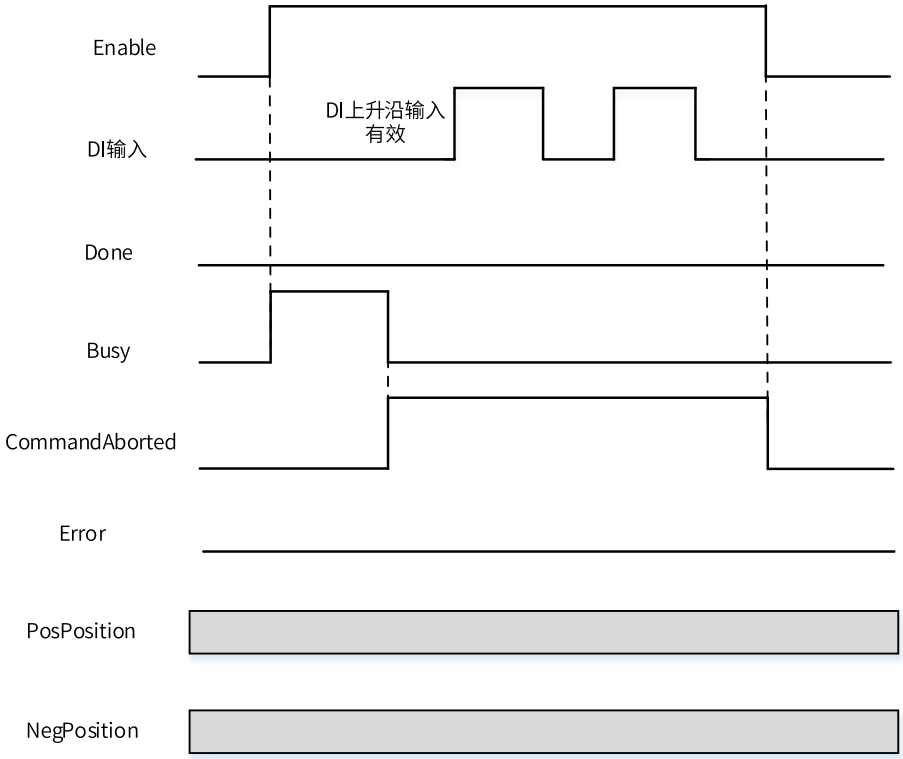
- 探针1上升沿有效，DI端子触发，连续触发，窗口功能无效。



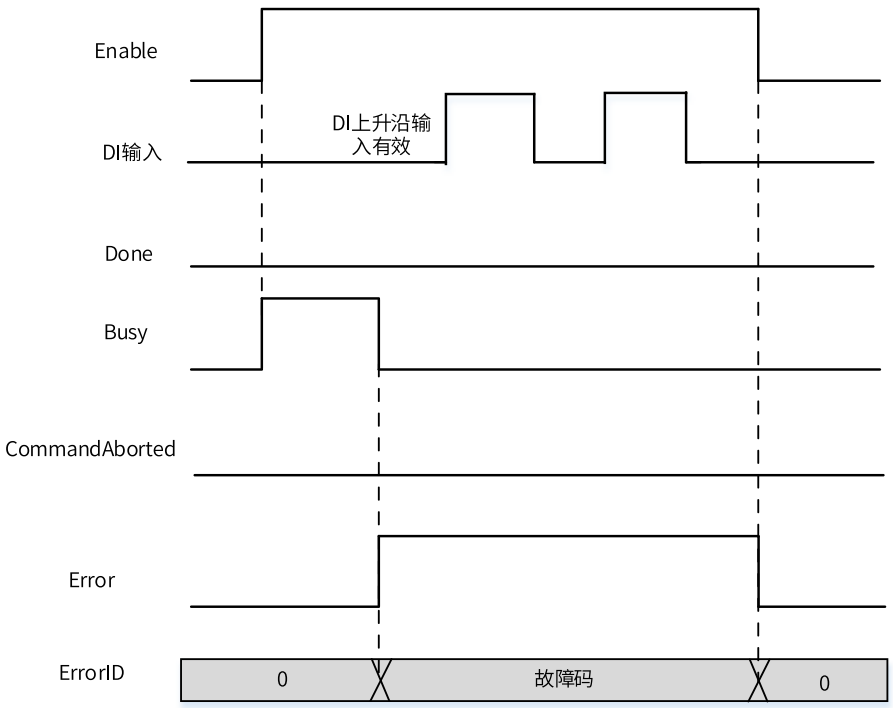
- 探针1上升沿和下降沿有效，DI端子触发，连续触发，当DI端子的上升沿和下降沿都有效之后Done产生一个周期的有效信号，窗口功能无效。



- 探针1被其他跟探针有关的指令打断，窗口功能无效。



- 探针1指令故障



3.14.13 MC_MoveRelative

MC_MoveRelative — 相对定位指令

图形块

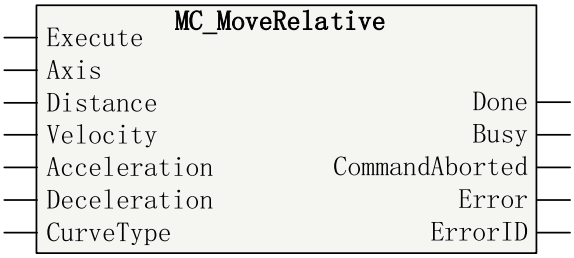


表3–245 指令列表格式

16位指令	—					
32位指令	MC_MoveRelative 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Distance	目标位置	否	-	正数/负数/0	REAL
S3	Velocity	目标速度	否	-	正数，小于最大速度	REAL
S4	Acceleration	加速度	否	-	正数，小于最大加速度	REAL
S5	Deceleration	减速度	是	加速度	正数，小于最大加速度	REAL

S6	CurveType	曲线类型 0：T型速度曲线 1：5段S曲线	是	0	0-1	INT
D1	Done	目标位置到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	CmdAborted	终止执行	是	FALSE	TRUE/FALSE	BOOL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–246 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
S3	-	-	-	✓	✓	✓	-	✓	-
S4	-	-	-	✓	✓	✓	-	✓	-
S5	-	-	-	✓	✓	✓	-	✓	-
S6	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

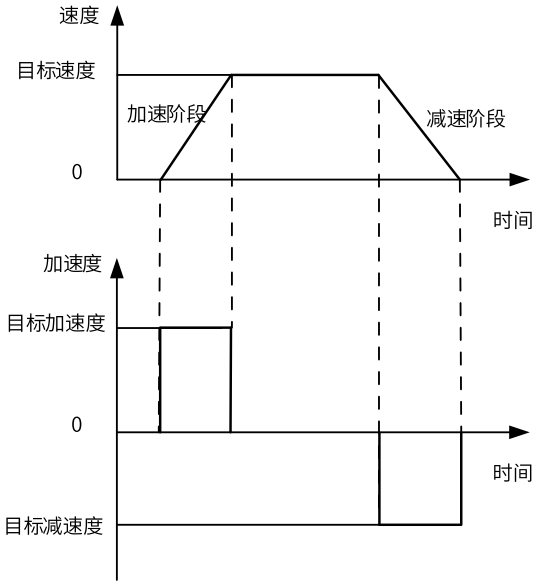
该指令用于控制EtherCAT总线轴或者本地脉冲轴实现相对定位功能，上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在Execute输入的上升沿，指令锁存Distance、Velocity等左侧的输入参数，并触发相对定位功能，并将轴的PLCOpen状态机切换到DiscreteMotion状态。

- Distance用于设定相对定位的距离。不管在线性模式还是环形模式下，如果Distance为正数，轴正向运行Distance指定的距离，如果Distance为负，则轴负向运行|Distance|指定的距离。
- CurveType用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动。



目标位置：在绝对定位指令中指轴最终要达到的位置，单位是Unit(用户单位)。

目标速度：在轴运行过程中能达到的最大速度，单位是Unit/s(用户单位/秒)。

目标加速度：轴在做加速运行时每秒钟速度的变化量，单位是Unit/s^2。

目标减速度：轴在做减速运行时每秒钟速度的变化量，单位是Unit/s^2。

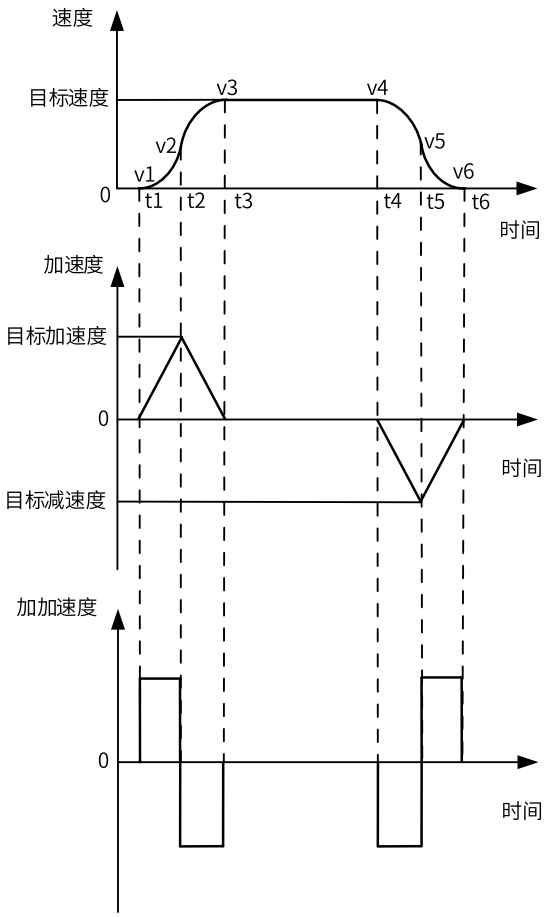
加速阶段，假设轴的初始速度为Vs，目标速度为Vt，目标加速度为Acc，则加速阶段的时间：

“Tacc = (Vt - Vs)/Acc”

减速阶段，假设轴的初始速度为Vs，目标速度为Ve，目标减速度为Dec，则减速阶段的时间：

“Tdec = (Vs - Ve)/Dec”

- CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。



在5段S型速度曲线中，根据加速度的状态分为加加速、减加速、匀速、加减速、减减速5个阶段，一定不存在匀加速和匀减速阶段，在加加速、加减速等变加速度阶段中实际的Jerk是H5u内部计算得到的，用户不可以设置。

目标位置：在绝对定位指令中指轴最终要达到的位置，单位是Unit(用户单位)。

目标速度：在轴运行过程中能达到的最大速度，单位是Unit/s(用户单位/秒)。

目标加速度：轴在做变加速运行时每秒钟速度的最大变化量，单位是Unit/s^2，速度曲线中速度由加加速阶段变成减加速这一时刻（t2）的加速度必然是目标加速度。

目标减速度：轴在做变减速运行时每秒钟速度的最大变化量，单位是Unit/s^2，速度曲线中速度由减加速阶段变成减减速这一时刻（t5）的减速度必然是目标减速度。

加速阶段，假设轴的初始速度为V1，目标速度为V3，目标加速度为Acc，则加速阶段的时间：

“Tacc = 2*(V3 – V1)/Acc”

减速阶段，假设轴的初始速度为V4，目标速度为V6，目标减速度为Dec，则减速阶段的时间：

“Tdec = 2* (V4 – V6) / Dec”

报错

当轴号不存在或轴类型不匹配时指令报错9101。

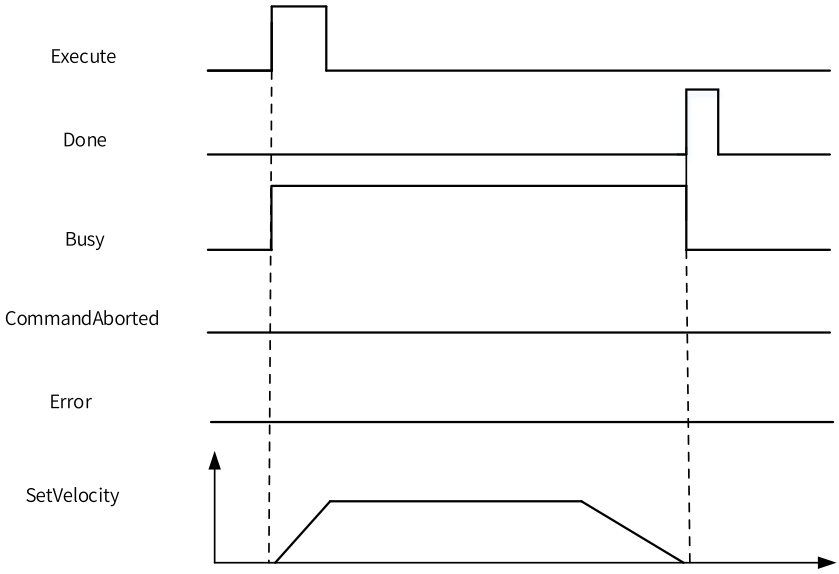
当轴初始化失败时指令报错9102。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

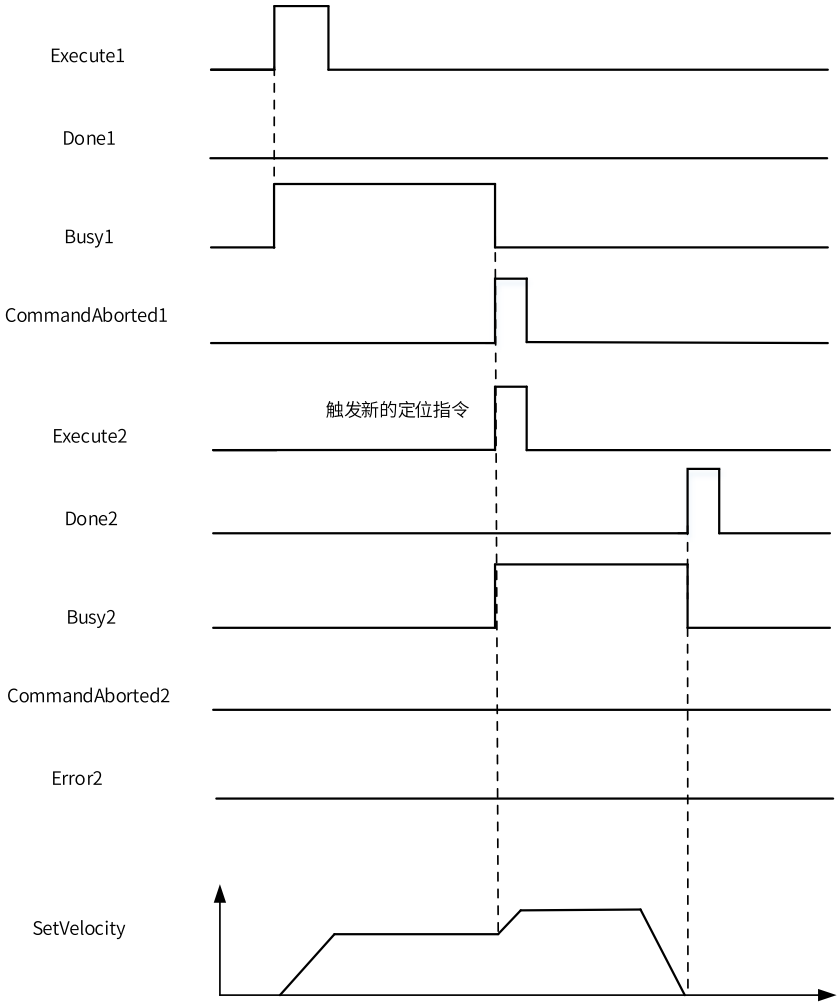
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute= TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

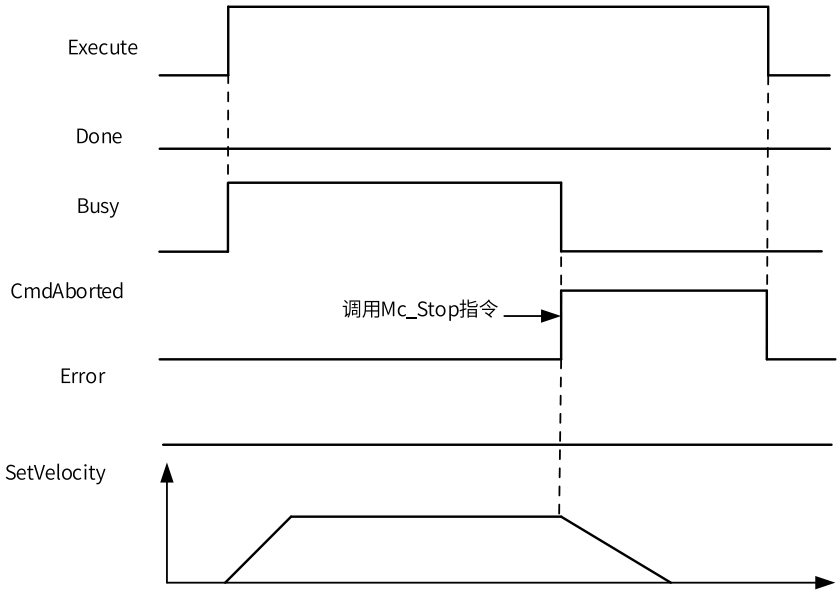
- 当轴在StandStill状态下调用本指令做T型曲线下的相对定位运动。



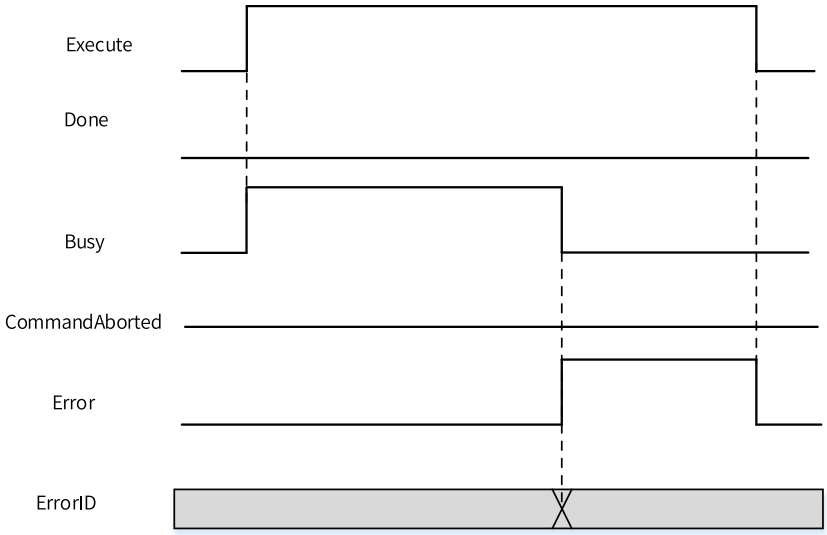
- 相对定位期间触发另一条相对定位指令。



- 当轴在相对定位运动期间被Mc_Stop指令打断。



- 当轴在运动期间驱动器发生了故障。



3.14.14 MC_MoveVelocity

MC_MoveVelocity — 速度指令

图形块

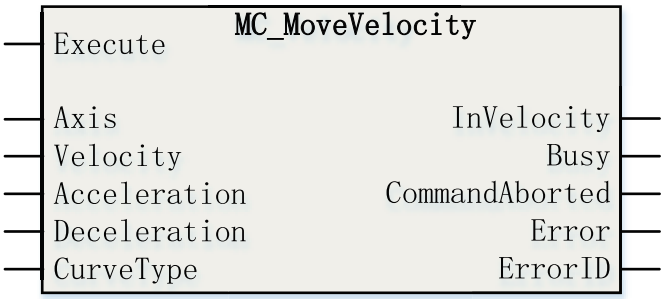


表3-247 指令列表格式

16位指令	—					
32位指令	MC_MoveVelocity 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Velocity	目标速度	否	-	正数/负数/0，绝对值小于最大速度	REAL
S3	Acceleration	加速度	否	-	正数，小于最大加速度	REAL
S4	Deceleration	减速度	是	加速度	正数，小于最大加速度	REAL
S5	CurveType	速度曲线类型 0：T型速度曲线 1：5段S曲线	是	0	0-1	INT
D1	InVelocity	速度到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL

D3	CmdAborted	终止执行	是	FALSE	TRUE/FALSE	BOOL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–248 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
S3	-	-	-	✓	✓	✓	-	✓	-
S4	-	-	-	✓	✓	✓	-	✓	-
S5	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于控制EtherCAT总线轴或者本地脉冲轴实现绝对定位功能，上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在Execute输入的上升沿，指令锁存Velocity、Acceleration等左侧的输入参数，触发轴按照Velocity设定的速度一直运行，并将轴的PLCOpen状态机切换到ContinuousMotion状态。

  CurveType用于设定速度曲线的类型。

  - CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动。
  - CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。

调用本指令后如果想停止轴的运动可以调用MC_Stop、Mc_Halt和MC_ImmediateStop（驱动器支持）指令。

打断

在PLCOpen中该指令有效时轴将处于Continuous Motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Enable=TRUE且Busy信号有效时MC_Power指令无效导致轴去使能时CommandAboarted有效。

在Enable=TRUE且Busy信号有效时碰到限位需要执行减速停机时CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

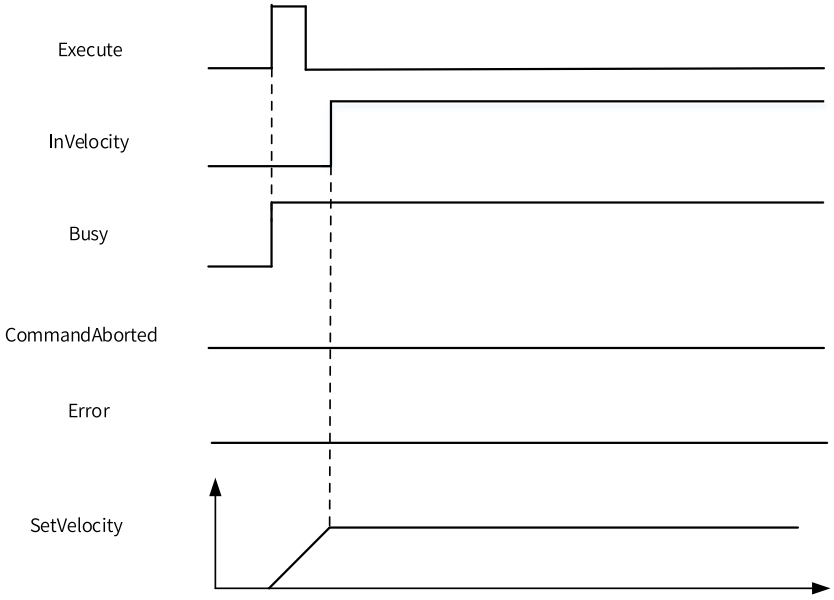
当轴初始化失败时指令报错9102。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

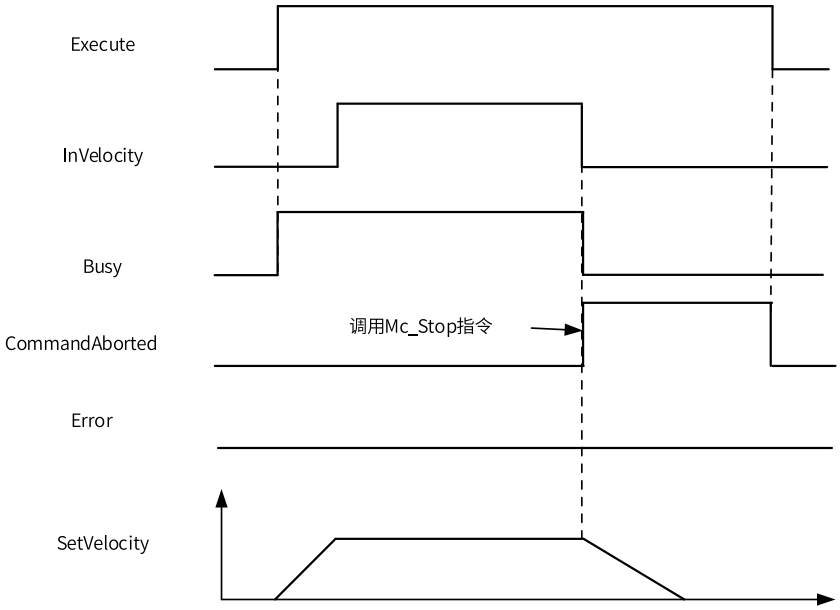
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute= TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

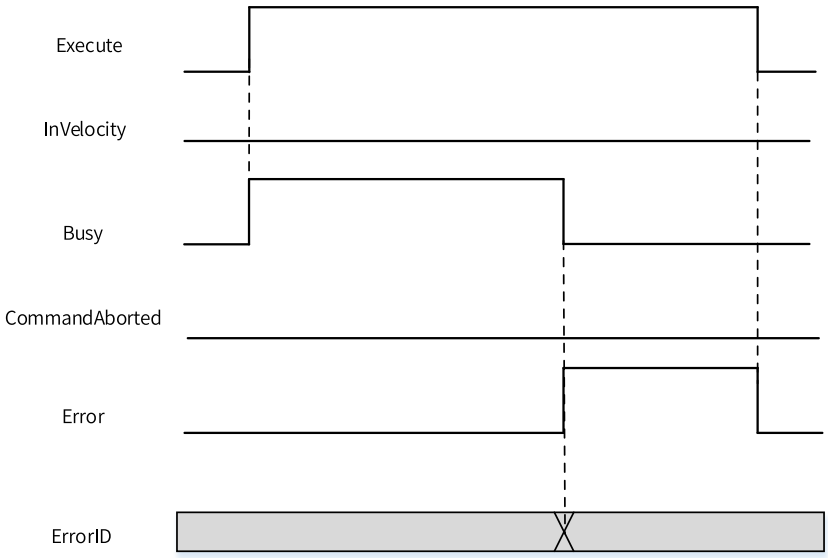
- 当轴在StandStill状态下调用本指令做T型曲线下的连续运动。



- 当轴在运动期间被Mc_Stop指令打断。



- 当轴在加速期间驱动器发生了故障。



3.14.15 MC_MoveAbsolute

MC_MoveAbsolute — 绝对定位指令

图形块

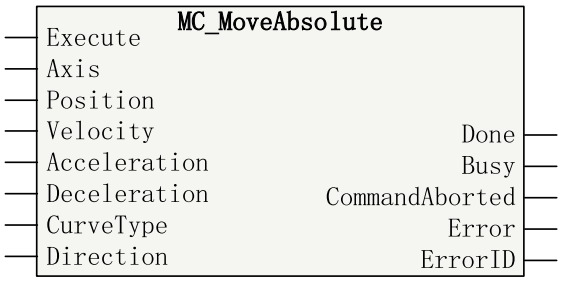


表3–249 指令列表格式

16位指令	—					
32位指令	MC_MoveAbsolute 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Position	目标位置	否	-	正数/负数/0	REAL
S3	Velocity	目标速度	否	-	正数，小于最大速度	REAL
S4	Acceleration	加速度	否	-	正数，小于最大加速度	REAL
S5	Deceleration	减速度	是	加速度	正数，小于最大加速度	REAL
S6	CurveType	速度曲线类型 0：T型速度曲线 1：5段S曲线	是	0	0-1	INT
S7	Direction	模式，仅环形模式下 0：正向(速度大于0) 1：负向（速度小于0） 2：最短距离 3：当前方向	是	0	0-3	INT
D1	Done	目标位置到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	CmdAborted	终止执行	是	FALSE	TRUE/FALSE	BOOL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–250 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
S3	-	-	-	✓	✓	✓	-	✓	-
S4	-	-	-	✓	✓	✓	-	✓	-
S5	-	-	-	✓	✓	✓	-	✓	-
S6	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

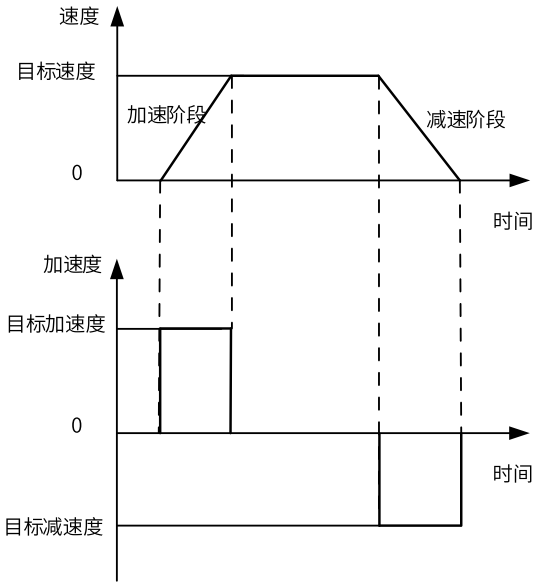
功能和指令说明

该指令用于控制EtherCAT总线轴或者本地脉冲轴实现绝对定位功能，上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

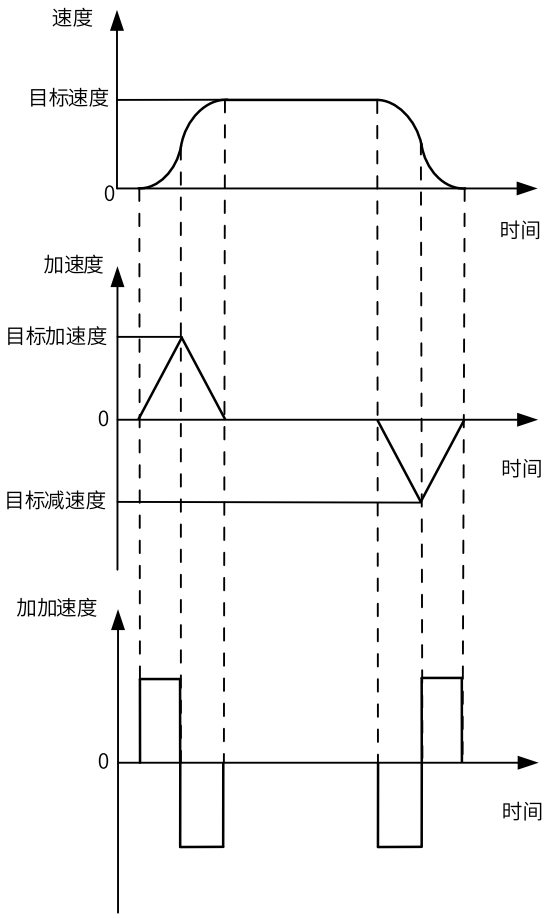
  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在Execute输入的上升沿，指令锁存Position、Velocity等左侧的输入参数，触发绝对定位功能，并将轴的PLCOpen状态机切换到DiscreteMotion状态。
  - 线性模式下，Position用于设定对对定位的目标位置。如果当前位置小于目标位置，轴将正向运动，最后到达Position设定的位置，如果当前位置大于目标位置，轴将反向运动，最后到达Position设定的位置。
  - CurveType用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动。



T型曲线的详细介绍请查看相对定位指令。

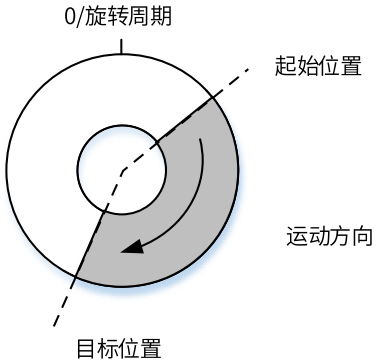
- CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。



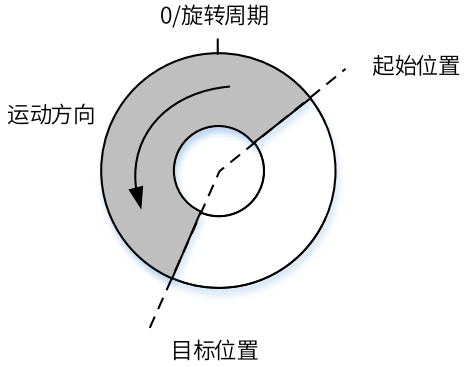
S型曲线的详细介绍请查看相对定位指令。

在环形模式下，指令首先将Position对旋转周期求模，得到一个旋转周期内的绝对位置Position_p,轴实际的运动方向由以下4中情况决定：

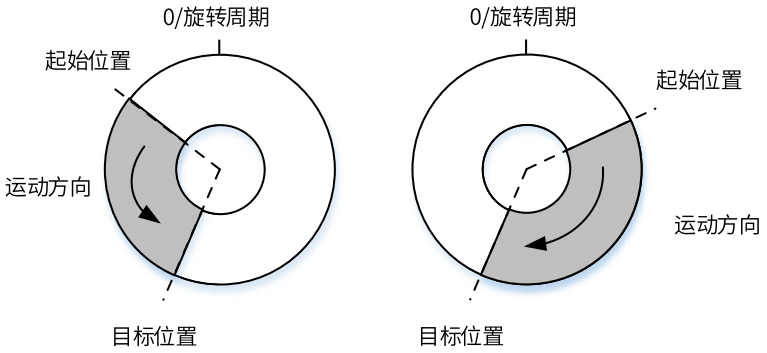
1. Direction = 0表示正向运动（目标速度大于0）。如果当前速度大于0，则继续朝当前方向运行，然后到Position_p设定的位置停止；如果当前速度小于0，则先做减速运动，减速到0后开始速度反向运动，直到Position_p设定的位置停止；如果Position_p和当前位置相等***1**，则不运动。



2. Direction = 1表示反向运动（目标速度小于0）。果当前速度小于0，则继续朝当前方向运行，然后到Position_p设定的位置停止；如果当前速度大于0，则先做减速运动，减速到0后开始速度反向运动，直到Position_p设定的位置停止；如果Position_p和当前位置相等***1**，则不运动。



3. Direction = 2表示最短距离，在Execute的上升沿，轴将记录当前位置，并假设当前速度是0，推算出从0速启动轴正向运动到Position_p的位移Distance，如果Distance小于等于0.5*旋转周期，则正向运动，如果大于0.5*旋转周期则反向运动; 如果Position_p和当前位置相等***1**，则不运动。



4. Direction = 3表示沿当前方向，在Execute的上升沿，轴将按照Execute上升沿到达之前最近的一次运动方向运动，直到Position_p设定的位置停止，如果是第一次上电，则正向运动（目标速度大于0）。如果Position_p和当前位置相等***1**，则不运动。

说明

***1**：在环形模式下，如果目标位置大于环形周期，则将目标位置对环形周期求模，后得到新的目标位置，如果新的目标位置和轴的setposition差值的绝对值小于0.001，则认为两者相等。

打断

在PLCOpen中该指令有效时轴将处于Discrete motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Enable=TRUE且Busy信号有效时MC_Power指令无效导致轴去使能时CommandAboarted有效。

在Enable=TRUE且Busy信号有效时碰到限位需要执行减速停机时CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

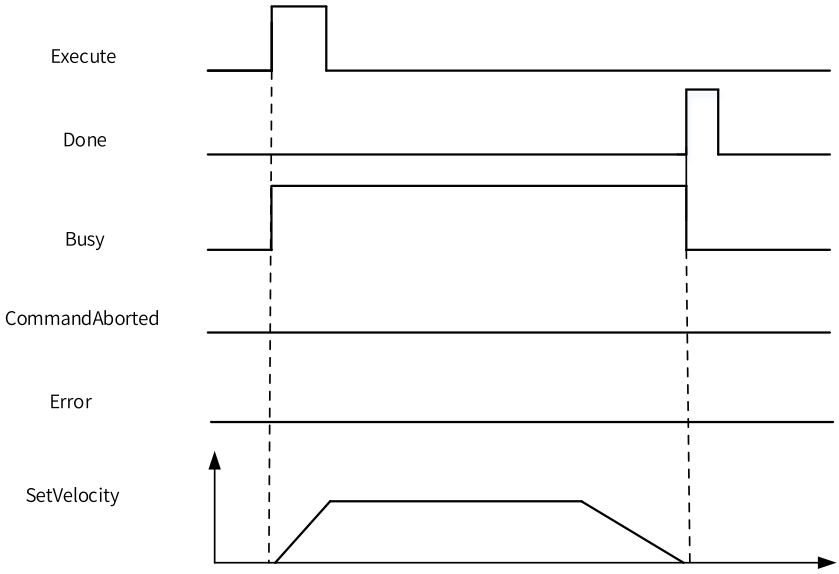
当轴初始化失败时指令报错9102。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

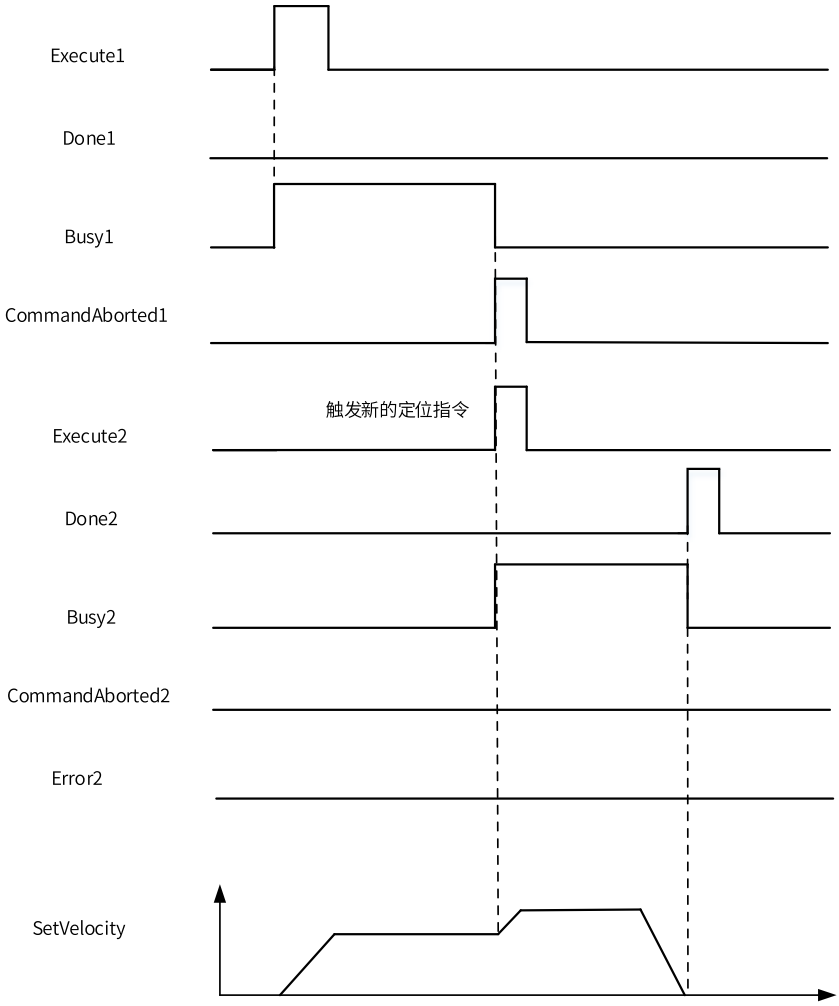
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute= TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

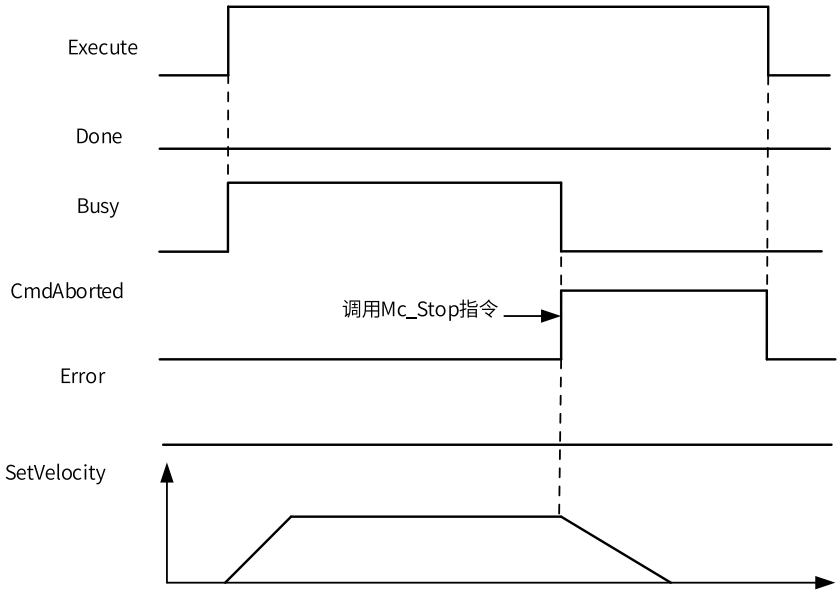
- 当轴在StandStill状态下调用本指令做T型曲线下的绝对定位运动。



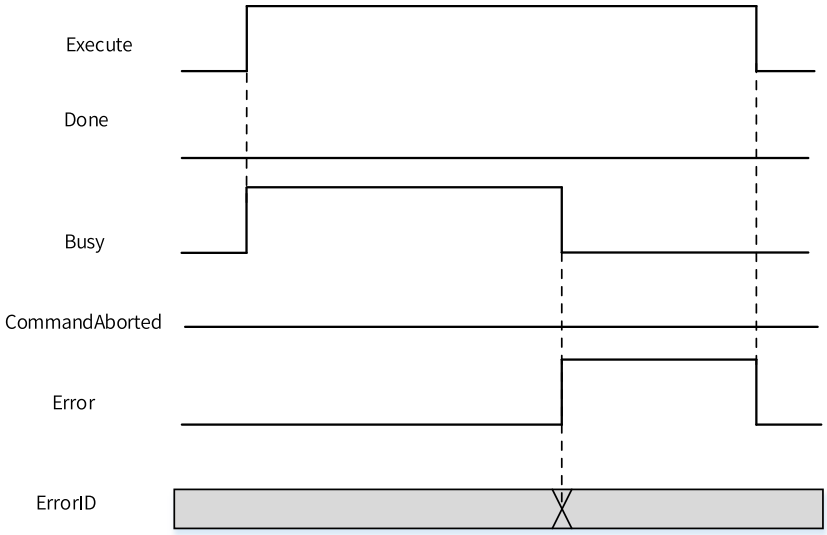
- 绝对定位期间触发另一条绝对定位指令。



- 当轴在绝对定位运动期间被Mc_Stop指令打断。



- 当轴在运动期间驱动器发生了故障。



3.14.16 MC_Jog

MC_Jog — 点动运动指令

图形块

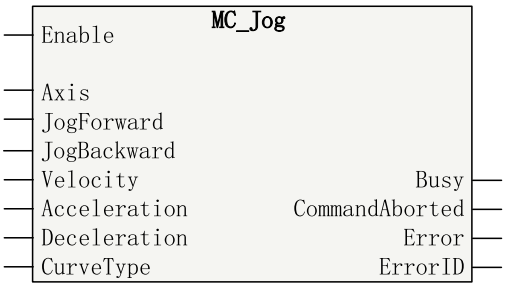


表3-251 指令列表格式

16位指令	—					
32位指令	MC_Jog 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	JogForward	正向运动，上升沿触发电平有效	否	-	-	BOOL
S3	JogBackward	反向运动，上升沿触发电平有效	否	-	-	BOOL
S4	Velocity	目标速度	否	-	正数，小于最大速度，小于点动最大速度	REAL
S5	Acceleration	加速度	否	-	正数，小于最大加速度	REAL
S6	Deceleration	减速度	是	加速度	正数，小于最大加速度	REAL

S7	CurveType	曲线类型 0：T型速度曲线 1：5段S曲线	是	0	0-1	INT
D1	Busy	忙标志	是	FALSE	-	BOOL
D2	CommandAborted	终止执行	是	FALSE	-	BOOL
D3	Error	指令故障标志	是	FALSE	-	BOOL
D4	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–252 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	✓	✓	✓	-	-	-	-	-	-
S3	✓	✓	✓	-	-	-	-	-	-
S4	-	-	-	✓	✓	-	-	✓	-
S5	-	-	-	✓	✓	-	-	✓	-
S6	-	-	-	✓	✓	-	-	✓	-
S7	-	-	-	✓	✓	-	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现EtherCAT总线轴或本地脉冲轴的点动功能，高电平有效。

- 轴号的设置
  - 在Enable输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Enable=TRUE期间修改Axis，原来控制的轴进入ErrorStop状态。
  - 如果Axis设置的是轴号，在Enable=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在指令的上升沿，功能块锁存Velocity、Acceleration、Deceleration、CurveType等输入参数，并将轴的状态机切换到Continuous Motion模式，进入点动状态。

- 在Enable=TRUE时，如果调用MC_Stop，MC_MoveRelative之类的指令，将打断jog指令，jog执行的CommandAborted输出有效。
- 当JogForward有效时，轴按照Velocity设定的速度正向运动，当JogBackward有效时，轴按照Velocity设定的速度做反向运动。当JogForward和JogBackward同时有效时，轴停止运动，指令报故障，轴停止运动，但是不会进入ErrorStop状态。
- 在Enable=TRUE时，轴往一个方向运行碰到限位，指令报故障，停止轴的运动但不会进入ErrorStop状态。重新触发Jog指令后可以将轴往相反方向运动。
- CurveType用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动。CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。

打断

在PLCOpen中该指令有效时轴将处于Continuous Motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Enable=TRUE且Busy信号有效时MC_Power指令无效导致轴去使能时CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

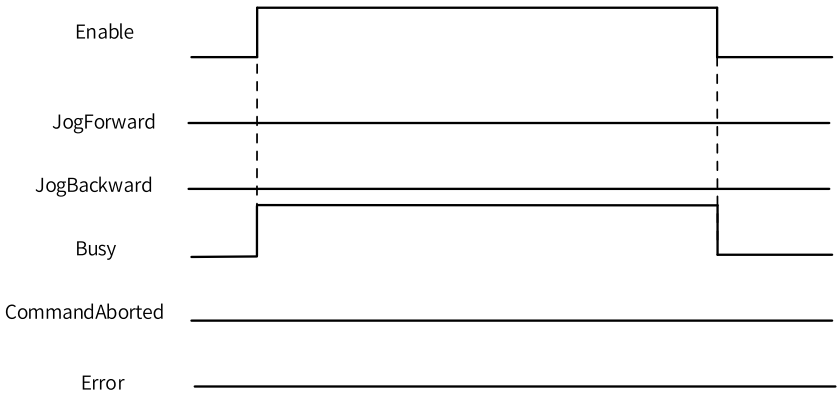
当轴初始化失败时指令报错9102。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

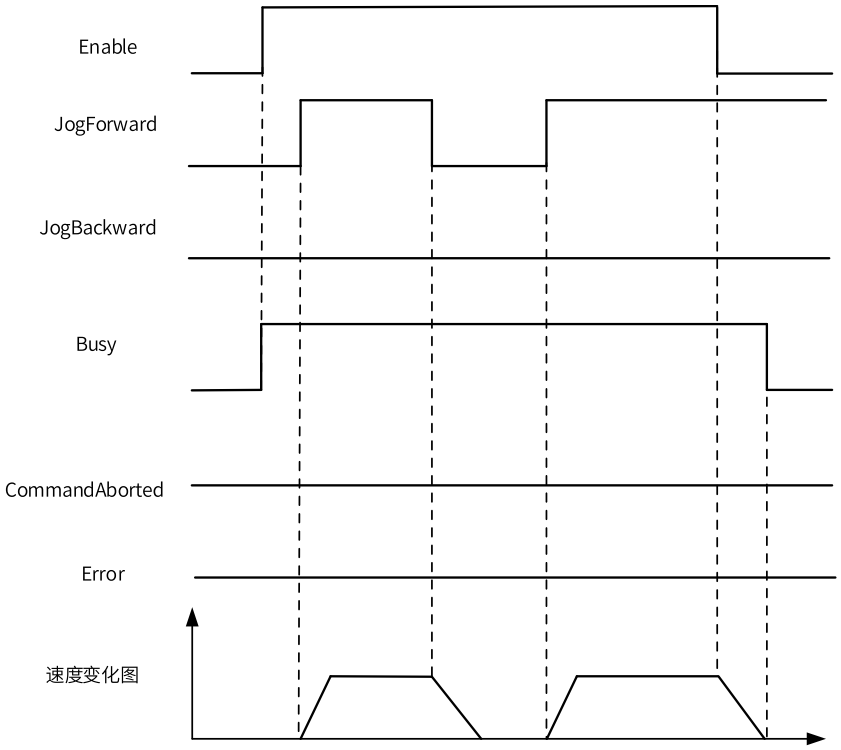
- 在Enable输入的上升沿如果轴处于在线调试状态报错9116。
- 在Enable输入的上升如果轴正在执行故障减速报错9106。
- 在Enable输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Enable=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Enable= TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

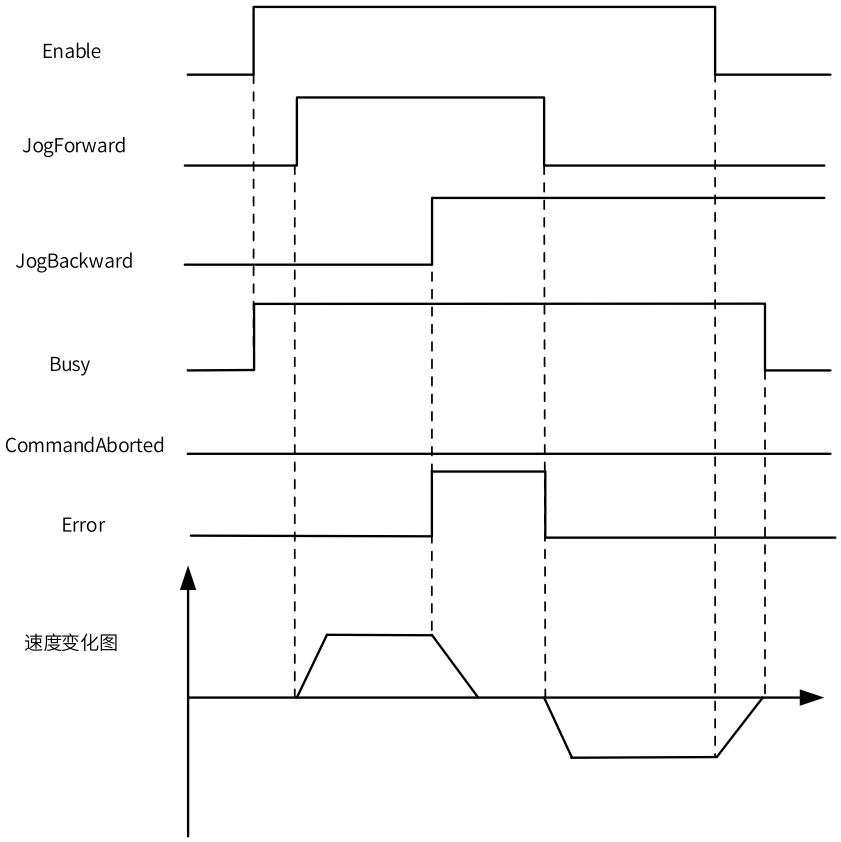
- 仅Enable输入有效时，轴进入ContinousMotion状态，Busy输出有效。



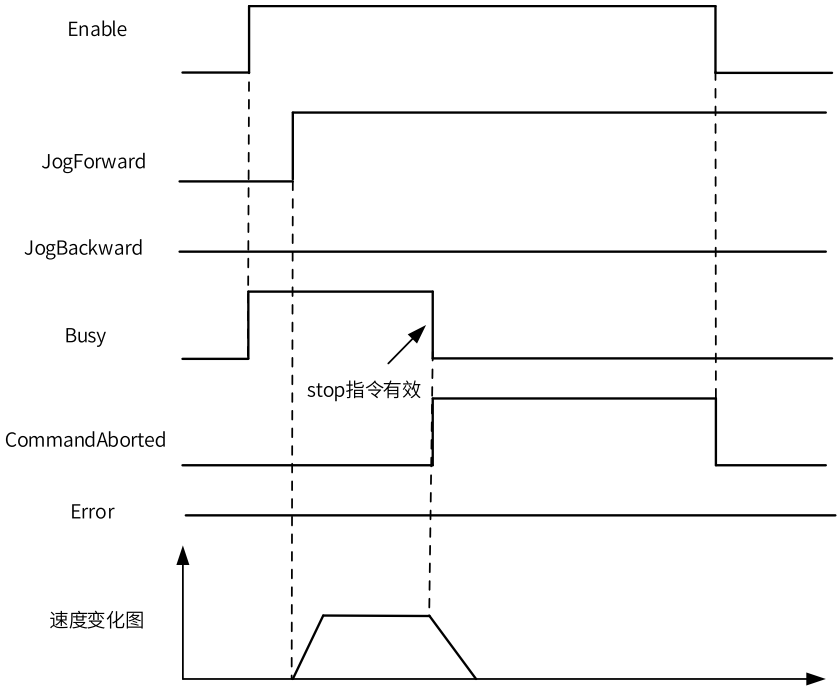
- 当Enable、JogForward输入有效时。



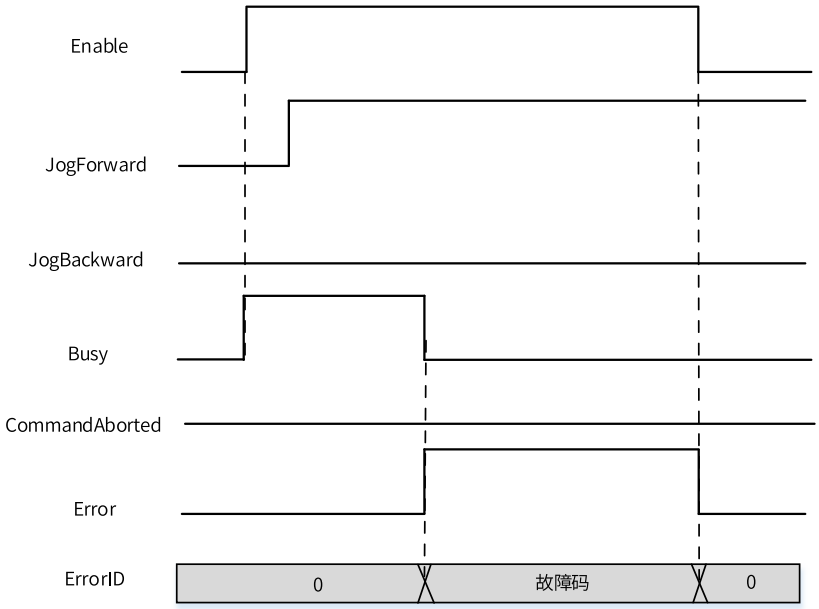
- 当Enable、JogForward输入有效后，将JogBackward写入TRUE时。



- 当指令执行期间被Stop指令打断时。



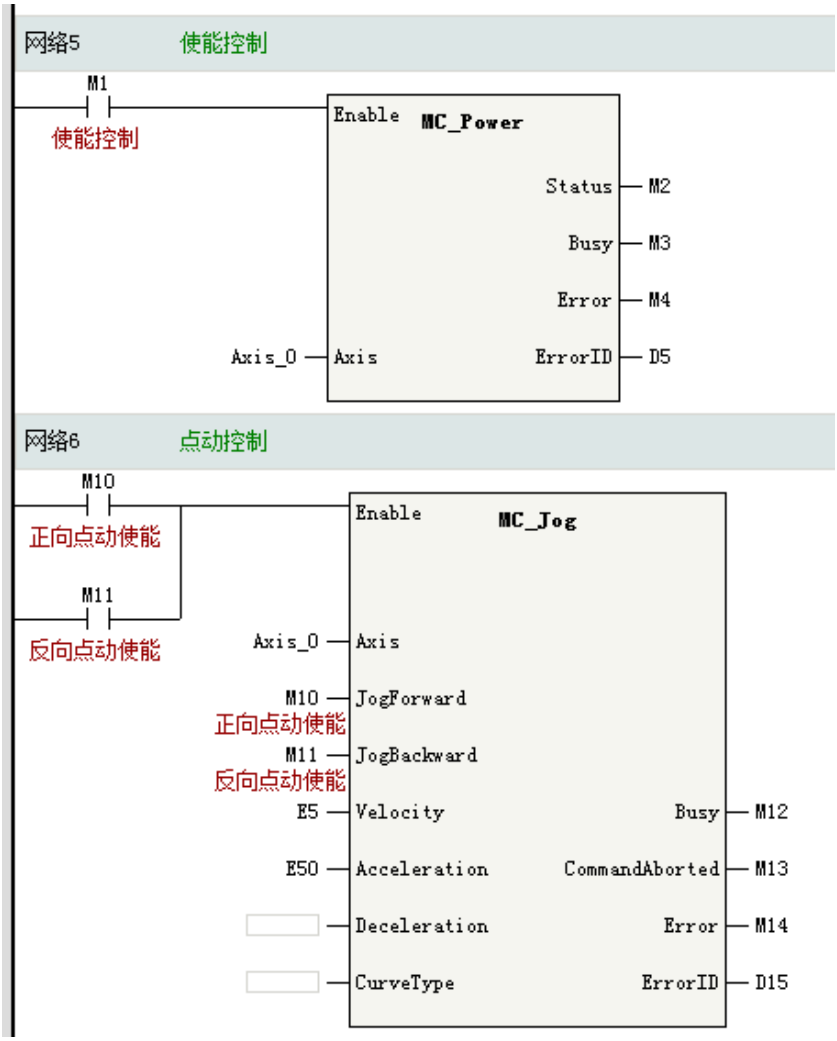
- 当轴报错时。



例程

通过以下例程实现如下功能。

1. 将M1设置为ON后使能轴Axis_0。
2. 将M10设置为ON后轴Axis_0以5Unit/s的速度正向运行。
3. 将M10设置为OFF后轴Axis_0停止运行。
4. 将M11设置为ON后轴Axis_0以5Unit/s的速度反向运行
5. 将M11设置为 OFF后轴Axis_0停止运行。



3.14.17 MC_TorqueControl

MC_TorqueControl — 力矩控制指令

图形块

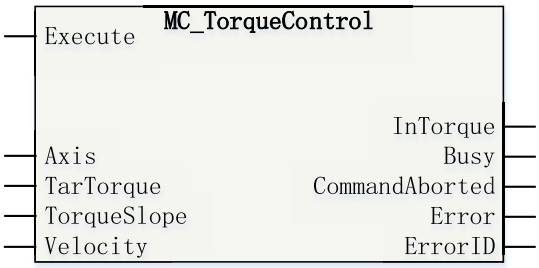


表3-253 指令列表格式

16位指令	—					
32位指令	MC_TorqueControl 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型

S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	TarTorque	目标扭矩（单位是1%）	否	-	正数，负数，0	REAL
S3	TorqueSlope	力矩斜坡（单位是1%）	否	-	正数	REAL
S4	Velocity	限制速度	否	-	正数,0	REAL
D1	InTorque	扭矩到达 当设置力矩达到目标力矩且反馈力矩和目标力矩的差值的绝对值在5%以内时输出有效	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	CommandAborted	终止执行	是	FALSE	TRUE/FALSE	BOOL
D4	Error	指令故障标志	是	FALSE	TRUE/FALSE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–254 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	✓	-	-
S3	-	-	-	✓	✓	✓	✓	-	-
S4	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现力矩控制功能，仅用于总线线EtherCAT总线轴。指令上升沿有效，不支持虚轴模式。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述
  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  力矩指令需要驱动器映射0x6040,0x6041,0x6060,0x6061,0x6071,0x6077几个PDO才可以使用，否则报故障。

  本指令采用驱动器同步力矩模式实现力矩控制功能。

  在指令的上升沿， 功能块锁存TarTorque、TorqueSlope和Velocity输入参数，轴处于Continuous Motion状态并做力矩运动。

  - TarTorque：目标力矩，单位是1%，程序中小数点后一位小数有效，后面的直接舍弃，驱动器实际的力矩受配置参数中设置的最大正负力矩限制。
  - TorqueSlope：力矩斜坡，单位是1%，程序中小数点后一位小数有效，后面的直接舍弃。
- 转矩模式下的速度控制
  对于汇川伺服驱动器，如果轴参数中映射了0x607f，则本指令通过0x607f限制伺服电机的最大转速，如果没有映射0x607f，则速度限制无效。

  在Execute的上升沿，指令将Velocity设定的限制速度转换成脉冲单位通过PDO写入0x607f中。

  当力矩指令被其他指令打断时，轴的最大速度将通过配置界面中的“最大速度”一项进行限制。

  对于第三方驱动器，满足以下几种情况才可以将Velocity用作速度限制：

  - 可以通过0x607F限制伺服电机的最大速度。
  - 0x607F可以配置到PDO中。
  - 0x607F的单位为脉冲单位，非转速单位。
- 转矩模式下的停止控制
  在力矩模式下调用MC_Stop指令执行停机操作，在接收到Stop指令之后，驱动器转换到同步位置模式，按照Stop指令设定的减速度执行减速。

打断

在PLCOpen中该指令有效时轴将处于Continuous Motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Execute=TRUE且Done信号无效时即将碰到限位信号执行需要执行减速时CommandAboarted有效。

在Execute=TRUE且Done信号无效时轴去使能CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

当轴使能虚轴模式后调用本指令轴报错9113。

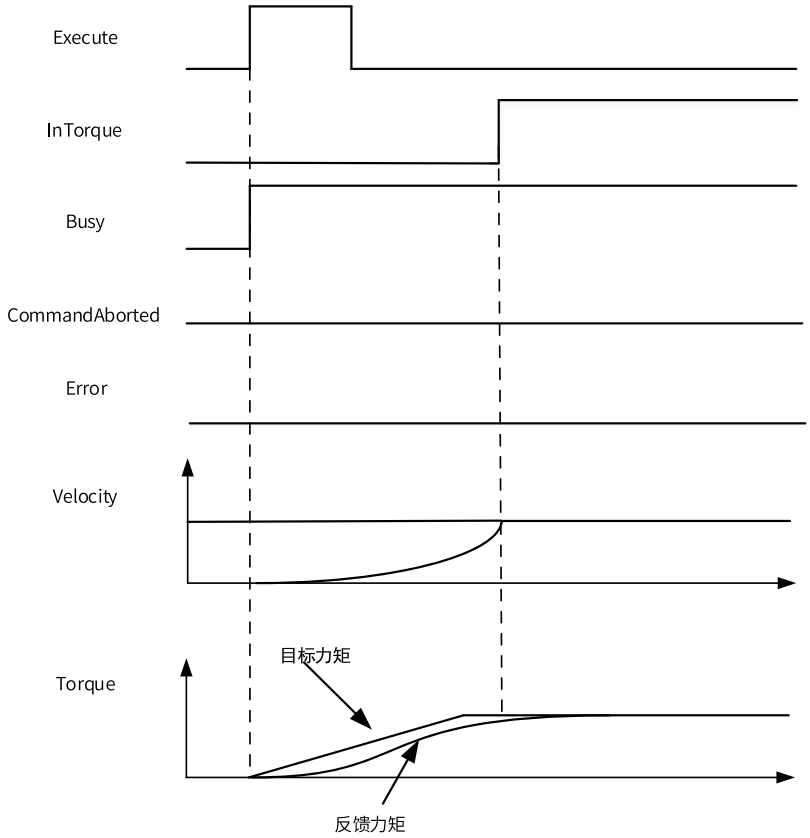
当轴缺少相应的PDO配置时报缺少PDO配置故障。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

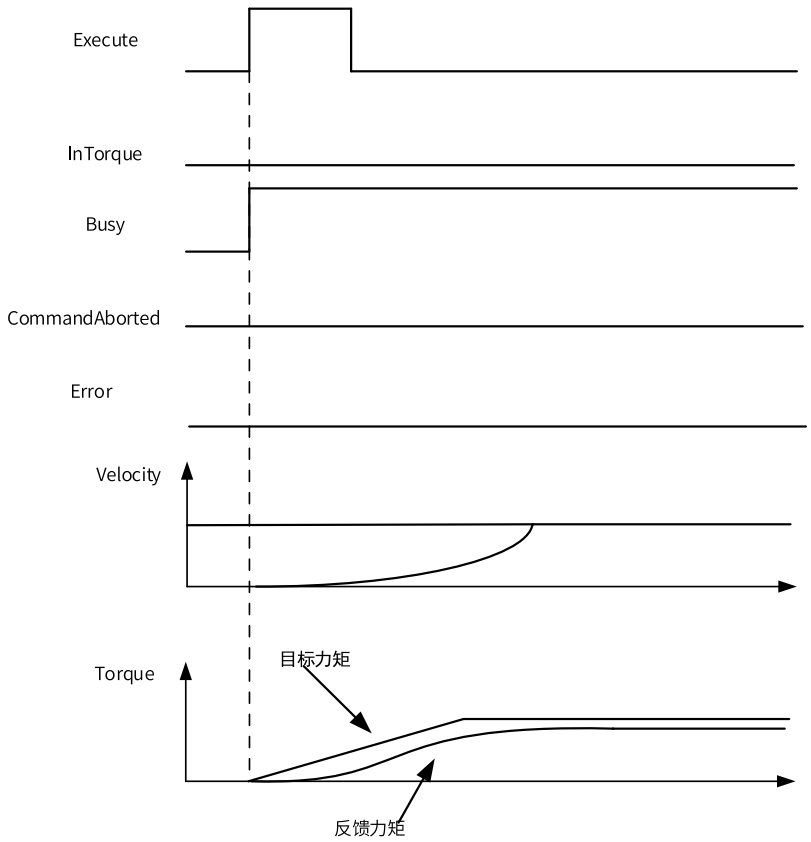
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

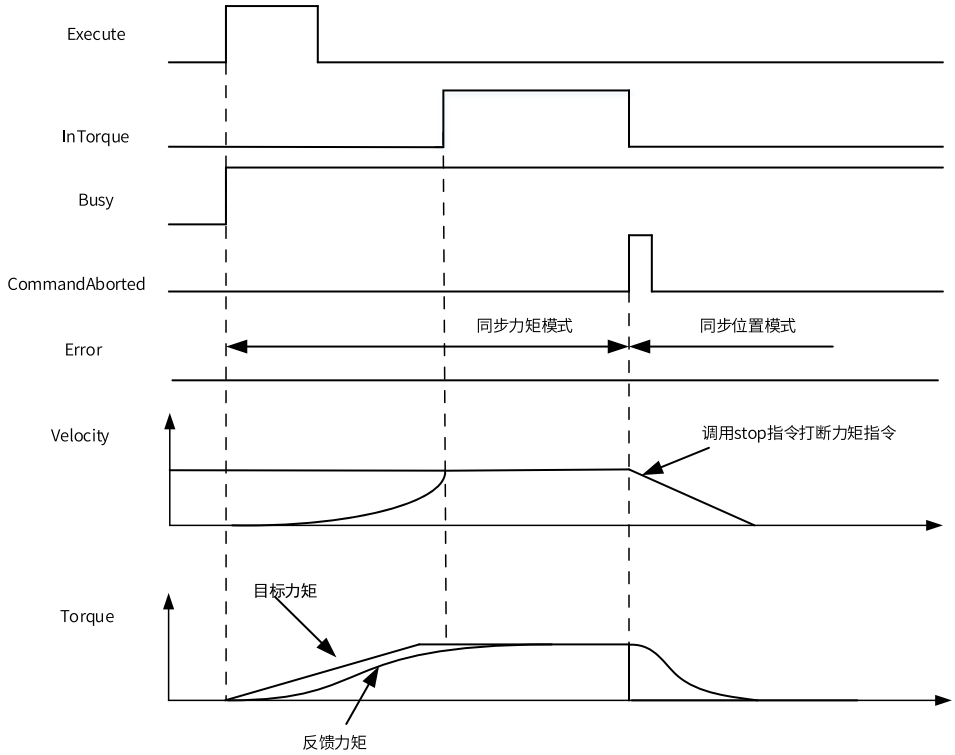
- 设定目标力矩后触发指令，在实际输出力矩可以达到目标力矩的情况下。



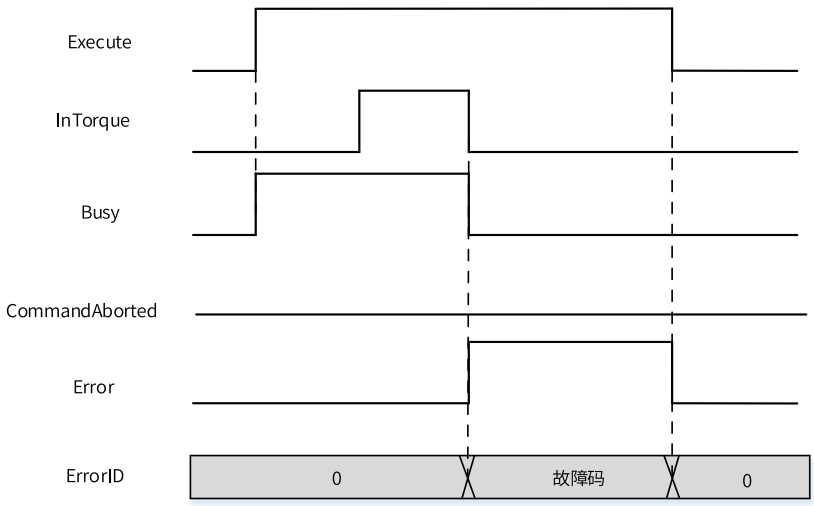
- 设置目标力矩后触发指令，最后实际输出力矩不能达到目标力矩的情况下。



- 在力矩运行过程中，Mc_Stop指令打断指令的运行。



- 在力矩运行过程中，驱动器报错。



3.14.18 MC_Home

MC_Home — 原点回归指令

图形块

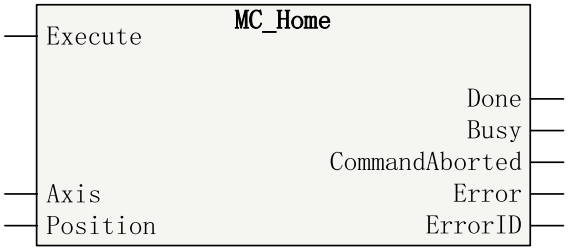


表3–255 指令列表格式

16位指令	—					
32位指令	MC_Home 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Position	原点偏移	是	0	正数/负数/0	REAL
D1	Done	原点回归完成	是	FALSE	FALSE/TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE/TRUE	BOOL
D3	CommandAborted	终止执行	是	FALSE	FALSE/TRUE	BOOL
D4	Error	指令故障标志	是	FALSE	FALSE/TRUE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–256 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	✓[1]	✓	✓	-	-	✓	-	-	-
D5	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现EtherCAT总线轴和本地脉冲轴的原点回归功能，指令上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述
  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在指令的上升沿， 功能块锁存Position输入参数，轴处于Homing状态并做原点回归运动。

  Position用于设定原点偏移。

  在虚轴模式下调用本指令，将按照402协议中35号绝对模式回零。
- 多重调用
  回原指令不允许重复调用，当调用一条MC_Home指令使轴处于Homing状态后在调用其他的MC_Home指令，后调用的指令报错。

打断

在PLCOpen中该指令有效时轴将处于Homing状态，可以让轴处于Stopping状态的MC_Stop指令和MC_ImmediateStop指令可以打断本指令，指令被打断时CommandAboarted有效。

在Execute=TRUE且Done信号无效时MC_Power指令无效，这会导致轴去使能时CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

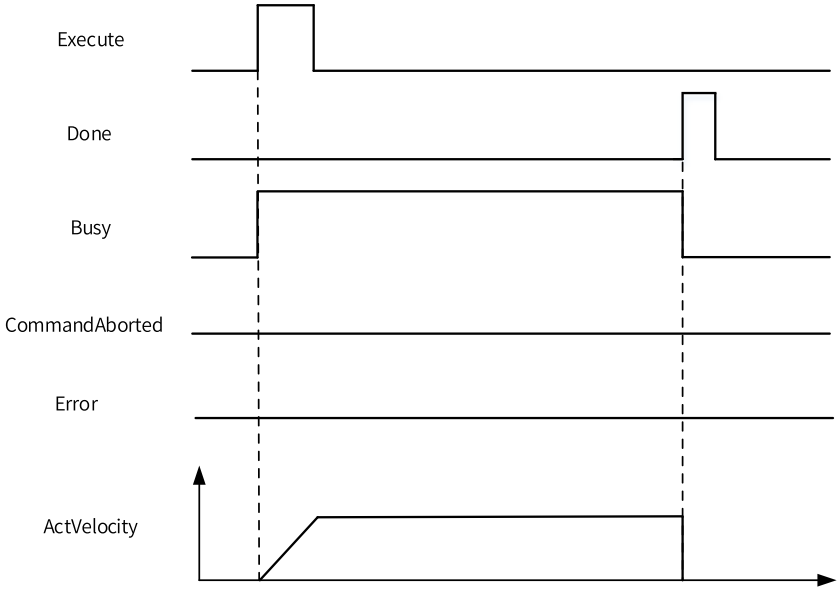
当轴初始化失败时指令报错9102。

只有当轴处于StandStill状态时调用本指令有效，其他状态报错9108。

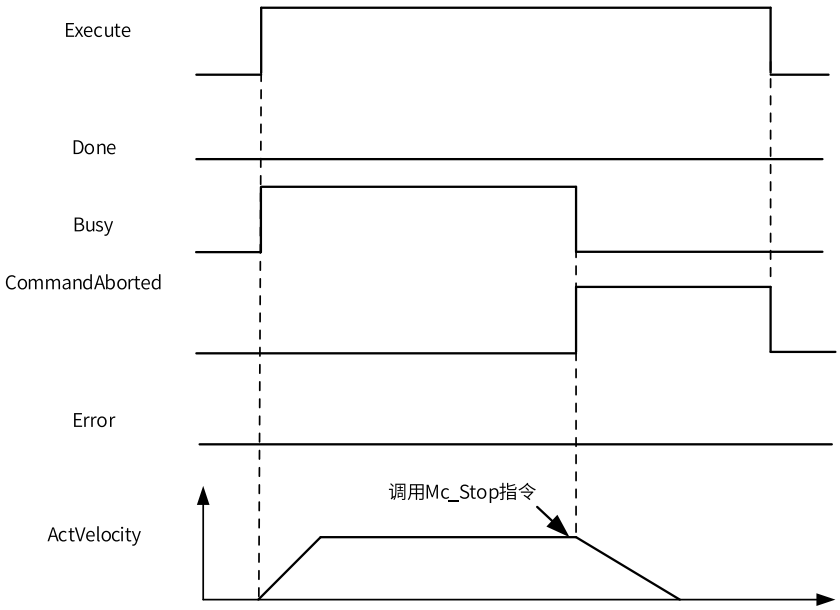
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

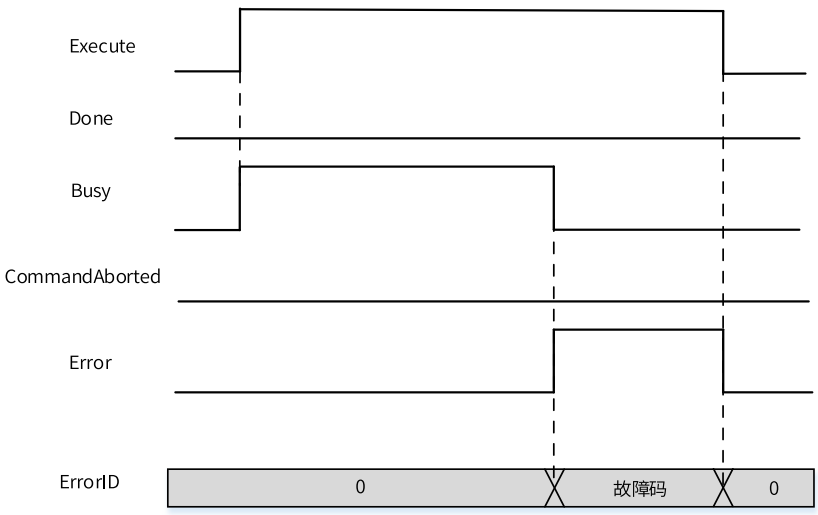
- 启动原点回归，驱动器正常执行原点回归动作。



- 原点回归过程中调用MC_Stop指令打断原点回归动作。



- 原点回归过程中驱动器发生故障。



3.14.19 MC_Stop

MC_Stop — 停止指令

图形块

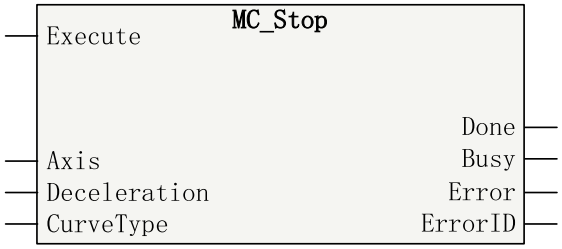


表3–257 指令列表格式

16位指令	—					
32位指令	MC_Stop 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Deceleration	减速度	否	-	正数，小于最大加速度	REAL
S3	CurveType	曲线类型 0：T型速度曲线 1：5段S加减速度	是	0	0-1	INT
D1	Done	停机完成	是	FALSE	FALSE/TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE/TRUE	BOOL
D3	Error	指令故障标志	是	FALSE	FALSE/TRUE	BOOL
D4	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3-258 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	-	-	✓	-
S3	-	-	-	✓	✓	-	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于EtherCAT总线轴或本地脉冲轴实现停止功能，指令上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在Execute输入的上升沿， 功能块锁存Deceleration 、CurveType等输入参数，轴处于Stopping状态并做减速运动。

  减速完成后Done信号有效且在Execute=TRUE期间一直保持在Stopping状态。

  Execute=FALSE且Done=TRUE时轴由Stopping状态切换到Standstill状态。

  调用该指令时轴所处的运行状态不同，停止方式也不同。

  1. 当轴正在执行定位指令或者连续运行时调用该指令，参数CurveType用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Deceleration设定的值做减速运动；CurveType = 1表示5段S曲线，此时Deceleration表示轴在减速过程中达到的最大减速度。
  2. 当轴处于原点回归状态时本指令触发驱动器的控制字的Halt标志位，驱动器按照预设的参数执行减速，此时CurveType、Deceleration无效。

重启指令

同一条stop指令重复执行，在减速期间重新触发同一条stop指令则按照最后一次触发时的减速度减速停机。

多重调用

MC_Stop指令不可以多重调用，在一条stop指令有效期间调用其他的stop指令，其他的stop指令报故障。

打断

指令有效时轴处于Stopping状态，其他运动指令运不可以打断该指令，指令无效时轴由Stopping状态切换到Standstill状态，其他运动控制指令可以运行。

本指令可以被MC_ImmediateStop指令打断，当MC_Stop指令有效时调用MC_ImmediateStop指令，MC_Stop指令报错。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

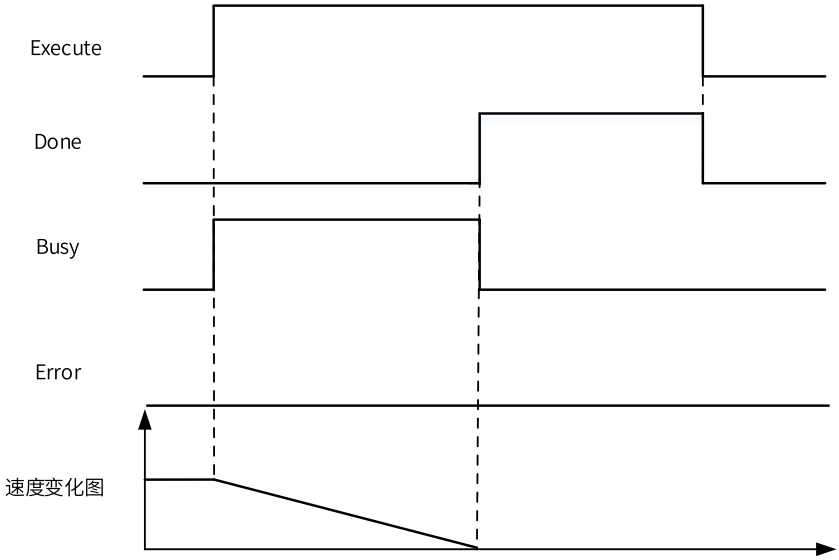
只有当轴处于Disabled 和errorstop状态时报错9108。

当调用MC_ImmediateStop指令让轴处于Stopping状态时调用本指令报错9142。

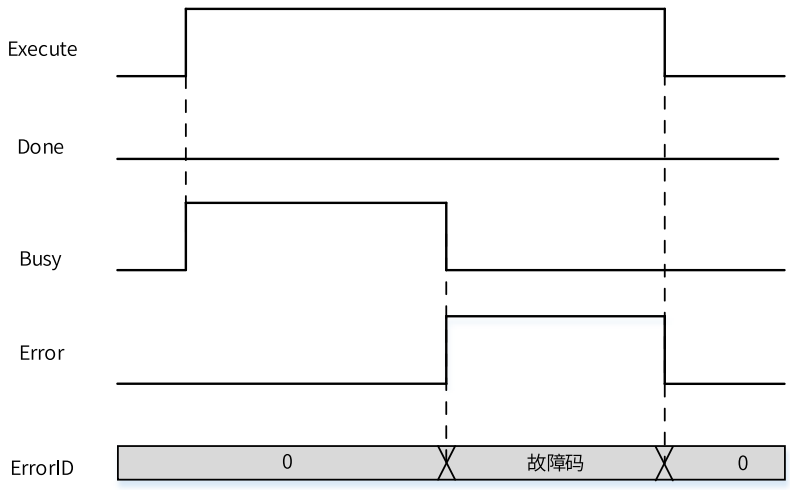
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

- 调用MC_MoveVelocity指令之后再调用本指令。



- 指令运行期间驱动器发生故障。



3.14.20 MC_Halt

MC_Halt — 暂停指令

图形块

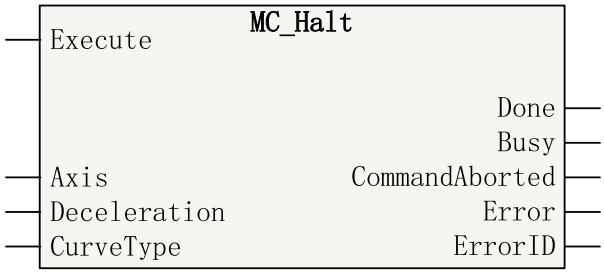


表3–259 指令列表格式

16位指令	—					
32位指令	MC_Halt 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Deceleration	减速度	否	-	正数，小于最大加速度	REAL
S3	CurveType	曲线类型 0：T型速度曲线 1：5段S加减速度	是	0	0-1	INT
D1	Done	停机完成	是	FALSE	FALSE/TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE/TRUE	BOOL
D3	CmdAborted	终止执行	是	FALSE	FALSE/TRUE	BOOL

D4	Error	指令故障标志	是	FALSE	FALSE TRUE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–260 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	✓	-
S3	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现总线EtherCAT总线轴或本地脉冲轴的暂停功能，指令上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在指令的上升沿， 功能块锁存Deceleration 、CurveType等输入参数并做减速运动，该指令可被其他指令打断。

  CurveType用于设定速度曲线的类型。

  1. CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动。
  2. CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。

打断

在PLCOpen中该指令有效时轴将处于Discrete motion状态，其他可以让轴处于Continuous motion、Discrete motion或者其他符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

- 在Execute=TRUE且Done信号无效时即将碰到限位信号执行需要执行减速时CommandAboarted有效。
- 在Execute=TRUE且Done信号无效时MC_Power指令无效导致轴去使能CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

只有当轴处于Disabled 、errorstop、Homing、状态时报错9108。

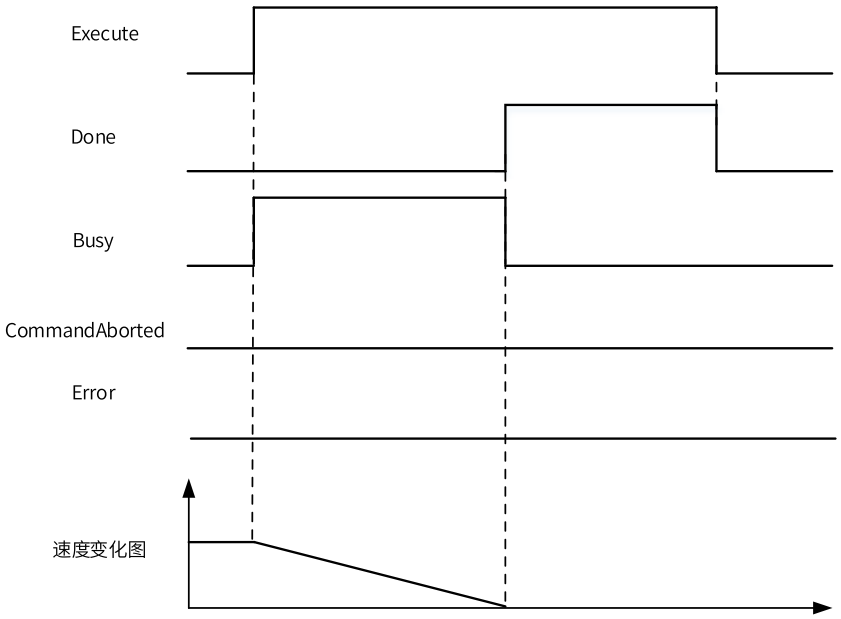
当调用MC_Stop指令让轴处于Stopping状态时调用本指令报错9115。

当调用MC_ImmediateStop指令让轴处于Stopping状态时调用本指令报错9142。

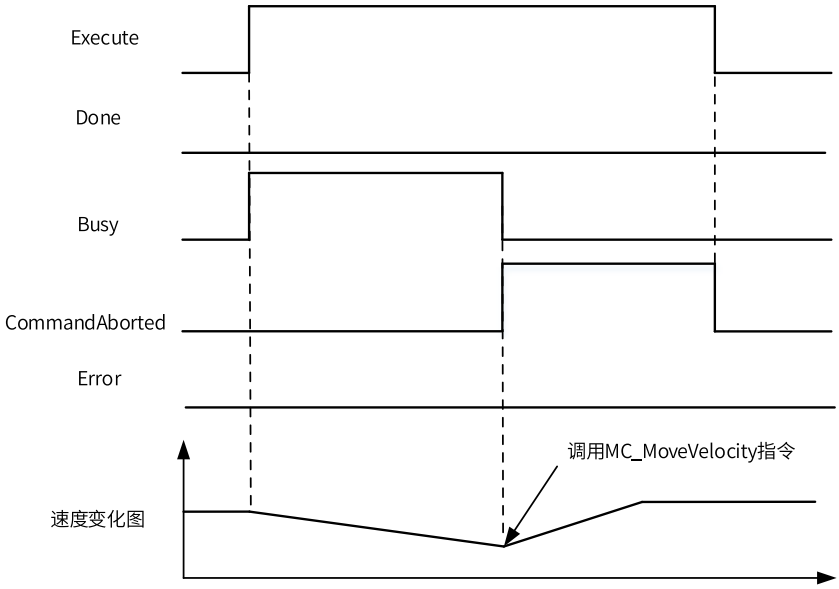
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

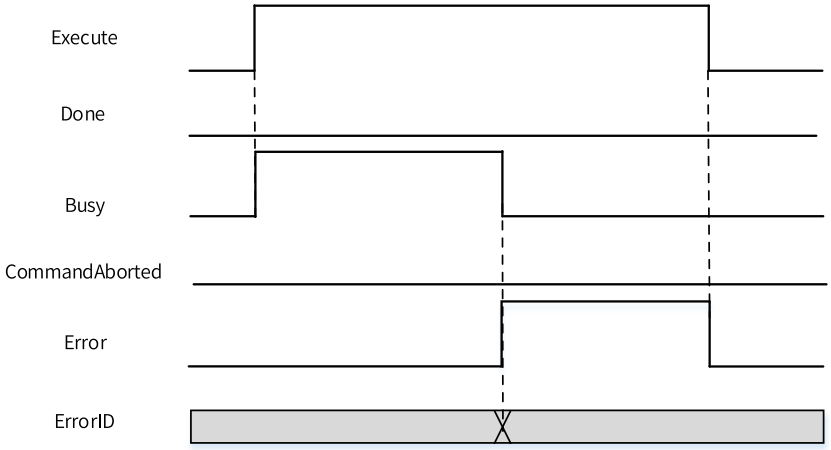
- 调用定位之后，触发MC_Halt指令。



- 在触发MC_Halt指令后，重新调用速度指令打断MC_Halt指令的执行。



- 在MC_Halt指令执行期间，发生故障停机。



3.14.21 MC_MoveFeed

MC_MoveFeed — 中断定长指令

图形块

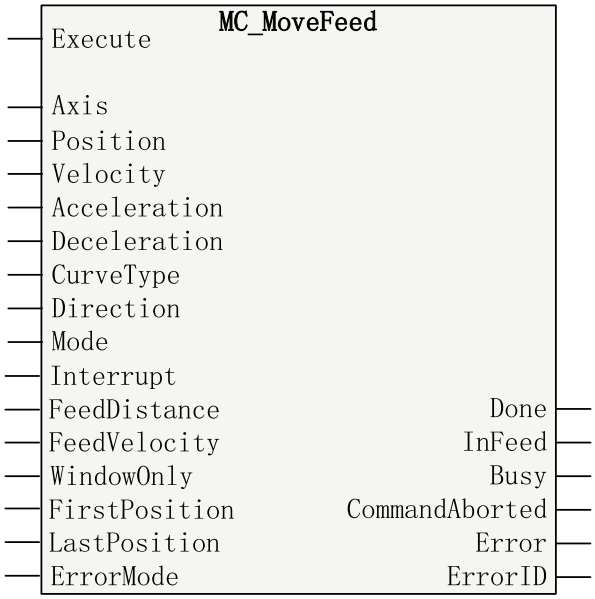


表3-261

16位指令	—					
32位指令	MC_MoveFeed 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Position	目标位置	否	-	正数/负数/0	REAL
S3	Velocity	目标速度	否	-	正数，小于最大速度	REAL
S4	Acceleration	加速度	否	-	正数，小于最大加速度	REAL
S5	Deceleration	减速度	是	同加速度	正数，小于最小加速度	REAL
S6	CurveType	速度曲线类型 0：T型加速 1:5段S曲线加速	是	0	0-1	INT
S7	Direction	环形模式绝对定位下的运动方向 0：正方向（目标速度为正） 1：负方向（目标速度为负） 2：最短距离 3：当前方向	是	0	0-3	INT

S8	Mode	模式 0：绝对定位模式 1：相对定位模式 2：速度模式	是	0	0-2	INT
S9	Interrupt	中断源选择 0：探针1 1：探针2	是	0	0-1	INT
S10	FeedDistance	中断源到达后的位移 正数表示中断源到达后按照当前的运动方向运行FeedDistance指定的距离 负数表示中断源到达后减速并向相反的方向运行-FeedDistance指定的距离	否	-	正数/负数/0	REAL
S11	FeedVelocity	中断到达后的目标速度	是	同Velocity	正数，小于最大速度	REAL
S12	WindowOnly	使能中断源窗口 0：不使能窗口检测功能 1：使能窗口检测功能	是	FALSE	FALSE/TRUE	BOOL
S13	FirstPosition	中断源窗口开始位置	是	0	正数/负数/0	REAL
S14	LastPosition	中断源窗口结束位置	是	0	大于FirstPosition	REAL
S15	ErrorMode	故障模式 FALSE：Position指定的位置到达后如果没有检测到中断信号则Done信号置位，指令不报故障 TRUE：Position指定的位置到达后如果没有检测到中断信号则Error信号置位，指令报故障	是	FALSE	FALSE/TRUE	BOOL
D1	Done	目标位置到达	是	FALSE	-	BOOL
D2	InFeed	中断信号有效	是	FALSE	-	BOOL
D3	Busy	忙标志	是	FALSE		BOOL
D4	CmdAborted	终止执行	是	FALSE	-	BOOL

D5	Error	指令故障标志	是	FALSE	-	BOOL
D6	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3-262 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	-	-	✓	-
S3	-	-	-	✓	✓	-	-	✓	-
S4	-	-	-	✓	✓	-	-	✓	-
S5	-	-	-	✓	✓	-	-	✓	-
S6	-	-	-	✓	✓	-	✓	-	-
S7	-	-	-	✓	✓	-	✓	-	-
S8	-	-	-	✓	✓	-	✓	-	-
S9	-	-	-	✓	✓	-	✓	-	-
S10	-	-	-	✓	✓	-	-	✓	-
S11	-	-	-	✓	✓	-	-	✓	-
S12	✓	✓	✓	-	-	-	-	-	-
S13	-	-	-	✓	✓	-	✓	-	-
S14	-	-	-	✓	✓	-	✓	-	-
S15	✓	✓	✓	-	-	-	-	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	✓[1]	✓	✓	-	-	-	-	-	-
D6	-	-	-	✓	✓	-	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现总线EtherCAT总线轴或本地脉冲轴的中断定长功能，指令上升沿有效，本指令不支持虚轴模式。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。

- 功能描述
  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在指令的上升沿，功能块锁存Position、Velocity，Direction ，Acceleration，Deceleration等输入参数。

  在中断到达前，轴将按照Position、Velocity，Direction ， Mode等设定的参数做绝对定位（Mode = 0），相对定位（Mode =1）或者连续运动（Mode = 2），在Interrupt设置的中断源产生中断信号后，轴将按照FeedDis和FeedVelo设置的参数在中断到达时的位置做相对运动。

  - Position：目标位置。当Mode = 0（绝对定位）或者Mode =1（相对定位）时用于设定轴在中断到达前的目标位置。
  - Velocity：目标速度，用于设置轴在中断到达前的目标速度。
  - CurveType：用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration：设定的值做加速或者减速运动；CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。
  - Direction：运动方向，意义同MC_MoveAbsolute中的Direction。当Mode = 0（绝对定位）该参数用于设定环形模式下轴的运动方向。当设置为0时表示正向运行，设置为1时表示反向运动，设置为2时表示最短路径，设置为3时表示当前方向。
  - Mode：中断到达前的运动模式。当Mode=0时，轴在中断到达前将做绝对定位；当Mode=1时1，轴在中断到达前将做相对定位；当Mode=2时，轴在中断到达前将做连续运动。
  - Interrupt：中断源选择，当Interrupt=0时，中断源为探针1，并在探针1的上升沿有效；当interrupt=1时，中断源为探针2，并在探针2的上升沿有效。
  - FeedDistance:中断到达后的目标位移。为正时表示当中断信号到达时，按照当前运动方向运行FeedDistance设定的距离，为负时则表示中断信号到达时，向相反方向运行FeedDistance设定的距离。
  - FeedVelocity：中断到达后的目标速度。
  - ErrorMode，无中断时的故障处理模式。在绝对定位和相对定位模式下，如果Position指定的位置（位移）到达后仍然没有检测到中断信号，当ErrorMode = FALSE时指令不报故障，当ErrorMode = TRUE时指令报故障。
  - InFeed：中断信号到达后InFeed输出有效。

打断

在PLCOpen中该指令有效时轴将处于Discrete motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Execute=TRUE且Done信号无效时即将碰到限位信号执行需要执行减速时CommandAboarted有效。

在Execute=TRUE且Done信号无效时轴去使能CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

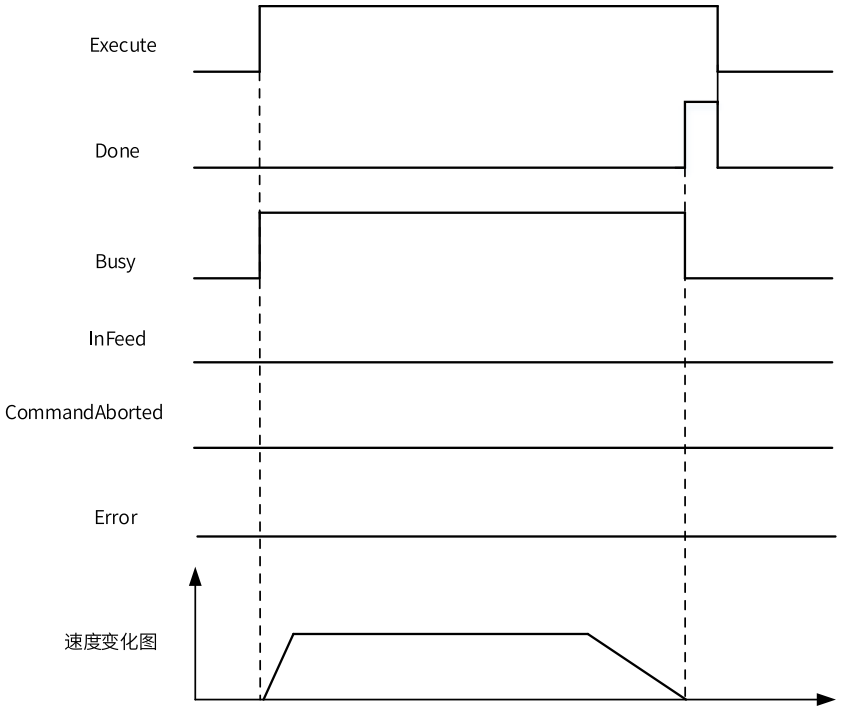
只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升沿如果轴使能了虚轴模式则轴报9133。

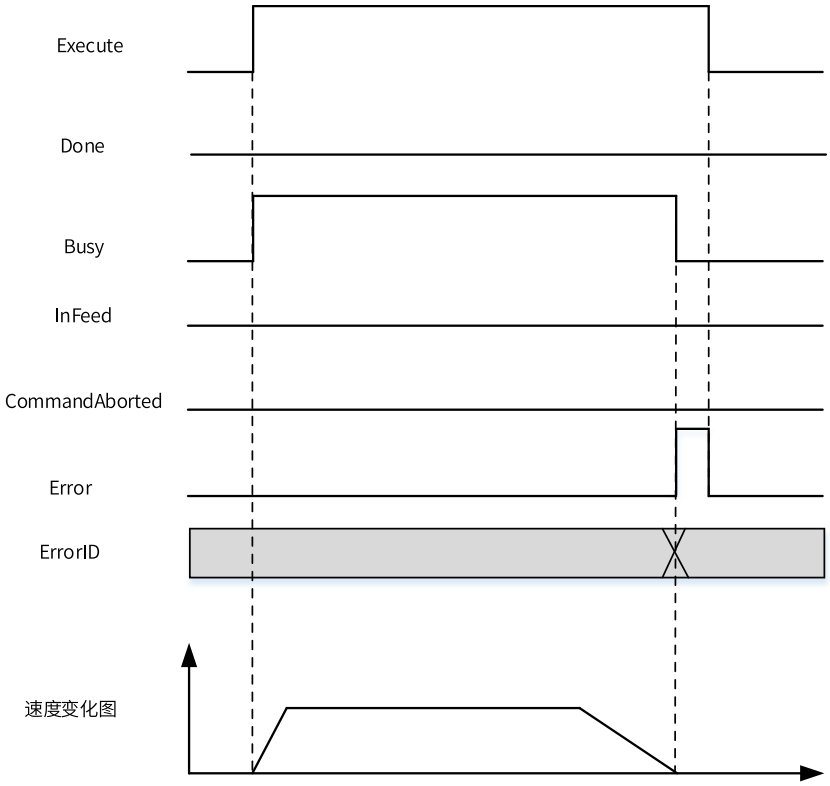
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

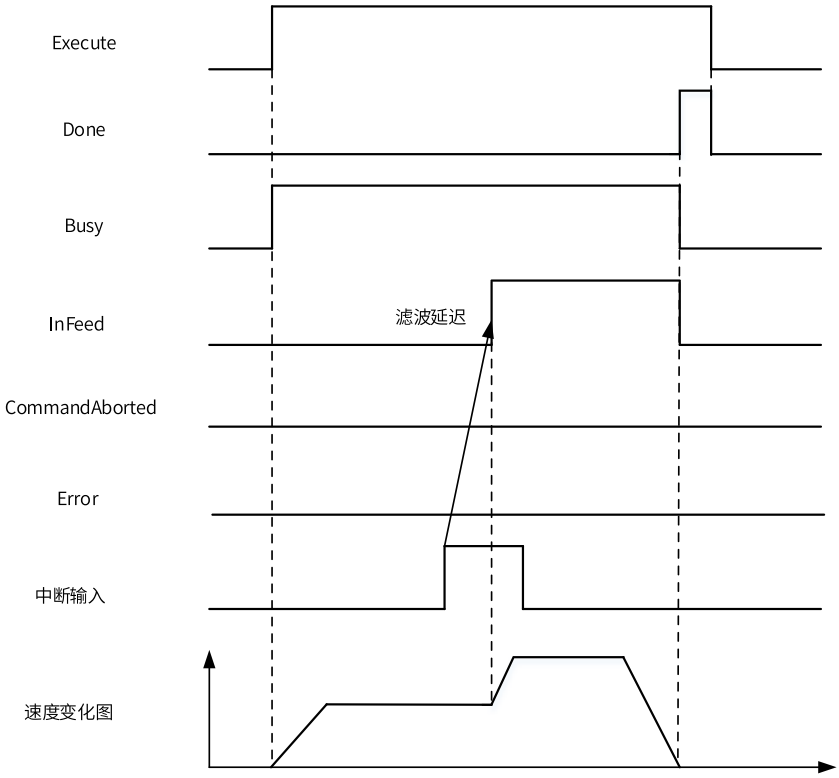
- 当选择相对定位、绝对定位模式，没有触发中断信号且ErrorMode = FALSE时。



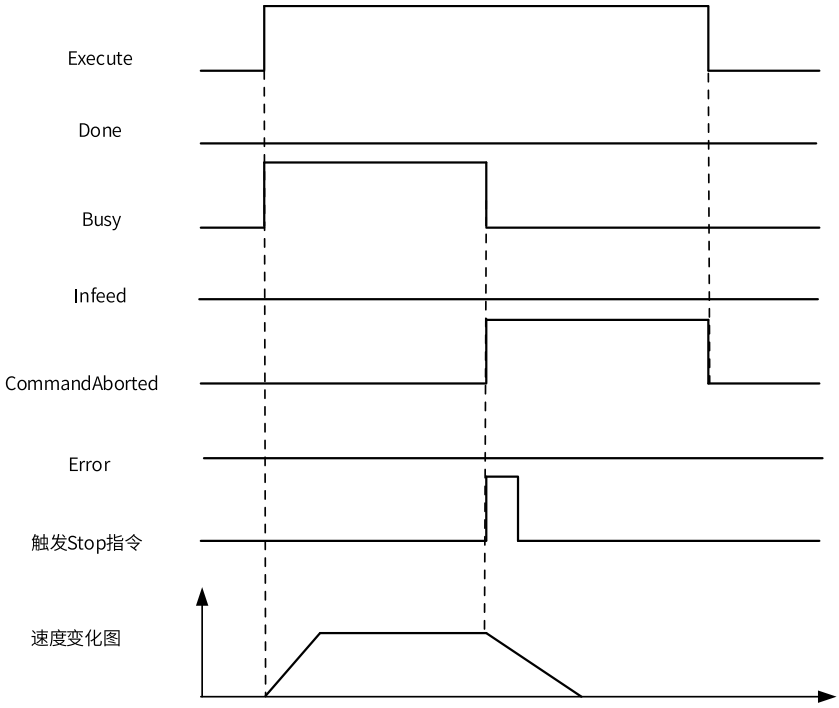
- 当选择相对定位、绝对定位模式，没有触发中断信号且ErrorMode = TRUE时。



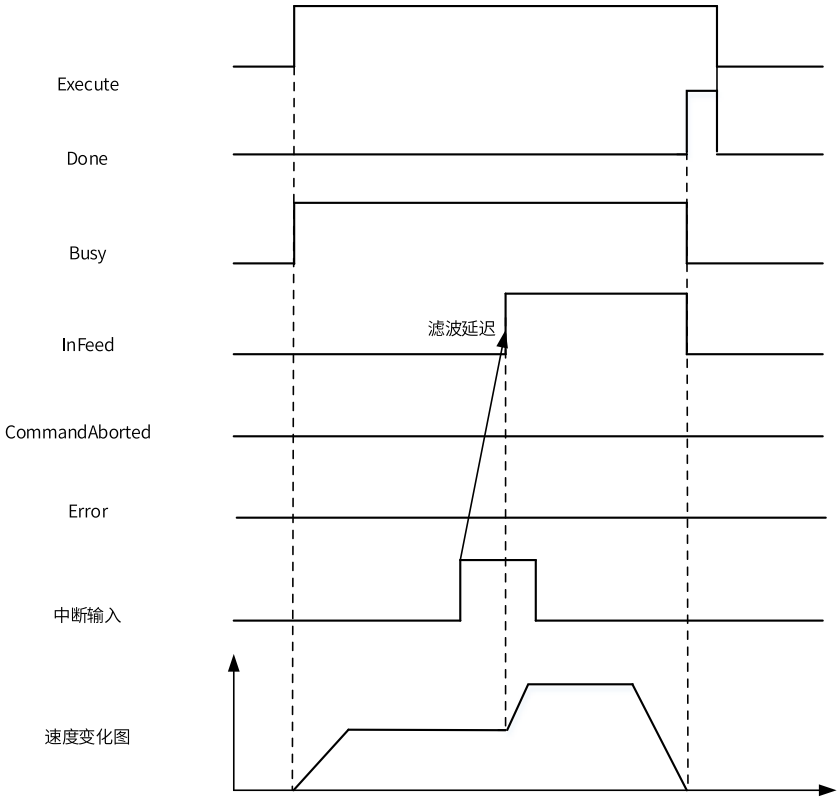
- 当选择相对定位、绝对定位模式，有中断信号时。



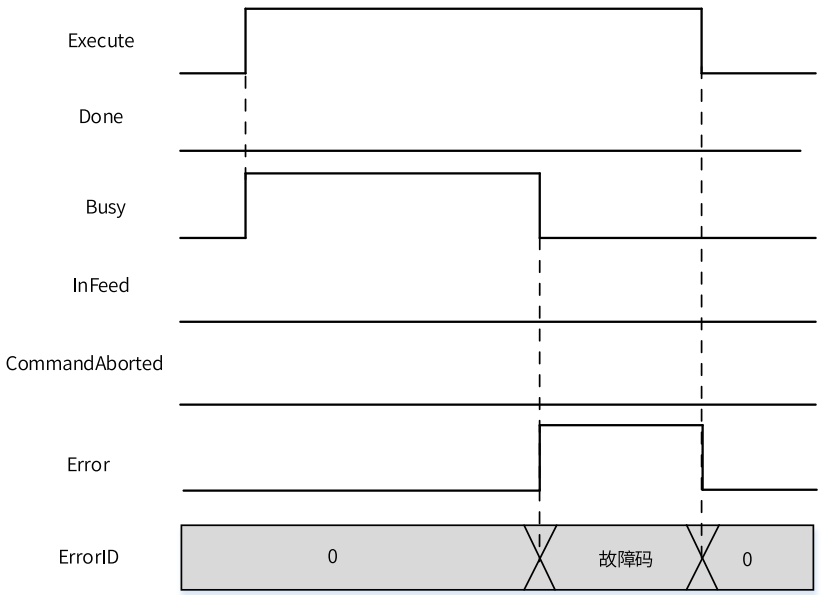
- 选择速度模式，没有触发中断，运行一段时间以后用MC_Stop指令打断。



- 选择速度模式，触发中断。



- 指令运行中发生故障。



3.14.22 MC_MoveBuffer

MC_MoveBuffer — 多段位置定位指令

图形块

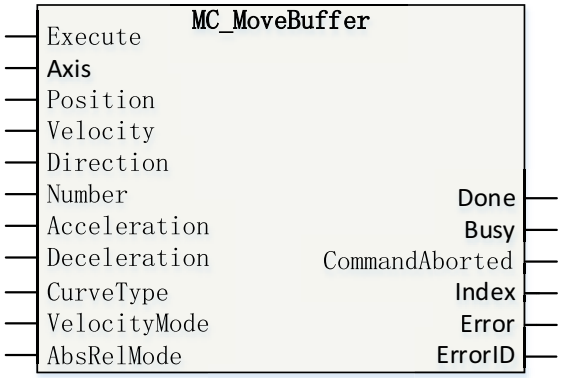


表3-263 指令列表格式

16位指令	—					
32位指令	MC_MoveBuffer 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
S2	Position	目标位置起始地址	否	-	正数 负数 0	FLT32，数组*16
S3	Velocity	目标速度起始地址	否	-	正数	FLT32，数组*16

S4	Direction	环形模式绝对定位方向起始地址 0：正方向（目标速度大于0） 1：负方向（目标速度小于0） 2：最短距离 3：当前方向	是	0	0-3	INT，数组*16
S5	Number	缓冲对数量	否	-	1-16	INT
S6	Acceleration	加速度	否	-	-	FLT32
S7	Deceleration	减速度	是	加速度	-	FLT32
S8	CurveType	速度曲线类型 0：T型加速度 1:5段S型加速 其他：T型速度曲线	是	0	-	INT
S9	VelocityMode	速度切换模式 0：减速到0然后启动下一段 1：保持当前速度启动下一段	是	0	0-1	INT
S10	AbsRelMode	定位模式 0：绝对定位 1：相对定位	是	0	0-1	INT
D1	Done	停机完成	是	FALSE	-	BOOL
D2	Busy	忙标志	是	FALSE	-	BOOL
D3	CommandAborted	终止执行	是	FALSE	-	BOOL
D4	Index	当前正在执行的段	是	0	0-15	INT
D5	Error	指令故障标志	是	FALSE	-	BOOL
D6	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第370页“3.14.24 轴故障码”*。

表3–264 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
S2	-	-	-	✓	✓	✓	-	-	-
S3	-	-	-	✓	✓	✓	-	-	-
S4	-	-	-	✓	✓	✓	-	-	-
S5	-	-	-	✓	✓	✓	✓	-	-
S6	-	-	-	✓	✓	✓	-	✓	-

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S7	-	-	-	✓	✓	✓	-	✓	-
S8	-	-	-	✓	✓	✓	-	✓	-
S9	-	-	-	✓	✓	✓	✓	-	-
S10	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	✓	-	-	-
D2	✓[1]	✓	✓	-	-	✓	-	-	-
D3	✓[1]	✓	✓	-	-	✓	-	-	-
D4	-	-	-	✓	✓	✓	-	-	-
D5	✓[1]	✓	✓			✓	-	-	-
D6	-	-	-	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现总线EtherCAT总线轴或本地脉冲轴的多段位置定位功能，上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

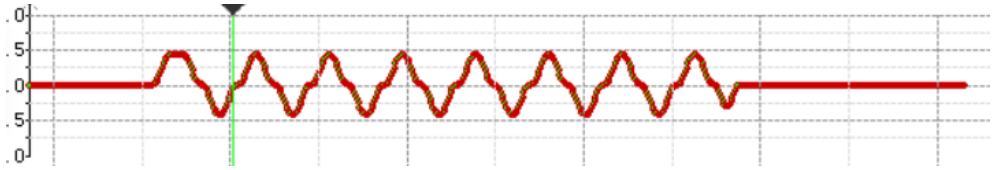
  该Execute输入的上升沿，功能块锁存Position，Velocity，Direction ，Number，Acceleration，Deceleration等输入参数。

  轴将按照AbsRelMode 设定的值做缓冲模式下的绝对定位（AbsRelMode = 0）或相对定位（（AbsRelMode = 1）。该指令最多实现16段位置。

  - Position：目标位置，数组类型，最大16级。绝对定位模式下在用于设定轴的目标绝对位置，相对定位模式下用于设定轴的目标位移。
  - Velocity：目标速度，数组类型，最大16级，用于设置目标速度。
  - Direction：环形绝对定位模式下的目标方向。意义同MC_MoveAbsolute的Direction。
  - Number：缓冲的目标位置、目标速度和方向的组数，范围是1-16，超出该范围将报参数不合理故障。
  - CurveType：用于设定速度曲线的类型。CurveType = 0表示T型曲线，此时轴的速度将按照Acceleration和Deceleration设定的值做加速或者减速运动；CurveType = 1表示5段S曲线，此时Acceleration和Deceleration表示轴在加速和减速过程中达到的最大加速度和最小减速度。
  - VelocityMode ：速度切换模式，当设置为0时，轴在到达前一段目标值时，轴首先减速到0，然后从0速开始启动运行到后一段目标位置；当设置为1时，轴将按照前一段的目标速度运行到目标位置，然后按照加速度（减速度）切换到新的速度并向新的目标位置运动。

说明

在绝对定位速度保持的模式下可能出现一种情况，如设置三段位置，第一段为10，第二段为10.1，第三段为10.2，目标速度设置为100，反馈速度也是100，ethercat任务周期是8ms，此时每一个ethercat周期目标位移的增量是0.8。在运行第一段时如果某一个ethercat周期开始时当前位置是9.9，则下一个ethercat周期当前位置变成10.7，此时要已经超过了第二段的目标位置，因此速度需要减速反向，此时可能产生如下图所示的速度曲线：



打断

在PLCOpen中该指令有效时轴将处于Discrete motion状态，其他可以让轴处于Discrete motion或者符合PLCOpen状态机切换的指令均可以打断该指令，指令被打断时CommandAboarted有效。

在Execute=TRUE且Done信号无效时即将碰到限位信号执行需要执行减速时CommandAboarted有效。

在Execute=TRUE且Done信号无效时轴去使能CommandAboarted有效。

报错

当轴号不存在或轴类型不匹配时指令报错9101。

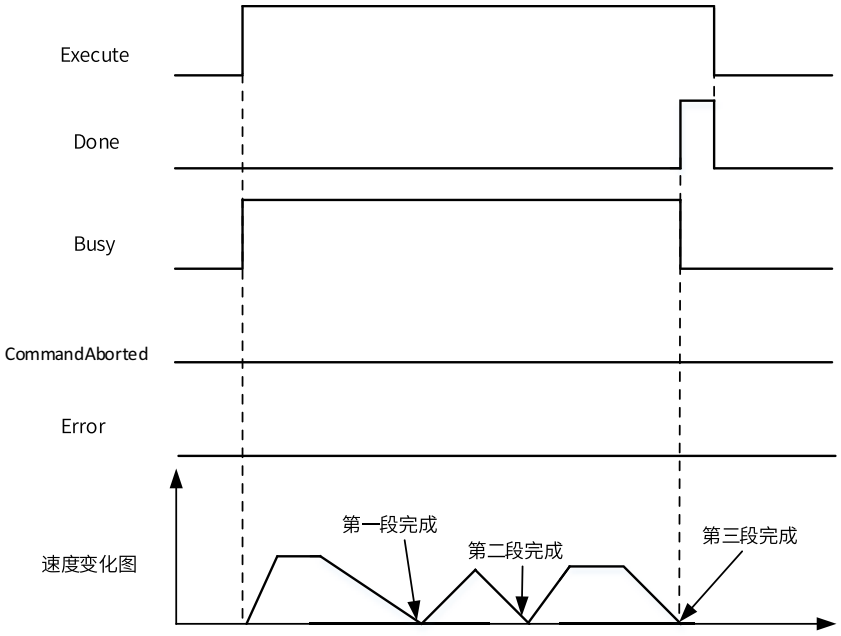
当轴初始化失败时指令报错9102。

只有当轴处于StandStill、DiscreteMotion、ContinuousMotion状态时调用本指令有效，其他状态报错9108。

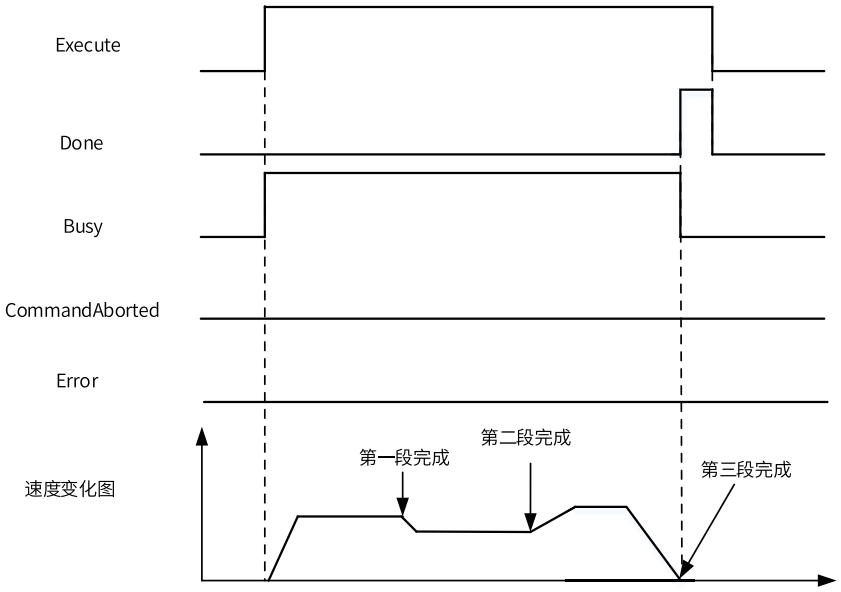
- 在Execute输入的上升沿如果轴处于在线调试状态报错9116。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute输入的上升如果指令左侧的参数超范围或不合理则指令报故障。
- 在Execute=TRUE且Done信号无效时轴进入调试状态报错9116。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

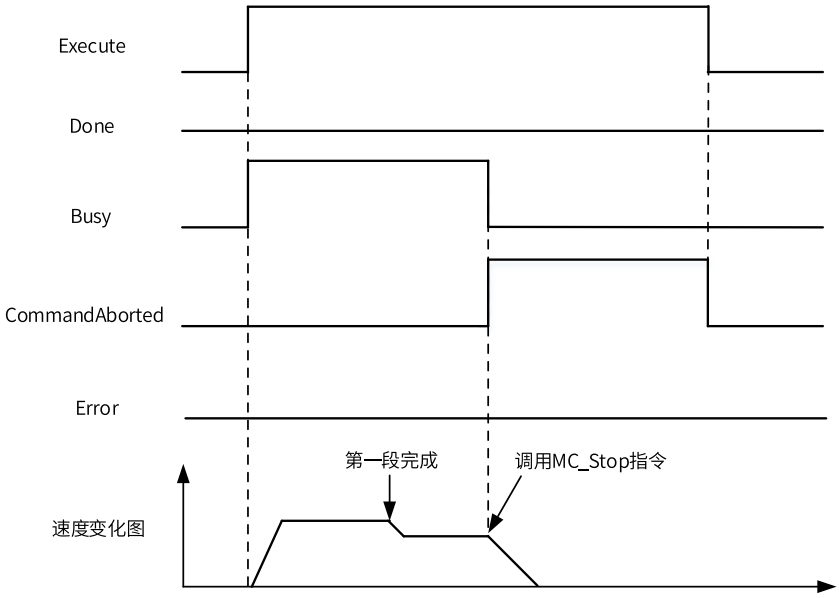
- 设置为3段缓冲，当VelocityMode =0时。



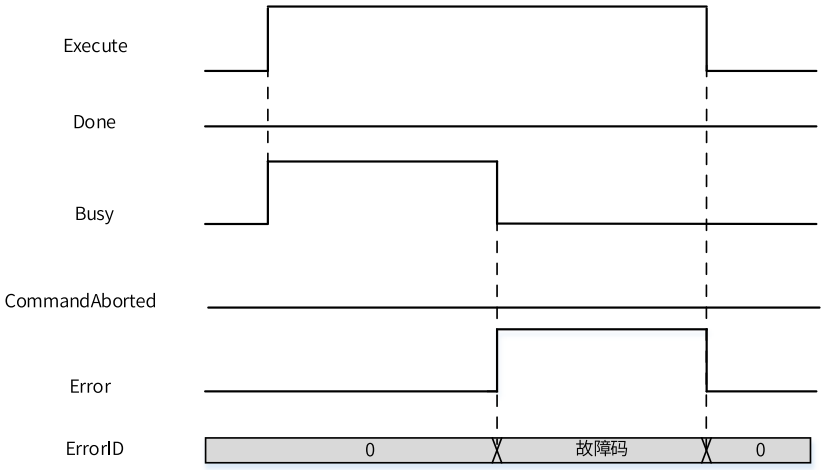
- 设置为3段缓冲，当VelocityMode =1时。



- 设置为3段缓冲，当运行期间被MC_Stop指令打断时。



- 设置为3段缓冲，但运行期间报错时。



3.14.23 MC_ImmediateStop

MC_ImmediateStop — 急停指令

图形块

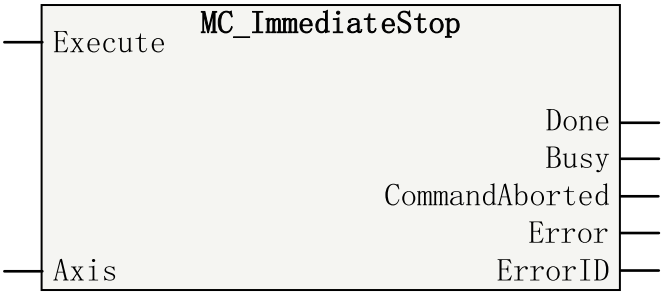


表3–265 指令列表格式

16位指令	—
32位指令	MC_ImmediateStop 连续执行

操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Axis	轴名称/轴ID	否	-	0-32767	INT _sMCAXIS_INFO
D1	Done	停机完成	是	FALSE	FALSE/TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE/TRUE	BOOL
D3	CommandAborted	终止执行	是	FALSE	FALSE/TRUE	BOOL
D4	Error	指令故障标志	是	FALSE	FALSE/TRUE	BOOL
D5	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第370页“3.14.24 轴故障码”。

表3–266 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	✓	✓	✓	✓	-	-
D1	✓[1]	✓	✓	-	-	-	-	-	-
D2	✓[1]	✓	✓	-	-	-	-	-	-
D3	✓[1]	✓	✓	-	-	-	-	-	-
D4	✓[1]	✓	✓	-	-	-	-	-	-
D5	-	-	-	✓	✓		-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于EtherCAT总线轴或本地脉冲轴实现紧急停止功能，上升沿有效。

- 轴号的设置
  - 在Execute输入的上升沿锁存Axis。
  - 如果Axis设置的是轴号，在Execute=TRUE期间修改Axis无效。
  - 如果Axis设置的是轴号，在Execute=FALSE期间修改Axis无效。
- 功能描述

  只有使用MC_Power指令将轴切换到使能状态才可以调用本指令。

  在指令的上升沿， 功能块将轴的PLCOpen状态机切换到Stopping状态，将驱动器的402协议状态机切换到快速停机状态，驱动器根据0x605A设定的停止方式停止运行。对于汇川伺服IS620N来说，停止方式如下：

- 当伺服处于CSP模式时：

设定值	停机方式
0	自由停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
1	以 2007-10h 紧急停止转矩停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
2	
3	
4	NA
5	以 2007-10h 紧急停止转矩停机，保持位置锁定状态
6	
7	

- 当伺服处于HM模式时：

设定值	停机方式
0	自由停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
1	以6084h(HM：609Ah) 斜坡停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
2	以6085h 斜坡停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
3	急停转矩停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
4	N/A
5	以6084h(HM：609Ah) 斜坡停机，保持位置锁定状态
6	以6085h 斜坡停机，保持位置锁定状态
7	以2007-10h 紧急停止转矩停机，保持位置锁定状态

- 当伺服处于CST模式时：

设定值	停机方式
0	自由停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
1	以6087h 斜坡停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
2	
3	自由停机，保持自由运行状态 选择该模式，指令Done信号输出有效后伺服的反馈速度不一定为0
4	N/A
5	以6087h 斜坡停机，保持位置锁定状态
6	
7	自由停机，保持位置锁定状态

- 重启指令

  同一条MC_ImmediateStop指令重复执行，在减速期间重新触发同一条MC_ImmediateStop指令则按照最后一次触发停机。

- 多重调用
  调用多条MC_ImmediateStop指令时以最先触发上升沿的那条指令为准，其他指令报故障，故障码为9143（重复调用急停指令）。

当轴处于Stopping状态，其他运动指令运不可以打断该指令。

指令能流的下降沿轴由Stopping状态切换到Standstill状态，其他运动控制指令可以运行。

本指令的优先级高于MC_Stop指令，在MC_Stop指令有效期间调用本指令，MC_Stop指令指令报错。

打断

在Execute = TRUE期间轴去使能打断信号输出有效。

报错

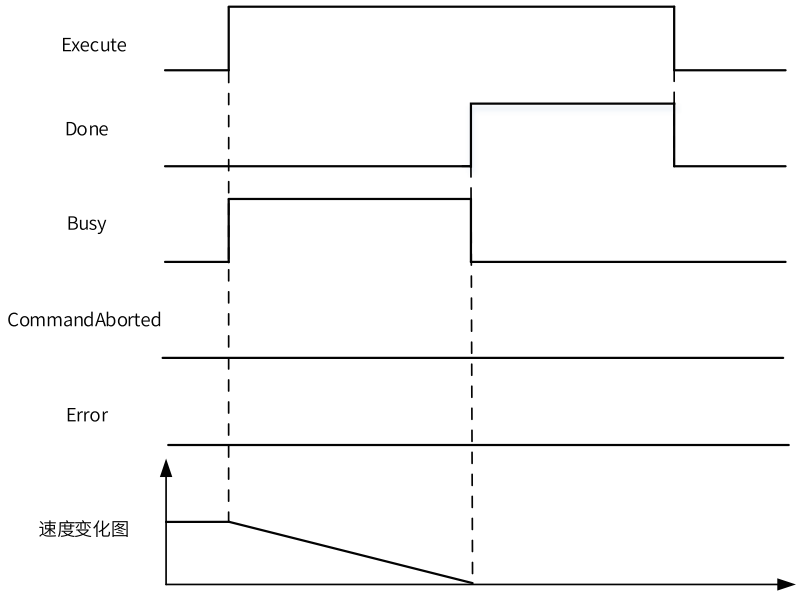
当轴号不存在或轴类型不匹配时指令报错9101。

当轴初始化失败时指令报错9102。

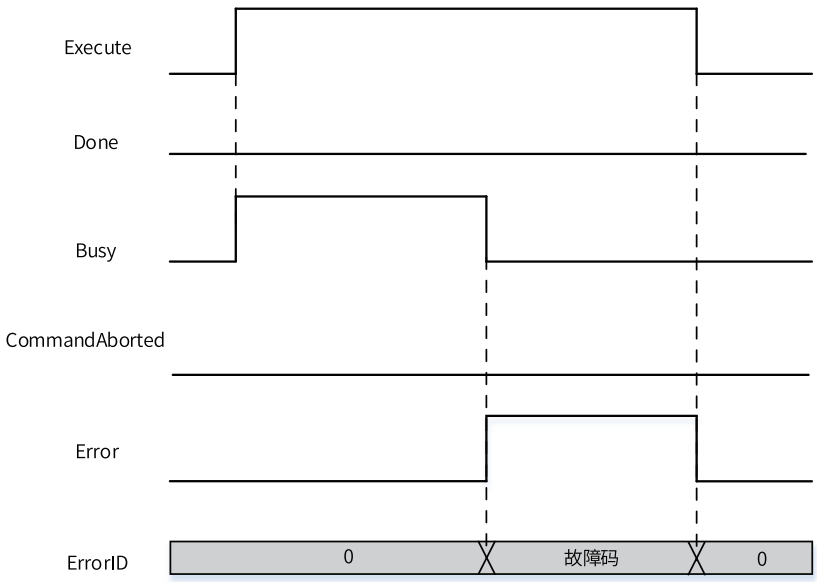
- 在Execute输入的上升沿如果轴处于disabled、errorstop两种状态时报错9108。
- 在Execute输入的上升如果轴正在执行故障减速报错9106。
- 在Execute = TRUE期间轴发生故障进入errorstop状态指令显示轴的在errorstop状态的故障码。

时序图

- 调用MC_MoveVelocity指令之后再调用本指令。



- 指令运行期间驱动器发生故障。



3.14.24 轴故障码

轴故障码分为本地脉冲轴故障码和运动控制轴故障码两类。使用运动控制轴指令时，若指令报故障，请查看相应的故障代码释义。

本地脉冲轴故障码

故障码	原因	解决方案	触发停机
9001 (0x2329)	急停 触发了急停端子输入	急停输入有效，请在关闭急停端子输入后调用MC_Reset指令复位故障	是
9002 (0x232a)	跟随误差过大（保留）	请调整加速度（减速度）目标速度	是
9003 (0x232b)	超速 脉冲输出频率超过200K	脉冲输出频率超过200K	是
9020 (0x233c)	原点回归错误 没有映射负限位	该原点回归方式需要映射负限位，请在配置界面映射负限位	是
9021 (0x233d)	原点回归错误 没有映射正限位	该原点回归方式需要映射正限位，请在配置界面映射正限位	是
9022 (0x233e)	原点回归错误 没有映射原点信号	该原点回归方式需要映射原点开关，请在配置界面映射原点开关	是
9023 (0x233f)	原点回归错误 以原点返回速度运行时输出频率超过200K 以原点返回接近速度运行时输出频率超过200K	修改单位换算设置确保原点返回速度和原点返回接近速度不超过200K 修改原点返回速度确保输出频率不超过200K 修改原点返回接近速度确保输出频率不超过200K	是
9024 (0x2340)	原点回归错误 原点返回过程中超时	检查限位信号能否正常导通 检查原点返回超时时间设置是否过小	是

故障码	原因	解决方案	触发停机
9025 (0x2341)	原点返回错误 原点返回过程中限位信号错乱	检查是否触发了当前原点回归方式没有用的限位信号	是
9030 (0x2342)	限位有效	检测是否在正常运行过程中碰到限位	否
9031 (0x2343)	同步异常 目标发送脉冲数和实际发送脉冲数不匹配	检测是否在正常定位过程中碰到限位	否

说明 本地脉冲轴发生故障时，请查阅以上故障码了解故障信息。

运动控制轴故障码

故障码	原因	解决方案	触发停机
9101	AxisID指定的轴类型错误 AxisID指定的轴不存在	检查指令是否支持AixsID指定的轴 检查AxisID指定的轴是否存在	否
9102	轴配置数据丢失 轴配置参数不合理	检查配置参数是否有误	否
9103	在轴无故障的情况下调用了MC_Reset指令	检查是否在轴没有切换到ErrorStop状态到的情况下故障的情况下调用了MC_Reset指令	否
9104	调用MC_ReadStatus指令时轴处于未知状态	通过在线监控功能查看轴当前的状态是否是不可控状态	否
9105	不允许设置当前位置	检测是否已经调用MC_SetPositon指令	否
9106	轴正在执行故障停机	等待轴故障停机完成后解决了故障后在执行命令	否
9107	参数不合理	检查指令左侧的参数是否合理	是
9108	PLCOpen状态机不合理	检查当前PLCOpen的状态机是否满足本指令运行的条件，如果不满足，调用相应的指令将轴切换到满足状态	否
9109	指令执行过程中轴进入Disabled状态	查看轴是否进入了Disabled状态	否
9110	在停止过程中重复调用了MC_Stop指令	检查程序查看是否重复调用了MC_Stop指令	否
9111	指令链表丢失	检查后台版本和单板版本是否匹配	否
9112	轴号变更 在指令能流有效期间轴号发生变更	MC_Power、MC_Jog等Enable型指令在能流有效期间不能变更轴号	是
9113	MC_Reset指令复位超时	查看驱动器故障能否复位 查看轴的故障类型能否复位	否
9114	轴写0x6060失败	检查网络通信是否收到干扰	否
9115	Stopping状态下调用了MC_Halt指令	Stopping状态下不允许调用MC_halt指令	否
9116	当前轴处于在线调试模式	检查当前轴是否处于在线调试模式	否
9117	指令暂未开发	指令功能还没有开发	否
9118	指令加（减）速超过最大加速度	检查指令的加（减）速度是否超过最大加速度	是

故障码	原因	解决方案	触发停机
9119	MC_Jog指令的目标速度超过最大点动速度	检查Jog指令的目标速度是否超过最大点动速度	是
9120	目标速度超过最大速度	检查指令的目标速度是否超过最大速度	是
9121	Jog指令正向运动和负向运动信号同时有效	Jog指令的JogForward和JogBackward信号不能同时有效	是
9122	EtherCAT总线轴中没有映射控制字	PDO中添加控制字然后映射到轴中	否
9123	EtherCAT总线轴中没有映射目标位置	PDO中添加目标位置后映射到轴中	否
9124	EtherCAT总线轴中没有映射目标力矩	POD中添加目标力矩然后映射到轴中	否
9125	EtherCAT总线轴中没有映射状态字	PDO中添加状态字然后映射到轴中	否
9126	EtherCAT总线轴中没有映射当前位置	PDO中添加反馈位置然后映射到轴中	否
9127	EtherCAT总线轴中没有映射0x60fd	PDO中添加0x60fd然后映射到轴中	否
9128	EtherCAT总线轴中没有映射当前力矩	PDO中添加当前力矩然后映射到轴中	否
9129	EtherCAT总线轴中没有映射探针控制字	PDO中添加探针控制字然后映射到轴中	是（定长） 否（探针）
9130	EtherCAT总线轴中没有映射探针状态字	PDO中添加探针状态字然后映射到轴中	是（定长） 否（探针）
9131	EtherCAT总线轴中没有映射探针位置	PDO中添加探针状态机然后映射到轴中	是（定长） 否（探针）
9132	正在运行中断定长指令，探针通道被占用	探针指令和中断定长指令不可同时占用同一个探针通道，程序中同时调用时中断定长指令优先级更高	否
9133	启用了虚轴模式	当前指令不支持虚轴模式	否
9134	保留	-	-
9135	中断定长没有触发中断信号	调用中断定长指令，在定位完成后没有检测到中断信号	否
9136	中断定长过程中探针通道被其他指令占用	中断定长过程中探针通道不能被占用	是
9137	总线驱动器中没有映射控制模式0x6060	PDO中添加0x6060然后映射到轴中	否
9138	总线驱动器中没有映射控制模式0x6061	PDO中添加0x6061然后映射到轴中	否
9139	在原点回归过程中重复调用MC_home指令	在回原过程中不允许重复调用回原指令	否
9140	指令的目标力矩超过最大值	检查指令的目标力矩是否超过正负力矩限制值	是
9141	总线驱动器中没有映射最大速度	在PDO中添加0x607f,然后映射到轴中	否
9142	急停指令有效	查看是否已经调用了急停指令	否
9143	重复调用急停指令	查看是否重复调用了急停指令	否
9144	点动运行过程中碰到限位	查看限位是否有效	否
9145	目标位置超过9999999 单精度浮点数超过9999999后会丢失精度，目标位置不能超过该值	检查目标位置是否正确，重新设定目标位置 修改齿轮比，保证需要设定的目标位置不大于9999999	是
9146	目标速度超过9999999 单精度浮点数超过9999999后会丢失精度，目标速度不能超过该值	检查目标速度是否正确，重新设定目标速度 修改齿轮比，保证需要设定的目标速度不大于9999999	是

故障码	原因	解决方案	触发停机
9147	目标加速度超过9999999 单精度浮点数超过9999999后会丢失精度，目标加速度不能超过该值	检查加目标速度是否正确，重新设定目标加速度 修改齿轮比，保证需要设定的目标加速度不大于9999999	是
9148	目标减速度超过9999999 单精度浮点数超过9999999后会丢失精度，目标减速度不能超过该值	检查目标减速度是否正确，重新设定目标减速度 修改齿轮比，保证需要设定的目标减速度不大于9999999	是
9149	轴处于同步控制模式，不允许调用单轴运动指令	检查轴是否正在执行插补指令，在执行插补期间不允许调用单轴运动类指令	否
9501	驱动器故障	EtherCAT总线轴通过0x603f确定从站故障类型后排出故障。 本地轴查看本地轴故障列表排查故障。	是
9502	驱动器断使能	在运动期间驱动器状态字0x6041切换到非使能状态。	是
9503	驱动器碰到限位	检查是否配置限位且限位信号有效	是
9504	保留	-	-
9505	写0x6060失败	检查网络通信是否收到干扰	是
9506	保留	-	-
9507	保留	-	-
9508	原点回归故障	查找驱动器原点回归失败原因 原点回归是否超时 限位信号是否错误	是
9509	精度丢失	检查指令的浮点型数据是否超过单精度浮点数的精度范围	是
9510	跟随误差过大 设定位置和反馈位置的差值超过设定值	检查加速度是否过大 检查跟随误差设置是否偏小	是
9511	保留	-	-
9512	在运行过程中驱动器通讯失败	检查驱动器是否正常 检查网线是否正常 检查通讯是否收到强烈干扰	是
9513	驱动器故障导致原点回归失败	查看驱动器的故障码解决故障	是
9514	原点回归偏移超过32位导致回原失败	计算当前原点回归的偏移乘以齿轮比是否超过32位，如果超过32需要修改齿轮比	是
9515	从站丢失导致原点回归失败	内部故障，寻求厂家帮助	是
9516	SDO写对象字典0x607C失败导致原点回归失败	1、检查驱动时是否支持0x607C 2、检查网络通信质量	是
9517	SDO写对象字典0x6060为6时失败导致原点回归失败	1、将0x6060配置到PDO中 2、检查网络通信质量	是
9518	SDO读对象字典0x6061失败导致原点回归失败	1、将0x6061配置到PDO中 2、检查网络通信质量	是

故障码	原因	解决方案	触发停机
9519	SDO写对象字典0x6060为8时失败导致原点回归失败	1、将0x6060配置到PDO中 2、检查网络通信质量	是
9551	状态切换失败	检查网络通信是否收到干扰	是
9552	目标速度小于0	检查位置类指令的目标速度是否合理	是
9601	绝对定位指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9602	相对定位指令报错导致停止	检查报错的指令，根据指令的报错码进一步确定故障	是
9603	速度指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9604	点动指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9605	保留	-	-
9606	缓冲指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9607	中断定长指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9608	Stop指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9609	力矩指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9610	Halt指令报错导致停机	检查报错的指令，根据指令的报错码进一步确定故障	是
9800	获取轴的数量失败	更换后台版本	是
9801	轴的数量大于32个	H5U最多支持32个轴，减小轴的数量	是
9802	申请内存失败	检查内存是否用完	是
9803	获取参数失败	检查单板和后台版本是否匹配	是
9804	获取从站失败	无	是

说明 运动控制轴发生故障时，请查阅以上故障码了解故障信息。

3.15 轴组控制指令

3.15.1 指令列表

轴组控制指令涵盖以下指令条目：

指令类别	名称	功能
轴组控制指令	MC_MoveLinear	直线插补
	MC_MoveCircular	圆弧插补
	MC_GroupStop	停止轴组运行
	MC_GroupPause	暂停轴组运行

3.15.2 MC_MoveLinear

MC_MoveLinear — 直线插补指令

图形块

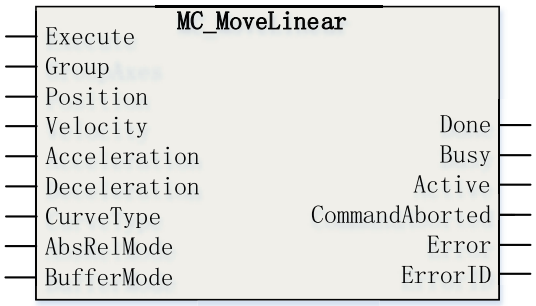


表3–267 指令表格式

16位指令	—					
32位指令	MC_MoveLinear 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Group	轴ID	否	-	0-32767	INT
S2	Position	目标位置	否	-	正数 负数 0	REAL[0-3]或 sMCGROUP_INFO
S3	Velocity	目标速度	否	-	正数	REAL
S4	Acceleration	加速度	否	-	正数	REAL
S5	Deceleration	减速度	是	加速度	正数	REAL
S6	CurveType	速度曲线类型 0：T型速度曲线 其他：T型速度曲线	是	0	0	INT
S7	AbsRelMode	绝对定位与相对定位模式 0：绝对定位 1：相对定位	是	0	0-1	INT
S8	BufferMode	缓冲模式 0：打断 + 无过渡 1：缓冲 + 无过渡 2：前一个速度 + 无过渡 3：附加角过渡	是	0	0-3	INT
D1	Done	目标位置到达 目标位置到达后为True	是	FALSE	TRUE FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE FALSE	BOOL

D3	Active	控制中 开始执行本段曲线时为True	是	FALSE	TRUE FALSE	BOOL
D4	CommandAborted	终止执行	是	FALSE	TRUE FALSE	BOOL
D5	Error	指令故障标志	是	FALSE	TRUE FALSE	BOOL
D6	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第391页“3.15.6 故障码”。

功能和指令说明

MC_MoveLinear指令用于控制轴组直线插补功能，上升沿有效。

- 轴号的设定
  在Execute输入的上升沿锁存Group。

  在Execute=TRUE期间修改Group无效。

  在Execute=FALSE期间修改Group无效。

- 与单轴控制指令的关系
  只有调用MC_Power指令将轴切换到StandStill状态才可以触发本指令。

  在单轴运动期间（例如正在做点位运动、力矩控制、回原、停止等）触发本指令无效。

  触发本指令后单轴的PLCOpen状态机处于同步运动模式，运动期间不可以用单轴运动类指令打断，插补曲线完成后单轴的PLCOpen状态机进入StandStill状态，此时可以执行单轴的运动类指令。

  Velocity表示插补器的目标速率，其中坐标轴的标速度按照公式（1）、（2）、（3）分解。

- 参数说明
  Position用于设定目标位置或移位，Position[0]表示x轴的位置位移分量，Position[1]表示y轴的位置位移分量，Position[2]表示z轴的位置位移分量，Position[3]表示辅助轴的位置位移分量。

  Velocity表示插补器的目标速率，其中坐标轴的标速度按照公式（1）、（2）、（3）分解。

$$Vx = V \times \cos\alpha \quad (1)$$

$$Vy = V \times \cos\beta \quad (2)$$

$$Vz = V \times \cos\gamma \quad (3)$$

$$V = \sqrt[2]{Vx^2 + Vy^2 + Vz^2} \quad (4)$$

  辅助轴的插补速度分两种情况：

  1. 当坐标轴上的点不动而单独移动辅助轴时，辅助轴按照Velocity设定的目标速度运动。
  2. 当坐标轴上的点移动时，辅助轴和将和坐标轴上的点同时到达目标位置。假设插补直线的长度为L1，辅助轴的目标位移为L2，某一个时刻插补直线的速率为V0，则辅助轴的速度Va计算方式如下：

$$Va = V0 \times \frac{L2}{L1} \quad (5)$$

- 相对绝对模式选择
  AbsRelMode = 0表示相对定位模式。触发本指令后三个坐标轴轴组最终运动到(Position[0],Position[1],Position[2])指定的位置，辅助轴运动到Position[3]指定的位置。

  AbsRelMode = 0表示相对定位模式。设轴组的三个坐标轴的位置为(Px,Py,Pz)，辅助轴当前位置为Pa，触发本指令后三个坐标轴最终运动到(Px+ Position[0],Py+ Position[1],Pz+ Position[2])。辅助轴最终位置为Pa+Position[3]。

- 缓冲与过渡
  缓冲与过渡的可选模式有以下四种，具体用法请参见《H5U系列可编程逻辑控制器编程与应用手册》中的“插补功能”章节。

序号	缓冲模式	描述
0	打断 + 无过渡	立即切换到下一个功能块动作，无过渡曲线
1	缓冲 + 无过渡	第一段减速完成开始执行缓冲的功能块，无过渡曲线
2	前一个速度 + 无过渡	以当前速度走到第一段结束并按照第一段的速率开始执行第二段
3	附加角过渡	有过渡曲线，在第一段开始执行减速时加入第二段的加速

  当选择1、2、3这三种缓冲模式时，插补指令最多允许缓冲8条曲线。当指令进入缓冲状态时，Busy信号有效，当本条指令开始执行时，Active输出有效，当指令执行完成时，Done信号输出有效。

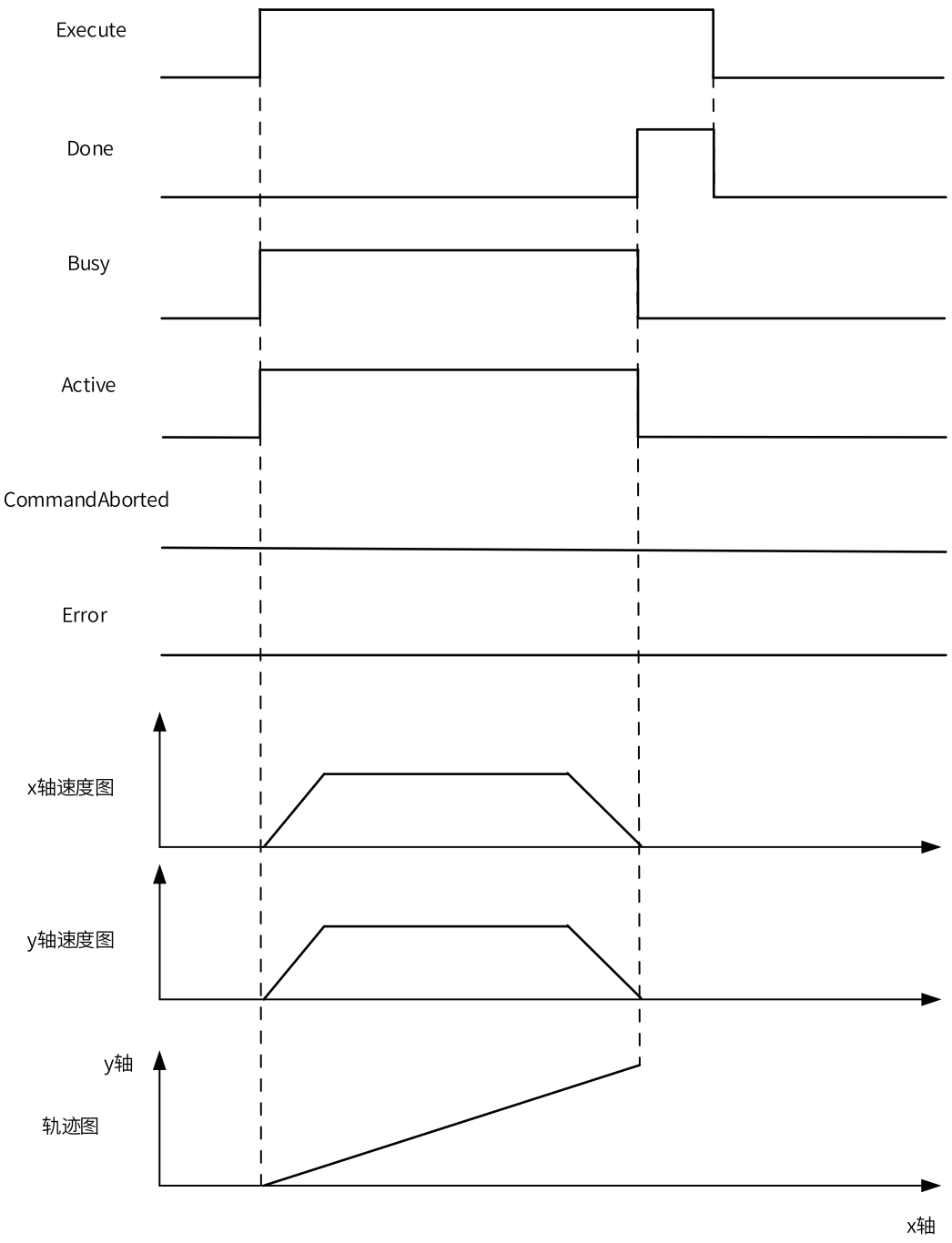
  当新加入的插补指令选择0（打断 + 无过渡）模式时，本条指令将打断正在执行的和处于缓冲的所有插补指令，被打断的插补指令CommandAborted输出有效。

- 重启本指令
  无法重启本指令。

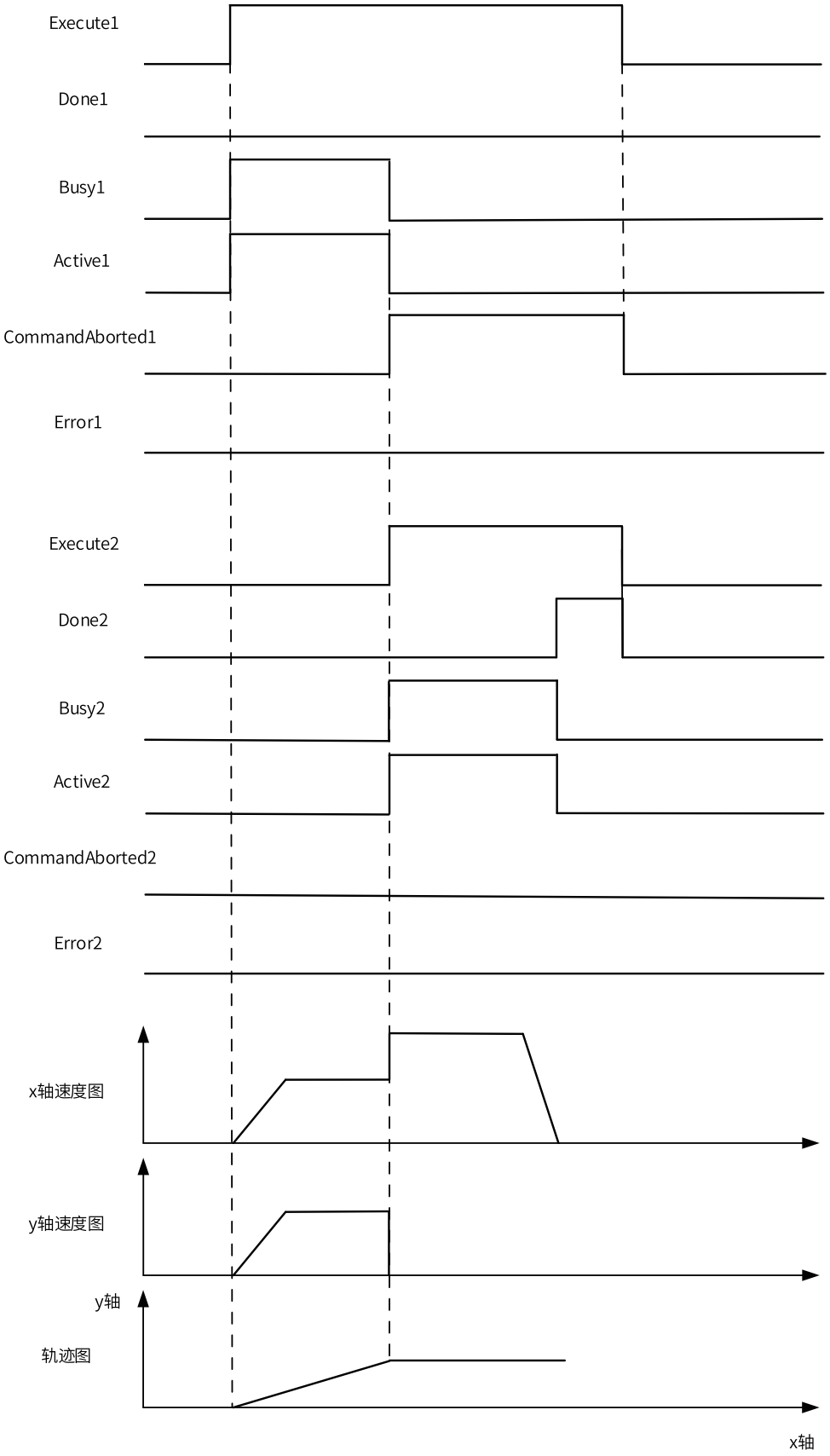
  在Busy输出为TRUE时重复触发本指令，轴将报故障9421（重复触发指令故障），同时所有轴立即停止运动，进入ErrorStop状态。

时序图

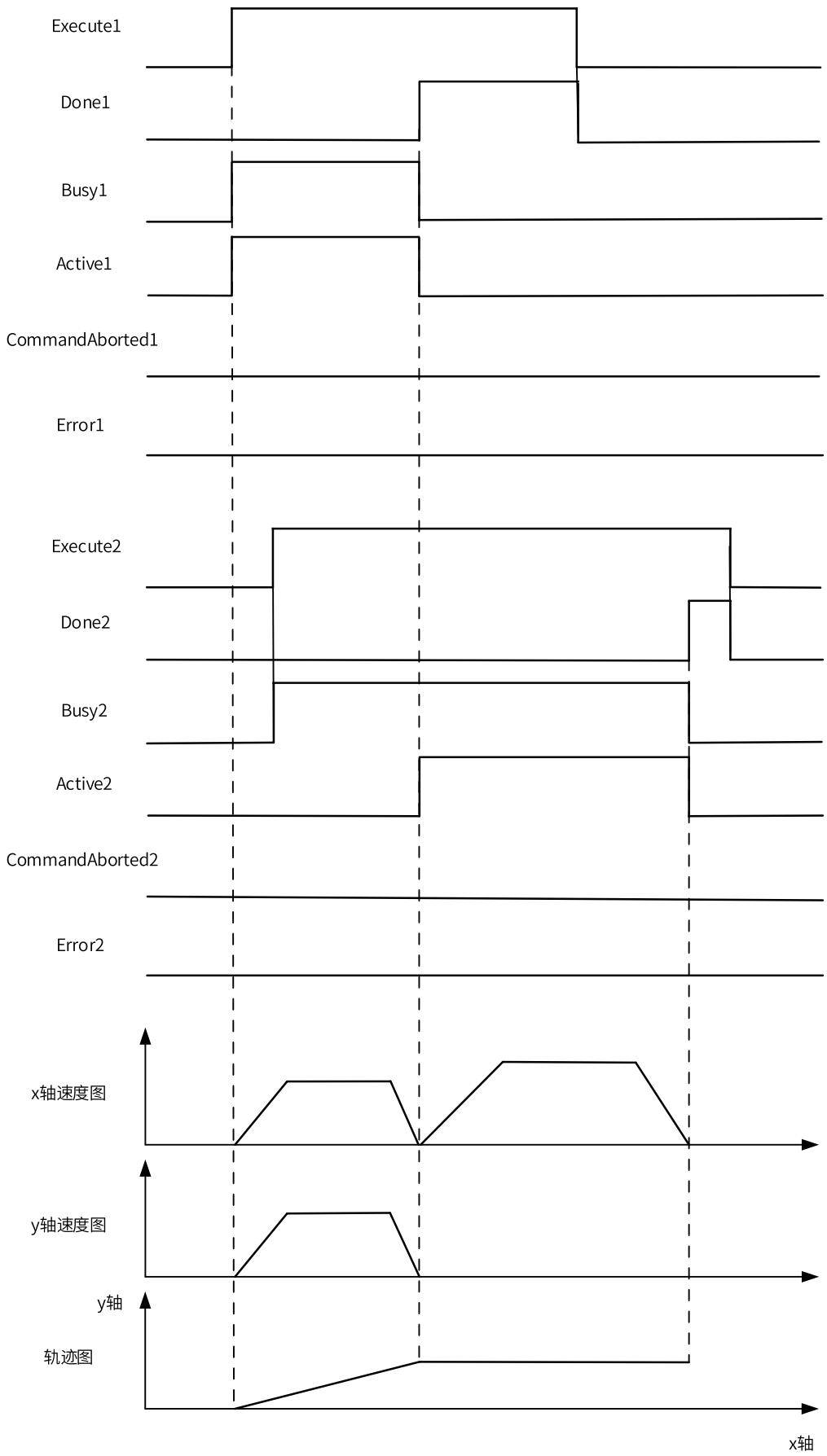
- 调用一个直线插补指令，在X轴、Y轴方向做插补。



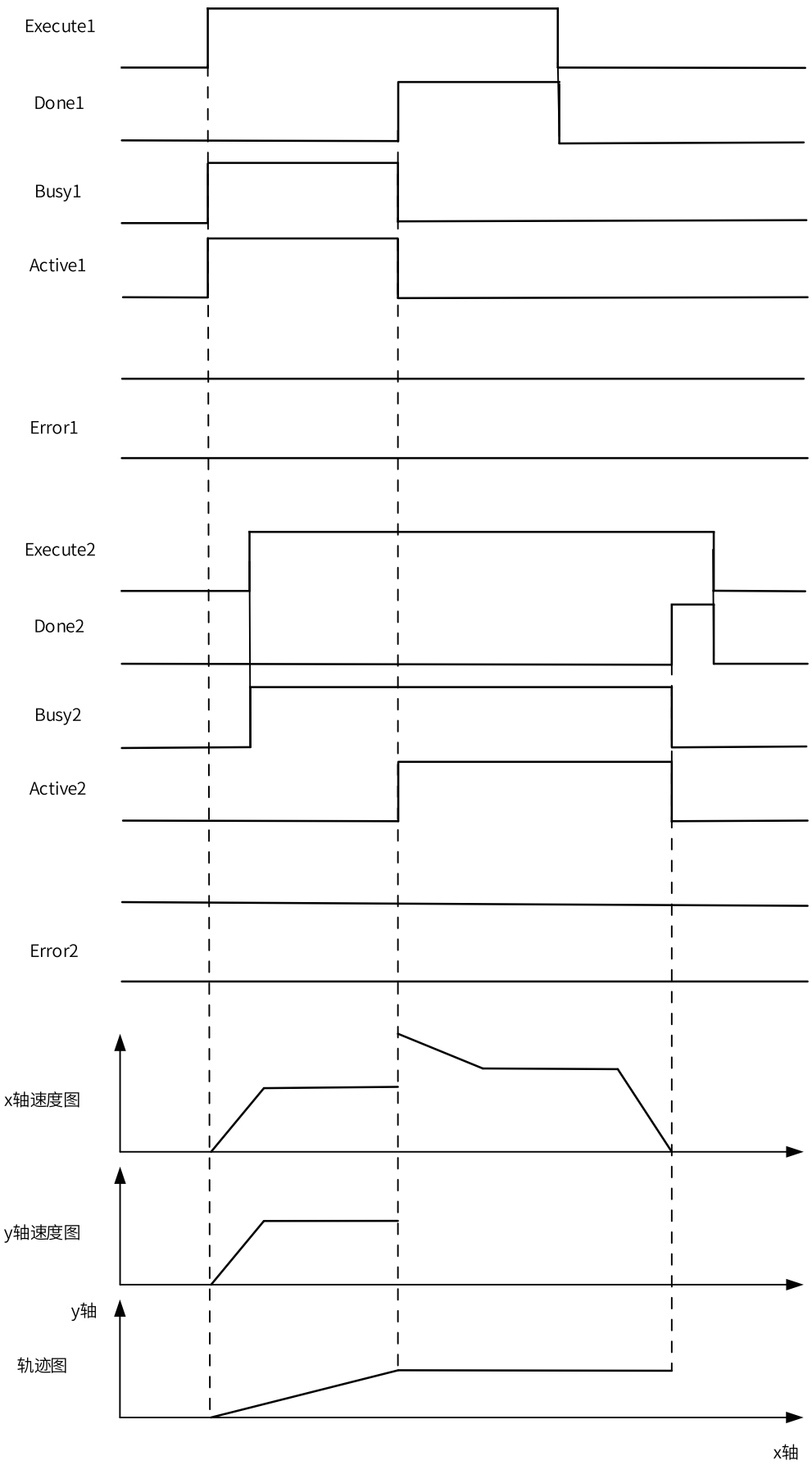
- 调用两个直线插补指令，在第一个插补指令运行过程中触发第二个插补指令，打断第一个插补指令。



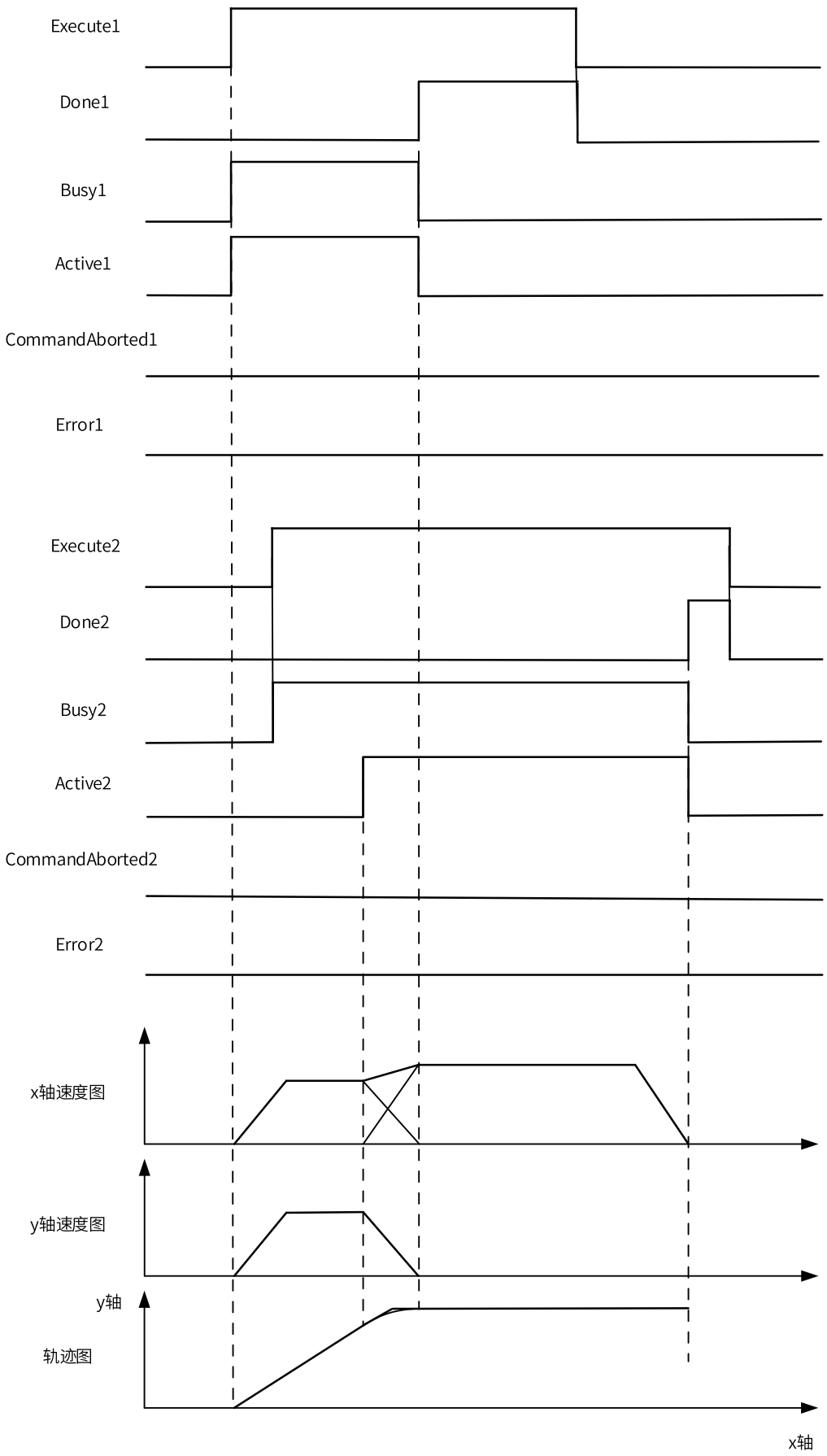
- 调用两个直线插补指令，第二条指令采用“缓冲+无过渡”模式执行。



- 调用两个直线插补指令，第二条指令采用“前一个速度+无过渡”模式执行。



- 调用两个直线插补指令，第二条指令采用“附加角过渡”模式执行。


Execute1
Done1
Busy1
Active1
CommandAborted1
Error1
Execute2
Done2
Busy2
Active2
CommandAborted2
Error2
x轴速度图
y轴速度图
y轴
轨迹图
x轴

3.15.3 MC_MoveCircular

MC_MoveCircular— 圆弧插补指令

图形块

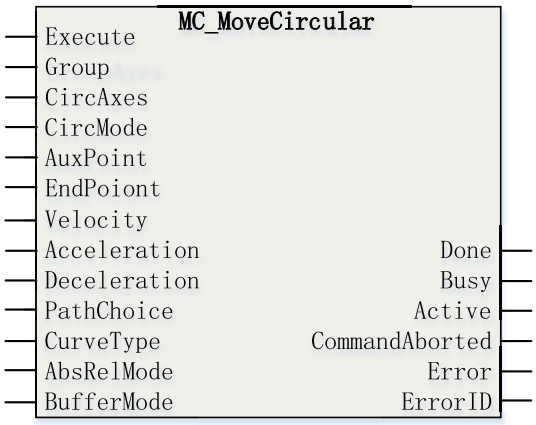


表3-268 指令表格式

16位指令	—					
32位指令	MC_MoveCircular 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Group	轴ID	否	-	0-32767	INT
S2	CircAxes	圆弧轴指定 0：x-y轴平面 1：y-z轴平面 2：x-z轴平面	是	0	0-2	INT
S3	CircMode	圆弧插补模式 0：指定为通过点 1：指定为中心点 2：指定为半径	是	0	0-2	INT
S4	AuxPoint	辅助点	否	-	正数 负数 0	REAL[0-3]
S5	EndPoint	终点	否	-	正数 负数 0	REAL[0-3]或 _sMC_ GROUP_POS
S6	Velocity	目标速度	否	-	正数	REAL
S7	Acceleration	加速度	否	-	正数	REAL
S8	Deceleratio n	减速度	是	加速度	正数	REAL
S9	PathChoice	路径选择 0：CW 1：CCW	是	0	0-1	INT

S10	CurveType	速度曲线类型 0：T型速度曲线 其他：T型速度曲线	是	0	0	INT
S11	AbsRelMode	绝对定位与相对定位模式 0：绝对定位 1：相对定位	是	0	0-1	INT
S12	BufferMode	缓冲模式 0：打断 + 无过渡 1：缓冲 + 无过渡 2：前一个速度 + 无过渡 3：附加角过渡	是	0	0-3	INT
D1	Done	目标位置到达 目标位置到达后为True	是	FALSE	TRUE FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE FALSE	BOOL
D3	Active	控制中 开始执行本段曲线时为True	是	FALSE	TRUE FALSE	BOOL
D4	CommandAborted	终止执行	是	FALSE	TRUE FALSE	BOOL
D5	Error	指令故障标志	是	FALSE	TRUE FALSE	BOOL
D6	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第391页“3.15.6 故障码”*。

功能和指令说明

MC_MoveCircular指令用于控制轴组直线插补功能，上升沿有效。

- 轴号的设定
  在Execute输入的上升沿锁存Group。

  在Execute=TRUE期间修改Group无效。

  在Execute=FALSE期间修改Group无效。

- 与单轴控制指令的关系
  只有调用MC_Power指令将轴切换到StandStill状态才可以触发本指令。

  在单轴运动期间（例如正在做点位运动、力矩控制、回原、停止等）触发本指令无效。

  触发本指令后单轴的PLCOpen状态机处于同步运动模式，运动期间不可以用单轴运动类指令打断，插补曲线完成后单轴的PLCOpen状态机进入StandStill状态，此时可以执行单轴的运动类指令。

- 圆弧轴的指定
  CircAxes指定坐标轴平面。意义如下：

  ircAxes = 0代表选择x-y坐标轴平面。AxisID_x和AxisID_y指定的运动轴执行圆弧插补，AxisID_z和AxisID_a指定的轴为辅助轴，执行直线插补。

  CircAxes = 1代表选择y-z坐标轴平面。AxisID_y和AxisID_z指定的运动轴执行圆弧插补，AxisID_x和AxisID_a指定的轴为辅助轴，执行直线插补。

  CircAxes = 2代表选择x-z坐标轴平面。AxisID_x和AxisID_z指定的运动轴执行圆弧插补，AxisID_y和AxisID_a指定的轴为辅助轴，执行直线插补。

- 插补模式选择

  1. CircMode = 0代表根据通过点进行圆弧插补。

     选择x-y平面时通过点为(AuxPoint[0], (AuxPoint[1])，终点为(EndPoint[0], EndPoint[1]); 选择y-z平面时通过点为(AuxPoint[1], (AuxPoint[2])，终点为(EndPoint[1], EndPoint[2])；选择x-z平面时通过点为(AuxPoint[0], (AuxPoint[2])，终点为(EndPoint[0], EndPoint[2])。

     以x-y平面为例，x轴的起始位置为Px，y轴的起始位置为Py，触发指令后将执行以(Px, Py)为起点，以(EndPoint[0], EndPoint[1])为终点并通过点(AuxPoint[0], (AuxPoint[1])的圆弧插补。

     当起点和终点为同一点时，以起点(Px, Py)和通过点(AuxPoint[0], (AuxPoint[1])为直径绘制正圆。这种情况下，通PathChoice (路径选择) 指定圆弧的旋转方向。

     当起点、通过点与终点在同一条直线上时不能构成圆，指令报错，停止插补指令的执行。

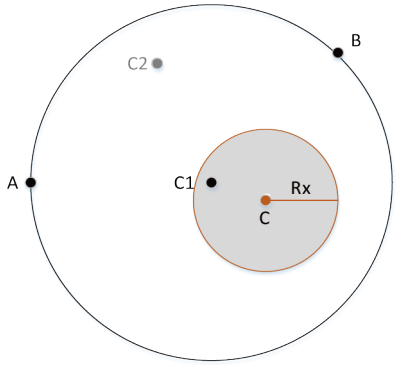
     当通过点与终点为同一点或者起点和通过点位置为同一点时，指令报错，停止插补指令的执行。
  2. CircMode = 1代表根据中心点进行圆弧插补。

     选择x-y平面时中心点为(AuxPoint[0], (AuxPoint[1])，终点为(EndPoint[0], EndPoint[1])；择y-z平面时中心点为(AuxPoint[1], (AuxPoint[2])，终点为(EndPoint[1], EndPoint[2])；择x-z平面时中心点为(AuxPoint[0], (AuxPoint[2])，终点为(EndPoint[0], EndPoint[2])；

     以x轴和z轴为例，x轴的起始位置为Px，z轴的起始位置为Pz，触发指令后将执行以(Px, Pz)为起点，以 (AuxPoint[0], (AuxPoint[2]) 为圆心，以 (EndPoint[0], EndPoint[2])为终点的圆弧插补，圆弧的绘制旋转方向PathChoice决定。

     当从指定的中心位置(AuxPoint[0], (AuxPoint[1])到起点(Px, Pz)的距离R1与到终点(EndPoint[0], EndPoint[2])的距离R2不同时(R1与R2的差值大于1)，将用R1和R2计算出平均值R，并按照与半径指定同样的方法计算中心位置（Cx,Cy），使用该半径和中心位置画圆弧。

     需要注意的是，在重新调整中心位置时，如果算出两个圆心，首先计算出两个圆心到设定中心点的距离，选出距离短的那个且该点必须位于以(AuxPoint[0], (AuxPoint[1])为圆心以AuxPoint[3]为半径的圆的内部。如下图所示，重新调整时以C1点为新的圆心。



3. CircMode = 2代表根据指定半径进行圆弧插补。

   不管选择哪个平面，圆弧的半径大小始终由|AuxPoint[0]|决定。选择x-y平面时终点为(EndPoint[0], EndPoint[1])；择y-z平面时终点为(EndPoint[1], EndPoint[2])；择x-z平面时终点为(EndPoint[0], EndPoint[2])；

   以y轴和z轴为例，y轴的起始位置为Py，z轴的起始位置为Pz，y轴和z轴执行以(Py, Pz)为起点，以|AuxPoint[0]|为半径，以 (EndPoint[1], EndPoint[2])为终点的圆弧插补。

   半径符号为负时，绘制出较长的圆弧；半径符号为正时，绘制出较短的圆弧。圆弧的旋转方向通过PathChoice(路径选择) 指定。

- 定位模式的选择

  1. 绝对模式

     当选择通过点时，辅助点和终点代表坐标系中的绝对点。

     当选择中心点时，中心点和终点代表坐标系中的绝对点。

     当选择半径时，终点代表坐标系中的绝对点。

  2. 相对模式

     当选择通过点时，辅助点和终点代表相对于起始点的相对点。

     当选择中心点时，中心点和终点代表相对于起始点的相对点。

     当选择半径时，终点代表相对于起始点的相对点。

- 缓冲与过渡

  缓冲与过度的可选模式有四种，具体用法请参见第页 “”*《H5U系列可编程逻辑控制器编程与应用手册》*中的“插补功能”章节。

  当选择1、2、3这三种缓冲模式时，插补指令最多允许缓冲8条曲线。当指令进入缓冲状态时，Busy信号有效，当本条指令开始执行时，Active输出有效，当指令执行完成时，Done信号输出有效。

  当新加入的插补指令选择0（打断 + 无过渡）模式时，本条指令将打断正在执行的和处于缓冲的所有插补指令，被打断的插补指令CommandAborted输出有效。

- 重启本指令

  无法重启本指令。

  在Busy输出为TRUE时重复触发本指令，轴将报故障9421（重复触发指令故障），同时所有轴立即停止运动，进入ErrorStop状态。

时序图

参考直线插补指令。

3.15.4 MC_GroupStop

MC_GroupStop— 停止轴组运行指令

图形块

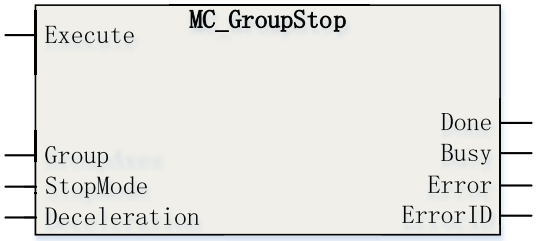


表3–269 指令表格式

16位指令	—					
32位指令	MC_MoveStop连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Group	轴ID	否	-	0-32767	INT
S2	StopMode	停止方式 0：减速停止 1：立即停止	是	1	0-1	INT
S3	Deceleration	减速度	是	1000	正数，小于最大加速度	REAL
D1	Done	停机完成	是	FALSE	FALSE TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE TRUE	BOOL
D3	Error	指令故障标志	是	FALSE	FALSE TRUE	BOOL
D4	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第391页“3.15.6 故障码”*章节“故障码”。

功能和指令说明

MC_GroupStop指令用于轴组实现停止功能，指令上升沿有效。

- 轴号的设定
  在Execute输入的上升沿锁存Group。

  在Execute=TRUE期间修改Group无效。

在Execute=FALSE期间修改Group无效。

- 生效范围
  MC_GroupStop指令仅能停止插补类指令（如MC_MoveLinear），不能停止单轴运动类指令（如MC_MoveAbsolute）。

  MC_Stop指令仅能停止单轴类运动指令，不能停止插补类指令。

- 状态转换
  在Execute的上升沿，插补器根据StopMode设置的停机方式执行停机，并打断所有处于缓冲状态的插补指令，停机完成后Done信号输出有效，单轴的PLCOpen状态机仍处于Synchronized Motion状态。

  在Execute=TRUE期间，插补器一直处于停止状态，此时触发新的插补指令无效。

  在Execute的下降沿，插补器将切换到非停止状态，单轴进入StandStill状态，此时可以触发新的插补指令。

- 停止模式
  StopMode设置为0时，将按照Deceleration设定的减速度减速停机；

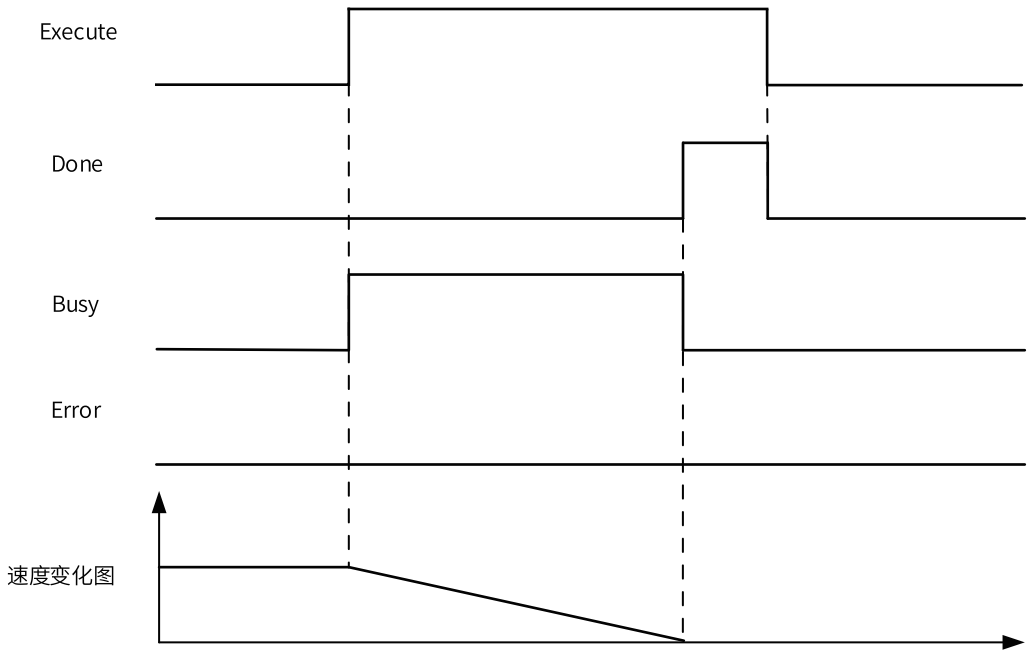
  StopMode设置为1时，将立即停机，无减速过程。

- 重启本指令
  在轴减速过程中再次触发本指令，轴组内的轴将按照新的减速度进行减速。

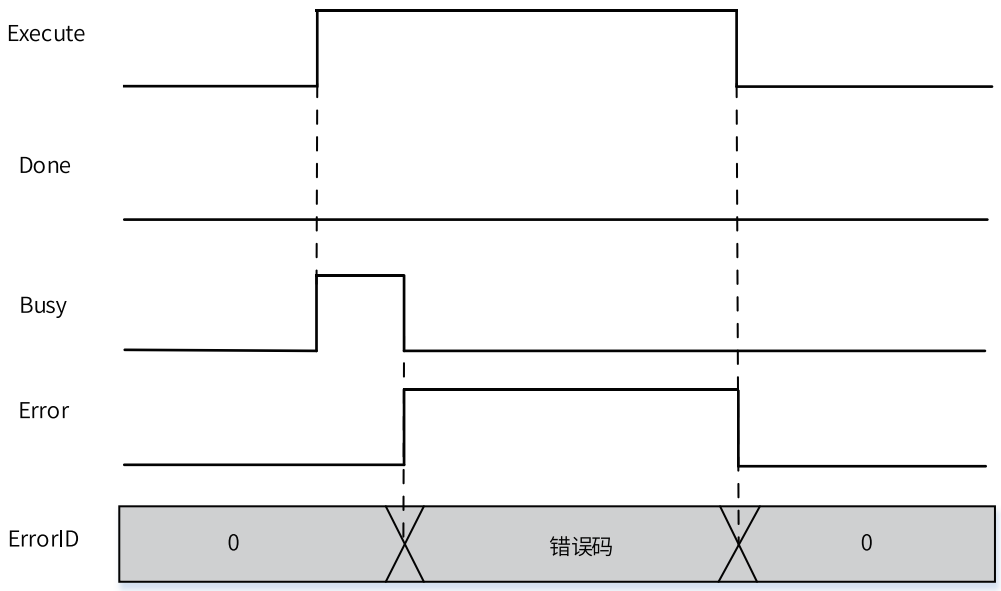
- 多重调用
  不允许多重调用本指令。当有MC_GroupStop指令正在执行且Execute输入为TRUE时触发其他的MC_GroupStop指令，新触发的MC_GroupStop指令报错9441（轴处于停止状态）。

时序图

- 有减速过程，能正常停止时。



- 减速过程中有轴发生故障时。



3.15.5 MC_GroupPause

MC_GroupPause— 暂停轴组运行指令

图形块

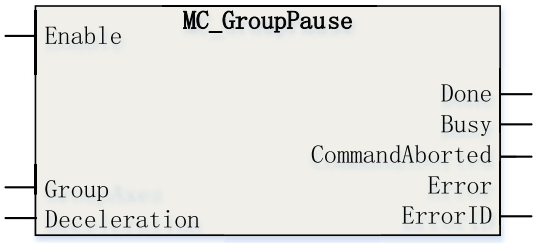


表3–270 指令表格式

16位指令	—					
32位指令	MC_MoveCircular 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	Group	轴组	否	-	0-32767	INT
S2	Deceleration	减速度	是	1000	正数，小于最大加速度	REAL
D1	Done	暂停完成	是	FALSE	FALSE TRUE	BOOL
D2	Busy	忙标志	是	FALSE	FALSE TRUE	BOOL
D3	CommandAborted	打断标志	是	FALSE	FALSE TRUE	BOOL
D3	Error	指令故障标志	是	FALSE	FALSE TRUE	BOOL
D4	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第391页“3.15.6 故障码”。

功能和指令说明

MC_GroupPause指令用于轴组实现暂停功能，指令上升沿有效。

- 轴号的设定
  在Execute输入的上升沿锁存Group。

  在Execute=TRUE期间修改Group无效。

  在Execute=FALSE期间修改Group无效。

- 生效范围
  MC_GroupPause指令仅能暂停插补类指令（如MC_MoveLinear），不能停止单轴运动类指令（如MC_MoveAbsolute）。

- 状态转换
  在Enable的上升沿，插补器根据Deceleration设置的减速度执行减速过程，减速完成后Done信号输出有效，单轴的PLCOpen状态机仍处于Synchronized Motion状态，暂停期间正在执行的插补指令的Busy信号和Valid信号保持输出有效。

  在Enable=TRUE期间，插补器一直处于暂停状态，此时触发新的插补指令新的插补指令处于缓冲状态。

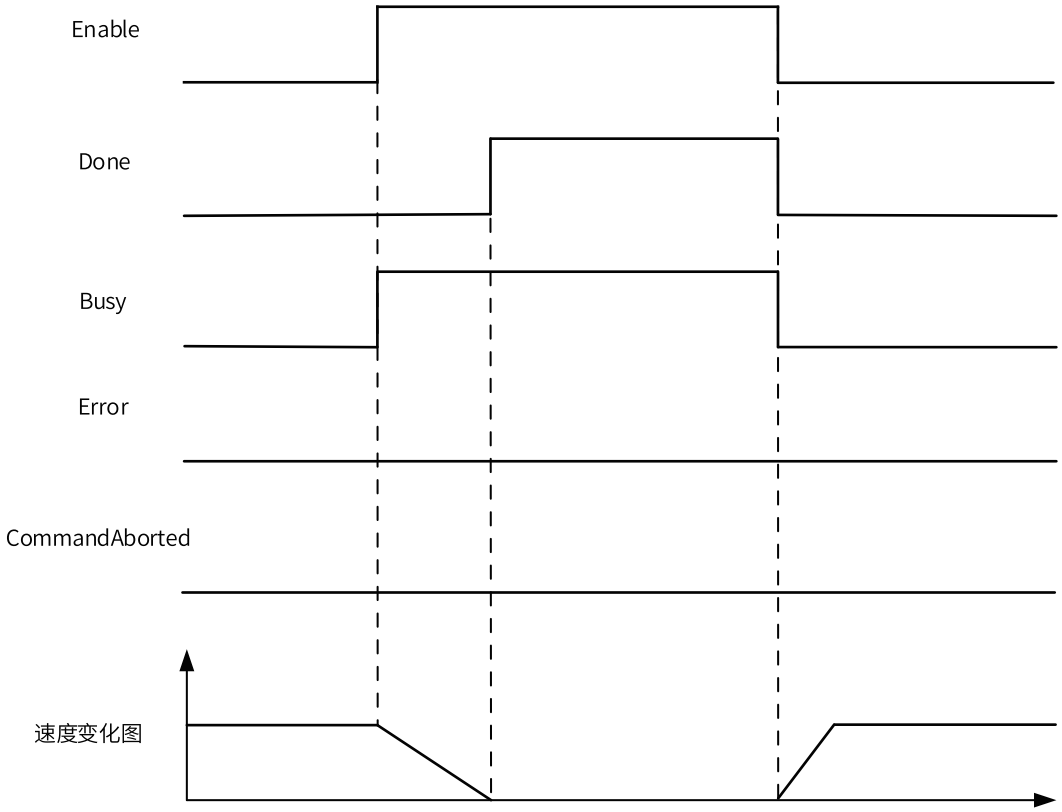
  在Enable的下降沿，插补器重新开始执行之前被暂停的插补指令。

- 重启本指令
  在轴减速过程中再次触发本指令，轴组内的轴将按照新的减速度进行减速。

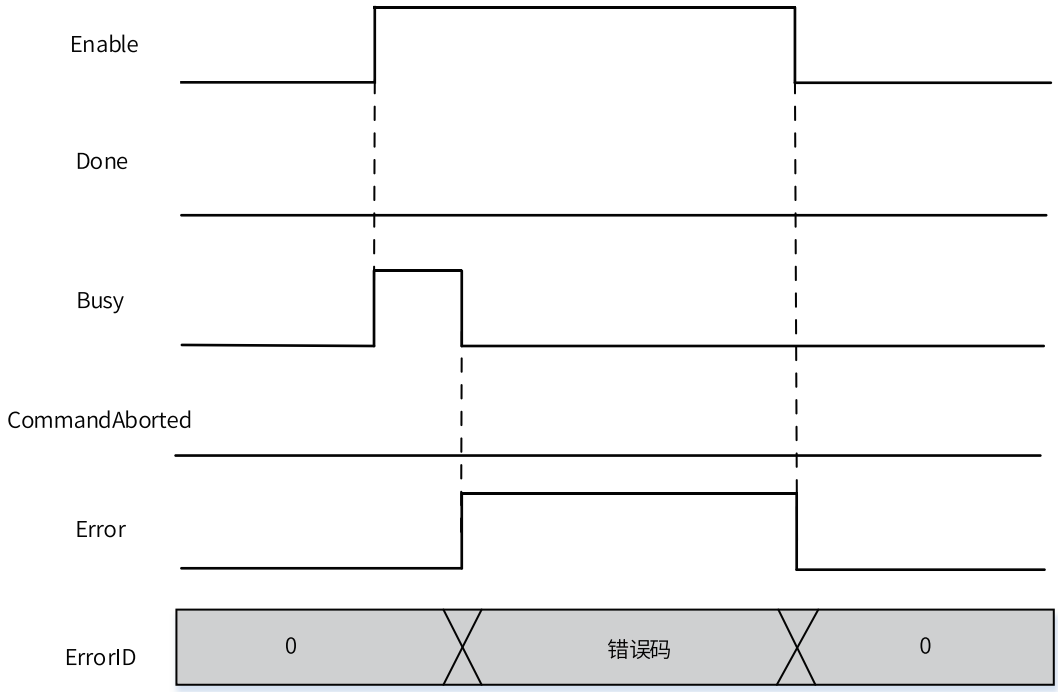
- 多重调用
  在第一条MC_GroupPause指令执行期间触发第二条MC_GroupPause指令，第一条暂停指令被打断，插补器按照第二条指令的减速度开始减速。

时序图

- 有减速过程，能正常停止时。



- 减速过程中有轴发生故障时。



3.15.6 故障码

使用插补功能发生故障时，请查阅本节所列故障码进行故障排查。

故障码	说明	解决方法
9400	轴组数量超过最大值	检查轴组数量是否大于8
9401	轴组内有轴处于故障状态	检查轴组中的轴是否进入Errorstop状态 根据单个轴的故障码排错
9402	缓冲的插补指令数量大于8条	插补指令最多缓冲8条，查看缓冲的指令是否超过了8条
9403	轴重复使用	查找到已经被使用的轴并更换成未被使用的轴
9404	不能构成轴组	X轴和y轴不能为空 请检查X轴或者Y轴是否不存在或者为空
9405	指定的Z轴不存在	检查AxisID_z参数指定的轴是否存在
9406	指定的辅助轴不存在	检查AxisID_a参数指定的轴是否存在
9407	轴组ID重复使用	检查GroupID是否重复
9408	轴配置失败	检查轴组中的是否有轴配置失败，如果是，则检查单板软件和后台是否匹配
9409	轴ID小于0	检查轴组中指定的轴的ID是否小于0
9410	短时间内重复触发同一条MC_SetAxesGroup指令，导致轴组没有释放	不允许在MC_SetAxesGroup的busy信号输出有效期间重复触发该指令
9411	MC_GroupStop指令被打断	查看是否在指令有效期间调用的比该指令优先等级更高的指令，导致指令被打断
9412	圆弧插补指令CircAxes超范围	检查圆弧插补指令CircAxes是否超范围
9413	圆弧插补指令CircMode超范围	检查圆弧插补指令CircMode是否超范围
9414	圆弧插补指令PathChoice超范围	检查圆弧插补指令PathChoice是否超范围
9415	停止指令StopMode超范围	检查停止指令StopMode是否超范围
9416	X坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9417	y坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9418	z坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9419	辅助轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9420	圆弧插补指令重复触发	同一条圆弧插补指令在Busy输出有效期间不允许重复触发
9421	直线插补指令重复触发	同一条直线插补指令在Busy输出有效期间不允许重复触发
9422	获取轴组失败	检查GroupID指定的轴组是否已经通过MC_SetAxesGroup创建
9423	轴配置失败	在轴还没有配置完成的情况下触发了指令。 检查轴组内的所有轴的通讯状态是否处于“Axis ready”状态。
9424	有轴处于非使能状态	在Disabled 状态下不允许调用插补指令
9425	有轴正在执行单轴类运动指令	当有轴正在执行单轴运动类指令而不处于StandStill状态时不允许调用插补指令
9426	有轴处于Stopping状态	当有轴执行了MC_Stop指令而处于Stopping状态时不允许调用插补指令
9427	轴组处于停止状态	MC_GroupStop指令有效，不允许执行插补指令
9428	有轴处于原点回归状态	当有轴执行了MC_Home指令而处于Homing状态时不允许调用插补指令
9429	有轴正在执行设置位置指令	当有轴执行了MC_SetPosition指令正在设置当前位置时不允许调用插补指令
9430	有轴处于调试状态	当有轴正在处于调试状态时不允许调用插补指令
9431	插补中有轴进入调试状态，打断其他轴的执行	插补过程中有轴进入了调试状态，打断了其他轴的运行

故障码	说明	解决方法
9432	内存申请失败	检查内存是否用完 联系厂家
9433	目标速度为0或小于0	指令的目标速度必须大于0
9434	目标加速度为0或小于0	指令的目标加速度必须大于0
9435	目标减速度为0或小于0	指令的目减标速度必须大于0
9436	曲线类型设置超范围	插补指令只支持T型曲线，检查是否设置了其他值。
9437	运动模式（AbsRelMode）设置不合理	仅支持相对定位和绝对定位两中模式，检查是否设置了其他值。
9438	缓冲模式(BufferMode)设置不合理	检查缓冲模式参数设置是否超范围
9439	插补模式(InsertMode)设置不合理	检查插补模式参数设置是否合理
9440	有轴处于故障状态导致停机	查找处于故障状态的轴，根据单轴故障码解决轴的故障
9441	重复调用MC_GroupStop指令	在一条MC_GroupStop指令有效期间不能重复触发或多次调用其他MC_GroupStop指令
9442	数据缓冲区非空	内部故障，联系厂家
9443	不能构成圆弧	-
9444	圆弧插补指令的起点、经过点和终点重合，不能构成圆	检查圆弧指令的给定参数，保证起点、经过点和终点可以构成一个圆
9445	指令缓冲区满	内部故障，联系厂家
9446	X轴超过最大速度	按照当前参数，x轴坐标轴的目标分速度将超过最大速度
9447	y轴超过最大速度	按照当前参数，y轴坐标轴的目标分速度将超过最大速度
9448	z轴超过最大速度	按照当前参数，z轴坐标轴的目标分速度将超过最大速度
9449	辅助轴超过最大速度	按照当前参数，辅助轴的目标分速度将超过最大速度
9450	获取轴组数量数量失败	后台版本过低，更实用最新发布的版本
9451	内部故障	联系厂家
9452	StandStill状态不允许调用本指令	本指令在StandStill状态下不允许调用
9453	超过最大速度	指令的目标速度超过轴组组态界面中的最大速度
9454	超过最大加（减）速度	指令中的目标加（减）速度超过组态界面中的最大加速度
9455	直线插补指令报错导致轴组故障	查找第一个报错的直线插补指令并根据插补指令的故障码进一步确认故障后解决
9456	圆弧插补指令报错导致轴组故障	查找第一个报错的圆弧插补指令并根据插补指令的故障码进一步确认故障后解决
9457	轴组停止指令报错导致轴组故障	查找第一个报错的轴组停止指令并根据指令的故障码进一步确认故障后解决
9458	轴组暂停指令报错导致轴组故障	查找第一个报错的轴组暂停指令并根据指令的故障码进一步确认故障后解决

3.16 CANopen轴控指令

3.16.1 指令列表

CANopen轴控指令涵盖以下指令条目：

指令类别	名称	功能
CANopen轴控指令	MC_Power_CO	通讯控制伺服轴使能
	MC_Reset_CO	通讯控制伺服轴故障复位
	MC_ReadActualPosition_CO	通讯控制读取轴当前实际位置
	MC_ReadActualVelocity_CO	通讯控制读取轴当前实际速度
	MC_Halt_CO	通讯控制伺服轴终止运动
	MC_Stop_CO	通讯控制伺服轴停止
	MC_MoveAbsolute_CO	通讯控制轴绝对定位
	MC_MoveRelative_CO	通讯控制轴相对定位
	MC_MoveVelocity_CO	通讯控制轴速度运行模式
	MC_Jog_CO	通讯控制轴点动
	MC_Home_CO	通讯控制轴原点回归
	MC_WriteParameter_CO	通讯控制写入轴参数
	MC_ReadParameter_CO	通讯控制读取轴参数

3.16.2 MC_Power_CO

控制伺服轴使能或解除使能。

MC_Power_CO — 通讯控制伺服轴使能

图形块

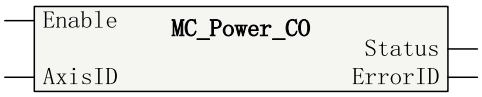


表3–271 指令列表格式

16位指令	MC_Power_CO 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
D1	Status	轴状态	是	FALSE	TRUE/FALSE	BOOL
D2	ErrorID	错误码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*章节“*CANopen轴控指令错误代码*”。

表3–272 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	√	-	-
D1	√[1]	-	√	-	√	√	-	-	-
D2	-	-	√	√	√	√	-	-	-

说明

- [1]不支持X元件。
- 每个轴仅可以使用一次MC_Power_CO指令。

功能和指令说明

轴号：指定控制轴的编号，范围：K1-K16。

轴状态：轴实际状态输出，ON表示轴已经使能，OFF表示轴未使能。

错误代码：请参考”指令错误代码定义”。

MC_Power_CO指令根据读取到的状态字（6041h），写入相应控制字（6040h），从而使轴进入使能状态。状态字（6041h）与控制字（6040h）的写入对应关系如下表所示：

能流状态	状态字（6041h）		控制字（6040h）	
ON	Not ready to switch on	xxxx xxxx x0xx 0000 b	Shutdown	0000 0000 0000 0110 b
	Switch on disabled	xxxx xxxx x1xx 0000 b		
	Ready to switch on	xxxx xxxx x01x 0001 b	Switch on	0000 0000 0000 0111 b
	Switched on	xxxx xxxx x01x 0011 b	Switch on + eable operation	0000 0000 0000 1111 b
	Fault reaction active	xxxx xxxx x0xx 1111 b	-	xxxx xx00 xx00 xxxx b
	Fault	xxxx xxxx x0xx 1000 b		
	其他		-	xxxx xxxx xxxx xxxx b
OFF	Ready to switch on	xxxx xxxx x01x 0001 b	Disable voltage	0000 0000 0000 0000 b
	Switched on	xxxx xxxx x01x 0011 b		
	Operation enabled	xxxx xxxx x01x 0111 b		
	其他		-	xxxx xx00 xx00 xxxx b

上表中，x表示任意值（状态字）或保持不变（控制字）。

3.16.3 MC_Reset_CO

复位轴相关错误，使轴进入“就绪”或“未使能”状态。

MC_Reset_CO — 通讯控制伺服轴故障复位

图形块

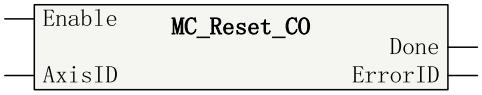


表3-273 指令列表格式

16位指令	MC_Reset_CO 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	ErrorID	错误代码	是	0	*1	INT

说明

*1请参见第414页“3.16.15 CANopen轴控指令错误代码”章节“CANopen轴控指令错误代码”。

表3-274 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	✓	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于复位CANopen总线轴的故障，使轴进入“就绪”或“未使能”状态。

轴号：指定控制轴的编号，范围：K1-K16。

完成：复位操作执行完成输出。

错误代码：请参考”指令错误代码定义”。

MC_Reset_CO指令根据读取到的状态字（6041h），写入相应控制字（6040h），从而复位轴故障。状态字（6041h）与控制字（6040h）的写入对应关系如下表所示：

能流状态	状态字（6041h）		控制字故障复位（6040h.bit7）
ON	Switch on disabled	xxxx xxxx x1xx 0000b	0
	Operation enabled	xxxx xxxx x01x 0111b	-
	Fault	xxxx xxxx x0xx 1000b	1
	-	其他	x
↓	-	xxxx xxxx xxxx xxxxb	0
OFF	-	xxxx xxxx xxxx xxxxb	x

上表中，x表示任意值（状态字）或保持不变（控制字）。

3.16.4 MC_ReadActualVelocity_CO

该指令读取轴当前的实际速度。

MC_ReadActualVelocity_CO — 通讯控制读取轴当前实际速度

图形块

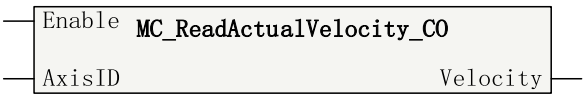


表3–275 指令列表格式

16位指令	—					
32位指令	MC_ReadActualVelocity_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
D1	Velocity	当前实际速度	是	0	-	REAL

表3–276 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
D1	-	-	✓	✓	✓	✓	-	-	-

指令和功能说明

该指令用于读取CANOpen总线轴的实际速度。

轴号：指定读取轴的编号，仅支持常数，范围：1-16。

位置：轴当前实际位置，32位浮点数。

3.16.5 MC_ReadActualPosition_CO

该指令可读取当前实际位置。

MC_ReadActualPosition_CO — 通讯控制读取轴当前实际位置

图形块

表3-277 指令列表格式

16位指令	—					
32位指令	MC_ReadActualPosition_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
D1	Position	当前实际位置	是	0	-	REAL

表3-278 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
D1	-	-	✓	✓	✓	✓	-	-	-

指令和功能说明

该指令用于读取CANOpen总线轴的实际位置。

轴号：指定读取轴的编号，仅支持常数，范围：1-16。

位置：轴当前实际位置，32位浮点数。

3.16.6 MC_Halt_CO

控制终止当前运动，完成后可以响应其他使轴运动的指令。

MC_Halt_CO — 通讯控制伺服轴终止运动

图形块

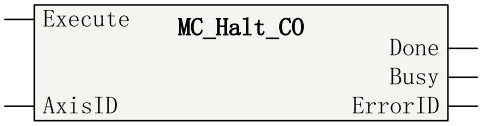


表3-279 指令列表格式

16位指令	MC_Halt_CO连续执行				
32位指令	—				
操作数	名称	描述	默认值	范围	数据类型
S1	Axis	轴号	-	-	INT
D1	Done	完成	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	故障码	-	-	INT

说明

*1请参见第414页“3.16.15 CANopen轴控指令错误代码”章节“CANopen轴控指令错误代码”。

表3-280 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	✓	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

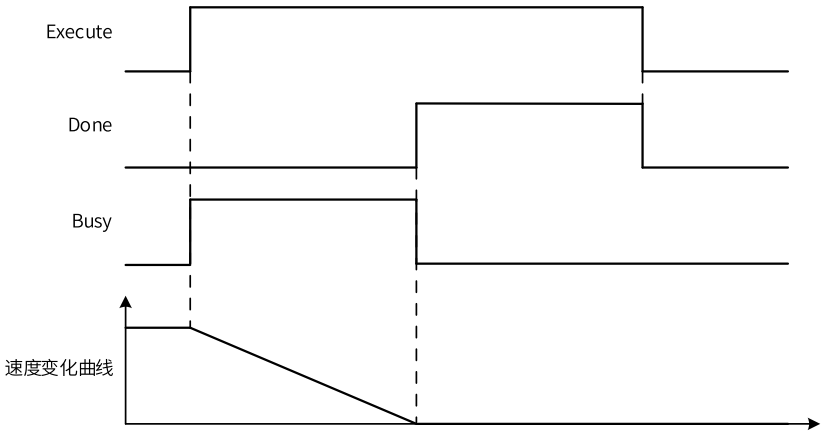
该指令用于控制CANOpen总线轴终止当前运动，完成后可以响应其他使轴运动的指令。

MC_Halt_CO指令可以被MC_MoveAbsolute_CO、MC_MoveRelative_CO、MC_MoveVelocity_CO、MC_Jog_CO打断。

表3-281 MC_Halt_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6040h.bit4 = 0	控制字触发运动停止 目标速度写零
	6040h.bit5 = 0	
	6040h.bit6 = 0	
	6040h.bit8 = 1	
	60FFh = 0	
2	606Ch = 0	等待停止完成
	6061h = 3 且 6041h.bit13 = 1	
	6061h != 3 且 6041h.bit10 = 1	
3	6060h = 1	切换到位置模式

时序图



3.16.7 MC_Stop_CO

控制轴停止并进入“停止”状态，不再响应任意使轴运动的指令。

MC_Stop_CO — 通讯控制伺服轴停止

图形块

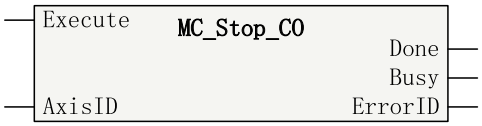


表3–282 指令列表格式

16位指令	MC_Stop_CO 连续执行					
32位指令	—					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	错误代码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*章节“*CANopen轴控指令错误代码*”。

表3–283 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S	-	-	-	-	-	-	✓	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

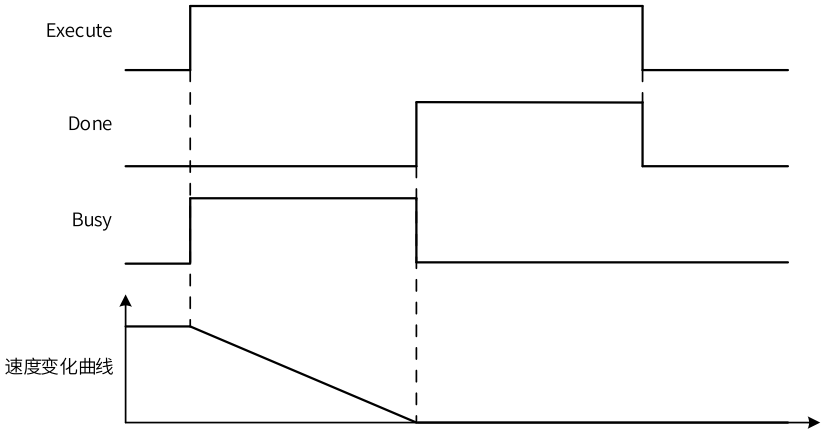
功能和指令说明

该指令用于控制CANOpen总线轴终止当前运动，并进入“停止”状态，不再响应任意使轴运动的指令。

表3-284 MC_Stop_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6040h.bit4 = 0	控制字触发运动停止 目标速度写零
	6040h.bit5 = 0	
	6040h.bit6 = 0	
	6040h.bit8 = 1	
	60FFh = 0	
2	606Ch = 0	等待停止完成
	6061h = 3 且 6041h.bit13 = 1	
	6061h != 3 且 6041h.bit10 = 1	
3	6060h = 1	切换到位置模式

时序图



3.16.8 MC_MoveVelocity_CO

MC_MoveVelocity_CO — 通讯控制轴速度运行模式

图形块

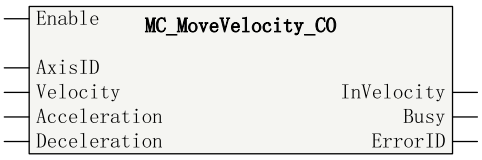


表3-285 指令列表格式

16位指令	—					
32位指令	MC_MoveVelocity_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	Velocity	速度	否	-	-	REAL
S3	Acceleration	加速度	否	-	-	REAL
S4	Deceleration	减速度	是	加速度	-	REAL
D1	InVelocity	速度到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*。

表3-286 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	✓	✓	✓	✓	-	-	-
S3	-	-	✓	✓	✓	✓	-	-	-
S4	-	-	✓	✓	✓	✓	-	-	-
S5	-	-	✓	✓	✓	✓	-	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令控制CANOpen总线轴轴以指定速度运动。当指定速度（Velocity）大于0时，轴正向运动，小于0时，轴反向运动。该指令支持运行中修改速度参数，并实时生效。

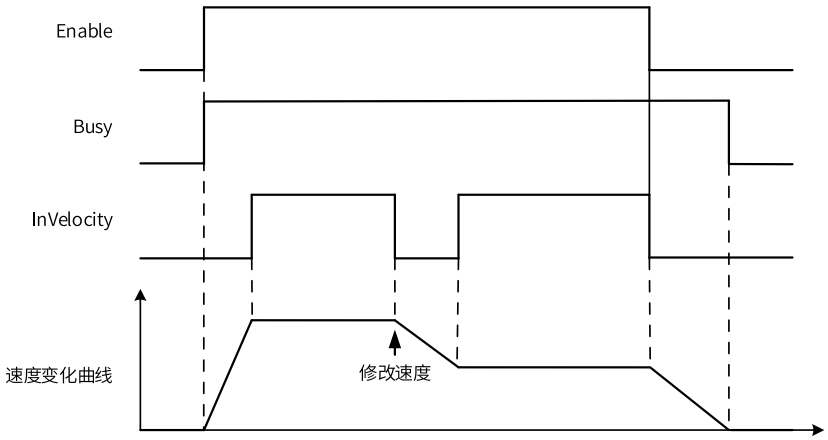
如果没有指定减速度（Deceleration），即减速度参数为空时，则默认减速度与指定的加速度（Acceleration）相等。

表3-287 MC_MoveVelocity_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6040h.bit8 = 0	复位控制字Halt位
2	6083h = 加速度	写加速度

步骤	操作/条件	说明
3	6084h = 减速度	写减速度
4	6060h = 3	切换到速度模式
5	6061h = 3	等待切换速度模式完成
6	60FFh = 目标速度	设置目标速度
	6041h.bit10 = 1	目标速度到达
	60FFh < 0 且6041h.bit11 = 1 且 60FDh.bit0 = 1： 60FFh = 0	负向运动遇负限位，运动结束
	607Ah > 0 且6041h.bit11 = 1 且 60FDh.bit1 = 1： 60FFh = 0	正向运动遇正限位，运动结束
	60FFh = 0	指令能流无效，运动结束

时序图



3.16.9 MC_MoveRelative_CO

MC_MoveRelative_CO— 通讯控制轴相对定位

图形块

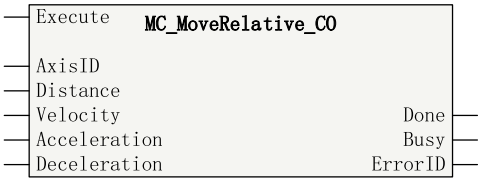


表3–288 指令列表格式

16位指令	—					
32位指令	MC_MoveRelative_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT

S2	Distance	目标距离	否	-	-	REAL
S3	Velocity	最大速度	否	-	-	REAL
S4	Acceleration	加速度	否	-	-	REAL
S5	Deceleration	减速度	是	加速度	-	REAL
D1	Done	完成，目标位置到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*章节“*CANopen轴控指令错误代码*”。

表3–289 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	✓	✓	✓	✓	-	-	-
S3	-	-	✓	✓	✓	✓	-	-	-
S4	-	-	✓	✓	✓	✓	-	-	-
S5	-	-	✓	✓	✓	✓	-	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令实现CANOpen总线轴相对定位功能，用于控制轴从当前位置开始运动指定距离。

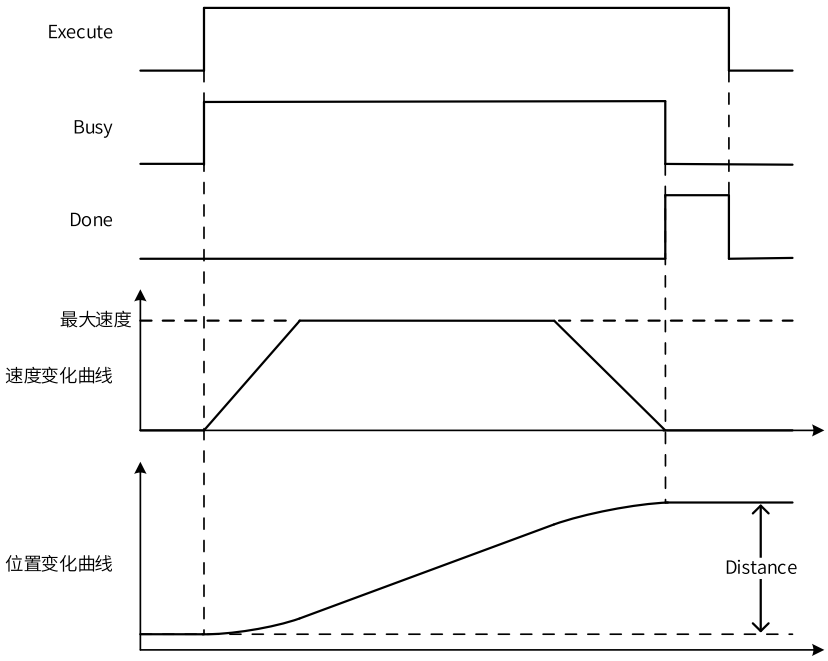
如果没有指定减速度（Deceleration），即减速度参数为空时，则默认减速度与指定的加速度（Acceleration）相等。

表3–290 MC_MoveRelative_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6060h = 1	切换到位置模式
2	6061h = 1	等待切换位置模式完成
3	6040h.bit5 = m	控制字写入相应模式。 缓存模式（参数编号：K1000）= 0，则m = 1；否则，m =0。
	6040h.bit6 = 1	
	6040h.bit8 = 0	
	6040h.bit9 = 0	

步骤	操作/条件	说明
4	607Ah = 位置	写（相对）目标位置、定位速度
	6081h = 速度	
5	6083h = 加速度	写加速度
6	6084h = 减速度	写减速度
7	6040h.bit4 = 1	触发定位
8	6041h.bit12 = 1	等待定位开始
9	6040h.bit4 = 0	复位定位触发
10	607Ah < 0 且6041h.bit11 = 1 且 60FDh.bit0 = 1	负向运动遇负限位，定位结束
	607Ah > 0 且6041h.bit11 = 1 且 60FDh.bit1 = 1	正向运动遇正限位，定位结束
	6041h.bit10 = 1 且 6041h.bit12 = 0	目标位置到达，定位完成

时序图



3.16.10 MC_MoveAbsolute_CO

MC_MoveAbsolute_CO — 通讯控制轴绝对定位

图形块

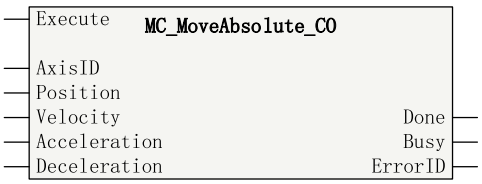


表3–291 指令列表格式

16位指令	—					
32位指令	MC_MoveAbsolute_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	Position	目标位置	否	-	-	REAL
S3	Velocity	最大速度	否	-	-	REAL
S4	Acceleration	加速度	否	-	-	REAL
S5	Deceleration	减速度	是	加速度	-	REAL
D1	Done	完成，目标位置到达	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*。

表3–292 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	✓	✓	✓	✓	-	-	-
S3	-	-	✓	✓	✓	✓	-	-	-
S4	-	-	✓	✓	✓	✓	-	-	-
S5	-	-	✓	✓	✓	✓	-	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

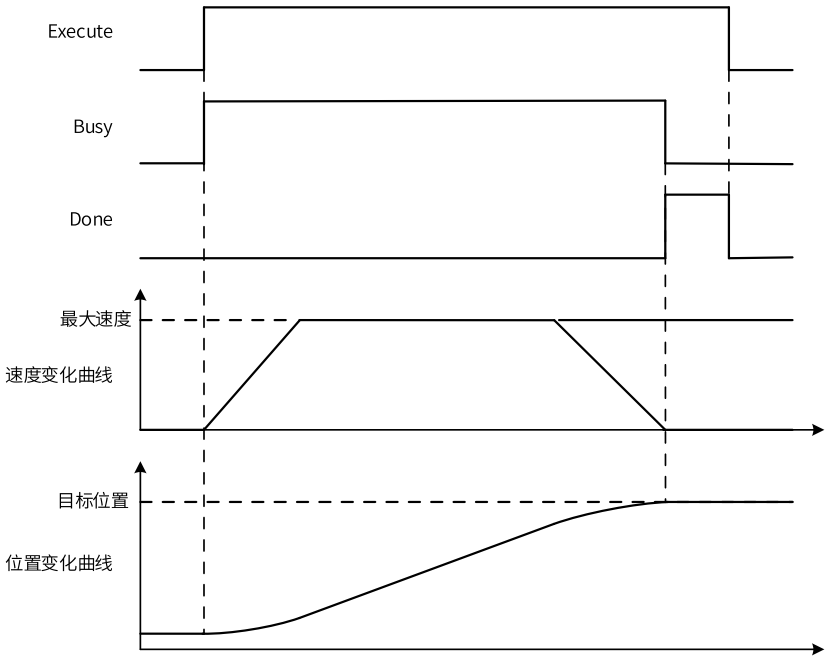
该指令实现CANOpen总线轴绝对定位功能，用于控制轴运动到指定位置。

如果没有指定减速度（Deceleration），即减速度参数为空时，则默认减速度与指定的加速度（Acceleration）相等。

表3-293 MC_MoveAbsolute_CO 指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6060h = 1	切换到位置模式
2	6061h = 1	等待切换位置模式完成
3	6040h.bit5 = m	控制字写入相应模式。 缓存模式（参数编号：K1000）= 0，则m = 1；否则，m = 0。
	6040h.bit6 = 0	
	6040h.bit8 = 0	
	6040h.bit9 = 0	
4	607Ah = 位置	写（绝对）目标位置、定位速度
	6081h = 速度	
5	6083h = 加速度	写加速度
6	6084h = 减速度	写减速度
7	6040h.bit4 = 1	触发定位
8	6041h.bit12 = 1	等待定位开始
9	6040h.bit4 = 0	复位定位触发
10	607Ah < 6064h 且6041h.bit11 = 1 且 60FDh.bit0 = 1	负向运动遇负限位，定位结束
	607Ah > 6064h 且6041h.bit11 = 1 且 60FDh.bit1 = 1	正向运动遇正限位，定位结束
	6041h.bit10 = 1 且 6041h.bit12 = 0	目标位置到达，定位完成

时序图



3.16.11 MC_Home_CO

MC_Home_CO — 通讯控制轴原点回归

图形块

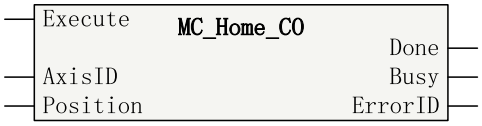


表3-294

16位指令	—					
32位指令	MC_Home_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	Position	原点回归后的目标位置	是	0	-	REAL
D1	Done	完成	是	FALSE	TRUE/FALSE	BOOL
D2	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D3	ErrorID	故障码	是	0	*1	INT

说明

*1请参见第414页“3.16.15 CANopen轴控指令错误代码”章节“CANopen轴控指令错误代码”。

表3-295 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	✓	✓	✓	✓	-	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	✓[1]	-	✓	-	✓	✓	-	-	-
D3	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

该指令用于实现CANOpen总线轴的原点回归。

原点回归方式、速度需在CANopen组态配置界面设置。各原点回归方式说明请参考伺服/电机驱动器相关手册。

表3-296 MC_Home_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6060h = 6	切换到原点回归模式
2	6061h = 6	等待切换原点回归模式完成
3	607Ch = 原点偏移	设置原点偏移
4	6040h.bit4 = 1	开始原点回归
5	6041h.bit10 = 1 且 6041h.bit13 = 1	原点回归失败
	6041h.bit10 = 1 且 6041h.bit12 = 1	原点回归成功

3.16.12 MC_Jog_CO

MC_Jog_CO — 通讯控制轴点动

图形块

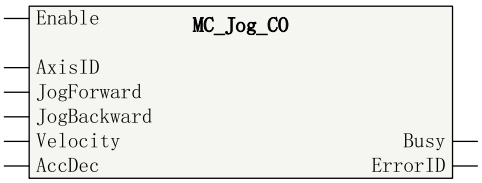


表3-297 指令列表格式

16位指令	—					
32位指令	MC_Jog_CO 连续执行					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	JogForward	正向运动，上升沿触发电平有效	否	-	TRUE/FALSE	BOOL
S3	JogBackward	反向运动，上升沿触发电平有效	否	-	TRUE/FALSE	BOOL
S4	Velocity	目标速度	否	-	-	REAL
S5	AccDec	加/减速度	否	-	-	REAL
D1	Busy	忙标志	是	FALSE	TRUE/FALSE	BOOL
D2	ErrorID	故障码	是	0	*1	INT

说明

*1请参见*第414页“3.16.15 CANopen轴控指令错误代码”*章节“*CANopen轴控指令错误代码*”。

表3–298 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	✓	-	✓	-	✓	✓	-	-	-
S3	✓	-	✓	-	✓	✓	-	-	-
S4	-	-	✓	✓	✓	✓	-	-	-
S5	-	-	✓	✓	✓	✓	-	-	-
D1	✓[1]	-	✓	-	✓	✓	-	-	-
D2	-	-	✓	✓	✓	✓	-	-	-

说明

[1]不支持X元件。

功能和指令说明

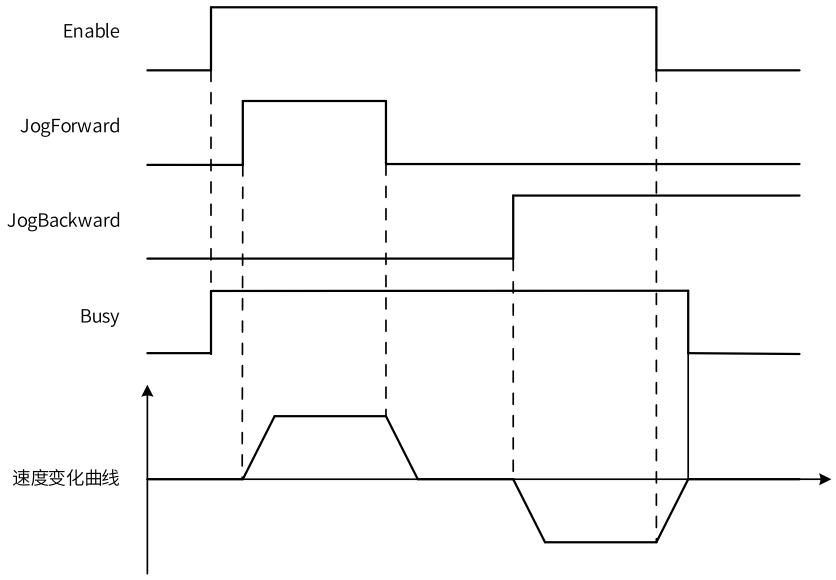
该该指令用于实现CANOpen总线轴的点动功能。当JogForward 有效时，轴按照Velocity 设定的速度正向运动，当JogBackward 有效时，轴按照Velocity

设定的速度做反向运动。当JogForward 和JogBackward 同时有效时，轴停止运动。

表3–299 MC_Jog_CO指令CANOpen对象操作步骤

步骤	操作/条件	说明
1	6040h.bit8 = 0	复位控制字Halt位
2	6083h = 加/减速度	写加速度
3	6084h = 加/减速度	写减速度
4	6060h = 3	切换到速度模式
5	6061h = 3	等待切换速度模式完成
6	正向点动：60FFh = 目标速度 反向点动：60FFh = -目标速度 其他：60FFh = 0	正反向点动
	60FFh < 0 且6041h.bit11 = 1 且 60FDh.bit0 = 1： 60FFh = 0	负向运动遇负限位，点动结束
	607Ah > 6040h 且6041h.bit11 = 1 且 60FDh.bit1 = 1： 60FFh = 0	正向运动遇正限位，点动结束
	60FFh = 0	指令能流无效，点动结束

时序图



3.16.13 MC_WriteParameter_CO

MC_WriteParameter_CO — 通讯控制写入轴参数

图形块

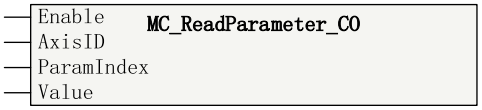


表3–300 指令列表格式

16位指令	—					
32位指令	MC_WriteParameter_CO					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	ParamIndex	参数编号	否	-	*1	INT
S3	Value	参数数值	否	-	-	DINT

说明

*1请参见下文“参数编号列表”。

表3–301 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	-	-	-	-	✓	-	-
S3	-	-	✓	✓	✓	✓	✓	-	-

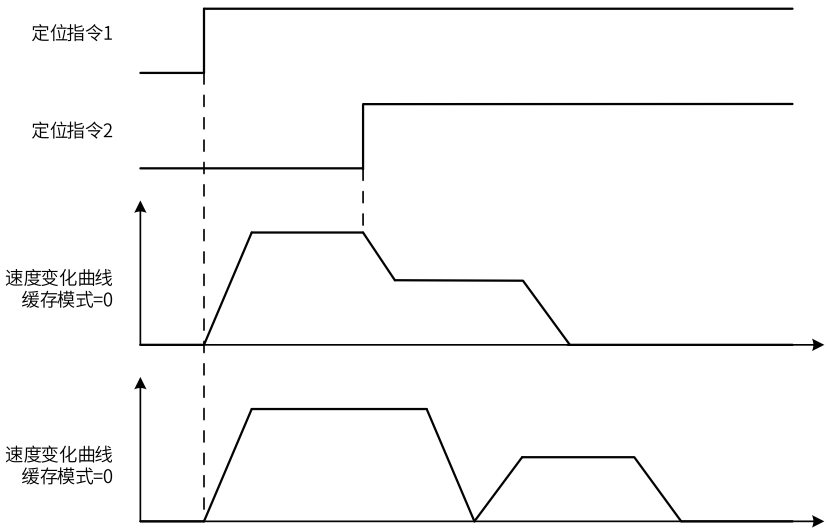
功能和指令说明

该指令用于实现设置CANOpen总线轴参数。参数编号，仅支持常数，请参考下文“参数编号列表”。

表3–302 参数编号列表

参数编号	名称	数据类型	读/写	说明
K1000	缓存模式	UINT32	读/写	定位缓存模式 0（默认）：无缓存，立即触发； 1：等待当前定位完成。 参考：缓冲模式时序图
K1001	DI输入状态	UINT32	读	DI输入状态 [31:16]：厂家自定义 [15:3]：保留 [1]：正向限位 0：无效 1：有效 [0]：反向限位 0：无效 1：有效
K1010	轴状态	INT32	读	当前轴状态 -1：未配置 0：未使能（Disabled） 1：就绪（Standstill） 2：停止（Stopping） 3：原点回归（Homing） 4：连续运动（COPntinue Motion） 5：定位/离散运动（Discrete Motion） 15：故障停止（Errorstop）

时序图



3.16.14 MC_ReadParameter_CO

MC_ReadParameter_CO — 通讯控制读取轴参数

图形块

表3–303 指令列表格式

16位指令	—					
32位指令	MC_ReadParameter_CO					
操作数	名称	描述	能否空	默认值	范围	数据类型
S1	AxisID	轴号，指定需要操作的CANOpen轴的轴号	否	-	1-16	INT
S2	ParamIndex	参数编号	否	-	*1	INT
D1	Value	参数数值	是	-	-	DINT

说明

*1请参见下文“参数编号列表”。

表3–304 软元件列表

操作数	位			字		指针	常数		其他
	X、Y、M、S、B	字元件的位	自定义位变量	D、R、W	自定义字变量	指针变量	K、H	E	
S1	-	-	-	-	-	-	✓	-	-
S2	-	-	-	-	-	-	✓	-	-
D1	-	-	✓	✓	✓	✓	✓	-	-

功能和指令说明

该指令用于实现读取CANOpen总线轴参数。参数编号，仅支持常数，请参考下文“参数编号列表”。

表3–305 参数编号列表

参数编号	名称	数据类型	读/写	说明
K1000	缓存模式	UINT32	读/写	定位缓存模式 0（默认）：无缓存，立即触发； 1：等待当前定位完成。 参考：缓冲模式时序图
K1001	DI输入状态	UINT32	读	DI输入状态 [31:16]：厂家自定义 [15:3]：保留 [1]：正向限位 0：无效 1：有效 [0]：反向限位 0：无效 1：有效
K1010	轴状态	INT32	读	当前轴状态 -1：未配置 0：未使能（Disabled） 1：就绪（Standstill） 2：停止（Stopping） 3：原点回归（Homing） 4：连续运动（COPntinue Motion） 5：定位/离散运动（Discrete Motion） 15：故障停止（Errorstop）

3.16.15 CANopen轴控指令错误代码

使用CANopen指令发生错误时，请参考以下错误代码信息。

表3–306 指令错误代码

代码	说明
0	没有错误。
1	轴号错误。 a) 轴号范围（1~16）； b) 轴号在CANOpen组态配置中不存在或PDO配置错误。
2	指令参数错误。 a) MC_MoveAbsolute_CO、MC_MoveRelative_CO、MC_MoveVelocity_CO、MC_Jog_CO指令加/减速度为小于等于0； b) MC_MoveAbsolute_CO、MC_MoveRelative_CO指令速度为小于等于0；
3	指令参数（位置、原点位置偏移）数值超范围。※1
4	指令参数（速度）数值超范围。※1
5	指令参数（加速度）数值超范围。※1

代码	说明
6	指令参数（减速度）数值超范围。※1
8	当前指令在执行过程被其他指令打断、使能丢失、掉线，导致指令无法完成而停止执行。
9	正向超程导致指令无法完成而停止执行。※2
10	反向超程导致指令无法完成而停止执行。※2
11	原点回归失败。
16	轴未使能，当前指令无法执行。
17	非“故障停止”状态， MC_Reset_CO指令无法执行。
18	轴在“停止”状态，指当前令无法执行。
19	轴正在原点回归，当前指令无法执行。
20	轴在连续运动，当前指令无法执行。
21	轴正在定位，当前指令无法执行。
31	轴在“故障停止”状态，当前指令无法执行。
250	轴使能超时。
251	伺服/电机驱动器出错。※3
255	伺服/电机驱动器掉线。※3

说明

- ※1数值转换脉冲单位后超出32位整数范围。
- ※2运动过程中超程，轴将会进入“故障停止”状态，需使用MC_Reset_CO指令复位后，才能触发轴反方向运动。
- ※3该错误代码仅在MC_Power_CO指令中指示。其他指令在执行过程中出现该故障，该指令会报指令被打断；而在出现该故障后再触发其他指令，该指令将会报轴未使能错误。

4 附录

4.1 ASCII码转换表

ASCII码表具体如下所示：

Bin (二进制)	Oct (八进制)	Dec (十进制)	Hex (十六进制)	缩写/字符	解释
0000 0000	00	0	0x00	NUL(null)	空字符
0000 0001	01	1	0x01	SOH(start of headline)	标题开始
0000 0010	02	2	0x02	STX (start of text)	正文开始
0000 0011	03	3	0x03	ETX (end of text)	正文结束
0000 0100	04	4	0x04	EOT (end of transmission)	传输结束
0000 0101	05	5	0x05	ENQ (enquiry)	请求
0000 0110	06	6	0x06	ACK (acknowledge)	收到通知
0000 0111	07	7	0x07	BEL (bell)	响铃
0000 1000	010	8	0x08	BS (backspace)	退格
0000 1001	011	9	0x09	HT (horizontal tab)	水平制表符
0000 1010	012	10	0x0A	LF (NL line feed, new line)	换行键
0000 1011	013	11	0x0B	VT (vertical tab)	垂直制表符
0000 1100	014	12	0x0C	FF (NP form feed, new page)	换页键
0000 1101	015	13	0x0D	CR (carriage return)	回车键
0000 1110	016	14	0x0E	SO (shift out)	不用切换
0000 1111	017	15	0x0F	SI (shift in)	启用切换
0001 0000	020	16	0x10	DLE (data link escape)	数据链路转义
0001 0001	021	17	0x11	DC1 (device control 1)	设备控制1
0001 0010	022	18	0x12	DC2 (device control 2)	设备控制2
0001 0011	023	19	0x13	DC3 (device control 3)	设备控制3
0001 0100	024	20	0x14	DC4 (device control 4)	设备控制4
0001 0101	025	21	0x15	NAK (negative acknowledge)	拒绝接收
0001 0110	026	22	0x16	SYN (synchronous idle)	同步空闲
0001 0111	027	23	0x17	ETB (end of trans. block)	结束传输块
0001 1000	030	24	0x18	CAN (cancel)	取消

Bin (二进制)	Oct (八进制)	Dec (十进制)	Hex (十六进制)	缩写/字符	解释
0001 1001	031	25	0x19	EM (end of medium)	媒介结束
0001 1010	032	26	0x1A	SUB (substitute)	代替
0001 1011	033	27	0x1B	ESC (escape)	换码(溢出)
0001 1100	034	28	0x1C	FS (file separator)	文件分隔符
0001 1101	035	29	0x1D	GS (group separator)	分组符
0001 1110	036	30	0x1E	RS (record separator)	记录分隔符
0001 1111	037	31	0x1F	US (unit separator)	单元分隔符
0010 0000	040	32	0x20	(space)	空格
0010 0001	041	33	0x21	!	叹号
0010 0010	042	34	0x22	"	双引号
0010 0011	043	35	0x23	#	井号
0010 0100	044	36	0x24	$	美元符
0010 0101	045	37	0x25	%	百分号
0010 0110	046	38	0x26	&	和号
0010 0111	047	39	0x27	'	闭单引号
0010 1000	050	40	0x28	(	开括号
0010 1001	051	41	0x29	)	闭括号
0010 1010	052	42	0x2A	*	星号
0010 1011	053	43	0x2B	+	加号
0010 1100	054	44	0x2C	,	逗号
0010 1101	055	45	0x2D	-	减号/破折号
0010 1110	056	46	0x2E	.	句号
0010 1111	057	47	0x2F	/	斜杠
0011 0000	060	48	0x30	0	字符0
0011 0001	061	49	0x31	1	字符1
0011 0010	062	50	0x32	2	字符2
0011 0011	063	51	0x33	3	字符3
0011 0100	064	52	0x34	4	字符4
0011 0101	065	53	0x35	5	字符5
0011 0110	066	54	0x36	6	字符6
0011 0111	067	55	0x37	7	字符7
0011 1000	070	56	0x38	8	字符8
0011 1001	071	57	0x39	9	字符9
0011 1010	072	58	0x3A	:	冒号
0011 1011	073	59	0x3B		分号
0011 1100	074	60	0x3C	<	小于
0011 1101	075	61	0x3D	=	等号
0011 1110	076	62	0x3E		大于
0011 1111	077	63	0x3F	?	问号
0100 0000	0100	64	0x40	@	电子邮件符号

Bin (二进制)	Oct (八进制)	Dec (十进制)	Hex (十六进制)	缩写/字符	解释
0100 0001	0101	65	0x41	A	大写字母A
0100 0010	0102	66	0x42	B	大写字母B
0100 0011	0103	67	0x43	C	大写字母C
0100 0100	0104	68	0x44	D	大写字母D
0100 0101	0105	69	0x45	E	大写字母E
0100 0110	0106	70	0x46	F	大写字母F
0100 0111	0107	71	0x47	G	大写字母G
0100 1000	0110	72	0x48	H	大写字母H
0100 1001	0111	73	0x49	I	大写字母I
01001010	0112	74	0x4A	J	大写字母J
0100 1011	0113	75	0x4B	K	大写字母K
0100 1100	0114	76	0x4C	L	大写字母L
0100 1101	0115	77	0x4D	M	大写字母M
0100 1110	0116	78	0x4E	N	大写字母N
0100 1111	0117	79	0x4F	O	大写字母O
0101 0000	0120	80	0x50	P	大写字母P
0101 0001	0121	81	0x51	Q	大写字母Q
0101 0010	0122	82	0x52	R	大写字母R
0101 0011	0123	83	0x53	S	大写字母S
0101 0100	0124	84	0x54	T	大写字母T
0101 0101	0125	85	0x55	U	大写字母U
0101 0110	0126	86	0x56	V	大写字母V
0101 0111	0127	87	0x57	W	大写字母W
0101 1000	0130	88	0x58	X	大写字母X
0101 1001	0131	89	0x59	Y	大写字母Y
0101 1010	0132	90	0x5A	Z	大写字母Z
0101 1011	0133	91	0x5B	[	开方括号
0101 1100	0134	92	0x5C	\	反斜杠
0101 1101	0135	93	0x5D	]	闭方括号
0101 1110	0136	94	0x5E	^	脱字符
0101 1111	0137	95	0x5F	_	下划线
0110 0000	0140	96	0x60	`	开单引号
0110 0001	0141	97	0x61	a	小写字母a
0110 0010	0142	98	0x62	b	小写字母b
0110 0011	0143	99	0x63	c	小写字母c
0110 0100	0144	100	0x64	d	小写字母d
0110 0101	0145	101	0x65	e	小写字母e
0110 0110	0146	102	0x66	f	小写字母f
0110 0111	0147	103	0x67	g	小写字母g
0110 1000	0150	104	0x68	h	小写字母h
0110 1001	0151	105	0x69	i	小写字母i

Bin (二进制)	Oct (八进制)	Dec (十进制)	Hex (十六进制)	缩写/字符	解释
0110 1010	0152	106	0x6A	j	小写字母j
0110 1011	0153	107	0x6B	k	小写字母k
0110 1100	0154	108	0x6C	l	小写字母l
0110 1101	0155	109	0x6D	m	小写字母m
0110 1110	0156	110	0x6E	n	小写字母n
0110 1111	0157	111	0x6F	o	小写字母o
0111 0000	0160	112	0x70	p	小写字母p
0111 0001	0161	113	0x71	q	小写字母q
0111 0010	0162	114	0x72	r	小写字母r
0111 0011	0163	115	0x73	s	小写字母s
0111 0100	0164	116	0x74	t	小写字母t
0111 0101	0165	117	0x75	u	小写字母u
0111 0110	0166	118	0x76	v	小写字母v
0111 0111	0167	119	0x77	w	小写字母w
0111 1000	0170	120	0x78	x	小写字母x
0111 1001	0171	121	0x79	y	小写字母y
0111 1010	0172	122	0x7A	z	小写字母z
0111 1011	0173	123	0x7B	{	开花括号
0111 1100	0174	124	0x7C	|	垂线
0111 1101	0175	125	0x7D	}	闭花括号
0111 1110	0176	126	0x7E	~	波浪号
0111 1111	0177	127	0x7F	DEL (delete)	删除

4.2 错误代码列表

用户编程发生错误时，后台软件会提示各个类别的错误代码。下表为错误代码分类及对应的解决措施：

表4–1 错误代码

错误代码	错误信息	详细说明	解决措施
用户程序			
1500	用户程序看门狗超时	用户程序执行时间过长，超过了设定的程序看门狗时间。	根据时间情况，调大看门狗时间，或排除用户程序是否存在不符合预期的执行时间过长的程序块。
1501	使用未定义指令	不支持该指令。	PLC需固件升级到支持该指令的版本。
1502	用户程序不完整，长度错误	用户程序不完整，长度错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1503	程序授权保护标识符错误，请确认标识符是否匹配	程序授权保护标识符错误，请确认设备是否设置了正确的授权保护标识符。	请联系设备提供商处理。
1505	程序块POU标志错误	程序块POU标志错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。

错误代码	错误信息	详细说明	解决措施
1510	子程序标志错误	子程序标志错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1511	子程序类型错误	子程序类型错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1512	子程序序号错误或超限	子程序序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1513	子程序地址错误，重复，冲突	子程序地址错误，重复，冲突	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1514	中断子程序序号错误或超限	中断子程序序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1515	中断子程序地址错误，重复，冲突	中断子程序地址错误，重复，冲突	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1516	中断子程序沿中断属性错误，非上升沿、下降沿等	中断子程序沿中断属性错误，非上升沿、下降沿等	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1517	中断子程序定时中断定时时间范围错误	中断子程序定时中断定时时间范围错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1520	OBprog程序标志错误	OBprog程序标志错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1521	OBprog程序类型错误	OBprog程序类型错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1522	OBprog程序序号错误或超限	OBprog程序序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1523	OBprog程序地址错误，重复，冲突	OBprog程序地址错误，重复，冲突	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1524	OBprog程序变量数量错误	OBprog程序变量数量错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1525	OBprog程序变量长度错误	OBprog程序变量长度错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1526	OBprog程序头数据错误	OBprog程序头数据错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1530	CJ-LBL指令LBL序号错误或超限	CJ-LBL指令LBL序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
1531	CJ-LBL指令LBL地址错误，重复，冲突	CJ-LBL指令LBL地址错误，重复，冲突	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5001	用户程序执行异常或指令返回值错误，部分指令不执行	用户程序执行异常或指令返回值错误，部分指令不执行；导致程序执行异常退出。	请检查用户程序逻辑，是否存在执行流程或执行逻辑异常。
5010	CALL指令的子程序序号错误或超限	CALL指令的子程序序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5011	CALL指令的子程序不存在或未初始化	CALL指令的子程序不存在或未初始化	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5012	CALL指令的子程序嵌套层次超范围或小于等于0	执行CALL指令时，检查发现子程序嵌套层次超范围。	修改程序逻辑，减少子程序嵌套层级。
5013	CALL指令的子程序返回关系错误	CALL指令的子程序返回关系错误	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。

错误代码	错误信息	详细说明	解决措施
5014	子程序调用和子程序返回不配套	子程序执行异常，子程序调用和子程序返回不配套。	检查用户程序是否异常退出，导致的子程序调用和返回错乱。
5015	中断子程序未定义	中断子程序执行时，发现对应的中断子程序未定义或不存在。	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5016	中断子程序定时中断队列已满，中断丢失	中断子程序定时中断队列已满，中断丢失	修改中断子程序属性或逻辑，酌情减少中断次数。
5020	FBFC的程序序号错误或超限	FBFC的程序序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5021	FBFC的程序不存在或未初始化	FBFC的程序不存在或未初始化	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5022	FBFC的程序的变量不存在或未初始化	FBFC的程序的变量不存在或未初始化	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5023	FBFC的程序嵌套层次超范围或小于等于0	执行FBFC调用时，检查发现FBFC程序嵌套层次超范围。	修改程序逻辑，减少FBFC程序嵌套层级。
5024	FBFC的程序返回关系错误	FBFC的程序返回关系错误	FBFC特殊指令使用位置错误，或重新编译并下载用户程序。
5025	OBprog程序调用和程序返回不配套	OBprog程序执行异常，调用和返回不配套	检查用户程序是否异常退出，导致的程序调用和返回错乱。
5030	CJ-LBL指令的LBL序号错误或超限	CJ-LBL指令的LBL序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5031	CJ-LBL指令的LBL不存在或未初始化	CJ-LBL指令的LBL不存在或未初始化	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
5032	FOR~NEXT指令的嵌套层次超范围或小于等于0	执行FOR~NEXT指令时，检查发现FOR~NEXT嵌套层次超范围。	修改程序逻辑，减少FOR~NEXT指令的嵌套层次。
5033	FOR~NEXT指令的循环次数超范围或小于等于0	执行FOR~NEXT指令时，检查发现FOR~NEXT指令的循环次数超范围或小于等于0	修改程序逻辑，修改FOR~NEXT指令的循环次数。
5034	FOR~NEXT指令的循环次数等于0	执行FOR~NEXT指令时，检查发现FOR~NEXT指令的循环次数等于0	修改程序逻辑，修改FOR~NEXT指令的循环次数。
5035	FOR~NEXT的关系不配套	FOR~NEXT的关系不配套	检查用户程序是否异常退出，导致的关系错乱。
5101	指令的参数变量的地址错误，变量不存在	指令的参数变量的地址错误，变量不存在	检查指令的参数变量的地址，是否异常，或变量不存在。
5102	指令的参数变量的大小错误，变量不存在或超范围	指令的参数变量的大小错误，变量不存在或超范围	检查指令的参数变量的数据长度，是否超出范围。
5103	xxxx0001错误	xxxx0001错误	xxxx0001错误
5104	指令的参数顺序错误或关系错误	指令的参数顺序错误或关系错误	检查指令的参数顺序是否错误，或大小关系错误。
5105	字符串指令，字符串数据错误或长度错误	字符串指令，字符串数据错误或长度错误	检查字符串指令，字符串数据，是否非法。
5110	Pointer指针变量序号错误或超限	Pointer指针变量序号错误或超限	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。

错误代码	错误信息	详细说明	解决措施
5111	Pointer指针变量未初始化，未指向有效的数据变量	Pointer指针变量未初始化，未指向有效的数据变量	检查Pointer指针变量是否初始化，或是否指向有效的变量地址。
5112	Pointer指针变量指向的变量不存在或超范围	Pointer指针变量指向的变量不存在或超范围	检查Pointer指针变量指向的变量地址，或重新初始化该指针变量。
5113	Pointer指针变量偏移值超范围	Pointer指针变量偏移值超范围	Pointer指针变量偏移值过大，需要调小。
5114	Pointer指针变量计算后指向的变量不存在或超范围	Pointer指针变量计算后指向的变量不存在或超范围	检查Pointer指针变量执行后的结果指向的变量地址，不存在或超范围
5120	counter计数器指令实例化失败	counter计数器指令实例化失败	counter计数器指令实例化失败，请重新编译并下载用户程序。
5121	counter计数器指令比较值错误或超范围	counter计数器指令比较值错误或超范围	counter计数器指令比较值错误或超范围
5130	timer定时器指令实例化失败	timer定时器指令实例化失败	timer定时器指令实例化失败，请重新编译并下载用户程序。
5131	timer定时器指令比较值错误或超范围	timer定时器指令比较值错误或超范围	timer定时器指令比较值错误或超范围
5140	SFC STL 并行分支/并行汇合/选择分支/选择汇合个数超范围	SFC STL 并行分支/并行汇合/选择分支/选择汇合个数超范围	SFC STL 并行分支/并行汇合/选择分支/选择汇合个数超范围
5150	功能块指令实例化失败	功能块指令实例化失败	功能块指令实例化失败，请重新编译并下载用户程序。
5160	数组下标变量编码错误或不存在	数组下标变量编码错误或不存在	请重新编译程序并下载。
5161	数组下标变量数据错误或超范围。	数组下标变量数据错误或超范围	修改数据下标变量的值，不要超出数组范围。
5600	SerialSR指令实例化失败	SerialSR指令实例化失败	SerialSR指令实例化失败，请重新编译并下载用户程序。
5601	SerialSR指令端口号超限	SerialSR指令端口号超限	修改SerialSR指令端口号。
5602	SerialSR指令协议错误	SerialSR指令协议错误	串口未设置成自由协议，通过后台软件的串口配置更改设置。
5603	SerialSR指令端口冲突	多条指令同时调用SerialSR指令，导致未抢占到端口的指令报错。	修改指令调度时序，分时复用。
5604	SerialSR指令发送数据长度超范围或小于0	SerialSR指令发送数据长度超范围或小于0	SerialSR指令发送数据长度超范围或小于0
5605	SerialSR指令发送数据缓冲区异常	获取SerialSR指令发送数据缓冲区获取失败。	获取SerialSR指令发送数据缓冲区失败，重新使能该指令。
5606	SerialSR指令接收数据长度超范围或小于0	SerialSR指令接收数据长度超范围或小于0	SerialSR指令接收数据长度超范围或小于0
5607	SerialSR指令接收数据缓冲区异常	获取SerialSR指令接收数据缓冲区失败。	获取SerialSR指令接收数据缓冲区失败，重新使能该指令。
6580	CANOpen轴指令使用了无效的轴号	CANOpen轴指令使用了无效的轴号	CANOpen轴指令使用了无效的轴号，修改轴号。

错误代码	错误信息	详细说明	解决措施
6701	无效内存地址，访问内存不存在	无效内存地址，访问的软元件或变量不存在。	修改指令参数，使用有效的软元件或变量。
6705	无效内存大小，访问内存不存在或超范围	无效内存大小，访问的软元件或变量的数量过大或超出范围。	修改指令参数，调整软元件或变量的数量。
6706	数据不合理或数据超范围	指令参数不合理，或指令参数超出允许的范围。	参考该条指令的手册说明，调整指令参数值。
6711	无效变量地址，访问的变量不存在	无效变量地址，访问的软元件或变量不存在。	修改指令参数，使用有效的软元件或变量。
6712	无效变量大小，访问的变量超范围	无效变量大小，访问的软元件或变量的数量过大或超出范围。	修改指令参数，调整软元件或变量的数量。
6713	无效变量编码	无效变量编码	请重新编译并下载用户程序，或更换后台软件后重新编译并下载用户程序。
系统			
1011	FPGA初始化失败	FPGA初始化失败。	设备硬件存在异常，请更换正常的设备使用，并将异常设备返厂维修。
1012	中断初始化失败	中断初始化失败。	设备硬件存在异常，请更换正常的设备使用，并将异常设备返厂维修。
1013	定时中断初始化失败	用户程序用定时中断，初始化失败。	请重启设备重试，或更换设备使用，并将异常设备返厂维修。
5200	断电保存数据错误	断电保存数据错误。	断电保存数据错误。
5238	2038年问题临近预警	设备在北京时间2038年1月19日中午11:14:07之后，将无法正常工作。	请修改设备时间。
5250	RTC时钟不可靠，电池电压低	检测到RTC时钟电池电压低，此时如果设备断电，系统时间将被恢复为初始值。	请在保持设备通电的情况下，更换RTC时钟电池。
5900	以太网IP地址冲突，网络已关闭	设备连接到网络、 修改IP或从停止切到运行后，将检测IP地址在当前网络中是否已经被其他设备使用。如果检测到IP已经被使用，为避免影响该设备，本设备将自动关闭网络。	请修改设备IP地址。
本机IO			
5300	初始化失败	初始化失败。	设备硬件存在异常，请更换正常的设备使用，并将异常设备返厂维修。
5301	数字输入滤波参数配置数据无效	数字输入滤波参数配置数据无效。	请修改数字输入滤波参数配置数据。
本地扩展模块			
5400	扩展模块接口硬件初始化失败	检测到扩展模块接口硬件出现异常，无法完成初始化操作。	设备硬件存在异常，请更换正常的设备使用，并将异常设备返厂维修。

错误代码	错误信息	详细说明	解决措施
5401	扩展模块配置数据解析失败	在升级新版本的Autoshop后，由于功能的升级，相对旧版本配置数据可能会发生变化。由于无法做到完全兼容，导致旧版本的设备无法解析新的组态配置数据。	请将设备升级到Autoshop指定兼容的固件版本。
5402	扩展模块接口插槽初始化失败	检测到扩展模块接口插槽出现异常，无法完成初始化操作。	1、检查扩展模块接口插槽是否有短路，如果有短路情况，请消除短路； 2、确认安装的模块硬件是否正常，如果有模块异常，请更换正常的模块。
5403	未安装扩展模块	组态配置有扩展模块，但实际检测到没有安装扩展模块。	请根据需求安装扩展模块，或者修改扩展模组态配置。
5404	插槽实际安装模块与组态配置模块不一致	扩展模块插槽实际安装的模块必须与组态配置模块一致，否则无法正常工作。	请根据需求安装扩展模块，并相应修改模块组态配置，确保两者完全一致。
5411	插槽的模块未供电	该模块要求提供外部电源才能正常工作，但实际检测到外部电源没有接通。	请按照模块规格说明要求正确接入外部电源。
5412	插槽的模块硬件故障	检测到模块内部存在故障，模块无法正常工作。	请更换正常的模块使用，并将故障模块返厂维修。
5413	插槽的模块温度过热	模块检测到其内部温度过高，可能工作不正常。	1、请不要将安装在超出模块规格说明要求温度范围的工作环境； 2、模块内部存在故障，请更换正常的模块使用，并将故障模块返厂维修。
5419	插槽的模块通道输入或输出上溢	对于输入通道而言，输入的信号已经超过其设置量程允许阈量的采样最大值，无法正常采样，且存在可能导致烧毁输入端口的可能性。 对于输出通道而言，设置到对应通道的输出值已经超过其设置量程允许阈量的最大值，信号无法确输出。	对于输入通道： 请检查实际输入信号值大小。 如果正常工作情况下，输入到该通道的信号已经超过其设置的采样量程，请修改采样量程，选择合适的采样量程。 如果信号异常，请检查该信号的输出设备或仪器。 对于输出通道： 检查设置的输出值大小，确保设置的例输出在设置的量程范围内，如果当前设置的量程无法满足实际需求，请修改合适的量程设置。

错误代码	错误信息	详细说明	解决措施
5420	插槽的模块通道输入或输出下溢	对于输入通道而言，输入的信号已经超过其设置量程允许阈量的采样最小值，无法正常采样。 对于输出通道而言，设置到对应通道的输出值已经超过其设置量程允许阈量的最小值，信号无法确输出。	对于输入通道： 请检查实际输入信号值大小。 如果正常工作情况下，输入到该通道的信号已经超过其设置的采样量程，请修改采样量程，选择合适的采样量程。 如果信号异常，请检查该信号的输出设备或仪器。 对于输出通道： 检查设置的输出值大小，确保设置的例输出在设置的量程范围内，如果当前设置的量程无法满足实际需求，请修改合适的量程设置。
5421	插槽的模块通道输入超上限或电流输出断线	对于输入通道而言，输入的信号已经超过其设置量程的采样最大值。此时，信号可以正常采样但精度无法保证。 对于电流输出通道而言，输出端口没有接入负载或者接入的负载阻抗过大，导致电流无法正常输出。	对于输入通道： 请检查实际输入信号值大小。 如果正常工作情况下，输入到该通道的信号已经超过其设置的采样量程，请修改采样量程，选择合适的采样量程。 如果信号异常，请检查该信号的输出设备或仪器。 对于电流输出通道： 请确保输出端口的负载连接正常、可靠，同是确保负载阻抗在模块规格说明指定的范围内。
5422	插槽的模块通道输入超下限或电压输出短路	对于输入通道而言，输入的信号已经超过其设置量程的采样最小值。此时，信号可以正常采样但精度无法保证。 对于电压输出通道而言，输出端口可能被短路或者接入的负载阻抗过小，导致电压无法正常输出。	对于输入通道： 请检查实际输入信号值大小。 如果正常工作情况下，输入到该通道的信号已经超过其设置的采样量程，请修改采样量程，选择合适的采样量程。 如果信号异常，请检查该信号的输出设备或仪器。 对于电压输出通道： 请确保输出端口的负载连接正常、可靠，同是确保负载阻抗在模块规格说明指定的范围内。
5423	插槽的模块通道输入断线或输出硬件故障	对于输入通道而言，输入端口没有连接输入信号或者输入的信号过小，无法检测或采样。 对于输出通道而言，检测到通道通道硬件存在故障，可能已经烧毁。	对于输入通道： 请确保输入端口的信号连接正常、可靠，同是确保信号正常有效。 对于输出通道： 模块通道通道硬件存在故障，可能已经烧毁，请更换正常的模块使用，并将故障模块返厂维修。
本地编码器轴			
6300	轴未分配输入设备或分配的输入设备无效的	本地编码器轴必需分配一个高速计数器，且每个高速计数器分配给一个轴，否无法正常工作。	请在该轴的“基本设置”页面“输入设备”中分配一个没有被分配的高速计数器。

错误代码	错误信息	详细说明	解决措施
6301	轴单位换算设置参数无效	高速计数器被分配轴后，其计数值(脉冲单位)，将按照单位换算设置参数转换为用户单位(Unit)。如果编码器旋转一圈的脉冲数、编码器旋转一圈的移动量或者变速器装置齿轮比参数设置不正确，轴将无法正常工作。	请在该轴的“单位换算设置”页面检查各项设置，并修复为正确的参数。
6302	轴软件限位或旋转周期参数配置数据无效	线性模式的负向限制值必需小于0，正向限制值必需大于0。 旋转模式的旋转周期必需大于0。 由于高速计数器为32位计数器，负向限制值、正向限制值及旋转周期换算为脉冲单位后必需在32位整数范围[-2147483648, 2147483647]。	线性模式: 请修改负向限制值与正向限制值，确保负向限制值小于0，正向限制值大于0，且换算为脉冲单位后在32位整数范围[-2147483648, 2147483647] 旋转模式: 请修改旋转周期，确保旋转周期大于0，且换算为脉冲单位后在32位整数范围[-2147483648, 2147483647]。
6303	轴计数模式或信号源参数配置数据无效	高速计数支持的计数模式和信号源如下。 A/B相1倍频： X0-A相，X1-B相 X2-A相，X3-B相 A/B相2倍频： X0-A相，X1-B相 X2-A相，X3-B相 A/B相4倍频： X0-A相，X1-B相 X2-A相，X3-B相 CW/CCW： X0-CW，X1-CCW X2-CW，X3-CCW 脉冲+方向： X0-脉冲，X1-方向 X2-脉冲，X3-方向	请选择支持的计数模式与信号源。
6304	轴预置功能设置了无效的输入端子	预置功能支持的输入端子有 X0、X1、X2、X3、X4、X5、X6、X7。	请选择支持预置功能支持的输入端子。

错误代码	错误信息	详细说明	解决措施
6305	轴探针1设置了无效的输入端子	探针1支持的输入端子有 X0、X1、X2、X3、X4、X5、X6、X7。	请选择探针1支持的输入端子。
6306	轴探针2设置了无效的输入端子	探针2支持的输入端子有 X0、X1、X2、X3、X4、X5、X6、X7。	请选择探针2支持的输入端子。
6307	轴比较输出设置了无效的输出端子	比较输出支持的输出端子有 Y0、Y1、Y2、Y3。	请选择比较输出支持的输出端子。
6308	轴比较输出设置了无效的脉冲宽度	单位选择ms时，设置的时间范围为0.1~6553.5ms。 单位选择Unit时，应确保设置的值转换为脉冲单位后在1~65535的范围。	请修改脉冲宽度设置，确保其在允许设置的范围内。
CANLink			
6400	站点地址冲突，检测到站点地址在网络中已经存在	在CANLink通讯中，实际接入到网络中的所有站点地址都不允许重复。 设备节点上电初始化完成或修改站点地址后执行地址冲突检测，如果检测到地址重复将报故障并停止该节点一切CANLink总线活动。	修改修改站点地址，确保网络中无重复的站点地址
6401	从站不在线	检测到从站不在线，无法与该从站通讯。	请检查CAN网络线路连接是否正常，确保线路连接可靠无短路或开路、CANH与CANL无接反、终端电阻正常。
6411	从站配置异常响应(1)“使用了未定义的编码”	配置从站时，从站返回异常响应(1) “使用了未定义的编码”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6412	从站配置异常响应(2)“配置索引超过节点支持最大值”	配置从站时，从站返回异常响应(2) “配置索引超过节点支持最大值”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6413	从站配置异常响应(3)“寄存器地址不存在或无法访问”	配置从站时，从站返回异常响应(3) “寄存器地址不存在或无法访问”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6415	从站配置异常响应(5)“寄存器数据长度无效”	配置从站时，从站返回异常响应(5) “寄存器数据长度无效”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6416	从站配置命令等待应答超时	配置从站时，等待从站应答响应超时。	请检查实际接入的设备类型/型号与组态配置是否相符。
6421	从站同步异常响应(1)“非法命令码”	向从站发送同步命令时，从站返回异常响应(1) “非法命令码”。	请检查实际接入的设备类型/型号与组态配置是否相符。

错误代码	错误信息	详细说明	解决措施
6422	从站同步异常响应(2)“寄存器地址不存在或不允许访问”	向从站发送同步数据时，从站返回异常响应(2) “寄存器地址不存在或不允许访问”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6423	从站同步异常响应(3)“数据值不在允许范围内”	向从站发送同步数据时，从站返回异常响应(3) “数据值不在允许范围内”。	1、请检查对应的寄存器地址设置的数值是否已经超范围。 2、请检查实际接入的设备类型/型号与组态配置是否相符。
6424	从站同步异常响应(4)“当前状态下该操作不可达或不允许”	向从站发送同步数据时，从站返回异常响应(4) “当前状态下该操作不可达或不允许”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6425	从站同步异常响应(5)“数据长度无效”	向从站发送同步数据时，从站返回异常响应(5) “数据长度无效”。	请检查实际接入的设备类型/型号与组态配置是否相符。
6426	从站同步命令等待应答超时	向从站发送同步命令时，等待从站应答响应超时。	请检查实际接入的设备类型/型号与组态配置是否相符。
CANopen			
6401	节点不在线	检测到节点不在线，无法与该节点通讯。	请检查CAN网络线路连接是否正常，确保线路连接可靠无短路或开路、CANH与CANL无接反、终端电阻正常。
Modbus RTU/ASC 主站			
5500	Modbus-RTU 串口必须使用8位数据位	Modbus-RTU 串口必须使用8位数据位。	Modbus-RTU 串口必须使用8位数据位。
6001	从站返回异常应答(01) “非法功能码”	对于服务器(或从站)来说，询问中接收到的功能码是不可允许的操作。这也许是因为功能码仅仅适用于新设备而在被选单元是不可实现的。同时，还指出服务器(或从站)在错误状态中处理这种请求，例如：因为它是未配置的，并且要求返回寄存器值。	请检查服务器(或从站)是否支持该功能码。
6002	从站返回异常应答(02) “非法数据地址”。	对于服务器(或从站)来说，询问中接收到的数据地址是不可允许的地址。特别是，参考号和传输长度的组合是无效的。对于带有100 个寄存器的控制器来说，带有偏移量96 和长度4的请求会成功，带有偏移量96和长度5的请求将产生异常码02。	请检查服务器(或从站)对应的功能码是否支持该配置所访问的所有地址。

错误代码	错误信息	详细说明	解决措施
6003	从站返回异常应答(03) “非法数据值”	对于服务器(或从站)来说，询问中包括的值是不可允许的值。这个值指示了组合请求剩余结构中的故障，例如：隐含长度是不正确的。并不意味着，因为Modbus协议不知道任何特殊寄存器的任何特殊值的重要意义，寄存器中被提交存储的数据项有一个应用程序期望之外的值。	请检查数值是否在有效范围。
6128	应答站号与请求站号不匹配	主站发送请求帧后，收到的应答帧站号与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6129	应答功能码与请求功能码不匹配	主站发送请求帧后，收到的应答帧功能码与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6130	应答数据地址与请求数据地址不匹配	主站发送请求帧后，收到的应答帧数据地址与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6131	应答数据值与请求数据值不匹配	主站发送请求帧后，收到的应答帧答数据值与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6240	配置中的缓存地址映射是无效的	配置中的缓存地址映射是无效的，此配置将无法正确执行。	请将配置中的缓存地址映射修改为有效的变量或元件地址。
6255	请求超时	主站发送请求帧后，在设置的超时时间内没有收到从站的应答，这时主站将按设置重试次数进行重试，当超过重试次数后，则认为从站异常，并报请求超时错误。	1、请确保通讯网络线路可靠连接。 2、请确保从站站号与配置中的从站站号一致。 3、请修改超时时间，确保主站能在超时时间内接收完成应答帧。 4、请检查接入的从站是否为正常的Modbus从站。
Modbus TCP 主站			
6000	连接断开	Modbus TCP客户机无法与服务器建立TCP连接。	1、请确保通讯网络线路可靠连接。 2、请检查从站IP地址及端口号与配置是否一致。 3、如果客户机与服务器通过网桥、路由器或网关联接，需要确保客户机与服务器网关设置正确。

错误代码	错误信息	详细说明	解决措施
6001	从站返回异常应答(01) “非法功能码”	对于服务器(或从站)来说，询问中接收到的功能码是不可允许的操作。这也许是因为功能码仅仅适用于新设备而在被选单元是不可实现的。同时，还指出服务器(或从站)在错误状态中处理这种请求，例如：因为它是未配置的，并且要求返回寄存器值。	请检查服务器(或从站)是否支持该功能码。
6002	从站返回异常应答(02) “非法数据地址”。	对于服务器(或从站)来说，询问中接收到的数据地址是不可允许的地址。特别是，参考号和传输长度的组合是无效的。对于带有100 个寄存器的控制器来说，带有偏移量96 和长度4的请求会成功，带有偏移量96和长度5的请求将产生异常码02。	请检查服务器(或从站)对应的功能码是否支持该配置所访问的所有地址。
6003	从站返回异常应答(03) “非法数据值”	对于服务器(或从站)来说，询问中包括的值是不可允许的值。这个值指示了组合请求剩余结构中的故障，例如：隐含长度是不正确的。并不意味着，因为Modbus协议不知道任何特殊寄存器的任何特殊值的重要意义，寄存器中被提交存储的数据项有一个应用程序期望之外的值。	请检查数值是否在有效范围。
6128	应答站号与请求站号不匹配	主站发送请求帧后，收到的应答帧站号与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6129	应答功能码与请求功能码不匹配	主站发送请求帧后，收到的应答帧功能码与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6130	应答数据地址与请求数据地址不匹配	主站发送请求帧后，收到的应答帧数据地址与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6131	应答数据值与请求数据值不匹配	主站发送请求帧后，收到的应答帧答数据值与发送的请求帧不一致。	请检查接入的从站是否为正常的Modbus从站。
6240	配置中的缓存地址映射是无效的	配置中的缓存地址映射是无效的，此配置将无法正确执行。	请将配置中的缓存地址映射修改为有效的变量或元件地址。

错误代码	错误信息	详细说明	解决措施
6255	请求超时	主站发送请求帧后，在设置的超时时间内没有收到从站的应答，这时主站将按设置重试次数进行重试，当超过重试次数后，则认为从站异常，并报请求超时错误。	1、请确保通讯网络线路可靠连接。 2、请确保从站站号与配置中的从站站号一致。 3、请修改超时时间，确保主站能在超时时间内接收完成应答帧。 4、请检查接入的从站是否为正常的Modbus从站。
运动控制轴			
9001	本地轴急停有效	急停端子输入有效，停止脉冲输出	在关闭急停端子输入后调用MC_Reset指令复位故障
9003	超速	脉冲输出频率超过200K	检查目标速度乘以齿轮比之后的脉冲频率是否超过200K
9020	原点回归错误	没有映射负限位	该原点回归方式需要映射负限位，请在配置界面映射负限位
9021	原点回归错误	没有映射正限位	该原点回归方式需要映射正限位，请在配置界面映射正限位
9022	原点回归错误	没有映射原点信号	该原点回归方式需要映射原点开关，请在配置界面映射原点开关
9023	原点回归错误	1、以原点返回速度运行时输出频率超过200K。 2、以原点返回接近速度运行时输出频率超过200K	1、修改单位换算设置确保原点返回速度和原点返回接近速速不超过200K。 2、修改原点返回速度确保输出频率不超过200K。 3、修改原点返回接近速度确保输出频率不超过200K
9024	原点回归错误	原点返回过程中超时	1、检查限位信号、原点信号能否正常导通。 2、检查原点返回超时时间设置是否过小
9025	原点回归错误	原点返回过程中限位信号有误	检查是否触发了当前原点回归方式没有用的限位信号
9030	限位有效	定位过程中限位信号输入有效	检测是否在正常运行过程中碰到限位
9031	同步异常	目标发送脉冲数和实际发送脉冲数不匹配	检测是否在正常定位过程中碰到限位
9101	轴类型错误或不存在	1、AxisID指定的轴类型错误 2、AxisID指定的轴不存在	1、检查指令是否支持AixsID指定的轴 2、检查AxisID指定的轴是否存在
9102	轴配置失败	1、轴配置数据丢失 2、轴配置参数不合理	检查配置参数是否有误
9103	在轴无故障的情况下调用了MC_Reset指令	在轴无故障的情况下调用了MC_Reset指令	检查是否在轴没有切换到ErrorStop状态到的情况下故障的情况下调用了MC_Reset指令
9104	调用MC_ReadStatus指令时轴处于未知状态	调用MC_ReadStatus指令时轴处于未知状态	通过在线监控功能查看轴当前的状态是否是不可控状态
9105	不允许设置当前位置	在运行或停止过程中调用了MC_SetPosition指令	再有当轴处于standstill、poweroff和errorstop状态下允许设置当前位置

错误代码	错误信息	详细说明	解决措施
9106	轴正在执行故障停机	轴正在执行故障停机	等待轴故障停机完成，解决完故障执行复位指令在执行本命令
9107	参数不合理	参数不合理	检查指令左侧的参数是否合理
9108	PLCOpen状态机不合理	PLCOpen状态机不合理	检查当前PLCOpen的状态机是否满足本指令运行的条件，如果不满足，调用相应的指令将轴切换到满足状态
9110	在停止过程中重复调用了MC_Stop指令	在停止过程中重复调用了MC_Stop指令	MC_Stop指令不予许多次调用，每次只能触发一个
9111	指令链表丢失	指令链表丢失	1、检查后台版本和单板版本是否匹配 2、联系厂家
9112	轴号变更	在指令有效期间AxisID的值发生变化	MC_Power、MC_Jog等Enable型指令在能流有效期间不能变更轴号
9113	MC_Reset指令复位超时	MC_Reset指令复位超时	1、查看驱动器故障能否复位 2、查看轴的故障类型能否复位
9114	轴写0x6060失败	轴写0x6060失败	1、检查网络通信是否收到干扰 2、检查从站是否支持对象字典0x6060
9115	Stopping状态下调用了MC_Halt指令	Stopping状态下调用了MC_Halt指令	Stopping状态下不允许调用MC_halt指令
9116	当前轴处于在线调试模式	当前轴处于在线调试模式	检查当前轴是否处于在线调试模式，在线调试模式下PLC运动类指令控制无效
9117	指令暂未开发	指令暂未开发	指令功能还没有开发
9118	指令加（减）速超过最大加速度	指令加（减）速超过最大加速度	检查指令的加（减）速度是否超过最大加速度
9119	MC_Jog指令的目标速度超过最大点动速度	MC_Jog指令的目标速度超过最大点动速度	检查Jog指令的目标速度是否超过最大点动速度
9120	目标速度超过最大速度	目标速度超过最大速度	检查指令的目标速度是否超过最大速度
9121	Jog指令正向运动和负向运动信号同时有效	Jog指令正向运动和负向运动信号同时有效	Jog指令的JogForward和JogBackward信号不能同时有效
9122	EtherCAT总线轴中没有映射控制字	EtherCAT总线轴中没有映射控制字	PDO中添加控制字然后映射到轴中
9123	EtherCAT总线轴中没有映射目标位置	EtherCAT总线轴中没有映射目标位置	PDO中添加目标位置后映射到轴中
9124	EtherCAT总线轴中没有映射目标力矩	EtherCAT总线轴中没有映射目标力矩	PDO中添加目标力矩然后映射到轴中
9125	EtherCAT总线轴中没有映射状态字	EtherCAT总线轴中没有映射状态字	PDO中添加状态字然后映射到轴中
9126	EtherCAT总线轴中没有映射当前位置	EtherCAT总线轴中没有映射当前位置	PDO中添加反馈位置然后映射到轴中
9127	EtherCAT总线轴中没有映射0x60fd	EtherCAT总线轴中没有映射0x60fd	PDO中添加0x60fd然后映射到轴中
9128	EtherCAT总线轴中没有映射当前力矩	EtherCAT总线轴中没有映射当前力矩	PDO中添加当前力矩然后映射到轴中
9129	EtherCAT总线轴中没有映射探针控制字	EtherCAT总线轴中没有映射探针控制字	PDO中添加探针控制字然后映射到轴中

错误代码	错误信息	详细说明	解决措施
9130	EtherCAT总线轴中没有映射探针状态字	EtherCAT总线轴中没有映射探针状态字	PDO中添加探针状态字然后映射到轴中
9131	EtherCAT总线轴中没有映射探针位置	EtherCAT总线轴中没有映射探针位置	PDO中添加探针状态机然后映射到轴中
9132	正在运行中断定长指令，探针通道被占用	正在运行中断定长指令，探针通道被占用	探针指令和中断定长指令不可同时占用同一个探针通道，程序中同时调用时中断定长指令优先级更高
9133	启用了虚轴模式	启用了虚轴模式	当前指令不支持虚轴模式
9135	中断定长没有触发中断信号	中断定长没有触发中断信号	调用中断定长指令，在定位完成后没有检测到中断信号
9136	中断定长过程中探针通道被其他指令占用	中断定长过程中探针通道被其他指令占用	中断定长过程中探针通道不能被占用
9137	总线驱动器中没有映射控制模式0x6060	总线驱动器中没有映射控制模式0x6060	PDO中添加0x6060然后映射到轴中
9138	总线驱动器中没有映射控制模式0x6061	总线驱动器中没有映射控制模式0x6061	PDO中添加0x6061然后映射到轴中
9139	在原点回归过程中重复调用MC_home指令	在原点回归过程中重复调用MC_home指令	在回原过程中不允许重复调用回原指令
9140	指令的目标力矩超过最大值	指令的目标力矩超过最大值	检查指令的目标力矩是否超过正负力矩限制值
9141	总线驱动器中没有映射最大速度	总线驱动器中没有映射最大速度	在PDO中添加0x607f,然后映射到轴中
9142	急停指令有效	急停指令有效	查看是否已经调用了急停指令
9143	重复调用急停指令	重复调用急停指令	查看是否重复调用了急停指令
9144	点动运行过程中碰到限位	点动运行过程中碰到限位	查看限位是否有效
9145	目标位置超过9999999	单精度浮点数超过9999999后会丢失精度，目标位置不能超过该值	1、检查目标位置是否正确，重新设定目标位置 2、修改齿轮比，保证需要设定的目标位置不大于9999999
9146	目标速度超过9999999	单精度浮点数超过9999999后会丢失精度，目标速度不能超过该值	1、检查目标速度是否正确，重新设定目标速度 2、修改齿轮比，保证需要设定的目标速度不大于9999999
9147	目标加速度超过9999999	单精度浮点数超过9999999后会丢失精度，目标加速度不能超过该值	1、检查目标加速度是否正确，重新设定目标加速度 2、修改齿轮比，保证需要设定的目标加速度不大于9999999
9148	目标减速度超过9999999	单精度浮点数超过9999999后会丢失精度，目标减速度不能超过该值	1、检查目标减速度是否正确，重新设定目标减速度 2、修改齿轮比，保证需要设定的目标减速度不大于9999999
9149	轴处于同步控制模式，且不允许被打断	1、当前轴正在执行插补执行，轴处于同步控制模式，调用了单轴运动类型指令，单轴运动指令报错	在插补期间不允许调用单轴运动类型指令
9400	轴组数量超过最大值	轴组数量超过允许的最大值	减少工程中轴组的数量，不要超过最大值

错误代码	错误信息	详细说明	解决措施
9401	轴组内有轴处于故障状态	轴组内有轴处于故障状态	查找轴组内哪些轴处于故障状态，查看单个轴的故障码并排除故障
9402	缓冲的插补指令数量大于8条	缓冲的插补指令数量大于8条	插补指令最多缓冲8条，查看缓冲的指令是否超过了8条
9403	轴重复使用	轴组内有轴被重复使用	每一个轴只能应用到一个轴组中，检查轴组内被重复使用的轴并更换成没有使用过的轴
9404	不能构成轴组	x轴或y轴不存在不能构成轴组	轴组内至少需要存在x轴和y轴，检查这两个轴是否存在
9405	指定的z轴不存在	指令中设定了z轴，但是组态中z轴不存在	检查指令中设置的z轴是否存在
9406	指定的辅助轴不存在	指令中设置了辅助轴，但是组态中的辅助轴不存在	检查指令中设置的辅助轴是否存在
9407	轴组ID重复使用	指定的轴ID已经被使用	每一个轴组ID只能被使用一次，请更换轴组ID
9408	轴配置失败	轴配置失败	检查轴组中的是否有轴配置失败，如果是，则检查单板软件和后台是否匹配
9409	轴ID小于0	轴ID小于0	检查轴组中指定的轴的ID是否小于0
9410	轴组没有释放	短时间内重复触发同一条MC_SetAxesGroup指令，导致轴组没有释放	不允许在MC_SetAxesGroup的busy信号输出有效期间重复触发该指令
9411	MC_GroupStop指令被打断	MC_GroupStop指令被打断	查看是否在指令有效期间调用的比该指令优先等级更高的指令，导致指令被打断
9412	圆弧插补指令CircAxes超范围	圆弧插补指令CircAxes超范围	检查圆弧插补指令CircAxes是否超范围
9413	圆弧插补指令CircMode超范围	圆弧插补指令CircMode超范围	检查圆弧插补指令CircMode是否超范围
9414	圆弧插补指令PathChoice超范围	圆弧插补指令PathChoice超范围	检查圆弧插补指令PathChoice是否超范围
9415	停止指令StopMode超范围	停止指令StopMode超范围	检查停止指令StopMode是否超范围
9416	X坐标轴设置为环形模式	X坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9417	y坐标轴设置为环形模式	y坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9418	z坐标轴设置为环形模式	z坐标轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9419	辅助轴设置为环形模式	辅助轴设置为环形模式	插补指令中的运动控制轴不允许设置为环形模式
9420	圆弧插补指令重复触发	圆弧插补指令重复触发	同一条圆弧插补指令在Busy输出有效期间不允许重复触发
9421	直线插补指令重复触发	直线插补指令重复触发	同一条直线插补指令在Busy输出有效期间不允许重复触发
9422	获取轴组失败	获取轴组失败	检查GroupID指定的轴组是否已经通过MC_SetAxesGroup创建
9423	轴配置失败	轴配置失败	在轴还没有配置完成的情况下触发了指令。\r\n检查轴组内的所有轴的通讯状态是否处于“Axis ready”状态。

错误代码	错误信息	详细说明	解决措施
9424	有轴处于非使能状态	有轴处于非使能状态	在Disabled 状态下不允许调用插补指令
9425	有轴正在执行单轴类运动指令	有轴正在执行单轴类运动指令期间触发了插补指令	当有轴正在执行单轴运动类指令而不处于StandStill状态时不允许调用插补指令
9426	有轴处于Stopping状态	有轴处于Stopping状态	当有轴执行了MC_Stop指令而处于Stopping状态时不允许调用插补指令
9427	轴组处于停止状态	轴组处于停止状态	MC_GroupStop指令有效，不允许执行插补指令
9428	有轴处于原点回归状态	有轴处于原点回归状态	当有轴执行了MC_Home指令而处于Homing状态时不允许调用插补指令
9429	有轴正在执行设置位置指令	有轴正在执行设置位置指令	当有轴执行了MC_SetPosition指令正在设置当前位置时不允许调用插补指令
9430	有轴处于调试状态	有轴处于调试状态	当有轴正在处于调试状态时不允许调用插补指令
9431	插补中有轴进入调试状态，打断其他轴的执行	插补中有轴进入调试状态，打断其他轴的执行	检查是否在插补过程中有轴进入了调试状态，从而打断了其他轴的运行
9432	内存申请失败	内存申请失败	检查内存是否用完。\r\n联系厂家
9433	目标速度为0或小于0	目标速度为0或小于0	指令的目标速度必须大于0
9434	目标加速度为0或小于0	目标加速度为0或小于0	指令的目标加速度必须大于0
9435	目标减速度为0或小于0	目标减速度为0或小于0	指令的目减标速度必须大于0
9436	曲线类型设置超范围	曲线类型设置超范围	插补指令只支持T型曲线，检查是否设置了其他值。
9437	运动模式（AbsRelMode）设置不合理	运动模式（AbsRelMode）设置不合理	仅支持相对定位和绝对定位两中模式，检查是否设置了其他值。
9438	缓冲模式(BufferMode)设置不合理	缓冲模式(BufferMode)设置不合理	检查缓冲模式参数设置
9439	插补模式(InsertMode)设置不合理	插补模式(InsertMode)设置不合理	检查插补模式参数设置是否合理
9440	有轴处于故障状态导致停机	有轴处于故障状态导致停机	查找处于故障状态的轴，根据单轴故障码解决轴的故障
9441	重复调用MC_GroupStop指令	重复调用MC_GroupStop指令	在一条MC_GroupStop指令有效期间不能重复触发或多次调用其他MC_GroupStop指令
9442	数据缓冲区非空	数据缓冲区非空,内部故障	联系厂家
9443	不能构成圆弧	参数设置不合理导致不能构成圆弧	更新参数
9444	不能构成圆弧	圆弧插补指令的起点、经过点和终点重合，不能构成圆	检查圆弧指令的给定参数，保证起点、经过点和终点可以构成一个圆
9445	指令缓冲区满	指令缓冲区满	内部故障，联系厂家
9446	x轴超过最大速度	x轴超过最大速度	按照当前参数，x轴坐标轴的目标分速度将超过最大速度
9447	y轴超过最大速度	y轴超过最大速度	按照当前参数，y轴坐标轴的目标分速度将超过最大速度

错误代码	错误信息	详细说明	解决措施
9448	z轴超过最大速度	按照当前参数，z轴坐标轴的目标分速度将超过最大速度	按照当前参数，z轴坐标轴的目标分速度将超过最大速度
9449	辅助轴超过最大速度	辅助轴超过最大速度	按照当前参数，辅助轴的目标分速度将超过最大速度
9450	获取轴组数量数量失败	获取轴组数量数量失败	后台版本过低，更实用最新发布的版本
9451	内部故障	内部故障	联系厂家
9452	StandStill状态不允许调用本指令	StandStill状态不允许调用本指令	本指令在StandStill状态下不允许调用
9453	超过最大速度	超过轴组设置界面的最大速度	检查指令的目标速度是否超过轴组组态界面中的最大速度
9454	超过最大加（减）速度	超过最大加（减）速度	检查指令中的目标加（减）速度是否超过组态界面中的最大加速度
9501	EtherCAT总线驱动器错误	驱动器错误,驱动器对象字典0x603F故障码的值为0x%x{16:16}	查看总线驱动器手册确定驱动器故障类型后排除故障。
9502	驱动器断使能	驱动器断使能	在运动期间驱动器状态字0x6041切换到非使能状态。
9503	碰到限位	碰到限位	1、检查是否配置软限位且到达软限位 2、检查是否触碰硬件限位信号
9505	修改控制模式失败	修改控制模式失败	1、检查网络通信是否收到干 2、检查驱动器是否支持对象字典0x6060
9508	原点回归失败	原点回归失败	1、通过驱动器故障码查找驱动器原点回归失败原因 2、检查原点回归是否超时
9509	轴内部计算精度错误	轴内部计算精度错误	检查指令的浮点型数据是否超过单精度浮点数的精度范围
9510	跟随误差超范围	跟随误差超范围	1、检查加速度是否过大 2、检查跟随误差设置是否偏小
9512	运行过程中伺服驱动器断开连接	运行过程中伺服驱动器断开连接	1、检查驱动器是否正常 2、检查网线是否正常 3、检查通讯是否收到强烈干扰
9513	驱动器故障导致原点回归失败	驱动器的状态字反馈故障导致原点回归失败	查看驱动器的故障码解决故障
9514	原点回归偏移超过32位导致回原失败	原点回归偏移超过32位导致回原失败	计算当前原点回归的偏移乘以齿轮比是否超过32位，如果超过32需要修改齿轮比
9515	从站丢失导致原点回归失败	EterhCAT驱动器丢失导致回原失败	内部故障，寻求厂家帮助
9516	SDO写对象字典0x607C失败导致原点回归失败	SDO写对象字典0x607C失败导致原点回归失败	1、检查驱动时是否支持0x607C 2、检查网络通信质量
9517	SDO写对象字典0x6060为6时失败导致原点回归失败	SDO写对象字典0x6060为6时失败导致原点回归失败	1、将0x6060配置到PDO中 2、检查网络通信质量

错误代码	错误信息	详细说明	解决措施
9518	SDO读对象字典0x6061失败导致原点回归失败	DO读对象字典0x6061失败导致原点回归失败	1、将0x6061配置到PDO中 2、检查网络通信质量
9519	SDO写对象字典0x6060为8时失败导致原点回归失败	SDO写对象字典0x6060为8时失败导致原点回归失败	1、将0x6060配置到PDO中 2、检查网络通信质量
9551	控制模式切换失败	控制模式切换失败	检查网络通信是否收到干扰
9552	目标速度为0	目标速度为0	检查位置类指令的目标速度是否合理
9601	MC_MoveAbsolute指令参数异常导致停机	MC_MoveAbsolute指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9602	MC_MoveRelative指令参数异常导致停机	MC_MoveRelative指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9603	MC_MoveVelocity指令异常导致停机	MC_MoveVelocity指令异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9604	MC_Jog指令异常导致停机	MC_Jog指令异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9606	MC_MoveBuffer指令异常导致停机	MC_MoveBuffer指令异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9607	MC_MoveFeed指令参数异常导致停机	MC_MoveFeed指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9608	MC_Stop指令参数异常导致停机	MC_Stop指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9609	MC_MoveTorque指令参数异常导致停机	MC_MoveTorque指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9610	MC_Halt指令参数异常导致停机	MC_Halt指令参数异常导致停机	检查报错的指令，根据指令的报错码进一步确定故障
9800	读不到轴运动控制轴的数量	读不到轴运动控制轴的数量	更换后台版本
9801	运动控制轴数量超范围	运动控制轴数量超范围	H5U最多支持32个轴，减小轴的数量
9802	轴内部申请空间失败	轴内部申请空间失败	1、检查内存是否用完 2、联系厂家
9803	轴获取参数失败	轴获取参数失败	更换后台版本
9804	获取从站失败	获取从站失败	更换后台版本
EtherCAT			
8001	读取主站配置信息失败	读取主站配置信息失败	查看单板软件和后台版本是否匹配
8002	获取从站配置参数失败	获取从站配置参数失败	查看单板软件和后台版本是否匹配
8003	EtherCAT启动超时	EtherCAT启动超时	1、查看网络是否接好 2、查看组态和实际接入的从站是否一致 3、查看从站类型是否匹配
8004	申请主站失败	申请主站失败	重启PLC
8200	从站启动参数写SDO失败	从站启动参数写SDO失败	1、检查启动参数列表中是否有从站不支持的对象字典 2、检查对象字典的值是否超范围

错误代码	错误信息	详细说明	解决措施
8201	运行过程中从站丢失	运行过程中从站丢失	1、检查从站之间的网络是否断线 2、检查从站是否断电
8202	从站状态机切换到非OP模式	从站状态机切换到非OP模式	1、检查从站之间的网络是否断线 2、检查从站是否断电
8203	从站状态机切换失败	从站状态机切换失败	—
8204	从站类型不匹配	从站类型不匹配	1、检查网线是否接反 2、检查组态中的设备和实际接入的设备是否匹配
8205	PDO地址错误	PDO地址错误	1、检查内存是否用尽 2、确认后台和单板软件的版本是否匹配 3、断电重启PLC
8206	PDO长度错误	PDO长度错误	确认后台和单板软件的版本是否匹配
8301	切换到INIT状态失败	切换到INIT状态失败	确认从站状态机是否支持状态转换
8302	切换到PerOP状态失败	切换到PerOP状态失败	确认从站是否支持CoE协议
8304	切换到SafeOP状态失败	切换到SafeOP状态失败	确认PDO通信配置是否正确
8308	切换到OP状态失败	切换到OP状态失败	1、确认网络通信质量 2、确认EtherCAT任务周期是否合理
8310	FMMU单元配置错误	FMMU单元配置错误	确认从站是否支持FMMU单元
8311	邮箱配置错误	邮箱配置错误	确认从站是否支持SM单元
8400	ECTA组态错误	ECTA组态错误	确认组态中扩展模块和实际接入的扩展模块是否相同
8401	ECTA硬件错误	ECTA硬件错误	1、确认ECTA和扩展模块之间的是否松动 2、更换ECTA
8402	ECTA扩展模块错误	ECTA扩展模块错误	1、查找ERR灯点亮的扩展模块 2、通过ETC_ReadParameter_CoE读取故障模块的诊断对象字典 3、查看ECTA应用手册中扩展模块的诊断对象字典的描述确认扩展模块的故障类型并排除错误

$ADD — 字符串的组合 213
$MOV — 字符串的传送 222

A

ABS — 绝对值指令 179
ACOS — 浮点数COS^{-1}运算 105
ADD — 二进制数据加法 79
ALT — 交替输出指令 47
ANB — 串联回路方块 49
AND<= — AND触点比较小于等于 63
AND<> — AND触点比较不等于 63
AND — 串联常开触点 39
AND^ — AND逻辑异或运算 76
AND&— AND逻辑与运算 76
AND= — AND触点比较等于 63
AND> — AND触点比较大于 63
AND>= — AND触点比较大于等于 63
AND|— AND逻辑或运算 76
AND< — AND触点比较小于 63
ANDF — 与脉冲(F)下降沿检测串行连接 39
ANDP — 与脉冲上升沿检测串行连接 39
ANDZ<> — 绝对值与触点比较不等于 72
ANDZ= — 绝对值与触点比较等于 72
ANDZ> — 绝对值与触点比较大于 72
ANDZ>= — 绝对值与触点比较大于等于 72
ANDZ<= — 绝对值与触点比较小于等于 72
ANDZ< — 绝对值与触点比较小于 72
ANI — 串联常闭触点 39
ASCI — HEX→ASCII转换指令 140
ASIN 浮点数SIN^{-1}运算 104
ATAN — 浮点数TAN^{-1}运算 106

B

BAND — 字或双字位数据位与触点指令 96
BANDI — 字或双字位数据位与非触点指令 .97
BCD — 二进制数据转BCD指令 126
BIN— BCD转二进制数据指令 127
BINDA — BIN→10进制ASCII的转换 131
BITW — 位元件赋值给字元件指令 138
BK- — 数据块减法运算 188
BK+ — 数据块加法运算 187
BKCMP<> — 矩阵不等于比较 196
BKCMP= — 矩阵等于比较 196
BKCMP> — 矩阵大于比较 196
BKCMP>= — 矩阵大于等于比较 196
BKCMP<= — 矩阵小于等于比较 196
BKCMP< — 矩阵小于比较 196
BLD — 字或双字位数据位触点指令 95
BLDI — 字或双字位数据位反触点指令 95
BMOV — 数据批量传送 146
BON — ON位数判断 176
BOR — 字或双字位数据位或触点指令 98
BORI — 字或双字位数据位或非触点指令 98
BOUT — 字或双字位数据输出指令 99
BRST — 字或双字位数据输出指令 101
BSET — 字或双字位数据置位指令 100
BTOW — 字节单位的数据结合 136
BZAND — 死区控制 114

C

CALL — 子程序调用 52
CCD — 校验码指令 181
CJ — 条件跳转指令 51
CML — 取反传送 149
CMP — 比较指令 150
COS — 浮点COS运算指令指令 104
COSH — 浮点数COSH运算 109
CRC — 累加和校验码指令 182

D

DABIN – 10进制ASCII→BIN的转换 130
DEC — 减一指令 88
DEG — 浮点数弧度转换角度 108
DI — 中断禁止 56
DIS – 16位数据的4位分离 135
DIV — 二进制除法指令 81
DWTOW – 32位字元件转换16位字元件指令 135

E

EABS — 浮点绝对值指令 180
EADD — 浮点加法指令 83
EBCD — 浮点→十进制浮点转换指令 129
EBIN — 十进制浮点→浮点转换指令 129
ECMP — 浮点比较指令 151
ECT_ReadParameter_CoE — 读取从站的SDO参数 288
ECT_WriteParameter_CoE — 写入从站的SDO参数 290
EDIV — 浮点除法指令 86
EI — 中断许可 56
EMOV — 浮点赋值指令 145
EMUL — 浮点乘法指令 85
ENEG — 浮点数符号取反 93
ESQR — 浮点开方指令 123
ESTR — 浮点数转换字符串 207
ESUB — 浮点减法指令 84
EVAL — 字符串转换浮点数 209
EXP — 浮点数指数运算 120
EZCP — 浮点区域比较指令 153

F

FAND<= — 浮点数与触点比较小于等于 68
FAND<> — 浮点数与触点比较不等于 68
FAND= — 浮点数与触点比较等于 68
FAND> — 浮点数与触点比较大于 68
FAND>= — 浮点数与触点比较大于等于 68
FAND< — 浮点数与触点比较小于 68
FDEL — 数据表的数据删除 159
FINS — 数据表的数据插入 160
FLDD<= — 浮点数触点比较小于等于 67
FLDD<> — 浮点数触点比较不等于 67
FLDD= — 浮点数触点比较等于 67
FLDD> — 浮点数触点比较大于 67
FLDD>= — 浮点数触点比较大于等于 67
FLDD< — 浮点数触点比较小于 67
FLT — 二进制整数至浮点数转换指令 128
FMOV — 多点传送 148
FOR<= — 浮点数或触点比较小于等于 69
FOR<> — 浮点数或触点比较不等于 69
FOR — 循环范围开始 57
FOR= — 浮点数或触点比较等于 69
FOR> — 浮点数或触点比较大于 69
FOR>= — 浮点数或触点比较大于等于 69
FOR< — 浮点数或触点比较小于 69

H

HC_ArrayCompare — 高速计数器数组比较 247
HC_Compare — 高速计数器比较 245
HC_Counter — 高速计数器使能 237
HC_Preset — 高速计数器预置值 234
HC_StepCompare — 高速计数器等间距离比较 249
HC_TouchProbe — 高速计数器探针 240
HEX — ASCII→HEX转换指令 142
HOUR — 计时表指令 233
HTOS — 小时/分/秒数据转换为秒 228

I

INC - 加一指令 87
INSTR — 字符串的检索 215
INT — 浮点→整数转换指令 125
INV — 运算结果取反 50
IRET — 中断返回 55

L

LBL — 标号指令 51
LD<= — 触点比较小于等于 65
LD<> — 触点比较不等于 64
LD — 加载常开触点 38
LD^ — LD逻辑异或运算 75
LD&— LD逻辑与运算 75
LD= — 触点比较等于 64
LD> — 触点比较大于 64
LD>= — 触点比较大于等于 65
LD|— LD逻辑或运算 75
LD< — 触点比较小于 64
LDF — 取脉冲下降沿 38
LDI — 加载常闭触点 38
LDP — 取脉冲上升沿 38
LDZ<= — 绝对值触点比较小于等于 71
LDZ<> — 绝对值触点比较不等于 71
LDZ= — 绝对值触点比较等于 71
LDZ> — 绝对值触点比较大于 71
LDZ>= — 绝对值触点比较大于等于 71
LDZ< — 绝对值触点比较小于 71
LEFT — 从字符串左侧开始取出 218
LEN — 检测字符串的长度 214
LIMIT — 上下限限位控制 112
LOG — 浮点数常用对数运算 121
LOGE — 浮点数自然对数运算 122
LRC — ASCII 模式LRC校验码运算 184

M

MAND — 矩阵与运算 189
MC_GroupPause— 暂停轴组运行指令 389
MC_GroupStop— 停止轴组运行指令 387
MC_Halt — 暂停指令 350
MC_Halt_CO — 通讯控制伺服轴终止运动 398
MC_Home — 原点回归指令 344
MC_Home_CO — 通讯控制轴原点回归 407
MC_ImmediateStop — 急停指令 366
MC_Jog — 点动运动指令 334
MC_Jog_CO — 通讯控制轴点动 409
MC_MoveAbsolute — 绝对定位指令 327
MC_MoveAbsolute_CO — 通讯控制轴绝对定位 405
MC_MoveBuffer — 多段位置定位指令 361
MC_MoveCircular— 圆弧插补指令 383
MC_MoveFeed — 中断定长指令 353
MC_MoveLinear — 直线插补指令 375
MC_MoveRelative — 相对定位指令 318
MC_MoveRelative_CO— 通讯控制轴相对定位 403
MC_MoveVelocity — 速度指令 324
MC_MoveVelocity_CO — 通讯控制轴速度运行模式 401
MC_Power — 使能控制指令 294
MC_Power_CO — 通讯控制伺服轴使能 394
MC_ReadActualPosition — 读取实际位置指令 304
MC_ReadActualPosition_CO — 通讯控制读取轴当前实际位置 397
MC_ReadActualTorque — 读取实际力矩指令 306
MC_ReadActualVelocity — 读取实际速度指令 307
MC_ReadActualVelocity_CO — 通讯控制读取轴当前实际速度 397
MC_ReadAxisError — 读取轴错误指令 301
MC_ReadDigitalInput — 读取数字量输入指令 302
MC_ReadParameter_CO — 通讯控制读取轴参数 413
MC_ReadStatus — 读取轴状态指令 298
MC_Reset — 复位故障指令 296
MC_Reset_CO — 通讯控制伺服轴故障复位 395
MC_SetPosition — 设置定位指令 308
MC_Stop — 停止指令 347
MC_Stop_CO — 通讯控制伺服轴停止 400
MC_TorqueControl — 力矩控制指令 339
MC_TouchProbe — 探针指令 311
MC_WriteParameter_CO — 通讯控制写入轴参数 411
MEAN — 平均值计算指令 111
MEF — 运算结果下降沿脉冲化 41
MEP — 运算结果上升沿脉冲化 41
MIDR — 从字符串中任意取出 221
MIDW — 字符串中任意替换 219
MINV — 矩阵反相运算 195
MOD — 除法求余指令 82
MOR — 矩阵或运算 191
MOV — 传送 144
MPP — 读出堆栈 49
MPS — 存入堆栈 49
MRD — 读出堆栈 49
MUL — 二进制数据乘法 81
MXNR — 矩阵用或运算 192
MXOR — 矩阵异或运算 193

N

NEG - 求补指令 92
NEXT — 循环范围结束 57
NOP — 无动作 50

O

OR<> — OR触点比较不等于 66
OR — 并联常开触点 40
OR^ — OR逻辑异或运算 77
OR&— OR逻辑与运算 77
OR= — OR触点比较等于 66
OR> — OR触点比较大于 66

OR>= — OR触点比较大于等于 66
OR|— OR逻辑或运算.....77
OR<= — OR触点比较小于等于 66
ORB — 并联回路方块 49
ORF — 与脉冲(F)下降沿检测并行连接 40
OR< — OR触点比较小于.....66
ORI — 并联常闭触点 40
ORP — 与脉冲上升沿检测并行连接 40
ORZ<= — 绝对值或触点比较小于等于.....73
ORZ<> — 绝对值或触点比较不等于 73
ORZ= — 绝对值或触点比较等于 73
ORZ> — 绝对值或触点比较大于 73
ORZ>= — 绝对值或触点比较大于等于.....73
ORZ< — 绝对值或触点比较小于 73
OUT — 驱动线圈 43
OUTSTL — 输出程序跳至副母线.....60

P

PID — PID运算指令 181
PLF — 脉冲下降沿检测线圈指令 47
PLS — 脉冲上升沿检测线圈指令 46
POP — 后入数据的读取.....161
POW — 浮点数权值指令.....124
PT* — 指针变量触点比较指令.....266
PTADD — 指针变量地址加法指令 263
PTDEC — 指针变量地址减1指令 263
PTGET — 指针变量赋值指令.....261
PTINC — 指针变量地址增1指令.....262
PTMOV — 指针变量相互赋值指令 265
PTSUB — 指针变量地址减法指令 264

R

RAD — 浮点数角度转换弧度 107
RAMP — 斜坡信号指令 185
RAND — 产生有范围限制的随机数据.....177
RCL — 数据带进位循环左移.....166
RCR — 数据带进位循环右移 165
RET — 程序返回主母线.....59
RIGHT — 从字符串右侧开始取出 216
ROL — 数据循环左移.....164
ROR — 数据循环右移 164
RST — 接点或缓存器清除 44
RSTSTL — 清除程序跳至副母线.....61

S

SCL — 定坐标(不同点坐标数据).....116
SCL2 — 定坐标2(X/Y坐标数据) 118
SER — 数据查找指令 157
SerialSR — 串口自由协议发送接收 268
SET — 置位动作保存线圈 43
SETSTL — 置位程序跳至副母线 60
SFL — 数据n位左移(带进位) 173
SFR — 数据n位右移(带进位).....172
SFRD — 数据“先进先出”读出 171
SFTL — 多个位变量左移.....168
SFTR — 多个位变量右移 167
SFWR — 数据“先进先出”写入.....171
SIN — 浮点SIN运算指令.....102
SINH — 浮点数SINH运算 108
SMOV — 移位传送 147
SORT — 数据排序指令 154
SORT2 — 数据排序2.....155
SQR — 开方运算指令.....123
SRET — 子程序返回 54
SSRET — 子程序带条件返回.....55
STL — 程序跳至副母线 59
STOH — 秒转换为小时/分/秒数据.....229
STR — 整数→字符串的转换.....198
STRMOV — 字符串赋值指令 202
SUB — 二进制数据减法 80
SUM — 统计ON位数指令 177
SWAP — 高低字节交换指令 175

T

TACR — 时间累加定时器.....259
TADD — 时钟数据加法运算指令.....226
TAN — 浮点TAN运算指令 103
TANH — 浮点数TANH运算 110
TCMP — 时钟数据比较指令 224
TCP_Accept TCP接受连接请求指令.....272
TCP_Close — TCP关闭连接指令.....275
TCP_Connect — TCP发起连接请求指令..273
TCP_Listen — TCP监听指令.....270
TCP_Receive — TCP接收数据指令 278
TCP_Send — TCP发送数据指令.....276
TOFR — 关断延时定时器.....257
TONR — 接通延时定时器.....256
TPR — 脉冲定时器.....254
TRD — 时钟数据读取指令.....231
TSUB — 时钟数据减法运算指令.....227
TWR — 时钟数据写入指令.....232

U

UDP_Bind — UDP套接字绑定指令.....284
UDP_Receive — UDP接收数据指令.....282
UDP_Send — UDP发送数据指令.....285
UNI – 16位数据的4位结合 134

W

WAND - 逻辑与指令 89
WBIT — 字元件赋值给位元件指令.....133
WDT — 监视定时器复位指令 56
WOR — 逻辑或指令.....90
WSFL — 多个字变量左移 170
WSFR — 多个字变量右移 169
WSUM — 算出数据合计值.....111
WTOB — 字节单位的数据分离.....137
WTODW — 16位字元件转换32位字元件指令.....139
WXOR — 逻辑异或指令.....91

X

XCH — 交换指令.....178

Z

ZCP — 区域比较.....152
ZONE — 区域控制.....115
ZRST — 全部数据复位.....45
ZSET — 批量置位指令.....44