MC系列PLC编程手册

黄石市科威自控有限公司

资料编号：202507-V1.00

联系电话：0714-3802747

- 此手册版权为黄石市科威自控有限公司所有
- 此手册内容支持产品系列如下表所列：

系类	详细名称	备注
MC	MC-16M16T	控制器产品

版本变更记录

版本	年份	修订内容	适用于内核版本（寄存器D8255）
V1.00	2025-07	第一版	V20042

目 录

目 录 3
一、 MC系列PLC硬件手册 1
1、 安全注意事项 1
2、 MC系列PLC选型 3
2.1 主要功能 3
2.2 型号说明 4
4
3、 MC系列PLC安装与尺寸 5
3.1 MC系列PLC安装 5
3.1.1 安装场所 5
3.1.2 环境条件 5
3.1.3 安装方向与间隔 5
3.2 MC系列PLC尺寸 6
4、 MC系列PLC接口 7
4.1 硬件接口及端子分布 7
4.2 电源配线 10
4.3 状态识别 10
4.4 开关量输入 11
4.5 开关量输出 11
二、基本应用指令手册 13
1. 前言 13
1.1 MC系列PLC编程软件 13
1.2 MC系列PLC编程语言 13
1.2.1 编程语言的种类 13
1.2.2 程序的互换性 14
2. 概要 14
2.1 基本功能介绍 14
2.2 软元件介绍 15
2.2.1 软元件一览表 15
2.2.2 软元件说明 17
2.3 应用指令介绍 19
2.3.1 主要应用指令 19
2.4 程序内存介绍 20
2.4.1 程序内存 20
2.4.2 程序加密 20
2.4.3 控制器状态 24
2.5 停电保持 24
2.5.1 停电保持 24
2.5.2 软元件停电保持动作 24
2.6 梯形图运行时序 26
2.6.1 扫描周期 26

2.6.2 梯形图运行时序图 26
3、指令一览 27
3.1 基本指令 27
3.2 步进梯形图指令 28
3.3 应用指令 29
4、软元件的作用和功能 35
4.1 软元件一览表 35
4.2 输入继电器 [X] 37
4.2.1 输入继电器编号 37
4.2.2 功能与动作说明 37
4.2.3 动作时序 38
4.2.4 输入处理便利功能 38
4.2.5 程序应用 39
4.3 输出继电器 [Y] 40
4.3.1 输出继电器编号 40
4.3.2 功能与动作说明 40
4.3.3 动作时序 40
4.3.4 Y 输出处理便利功能 41
4.3.5 程序应用 41
4.4 辅助继电器[M] 43
4.4.1 辅助继电器编号 43
4.4.2 功能与动作说明 43
4.4.3 程序应用 44
4.5 状态继电器[S] 45
4.5.1 状态继电器编号 45
4.5.2 功能与动作说明 45
4.5.3 步序状态 S 45
4.5.4 程序应用 46
4.6 定时器[T] 47
4.6.1 定时器编号（10 进制表示） 47
4.6.2 功能和动作说明 47
4.6.3 设定值指定方法 48
4.6.4 定时器精度 48
4.6.5 定时器做数据软元件使用 48
4.6.6 程序应用 48
4.7 计数器[C] 48
4.7 计数器[C] 49
4.7.1 计数器编号（10 进制表示） 49
4.7.2 计数器特征 49
4.7.3 程序应用 49
4.7.4 计数器计数方向 50
4.7.5 功能与动作说明 50
4.7.6 设定值指定方法 51

4.7.7 作为数据软元件使用 52
4.8 高速计数器[C] 53
4.8.1 高速计数器的类型和编号 53
4.8.2 高速计数器的外部输入分配 53
4.8.3 高速计数器的使用 54
4.8.4 当前值更新时序及比较 55
4.8.5 相关软元件 55
4.8.6 关于外部复位输入信号的逻辑变更 56
4.8.7 倍频 56
4.8.8 使用注意事项 56
4.8.9 高速计数器的响应频率 56
4.8.10 程序应用 56
4.9 数据寄存器[D] 58
4.9.1 数据寄存器[D]编号 58
4.9.2 数据寄存器构造 58
4.9.3 功能和动作说明 58
4.9.4 程序应用 60
4.10 扩展寄存器[R] 61
4.10.1 扩展寄存器的编号 61
4.10.2 扩展寄存器的构造 62
4.10.3 功能和动作说明 62
4.10.4 程序应用 63
4.11 变址寄存器[V、Z] 64
4.11.1 变址寄存器的编号 64
4.11.2 变址寄存器的构造 64
4.11.3 功能和动作说明 65
4.11.4 程序应用 65
4.12 指针[P]、[I] 66
4.12.1 指针的编号 66
4.12.2 分支用指针的功能和动作实例 66
4.12.3 程序应用 67
4.12.4 指针 I 的功能和动作说明 67
4.12.5 程序应用 68
5. 应用指令构造 69
5.1 指令格式 69
5.1.1 指令符号 69
5.1.2 指令操作数 69
5.2 指令形式和执行形式 69
5.3 操作数数值类型 70
5.3.1 数值种类 70
5.3.2 数值间转换 71
5.4 操作数对象软元件 71
5.4.1 常数 K，H，E 指定 71

5.4.2 位软元件指定 72
5.4.3 字软元件指定 73
5.4.4 变址修饰 73
5.5 指令说明 74
6. 编程前须知 75
6.1 编程方面基本注意事项 75
6.1.1 程序步骤及执行顺序 75
6.1.2 双线圈输出 76
6.2 输入输出响应延时 77
6.3 程序流程控制指令之间的相互关系 78
7. 基本逻辑指令 80
7.1 [LD]取,[LDI]取反指令 81
7.1.1 概要 81
7.1.2 功能和动作说明 81
7.1.3 注意要点 82
7.1.4 程序应用 82
7.2 [OUT]输出指令 83
7.2.1 概要 83
7.2.2 功能和动作说明 83
7.2.3 注意要点 85
7.2.4 程序应用 85
7.3 [AND]与,[ANI]与反指令 86
7.3.1 概要 86
7.3.2 功能和动作说明 86
7.3.3 注意要点 87
7.3.4 程序应用 87
7.4 [OR]或、[ORI]或反指令 88
7.4.1 概要 88
7.4.2 功能和动作说明 88
7.4.3 注意要点 89
7.4.4 程序应用 89
7.5 [LDP]、[LDF]、[ANDP]、[ANDF]、[ORP]、[ORF]指令 91
7.5.1 概要 91
7.5.2 功能和动作说明 92
7.5.3 注意要点 94
7.5.4 程序应用 95
7.6 [ORB]指令 97
7.6.1 概要 97
7.6.2 功能和动作说明 97
7.6.3 注意要点 97
7.6.4 程序应用 98
7.7 [ANB]指令 99
7.7.1 概要 99

7.7.2 功能和动作说明 99
7.7.3 注意要点 99
7.7.4 程序应用 100
7.8 [MPS]、[MRD]、[MPP]指令 101
7.8.1 概要 101
7.8.2 功能和动作说明 101
7.8.3 注意要点 103
7.8.4 程序应用 104
7.9 [INV]指令 105
7.9.1 概要 105
7.9.2 功能和动作说明 105
7.9.3 注意要点 106
7.9.4 程序应用 106
7.10 [ALT]交替输出指令 107
7.10.1 概要 107
7.10.2 功能和动作说明 107
7.10.3 注意要点 108
7.10.4 程序应用 108
7.11 [PLS]、[PLF]指令 110
7.11.1 概要 110
7.11.2 功能和动作说明 110
7.11.3 注意要点 112
7.11.4 程序应用 112
7.12 [SET]、[RST]指令 113
7.12.1 概要 113
7.12.2 功能和动作说明 113
7.12.3 注意要点 114
7.12.4 程序应用 115
7.13 对应计数器软元件C的[OUT],[RST]指令 116
7.13.1 概要 116
7.13.2 功能和动作说明 116
7.13.3 注意要点 116
7.13.4 程序应用 117
7.14 [NOP],[END]指令 118
7.14.1 概要 118
7.14.2 功能和动作说明 118
7.14.3 注意要点 119
7.14.4 程序应用 119
8. 传送·比较指令 120
8.1 [CMP]数据比较 121
8.1.1 概要 121
8.1.2 功能和动作说明 121
8.1.3 注意要点 122

8.1.4 程序应用 122
8.2 [ZCP]区间比较 124
8.2.1 概要 124
8.2.2 功能和动作说明 124
8.2.3 注意要点 125
8.2.4 程序应用 125
8.3 [MOV]传送 127
8.3.1 概要 127
8.3.2 功能和动作说明 127
8.3.3 注意要点 129
8.3.4 程序应用 130
8.4 [BMOV]成批传送 131
8.4.1 概要 131
8.4.2 功能和动作说明 131
8.4.3 注意要点 131
8.4.4 程序应用 132
8.5 [FMOV]多点传送 133
8.5.1 概要 133
8.5.2 功能和动作说明 133
8.5.3 注意要点 134
8.5.4 程序应用 135
8.6 [CML]反向传送 136
8.6.1 概要 136
8.6.2 功能和动作说明 136
8.6.3 注意要点 137
8.6.4 程序应用 137
9. 四则运算指令 138
9.1 [ADD]加法 139
9.1.1 概要 139
9.1.2 功能与动作说明 139
9.1.3 注意要点 139
9.1.4 程序应用 140
9.2 [SUB]减法 141
9.2.1 概要 141
9.2.2 功能与动作说明 141
9.2.3 注意要点 141
9.2.4 程序应用 142
9.3 [MUL]乘法 143
9.3.1 概要 143
9.3.2 功能与动作说明 143
9.3.3 注意要点 143
9.3.4 程序应用 144
9.4 [DIV]除法 145

9.4.1 概要 145
9.4.2 功能与动作说明 145
9.4.3 注意要点 146
9.4.4 程序应用 146
9.5 [INC]加一 147
9.5.1 概要 147
9.5.2 功能与动作说明 147
9.5.3 注意要点 147
9.5.4 程序应用 147
9.6 [DEC]减一 148
9.6.1 概要 148
9.6.2 功能与动作说明 148
9.6.3 注意要点 148
9.6.4 程序应用 148
9.7 [WAND]逻辑与 149
9.7.1 概要 149
9.7.2 功能与动作说明 149
9.7.3 注意要点 150
9.7.4 程序应用 150
9.8 [WOR]逻辑或 151
9.8.1 概要 151
9.8.2 功能与动作说明 151
9.8.3 注意要点 152
9.9 [WXOR]逻辑异或 153
9.9.1 概要 153
9.9.2 功能与动作说明 153
9.9.3 注意要点 153
9.9.4 程序应用 154
9.10 [NEG]求补 155
9.10.1 概要 155
9.10.2 功能和动作说明 155
9.10.3 注意要点 155
9.10.4 案例应用 155
9.11 [MEAN]求平均 157
9.11.1 概要 157
9.11.2 功能和动作说明 157
9.11.3 注意要点 157
9.11.4 案例应用 158
9.12 [SQR]开方 159
9.12.1 概要 159
9.12.2 功能与动作说明 159
9.12.3 注意要点 160
9.12.4 程序应用 160

10. 循环·移位 161
10.1 [ROL]循环左移 162
10.1.1 概要 162
10.1.2 功能和动作说明 162
10.1.3 注意要点 163
10.1.4 程序应用 164
10.2 [ROR]循环右移 165
10.2.1 概要 165
10.2.2 功能和动作说明 165
10.2.3 注意要点 167
10.2.4 程序应用 167
10.3[RCL]带进位循环左移 168
10.3.1 概要 168
10.3.2 功能和动作说明 168
10.3.3 注意要点 169
10.3.4 程序应用 169
10.4 [RCR]带进位循环右移 170
10.4.1 概要 170
10.4.2 功能和动作说明 170
10.4.3 注意要点 171
10.4.4 程序应用 172
10.5 [SFTL]位左移 173
10.5.1 概要 173
10.5.2 功能和动作说明 173
10.5.3 注意要点 174
10.5.4 程序应用 174
10.6 [SFTR]位右移 175
10.6.1 概要 175
10.6.2 功能和动作说明 175
10.6.3 注意要点 176
10.6.4 程序应用 176
10.7 [WSFL]字左移 177
10.7.1 概要 177
10.7.2 功能和动作说明 177
10.7.3 注意要点 178
10.7.4 程序应用 178
10.8 [WSFR]字右移 179
10.8.1 概要 179
10.8.2 功能和动作说明 179
10.8.3 注意要点 180
10.8.4 程序应用 180
10.9 [SFRD]移位读出 182
10.9.1 概要 182

10.9.2 功能和动作说明 182
10.9.3 注意要点 183
10.9.4 程序应用 183
10.10 [SFWR]移位写入 184
10.10.1 概要 184
10.10.2 功能和动作说明 184
10.10.3 注意要点 185
10.10.4 程序应用 185
11. 数据处理转换指令 186
11.1 [ZRST]批次复位 187
11.1.1 概要 187
11.1.2 功能和动作说明 187
11.1.3 注意要点 187
11.1.4 程序应用 188
11.2 [SWAP]高低字节交换 189
11.2.1 概要 189
11.2.2 功能与动作说明 189
11.2.3 程序应用 189
11.3 [XCH]数据交换 190
11.3.1 概要 190
11.3.2 功能与动作说明 190
11.3.3 注意要点 190
11.3.4 程序应用 191
11.4 [FLT]整型转浮点 192
11.4.1 概要 192
11.4.2 功能和动作说明 192
11.4.3 注意要点 192
11.4.4 程序应用 192
11.5 [INT]浮点转整型 193
11.5.1 概要 193
11.5.2 功能和动作说明 193
11.5.3 注意要点 193
11.5.4 程序应用 193
11.6 [BIN]BCD 转二进制 194
11.6.1 概要 194
11.6.2 功能和动作说明 194
11.6.3 注意要点 195
11.6.4 程序应用 196
11.7 [BCD]二进制转 BCD 197
11.7.1 概要 197
11.7.2 功能与动作说明 197
11.7.3 注意要点 198
11.7.4 程序应用 198

11.8 [ASCI][HEX→ASCII]十六进制 HEX 转 ASCII 199
11.8.1 概要 199
11.8.2 功能和动作说明 199
11.8.3 注意要点 200
11.8.4 程序应用 201
11.9 [HEX][ASCII→HEX]ASCII 转十六进制 HEX 202
11.9.1 概要 202
11.9.2 功能与动作说明 202
11.9.3 注意要点 203
11.9.4 程序应用 203
12. 浮点数运算指令 204
12.1 [DECMP]浮点比较 205
12.1.1 概要 205
12.1.2 功能和动作说明 205
12.1.3 注意要点 206
12.1.4 程序应用 206
12.2 [DEZCP]浮点区域比较 207
12.2.1 概要 207
12.2.2 功能和动作说明 207
12.2.3 注意要点 208
12.2.4 程序应用 208
12.3 [DEADD]浮点加法 209
12.3.1 概要 209
12.3.2 功能和动作说明 209
12.3.3 注意要点 209
12.3.4 程序应用 209
12.4 [DESUB]浮点减法 210
12.4.1 概要 210
12.4.2 功能和动作说明 210
12.4.3 注意要点 210
12.4.4 程序应用 210
12.5 [DEMUL]浮点乘法 211
12.5.1 概要 211
12.5.2 功能和动作说明 211
12.5.3 注意要点 211
12.5.4 程序应用 212
12.6 [DEDIV]浮点除法 213
12.6.1 概要 213
12.6.2 功能和动作说明 213
12.6.3 注意要点 213
12.6.4 程序应用 214
12.7 [DEBIN]浮点十进制转浮点二进制 215
12.7.1 概要 215

12.7.2 功能和动作说明 215
12.7.3 注意要点 216
12.7.4 程序应用 216
12.8 [DEBCD]浮点二进制转浮点十进制 217
12.8.1 概要 217
12.8.2 功能和动作说明 217
12.8.3 注意要点 217
12.8.4 程序应用 217
12.9 [DESQR]浮点开方 219
12.9.1 概要 219
12.9.2 功能和动作说明 219
12.9.3 注意要点 219
12.9.4 程序应用 220
12.10 [SIN]浮点 SIN 运算 221
12.10.1 概要 221
12.10.2 功能和动作说明 221
12.10.3 注意要点 221
12.10.4 程序应用 221
12.11 [COS]浮点COS 运算 223
12.11.1 概要 223
12.11.2 功能和动作说明 223
12.11.3 注意要点 223
12.11.4 程序应用 223
12.12 [TAN]浮点 TAN 运算 225
12.12.1 概要 225
12.12.2 功能和动作说明 225
12.12.3 注意要点 225
12.12.4 程序应用 225
12.13 [ASIN]浮点 ASIN 运算 227
12.13.1 概要 227
12.13.2 功能和动作说明 227
12.13.3 注意要点 227
12.13.4 程序应用 228
12.14 [ACOS]浮点 ACOS 运算 229
12.14.1 概要 229
12.14.2 功能和动作说明 229
12.14.3 注意要点 229
12.14.4 程序应用 229
12.15 [ATAN]浮点ATAN 运算 231
12.15.1 概要 231
12.15.2 功能和动作说明 231
12.15.3 注意要点 231
12.15.4 程序应用 231

12.16 [RAD]浮点数RAD运算 233
12.16.1 概要 233
12.16.2 功能和动作说明 233
12.16.3 注意要点 233
12.16.4 程序应用 233
12.17 [DEG]浮点DEG运算 234
12.17.1 概要 234
12.17.2 功能和动作说明 234
12.17.3 注意要点 234
12.17.4 程序应用 234
13. 专用指令 235
13.1 [PID]PID 控制 235
13.1.1 概要 235
13.1.2 功能和动作说明 235
13.1.3 程序应用 238
13.1.4 数据分析方法及步骤 243
13.1.5 PID参数的整定 245
14. 触点比较指令 247
14.1 [LD□]开始比较 248
14.1.1 概要 248
14.1.2 功能与动作说明 248
14.1.3 注意要点 248
14.1.4 程序应用 249
14.2 [OR□]并联比较 250
14.2.1 概要 250
14.2.2 功能与动作说明 250
14.2.3 注意要点 250
14.2.4 程序应用 251
14.3 [AND□]串联比较 252
14.3.1 概要 252
14.3.2 功能和动作说明 252
14.3.3 注意要点 252
14.3.4 程序应用 253
15 虚拟示波器 254
15.1 触发方式 254
15.2 触发单元 255
15.3 触发门限 255
15.4 采样周期 255
15.5 采样点数 255
15.6 记录起始寄存器序号 256
15.7 记录区间起始寄存器序号 257
15.8 记录区间终止寄存器序号 257
15.9 示波器状态设置/显示 257

15.10 当前采样点 258
15.11 通道工作模式 258
15.12 通道采样寄存器序号 259
16 特殊辅助继电器[M],特殊数据寄存器[D] 260
16.1 特殊辅助继电器[M]功能一览表 260
16.2 特殊辅助寄存器[D]功能一览表 265
16.3 实时时钟 270
16.4 固定周期脉冲 270
三、程序流程控制指令使用手册 271
1. 程序流程控制指令之间的相互关系 271
1.1. 程序流程控制指令之间的相互关系 271
2. 程序流程控制指令 272
2.1. 主控指令[MC-MCR] 272
2.1.1. 概要 272
2.1.2. 功能与动作说明 272
2.1.3. 注意要点 273
2.1.4. 使用说明 273
2.1.5. 案例分析 275
2.2. 子程序调用指令[CALL/CALLP] 276
2.2.1. 概要 276
2.2.2. 功能与动作说明 276
2.2.3. 注意要点 276
2.2.4. 使用说明 277
2.2.5. 案例分析 278
2.3. 循环指令[FOR-NEXT] 279
2.3.1. 概要 279
2.3.2. 功能与动作说明 279
2.3.3. 注意要点 279
2.3.4. 使用说明 280
2.3.5. 案例分析 281
2.4. 跳转指令[CJ-P] 282
2.4.1. 概要 282
2.4.2. 功能与动作要点 282
2.4.3. 注意要点 283
2.4.4. 使用说明 283
2.4.5. 案例分析 286
2.5. 步进控制指令[STL-RET] 287
2.5.1. 概要 287
2.5.2. 功能与动作要点 288
2.5.3. 注意要点 289
2.5.4. 使用说明 291
2.6. 主程序结束指令[FEND] 299
2.6.1. 概要 299

2.6.2. 功能与动作要点 299
2.6.3. 注意要点 300
2.7. 监视定时器[WDT] 301
四、中断使用手册 302
1、中断资源 302
1.1 中断种类 302
1.2 中断标志位 302
1.3 系统时间 303
2、中断处理流程 304
2.1 中断执行流程 304
2.2 中断执行条件 305
2.2.1 中断助记符 305
2.2.2 指令用法说明 306
2.2.3 进程语句 PROC 和 PEND 306
2.3 中断格式 306
2.3.1 中断在程序中的位置 306
2.3.2 中断的书写异常 307
2.4 中断优先级 308
2.5 中断嵌套 308
2.5.1 嵌套概念 308
2.5.2 嵌套的条件 308
2.5.3 嵌套执行流程 309
2.5.4 嵌套注意事项 310
2.6 多重中断 310
2.6.1 多重中断概念 310
2.6.2 多重中断实现方式 310
2.6.3 多重中断注意事项 310
2.7 中断溢出 311
2.7.1 中断溢出的概念 311
2.7.2 中断溢出的表达的方式 311
3、中断详解 312
3.1 软中断 312
3.1.1 软中断概述 312
3.1.2 软中断示例 312
3.1.3 注意事项 313
3.2 定时中断 314
3.2.1 定时中断概述 314
3.2.2 定时中断示例 314
3.2.3 注意事项 314
3.3 外部输入中断 316
3.3.1 外部输入中断概述 316
3.3.2 外部输入中断进入方式 316
3.3.3 外部输入中断捕获的轴脉冲 317

3.3.4 外部输入中断捕获特性 318
3.3.5 外部输入中断示例 319
3.3.6 注意事项 319
3.4 高级中断 320
3.4.1 高级中断概述 320
3.4.2 高级中断控制位 320
3.4.3 高级中断同其它中断的关系 320
3.4.4 高级中断和进程的书写格式 321
3.4.5 高级中断示例 323
3.4.6 高级中断任务和进程的时序 325
3.5 中断的应用和区别 327
3.5.1 1ms 中断 327
3.5.2 1ms 中断用法 327
3.5.3 I900 进程同 I010 软中断的执行关系 327
3.5.4 中断捕获高速计数值 328
附录：中断异常自查表 330
五、通信应用手册 331
1. 支持的通信协议介绍 331
1.1 下载监控协议 331
1.2 计算机链接协议 331
1.3 标准 MODBUS协议（RTU 模式） 331
1.4 自由协议 331
1.5 KLink总线协议 331
2. 串口初始化 332
2.1 串口介绍 332
2.2 串口初始化 334
2.2.1 串口初始化指令 334
2.2.2 串口初始化指令应用举例 335
2.2.3 串口协议与可配置格式对应表 336
3. 计算机链接协议 337
3.1 计算机链接协议介绍 337
3.2 计算机链接协议格式 338
3.2.1 通信格式 338
3.2.2 协议内容 340
3.2.3 协议示例 341
3.3 计算机链接协议主从互联 342
3.4 应用案例 348
4. MODBUS 协议 350
4.1 协议说明 350
4.2 MODBUS主站配置 350
4.2.1 支持 MODBUS 主站协议的端口 350
4.2.2 端口配置方法 350
4.2.3 从站通信协议的要求 352

4.2.4 任务调度配置[UINIT] 352
4.3 MODBUS 从站配置 354
4.4 MODBUS 协议主站的指令应用方法 354
4.4.1 RTU 模式 355
4.4.2 通信地址 355
4.4.3 功能码（Function ） 355
4.4.4 MODBUS 指令 360
4.5 MODBUS 应用注意事项 368
4.5.1 指令使用规范 368
4.5.2 与其他流程指令的相互关系 368
4.5.3 特殊情况下的端口配置方法 368
4.5.4 MC系列PLC作为 MODBUS 从站的响应时延 369
4.5.5 MC系列PLC作为 MODBUS 主站时，对其他从站的响应时间要求 369
4.6 MODBUS 通信应用案例 370
4.6.1 两台MC PLC一主一从通信分级别通信 370
4.6.2 三台MC PLC一主两从通信 371
4.6.3 两台MC系列PLC进行一主一从通信（定时通信方式） 373
4.6.4 MC系列PLC与LP2 系列PLC 进行一主一从通信（查询通信方式） 374
4.6.5 MC系列PLC与众辰TZ2400 变频器MODBUS 通信示例 375
4.6.6 使用MC系列 PLC组建两级 MODBUS 通信网络 377
4.6.7 三台MC系列PLC组成多机 MODBUS 通信 378
5. 自由协议 380
5.1 协议介绍 380
5.2 协议格式 380
5.2.1 通信格式 380
5.2.2 自由协议任务调度 382
5.3 USEND 指令 384
5.3.1 USEND 指令介绍 384
5.3.2 USEND 指令发送方法 385
5.4 URCV 指令 386
5.4.1 URCV 指令介绍 386
5.4.2 接收起始符和接收终止符的应用 387
5.4.3 发送自动转接收 388
5.5 应用案例 388
6. KLink总线协议 392
6.1KLink总线协议介绍 392
6.2 KLink总线协议格式 392
6.2.1 报文结构 392
6.2.2 RPT_M0--M3报文 392
6.2.3 RPT_M4--M7报文 393
6.2.4 RPT_M8报文 393
6.2.5 RPT_M9报文 394
6.3 KLink总线配置资源 394

6.3.1 主站配置资源 394
6.3.2 从站配置资源 398
6.3.3 主从站资源地址 399
6.4 KLink总线主从站互联 400
6.4.1 串口配置 400
6.4.2 RPT_M0号报文主从站互联 401
6.4.3 RPT_M4号报文主从站互联 403
6.4.4 RPT_M8号报文主从站互联 406
6.4.5 RPT_M9号报文主从站互联 407
6.5 KLink总线应用案列 408
6.5.1 自动铣刀磨刀系统 408
6.5.2 压力控制系统 411
6.5.3 五轴伺服同步控制系统 415
6.6 KLink总线应用注意事项 419
7. 其他协议 420
六、扩展模块应用手册 421
1、 扩展寄存器 421
2、 模拟量模块 422
2.1 模块规格 422
2.2 模块读写控制 423
2.3 输入模拟量 424
2.4 输出模拟量 424
3、 开关量模块 426
3.1 模块规格 426
3.2 模块读写控制 427
七、运动指令使用手册 428
1、绪论 428
1.1 运动系统类型 428
1.1.1 运动控制系统简述 428
1.1.2 运动控制系统类型 428
1.2 MC系列PLC运动功能简介 429
1.2.1 单轴定位 429
1.2.2 轴关联定位 431
1.2.3 运动的复合 433
2、运动控制模型 435
2.1 运动参数概念 435
2.1.1.运动指令 435
2.1.2.运动轴 436
2.1.3.运动轴之间的绑定关系 436
2.1.4.运动指令事件 437
2.1.5.硬件轴输入输出频率 438
2.2.轴参数定义 440
2.2.1. 输入轴参数 440

2.2.2. 输出轴参数 441
2.3.虚拟轴初始化 447
2.3.1 软件虚拟输入轴初始化 447
2.3.2 软件虚拟输出轴初始化 447
2.3.3 程序运行时如何修改轴配置参数 448
2.3.4 使用回原点指令时如何对软件虚拟输出轴进行初始化 448
2.4.运动指令执行过程 449
2.4.1. 运动指令的执行时延 449
2.4.2. 运动指令的启动条件 449
2.4.3. 运动指令的停止条件 449
2.4.4. 运动指令的运行状态 449
2.5.运动指令报警信息 450
3 运动指令 451
3.1 运动指令索引 451
3.1.1 运动指令一览表 451
3.1.2 输入端口掩码 452
3.1.3 轴输入事件定义 452
3.2 轴配置指令 454
3.2.1 轴当量指定[AXPU] 454
3.2.2 轴速度指定[AXSP] 456
3.2.3 轴端口指定[AXPO] 458
3.2.4 轴加减速指定[AXAS] 461
3.2.5 轴事件输入点指定[AXEI] 463
3.2.6 回原点速度指定[AXOSP] 466
3.2.7 回原点参数指定[AXOX] 468
3.3 高速计数指令 471
3.3.1 高速读出[DHMOV] 471
3.3.2 高速比较置位[DHSCS] 472
3.3.3 高速比较复位[DHSCR] 474
3.3.4 高速区间比较[DHSZ] 476
3.3.5 高速表格比较输出[DHSCT] 478
3.3.6 输入输出刷新[REF] 482
3.4 单轴基本运动指令 483
3.4.1 回原点[HOME] 483
3.4.2 定速定长[DPLSY] 493
3.4.3 加减速定长[DPLSR] 496
3.4.4 变速输出[DPLSV] 502
3.4.5 相对定位[DDRVI] 507
3.4.6 中断定位[DDVIT] 516
3.4.7 绝对定位[DDRVA] 527
3.4.8 插补输出[IPL] 532
3.4.9 简易回原点[HOMA] 535
3.4.10 多段速输出[DPTO] 539

3.4.11 定位输出[PSOUT] 550
3.4.12 普通制表[PTAB] 554
3.4.13 制表输出[PTOUT] 570
3.4.14 贴标机[TBJ] 577
3.4.15 停止输出[MSTOP] 581
3.5 单轴关联指令 583
3.5.1 电子齿轮[GEAR] 583
3.5.2 手摇轮[HAND] 586
3.5.3 追剪正程/滚切制表[PCTAB] 589
3.5.4 追剪返程制表[PBTAB] 594
3.5.5 电子凸轮输出[CAM] 600
3.6 两轴插补指令 622
3.6.1 设置坐标系[GCO] 622
3.6.2 选择坐标系[SELCO] 624
3.6.3两轴直线绝对位置插补[G90G01] 625
3.6.4两轴直线相对位置插补[G91G01] 628
3.6.5两轴顺圆弧绝对位置插补[G90G02] 631
3.6.6两轴顺圆弧相对位置插补[G91G02] 635
3.6.7两轴逆圆弧绝对位置插补[G90G03] 640
3.6.8两轴逆圆弧相对位置插补[G91G03] 644
3.7 FLASH指令 649
3.7.1 FLASH参数定义 649
3.7.2 FLASH格式化[FFMT] 651
3.7.3 FLASH状态查看[FDIR] 652
3.7.4 FLASH删除文件[FDEL] 652
3.7.5 FLASH打开文件[FOP] 653
3.7.6 FLASH关闭文件[FCLS] 655
3.7.7 FLASH读文件[FRD] 655
3.7.8 FLASH写文件[FWR] 657
3.7.9 FLASH错误码 659
八、C语言功能说明 661
1.概述 661
2.操作步骤 661
3、C语言变量 662
3.1 基本变量类型 663
3.2 整型变量 663
3.3实型变量 664
3.4字符变量 664
3.5 符号常量宏定义 664
3.6 数组 665
3.6.1 数组定义 665
3.6.2 数组引用 665
3.6.3 数组初始化 665

3.7 指针 665
3.7.1 指针定义 666
3.7.2指针的基本运算 666
3.8 结构体 666
3.8.1 结构的概念与定义 666
3.8.2 结构变量的定义和初始化 666
3.8.3 结构变量的使用 667
4 程序设计 668
4.1 基本运算符和表达式 668
4.2 if语句 669
4.3 Switch语句 670
4.4 循环结构顺序 672
4.4.1 while语句 672
4.4.2 do- while语句 672
4.4.3 For语句 672
4.4.4 break语句 673
4.4.5 continue语句 673
5 PLC资源与支持函数 673
5.1 字型资源访问 673
5.2 字型资源访问 674
6 应用要点 676
九、以太网通信篇 678
1. 以太网通讯概述 678
1.1 以太网的基本概念 678
1.2 TCP/IP协议 679
2. 以太网参数配置 681
2.1 以太网参数在编程软件中的配置 681
2.2 IP地址相关参数 682
3. 通讯协议 683
3.1 MODBUS TCP通讯协议 683
3.2 自由格式通讯协议 684
4. 以太网通讯指令 684
4.1 以太网通讯指令概述 684
4.2 通信任务创建指令(NOPEN) 684
4.3 通信任务关闭指令(NCLS) 687
4.4 自由通信发送指令(NSEND) 688
4.5 自由通信接收指令(NRCV) 689
4.6 读多个线圈指令(NRDC) 691
4.7 读多个输入线圈指令(NRDI) 692
4.8 读多个输出寄存器指令(NRDD) 693
4.9 读多个输入寄存器指令(NRDAI) 694
4.10 写单个线圈指令(NWRC) 695
4.11 写单个寄存器指令(NWRD) 696

4.12 写多个线圈指令(NWRCS) 697
4.13 写多个寄存器指令(NWRDS) 698
4.14 读多个寄存器指令(ERDD) 699
后续讲述ModbusTCP的扩展指令。 699
4.15 写多个寄存器指令(EWRD) 700
4.16 扩展读位区数据指令(ERDC) 701
4.17 扩展写位区数据指令(EWRC) 702
5. 以太网通讯案例 703
6. 以太网通讯相关标志位和寄存器 722
6.1 以太网通讯相关寄存器 722
6.2 以太网通讯相关标志位 723
7. 以太网远程连接 724
7.1 以太网远程连接概述 724
7.2 以太网远程连接步骤 724
7.3 以太网远程连接注意事项 725
附录 726
附录一 常用资源列表 726
附录二 特殊软元件功能一览表 727
特殊辅助继电器[M]功能一览表 727
特殊辅助寄存器[D]功能一览表 732

一、MC系列PLC硬件手册

1、安全注意事项

安全警告和标识

警告标识	含义
	该标识表示若操作错误则可能发生“死亡或重伤”
	该标识表示若操作错误则可能发生“人身伤害或财产损害”

安全标识	含义
	该图形表示“不可实施”的内容
	该图形表示“必须实施”的内容

安全注意事项

	
	切勿在有水的地方、存在腐蚀性、引火性气体的环境内以及可燃物旁使用，容易发生火灾
	不要在振动和冲击激烈的地方使用
	不要用湿手进行接线和设备操作
	切勿将手伸入MC系列PLC内部
	不要使导线受到损伤或承受过大的外力、重压、受夹
	切勿用裸手检查伺服接线故障，容易引发触电
	应在尘埃较少，不会接触到油、水的地方放置
	请进行正确的接线，否则可能发生火灾或故障
	导线应连接好，通电部位须通过绝缘套做到绝缘
	MC系列PLC必须实际接地
	外部应设置紧急停止电路，以确保紧急时可及时地停止运转、切断电源
	MC系列PLC的移动、接线、检查等要在切断电源并确保放电结束后没有触电危险的前提下进行
	必须设置过电流保护装置、漏电断路器、温度防护装置以及紧急停止装置

注意	
	不要在MC系列PLC上放置重物
	不要使MC系列PLC受到较强的冲击
	不要频繁地开、关主电源
	不要对MC系列PLC进行长时间极限情况测试
	切勿自行改造、分解、修理
	严格按照正确的安装方法安装MC系列PLC
	确保MC系列PLC在规定的温度和湿度范围内运行
	发生故障时请清除故障、确保安全后再重新运行
	MC系列PLC发生故障时请切断电源
	长时间不使用MC系列PLC时必须切断电源
	MC系列PLC的保养工作应由专业人员进行
	确保通风散热口的通畅、无障碍
	请在规定电压范围内运行

2、 MC系列PLC选型

2.1 主要功能

● 掉电数据保存

掉电保存数据功能，MC系列PLC可以在断电时自动保存用户数据。

● 丰富的输入输出扩展单元

可接各种输入输出扩展单元。

输入类型：DC输入（24V）

输出类型：继电器输出，晶体管输出

丰富的特殊扩展单元

模拟量输入单元　　模拟量输出单元

热电偶输入单元　　热电阻输入单元等

● 运动控制指令群

MC系列PLC支持简便指令和复杂运动控制指令：

HOME　回原点　　IPL　插补　　DPLSY/DPLSR　定速定长输出

DDRVI　相对定位　　DDVIT 中断定位　　DPLSV　变速输出

HOMA　简易回原点　　DPTO多段速输出

DHMOV 高速读出　　DHSCS　高速比较置位　　DHSCR 高速比较复位

DHSZ　高速区间比较　　DHSCT 高速表格比较输出

DGEAR 电子齿轮　　PTOUT 制表输出　　PCTAB 追剪滚切制表

PBTAB 追剪返程制表　　CAM 电子凸轮输出　　HAND 手摇轮

G90G01/G91G01 两轴直线插补　　G90G02/G91G02/G90G03/G91G03 两轴圆弧插补

● 高速处理

- 内装单相4点（MAX：200K），AB相4点（MAX：200K）高速计数器，分辨率为0.1MS。
- 脉冲捕捉功能
- 输入输出刷新功能
- 梯形图中断功能：4个外部输入中断，3个定时器中断，5个系统软中断，8个高级中断。
- 脉冲输出功能：4路高速脉冲输出可达200KHz。

● 通信

串口0、串口1、串口2通信速率可选择9600，19200，38400，57600，支持的通信协议包含：

编程口协议，标准MODBUS协议，计算机链接协议以及自由协议。

串口1支持科威高速通信总线协议KLink，波特率为2.5Mbps。

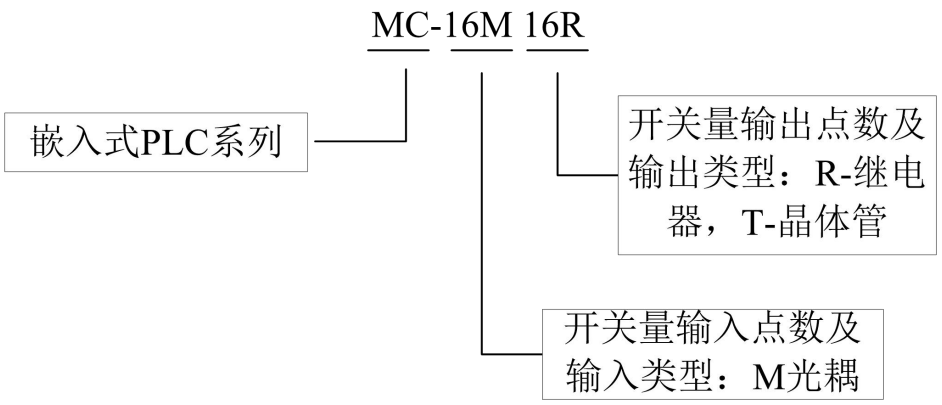
1个以太网接口，实现网络互连。

● 编程

支持 C 语言函数编写功能块，增加程序的灵活性和保密性，提高编程效率。

2.2 型号说明

以下是MC系列PLC的型号命名规则。



主机型号	合计点数	输入输出特性			
		普通输入	高速输入	普通输出	高速输出(T型)
MC-16M16R/T	32	16	单相：4路200KHz AB相：4路200KHz	16	4路200KHz

3、 MC系列PLC安装与尺寸

3.1 MC系列PLC安装

3.1.1 安装场所

安装注意事项
不能安装在含有腐蚀性气体的场所
不能安装在阳光直接照射的场所
不能安装在温度急剧上下变化的场所
不能安装在油，水，化合物容易侵入的场所
不能安装在有大量灰尘的场所
不能安装在震动大且会造成安装件移动的场所

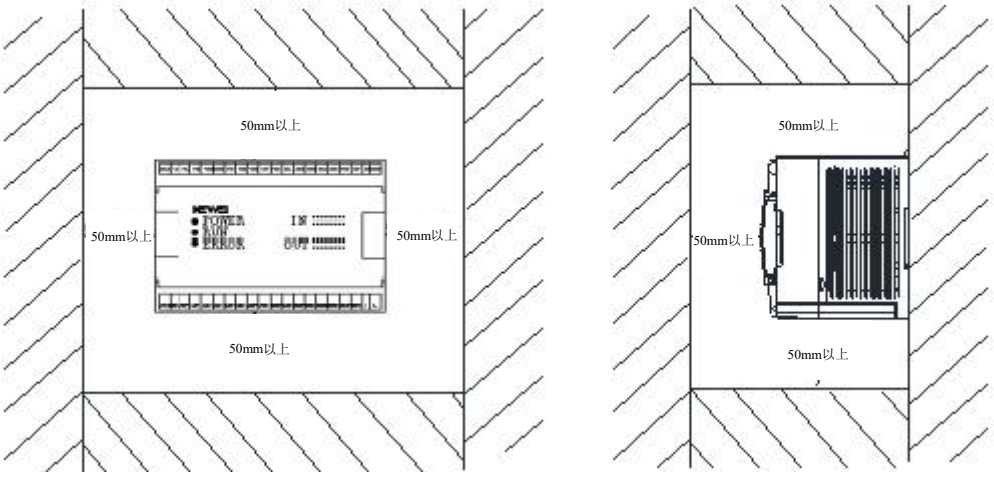
3.1.2 环境条件

项目	条件
使用环境温度	0℃～55℃不冻结
使用环境湿度	35 ～ 85[RH%]不凝露
存储温度	-20℃～70℃
存储湿度	35 ～ 85[RH%]不凝露
震动	JISC0911标准，10～55HZ，0.5mm(最大2G)， 3轴方向各2次(但用DIN导轨安装时为0.5G)
冲击	JISC0912标准，10G，3轴向各3次
高度	海拔1000m以下正常使用,海拔1000m以上降额使用

3.1.3 安装方向与间隔

注意
请留足够的空间，以便有效地降温
为保证控制柜内的温度分布均匀，请安装风扇
务必使MC系列PLC良好接地

为防止机器内部温度过热，请采用正常的壁挂方式安装，要求上下留有50mm以上空间作为散热空间。



MC系列PLC基本单元和其他设备或结构物之间留50mm以上的空隙。尽量远离高压线、高压设备和动力设备。

安装时有两种安装方法：

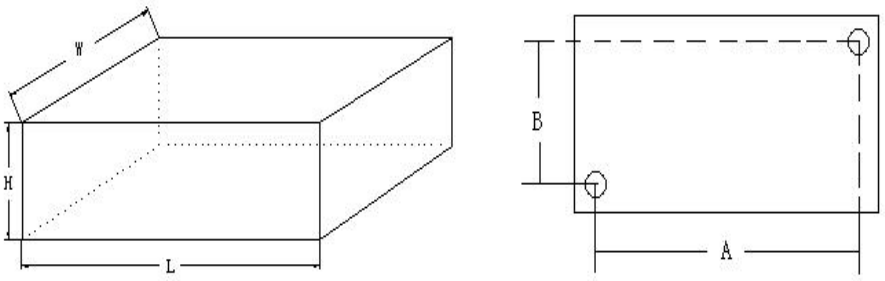
- DIN导轨安装：
  - 将DIN35mm槽水平固定于安装背板上
  - 将PLC底部下方的DIN槽卡扣拔出
  - 把PLC挂在DIN槽上，将卡扣压回原位，锁住PLC
  - 最后再将PLC的两端用DIN槽卡档件固定，避免左右滑动

  拆卸时，请将DIN导轨安装挂钩向外拉出。
- 直接安装：

  用M4螺钉对准螺孔固定即可。但各个器件之间需要留1-2mm的间隙。

3.2 MC系列PLC尺寸

MC系列PLC安装尺寸如图所示：



产品型号	L（mm）	W（mm）	H（mm）	A（mm）	B（mm）
MC-24M16T	143.5	102	77	119	98

A表示孔心距。孔的直径是4.5 mm。存在较大冲击的场合，建议采用四个M4螺钉固定。

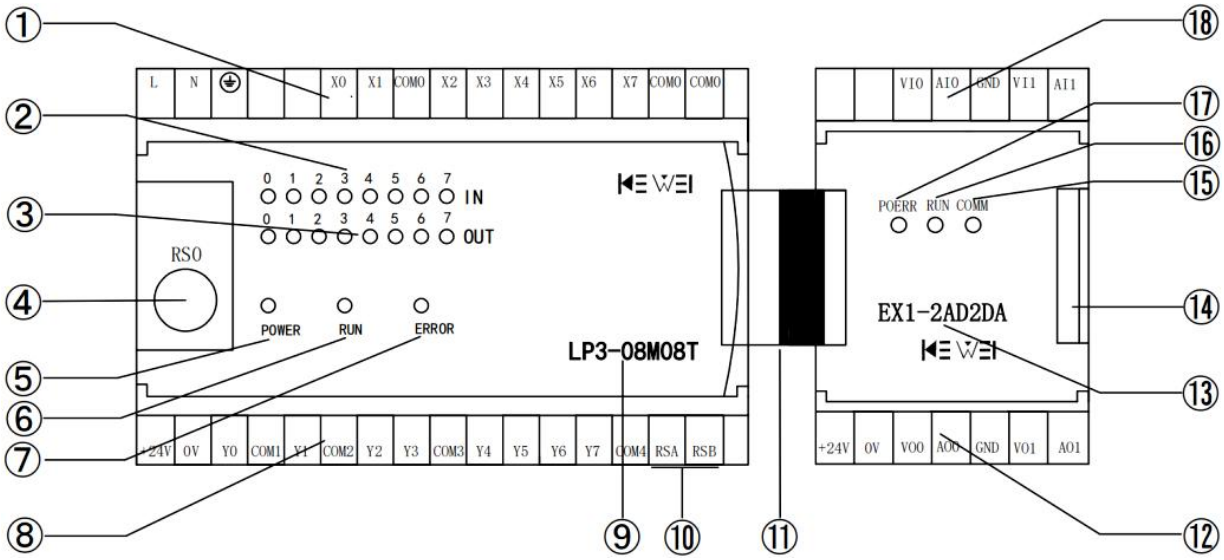
4、 MC系列PLC接口

4.1 硬件接口及端子分布

各部分说明

基本单元

扩展单元



①:输入端子，电源接入端子
②:输入动作指示灯
③:输出动作指示灯
④:RS232串口
⑤:电源工作灯
⑥:PLC运行灯
⑦:PLC错误灯
⑧:输出端子，24V电源输出
⑨:PLC型号
⑩:RS485串口
⑪:通讯模块扩展线
⑫:模块输出端子
⑬:扩展模块型号
⑭:扩展模块接入口
⑮:主机模块通讯指示灯
⑯:模块运行指示灯
⑰:模块电源指示灯
⑱:模块输入端子

接口端子分布

MC系列PLC的端子分布如下所示：

- MC-24M16R/MC-24M16T

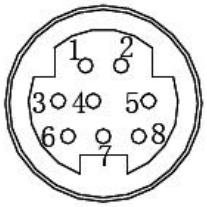


端子名称	说明
L、N	工作电源接入端子，电压范围为85~264VAC，50/60Hz。
G	接地端子。
+24V、0V	DC24V电源输出，额定负载电流为400mA。
X0 ～ Xn 、COM0	集电极输入，开关量输入端子，COM0为公共端。
Yn、COMn	集电极输出或继电器输出，普通开关量输出，COMn为公共端。
RSA、RSB	485通信端子

通信接口

MC系列PLC标配三个通讯口：COM0为标准RS232、COM2为标准RS485。

- **COM0（程序下载通讯端口）：**



Mini Din 8 芯插座（孔）

COM0端口定义

管脚号	信号	描述
1	NC	空脚
2	TXD	发送
3	RXD	接收
4	NC	空脚
5	GND	地线
6	NC	空脚
7	NC	空脚
8	NC	空脚

编程电缆内部接线如下图：



Mini Din 8 芯插头（针） DB9 插头（孔）

- **COM2（RS485通讯端口）：**



RSA2 RSB2

- **LAN2/LAN3（RS485通讯端口）：**



支持标准485通信，同时也支持科威高速总线KLINK

4.2 电源配线

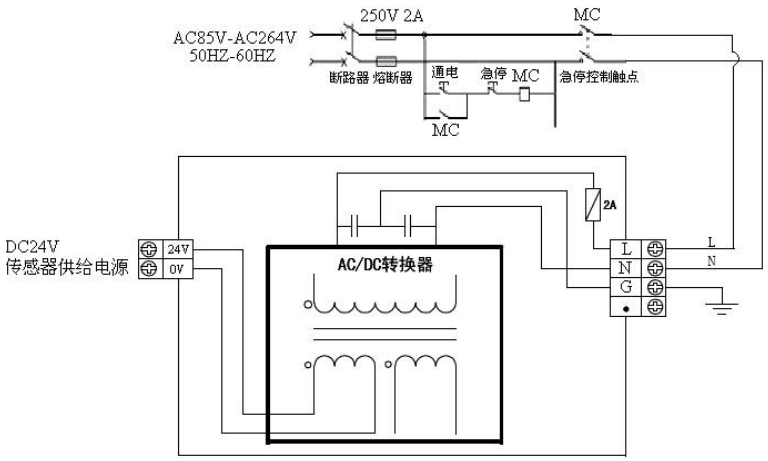
MC系列PLC 220V电源规格如下表所示：

额定电压	AC220V
电压允许范围	AC85V ～264V
额定频率	50/60Hz
允许瞬间断电时间	10ms
电源保险丝	250V 3.15A
冲击电流	最大40A 5ms以下/AC100V 最大50A 5ms以下/AC200V
最大消耗功率	30W
传感器用电源	24VDC±10% 最大400mA

● 电源线请用2mm² 以上的电线，以防止电压下降。

● 即使出现了10ms以内的断电，控制器仍可继续工作。当长时间地断电或电压异常下降时，控制器停止工作，输出也呈OFF状态，当电源恢复供电时，控制器就自动开始运行。

220V电源配线



4.3 状态识别

MC系列PLC状态指示LED有：POWER、RUN、ERROR。

MC系列PLC正常运行时，指示LED POWER和RUN常亮。

● 指示LED POWER熄，MC系列PLC电源不正常。

● 指示LED RUN熄，MC系列PLC处于设置状态。

● 指示LED ERROR亮，表示MC系列PLC运行时出错。在这种情况下，请先检查程序正确与否；若程序正确，再将MC系列PLC断电再复电，若MC系列PLC仍未恢复正常，则请检查有无高强度干扰源及MC系列PLC接地是否确实。排除以上因素后若ERROR仍亮，则MC系列PLC可能已经故障，应考虑送修。

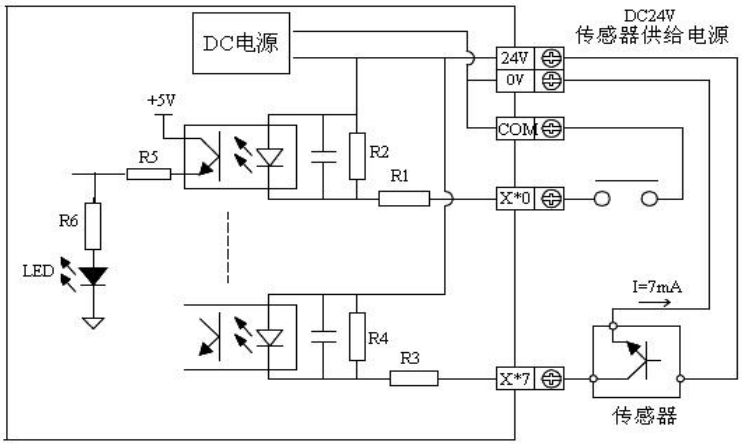
4.4 开关量输入

对于MC系列PLC而言，开关量输入为集电极输入电路。MC系列PLC的开关量输入规格表如下所示：

项目	规格		
	X000～X007	X010～X017	X020～X267
输入信号电压	DC24V±10%		
输入信号电流	11mA	11mA	4.5mA
输入信号阻抗	2KΩ	2kΩ	5.1kΩ
输入ON电流	4.5mA以上	4mA以上	2.3mA以上
输入OFF电流	4.4mA以下	3.9mA以下	2.2mA以下
输入响应时间	OFF-ON< 5 μS ON-OFF < 5 μS	OFF-ON< 50 μS ON-OFF < 50 μS	缺省0.5ms
输入信号形式	无电压触点，或NPN集电极开路晶体管		
电路绝缘	光电耦合器隔离		
输入状态显示	输入ON时LED灯亮		

集电极输入电路

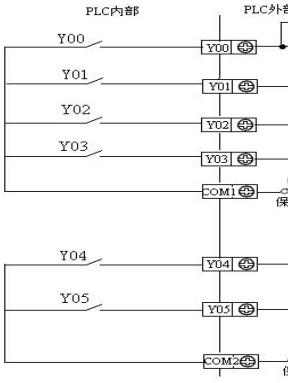
MC系列PLC的集电极输入端子连接电路如下所示：



4.5 开关量输出

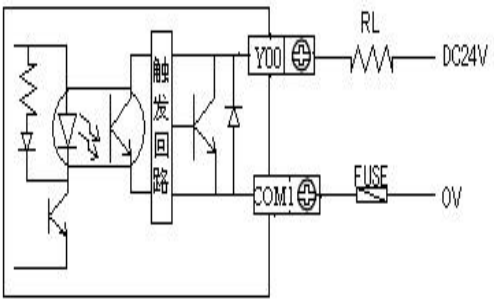
MC系列PLC开关量输出分为继电器输出（R）和晶体管输出（T）。T型PLC的Y0～Y3为高速脉冲输出，4路200K，Y4～Y7为脉冲方向，集电极输出。当不作高速输出和方向功能使用时，输出端子可以作一般输出使用。其余输出为集电极输出。

下表为MC系列PLC继电器输出规格：



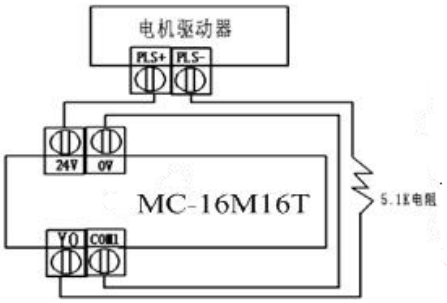
项目	继电器输出
外部电源	AC250V DC30V 以下
电路绝缘	机械绝缘
动作显示	继电器线圈通电时LED发光
最大电阻负载	2A/点 8A/4点公共端 8A/8点公共端
最大感性负载	80VA
最大指示灯负载	100W
开路漏电流	-
最小负载	DC5V 2 mA
响应时间OFF>ON	约10ms
响应时间ON>OFF	约10ms

下表为MC系列PLC晶体管输出规格：



项目	晶体管输出
外部电源	DC5~30V
电路绝缘	光耦绝缘
动作显示	光耦驱动时LED发光
最大电阻负载	0.5A/点 0.8A/4点公共端 1.6A/8点公共端
最大感性负载	12W/DC24V
最大指示灯负载	1.5W/DC24V
开路漏电流	0.1 mA /DC30V
最小负载	-
响应时间OFF>ON	0.2ms以下
响应时间ON>OFF	0.2ms以下

下表为MC系列PLC高速脉冲输出规格：



项目	高速脉冲输出
外部电源	DC5~30V
电路绝缘	不绝缘
动作显示	导通时LED发光
最大电阻负载	0.3A/点
最大感性负载	7.2W/DC24V
最大指示灯负载	0.9W/DC24V
开路漏电流	0.1 mA /DC30V
最小负载	-
响应时间OFF>ON	5us以下
响应时间ON>OFF	8us以下

二、基本应用指令手册

1. 前言

在本章中，说明了 MC系列PLC编程的基本事项。

1.1 MC系列PLC编程软件

MC系列PLC编程软件使用科威KWPro 软件，版本为 1.2.5.6 以上。

1.2 MC系列PLC编程语言

MC系列PLC编程语言有以下几点特点。

1.2.1 编程语言的种类

MC系列PLC支持 2 种编程语言。

1) 指令表编程

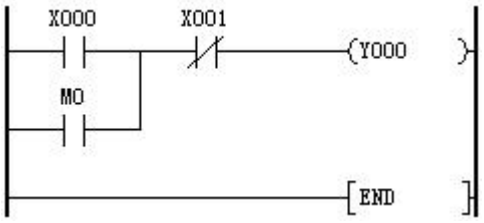
指令表编程方式，是通过在指令列表程序编辑器中输入“LD”“AND”“OR”等指令语言和操作数的方式。指令步号自动生成，不需要输入。

步 号	指令	软元件编号
0	LD	X000
2	OR	M0
4	ANI	X001
6	OUT	Y000
8	NOP	
9	NOP	
10	NOP	

指令表编程方式可以写入 NOP 指令，相似程序复制粘贴时没有行数限制。

2) 梯形图编程

梯形图编程方式，是使用符号和软元件编号在梯形图页面画顺控梯形图的方式。梯形图是一种和电气原理图非常相似的图形编程语言，逻辑也和电气原理图非常相似。控制器的输入点称为触点，功能和电气设备中的开关触点是一样的，意义导通或者关断。输入点存储为一个软元件，当输入点为 1 时，对应触点导通。控制器的输出点称为线圈，意义是输出的导通和关断，输出点软元件的值由输入点和控制逻辑决定， 输出点软元件的值被映射为电信号在输出点的继电器或晶体管输出。



上述指令表程序的梯形图形式。梯形图是通过触点符号和线圈符号来表现的，所以程序的内容更加直观易懂。相似程序复制粘贴时有行数限制。

3) C语言编程

MC系列PLC支持用户在科威编程软件上利用C语言函数编写功能块，在程序中以子程序形式调用，其中支持大

部分C语言函数。对用户来说，不仅增加了程序的灵活性，提高了编程效率，还进一步增强了程序的保密性。

1.2.2 程序的互换性

指令表编程和梯形图编程的程序都通过指令（指令表编程时的内容）保存到MC系列PLC的程序内存中。两种编程语言可以相互转换并显示、编程，编程人员可以根据使用习惯或者实际应用需求选择合适的编程语言。

2. 概要

在本章中，简要说明 MC系列PLC的基本功能。在基本功能中，包括了控制器的特点以及为了能让用户有效地使用该控制器的功能所需的典型的功能介绍、参数、存储运行等项目，所以请在设计程序之前阅读。

2.1 基本功能介绍

MC系列PLC具备以下功能：

● 掉电数据保存

掉电保存数据功能，MC系列PLC可以在断电时自动保存用户数据。

● 丰富的输入输出扩展单元

可扩展各种开关量输入输出单元和模拟量输入输出单元。

● 通信

串口 0 、串口 1和串口2 通信速率可选择设置。

支持编程口协议，MODBUS 协议，主从站链接协议以及自由协议等。

MODBUS 通信应用指令，编程方便。

串口1支持KLink协议通信，高达2M速度。
1个以太网接口，实现网络互连。

● 高速处理

内装单相4点（MAX：200K），AB相4点（MAX：200K）高速计数器，分辨率为0.1MS。

脉冲捕捉功能

输入输出刷新功能

梯形图中断功能：4个外部输入中断，3个定时器中断，5个系统软中断，8个高级中断。

● 运动控制

4 路200KHz 高速脉冲输出。丰富的运动控制指令群：

HOME 回原点	IPL 插补	DPLSY/DPLSR 定速定长输出
DDRVI 相对定位	DDVIT 中断定位	DPLSV 变速输出
HOMA 简易回原点	DPTO多段速输出	
DHMOV 高速读出	DHSCS 高速比较置位	DHSCR 高速比较复位
DHSZ 高速区间比较	DHSCT 高速表格比较输出	
DGEAR 电子齿轮	PTOUT 制表输出	PCTAB 追剪滚切制表
PBTAB 追剪返程制表	CAM 电子凸轮输出	HAND 手摇轮
G90G01/G91G01 两轴直线插补	G90G02/G91G02/G90G03/G91G03 两轴圆弧插补	

2.2 软元件介绍

MC系列PLC接收外部输入开关信号，运算处理后输出开关信号驱动外部负载。输入开关信号映射控制器内部软元件输入继电器 X，输出开关信号映射控制器内部软元件输出继电器 Y。MC系列PLC中还内置了多个辅助继电器M、状态继电器 S、定时器 T、计数器 C、数据寄存器 D 和扩展寄存器 R 等。这些软元件和指令一起构成回路从而实现对设备的控制。

2.2.1 软元件一览表

MC系列PLC中内置以下软元件：

简称	代号及点数范围	说明
输入 X	X000～X377(8 进制)	输入继电器，单机点数由MC系列PLC具体型号而定
输出 Y	Y000～Y377(8 进制)	输出继电器，单机点数由MC系列PLC具体型号而定
辅助 M	M0～M3071	辅助继电器
	M8000～M8255	特殊辅助继电器，具体定义见相关章
状态 S	S0～S1023	状态继电器
时间 T	T0～T255	T0～T199，100ms 时基定时器
		T200～T245，10ms 时基定时器
		T246～T249，1ms 时基累计定时器
		T250～T255，100ms 时基累计定时器
计数 C	C0～C255	C0～C199，16 位增计数器
		C200～C234，32 位可逆计数器，计数方向由 M8200～M8234 控制
		C235～C255，32 位高速计数器(与高速计数指令配合使用)
寄存器 D	D0～D7999	D200～D3999，掉电保持区
	D8000～D8255	系统寄存器区，具体定义见相关章节
扩展寄存器R	R0～R7999	R0～R59，内部保留区
		R60～R83，输出轴脉冲显示区
		R84～R99，内部保留区
		R100～R259，软件虚拟输入轴参数区
		R260～R299，内部保留区
		R300～R899，软件虚拟输出轴参数区
		R900～R999，内部保留区
		R1000～R1199，内部保留区
		R1200~R1899，内部保留区
		R1900~R1999，内部保留区
		R2000~R2049，FLASH参数区
		R2050~R2099，内部保留区
		R2100~R2259，KLink总线资源配置参数区
		R2300~R2323，以太网通信参数区
		R2324~R7999，内部保留区
变址 V，Z	V0～V7，Z0～Z7	与其他软元件配合使用，如 V0=10，D0V0 的地址为 D10
常数 K，H	带符号整型数范围	K 为 10 进制表达数，H 为 16 进制表达数，如 H10=K16

程序指针 P	P0～P127	做程序跳转或子程序调用使用
中断指针 I	I000～I999	适用于各类型中断的指示标签
程序容量	2K，4K，8K，12K，16K，20K，24K，28K，32K	MC系列PLC所适用的梯形图步长控制

2.2.2软元件说明

1) 输入X·输出Y 继电器

MC系列PLC基本单元分配了 8 进制的输入继电器、输出继电器编号，扩展单元的编号也是 8 进制编号，紧接基本单元地址。

输入继电器 X 接收外部开关信号，输出继电器 Y 被各种软元件的触点驱动，从而驱动外部负载。输入输出继电器响应均受扫描周期影响，需要更快的响应可使用外部输入中断、高速计数器、X 上升沿计数、脉冲输出等高速处理。

→请参考 4.2，4.3 节

2) 辅助继电器 M

辅助继电器 M 同输出继电器 Y 一样，可以被各种软元件的触点驱动。但辅助继电器没有映射到硬件端子， 不能直接驱动外部负载，仅起辅助逻辑运算的作用。

MC系列PLC断电情况下，保持用辅助继电器 M 能够记忆其 ON/OFF 状态。

请参考 4.4 节

3) 状态继电器 S

状态继电器 S 作为步进顺控(STL)指令中表示工程编号的继电器使用。

请参考 4.4 节

状态继电器 S 也可做一般使用，与辅助继电器 M 一样，作为一般的触点/线圈使用。MC系列PLC断电情况下，保持用状态继电器 S 能够记忆其 ON/OFF 状态。

→请参考 4.5 节

4) 定时器T

定时器 T（亦称时间继电器）是累计MC系列PLC中的 1ms、10ms、100ms 的时钟脉冲，当累计结果达到所指定的设定值时，定时器 T 的输出触点动作。

定时器 T 主要用于定时控制，可定时 0.001～3276.7 秒。定时器 T 也可以作为数据寄存器使用。MC系列PLC断电情况下，累计型定时器 T 能够记忆其状态。

→请参考 4.6 节

5) 计数器C

计数器 C 用于接收脉冲信号并计数，计数值达到设定值时其触点动作。根据用途可分为普通计数器和高速计数器。有 16 位计数器和 32 位计数器，16 位计数器可增计数，计数范围为 1～32767。32 位计数器是环形可逆计数器，可增、减计数。计数范围为-2147483648～2147483647。

高速计数器接收外部高速脉冲信号计数，快速响应。计数器 C 也可以作为数据寄存器使用。

MC系列PLC断电情况下，保持型计数器 C 能够记忆其状态。

→请参考 4.7，4.8 节

6) 数据寄存器 D

数据寄存器 D 是存储数据的软元件。MC系列PLC的数据寄存器都是 16 位（最高位为正、负号位），组合两个相邻地址的数据寄存器可做 32 位数据寄存器使用（最高位为正、负号位）。

MC系列PLC断电情况下，保持型数据寄存器 D 能够记忆其状态。

→请参考 4.9 节

7) 扩展寄存器 R

扩展寄存器 R 为数据寄存器 D 的扩展软元件，扩展寄存器 R 的使用方法与数据寄存器一致。MC系列PLC断电情况下，保持型扩展寄存器 R 能够记忆其状态。

→请参考 4.10 节

8) 变址寄存器 V、Z

寄存器中，被称为变址（修饰）用的有 V、Z 两种寄存器，V 与 Z 两者功能完全相同。

变址寄存器可作为普通的数据寄存器使用，还可以同其他的软元件编号或数值组合使用，间接指定软元件编号或改变数值的内容。

→请参考 4.11 节

9) 标记指针[P]、[I]

指针有分支用和中断用两种，分支用指针 P 用于指定 CJ 条件跳转和 CALL 子程序调用的对象目的地。中断用指针 I 用于指定外部输入中断、定时中断等中断程序。

→请参考 4.12 节

10) 常数[K]、[H]、[E]

MC系列PLC中使用的各种数值，K 表示 10 进制数，H 表示 16 进制数，E 表示浮点实数(KWPRO编程软件输入F前缀代表浮点数，PLC指令前缀DE代表单精度浮点数指令)。这些进制数均可用作设定值、当前值或应用指令的操作数。

→请参考第二章5节

2.3 应用指令介绍

MC系列PLC含括了数据的传送和比较、四则运算、逻辑运算、数据的循环和移位等基本应用指令， 和MODBUS 通信、高速计数、高速脉冲输出、中断等高速处理指令，还包括了浮点数运算\PID 控制指令和以太网指令。

2.3.1主要应用指令

列举了一些常用指令。

1、程序流程

条件跳转（CJ）
子程序调用（CALL）
允许中断（EI）
禁止中断（DI）
循环范围的起始（FOR）
主控指令（MC-MCR）
步进顺控（STL-RET）

2、传送·比较

比较（CMP）
触点比较（LD□）
浮点数比较（DECMP）
区间比较（ZCP）
数据传送（MOV）
成批传送（BMOV）
多点传送（FMOV）
反向传送（CML）

3、四则·逻辑运算

加法运算（ADD）
减法运算（SUB）
乘法运算（MUL）
除法运算（DIV）
加一（INC）
开方运算（SQR）
三角函数（SIN 等）
浮点数四则运算（DEADD等）
浮点数开方运算（DESQR等）

4、循环·移位

循环左移（ROL）
循环右移（ROR）
带进位循环左移（RCL）
带进位循环右移（RCR）
位左移（SFTL）
位右移（SFTR）
字左移（WSFL）
字右移（WSFR）

5、数据处理

成批复位（ZRST）
高低字节交换（SWAP）
两数据交换（XCH）
整型转浮点（FLT）
浮点转整型（INT）
BCD 转换（BCD）
BIN 转换（BIN）
十六进制转 ASCII（ASCI）
ASCII 转十六进制（HEX）

6、高速处理

高速读出（DHMOV）
高速比较置位（DHSCS）
高速比较复位（DHSCR）
高速区间比较（DHSZ）
高速表格比较输出（DHSCT）
输入输出刷新（REF）

7、高速脉冲输出

回原点（HOME）
定速定长输出（DPLSY）
加减速定长输出（DPLSR）
相对定位（DDRVI）
绝对定位（DDRVA）
中断定位（DDVIT）
插补输出（IPL）
变速输出（DPLSV）
简易回原点（HOMA）
多段速输出（DPTO）
电子齿轮（DGEAR）

8. **MODBUS 通信**

读线圈当前状态（MRDC）
读输入线圈当前状态（MRDI）
读寄存器当前值（MRDD）
读输入寄存器当前值（MRDAI）
强置一个线圈（MWRC）
强置多个线圈（MWRCS）
赋值一个寄存器（MWRD）
赋值多个寄存器（MWRDS）

9、复杂控制

PID 运算（PID）

10、以太网指令

通信任务创建（NOPEN）
通信任务关闭（NCLS）
读多个线圈指令（NRDC）

2.4 程序内存介绍

MC系列PLC内部分配了存储空间，分别存储程序和各种软元件变量。

2.4.1程序内存

MC系列PLC支持32K 的程序内存，即最大工程程序容量为32K 步。超过32K 步梯形图程序下载后控制器报错。程序容量有2K，4K，8K，12K，16K，20K，24K，28K，32K 9种。程序编写时请注意。

2.4.2程序加密

MC系列PLC支持程序全区加密以及分区加密功能，同时有着梯形图擦除功能，可以在擦除当前设备内部梯形图的同时解除设备的加密状态。

（请使用KWPro 1.2.7.0以上版本进行操作）

1、全区加密功能：

即对全部梯形图进行加密处理，梯形图无法被读取。

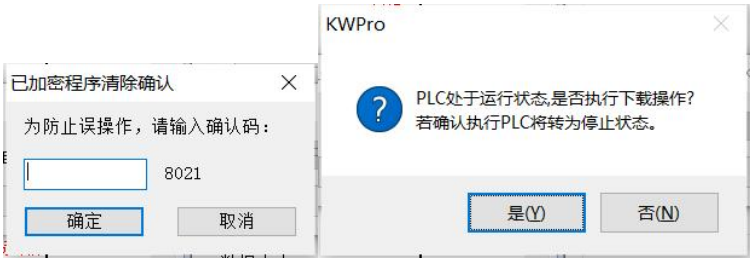
全区加密操作步骤：

① 点击  进入梯形图写入界面

② 选择加密下载→全区加密下载



③ 点击开始写入按钮→输入确认码→确认停止PLC，开始写入梯形图，梯形图写入后，即完成梯形图加密。



2、分区加密功能

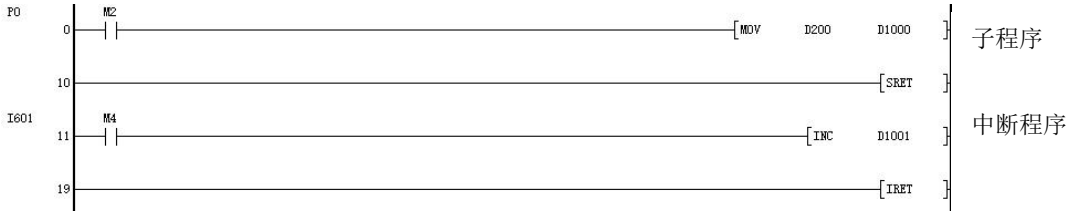
即对部分子程序进行加密，此时只能读取出非加密区梯形图。分区加密完成后仍可对非加密区梯形图进行修改，而不会影响到设备内部已加密部分的梯形图。

注意：

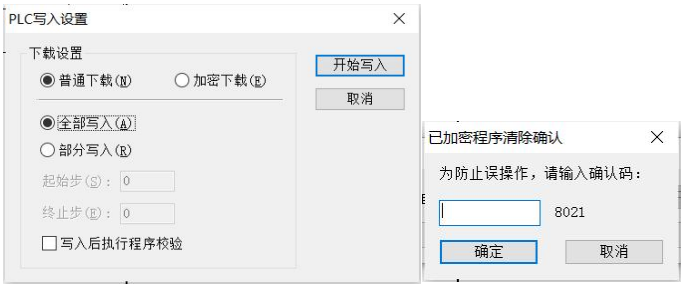
- 该操作需生成两个梯形图文件（非加密区梯形图以及加密区梯形图）。请使用KWPro 1.2.7.0以上版本进行编程（若为之前版本软件生成的梯形图，请使用KWPro 1.2.7.0以上版本打开并将其另存一次，KWPro 1.2.7.0以上版本保存MC程序时会自动生成4个文件，其中.kwz是加密后的程序文件，.kwp是未加密的普通程序文件，.cmt是注释文件，.rcp是配方文件）

非加密区梯形图：包括主程序以及无需加密的子程序。

加密区梯形图：只能为子程序或者中断程序，完整加密程序例程如下（无需增加FEND）：



- 需保证非加密梯形图和加密梯形图的总程序步数不超过当前工程程序容量。
- 分区加密状态下的设备选择普通下载模式下载梯形图时，会将设备内部的梯形图全部擦除，并解除加密状态。



- 分区加密状态下，可仅对非加密区程序进行修改且仅下载非加密区梯形图程序时，不会改变已加密部分的程序，具体操作步骤见下方“分区加密状态下修改非加密区程序操作步骤”
- 修改加密区梯形图后需将加密区梯形图文件保存再进行分区加密操作，否则写入的加密区梯形图仍为修改前的梯形图。

（1）分区加密操作步骤：

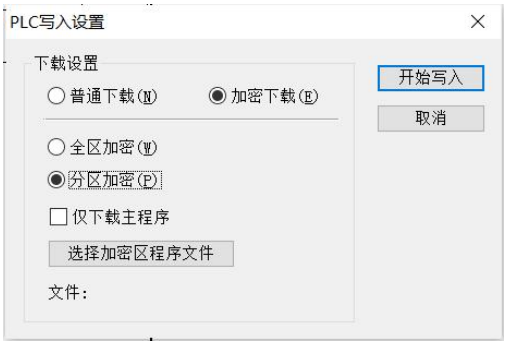
① 梯形图编写，分为非加密梯形图和加密梯形图两个梯形图文件。

注：请使用KWPro 1.2.7.0以上版本进行编程（若为之前版本软件生成的梯形图，请使用KWPro 1.2.7.0以上版本打开并将其另存为）

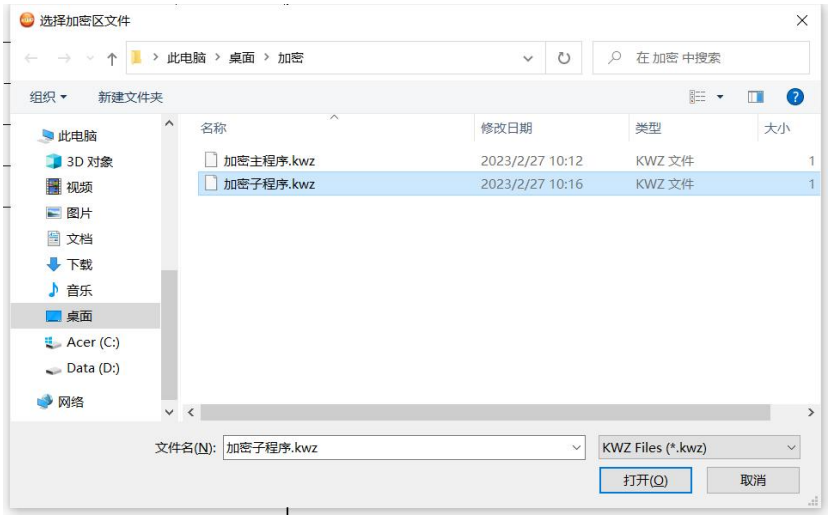
② 点击 工程(F)，选择工程参数，修改当前工程程序容量，需保证非加密梯形图和加密梯形图的总程序步数不超过当前工程程序容量，否则会导致加密梯形图下载失败。推荐将当前工程程序容量改为32000（MC系列PLC最大工程程序容量为32000）。



③ 在非加密梯形图界面，点击  ，进入梯形图写入界面



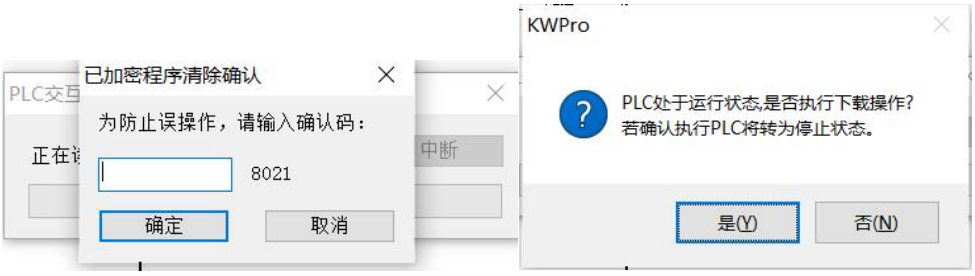
④ 选择加密下载→分区加密→取消框选‘仅下载主程序’→点击选择加密区程序文件，此时出现选择加密区文件弹窗，选中需要加密的.kwz格式的文件→点击打开



⑤ 此时加密文件选择完成，点击开始写入按钮。



⑥ 输入确认码→确认停止PLC，开始对非加密梯形图和加密梯形图进行下载，程序下载完成后，即完成分区加密操作。

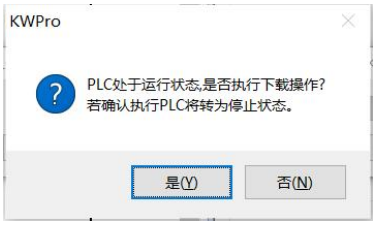


（2）分区加密状态下修改非加密区程序操作步骤

① 非加密区梯形图修改完成后，点击  ，进入PLC写入设置界面



② 选择加密下载模式→分区加密→勾选仅下载非加密区程序→点击开始下载→确认停止PLC，即开始下载非加密区梯形图程序。



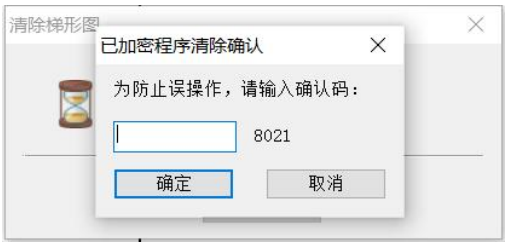
3、梯形图擦除功能

即解除当前设备的加密状态，同时擦除全部梯形图。

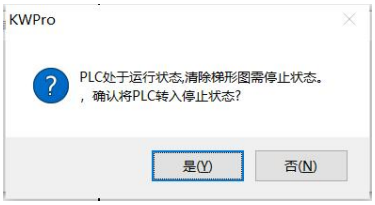
注意：当MC系列PLC中的的梯形图被擦除后，因为PLC中无程序，所以若此时运行PLC，PLC会报错。重新写入正确的梯形图即可恢复。

梯形图擦除操作步骤：

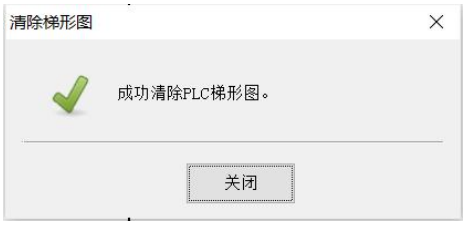
① 点击  →输入确认码



② 点击确认，停止PLC



③ 完成梯形图擦除操作



2.4.3控制器状态

MC系列PLC有两种状态，停止状态和运行状态，通过编程软件上的停止和运行按钮切换两种状态。点击停止按钮进入停止状态，停止状态下下载梯形图，控制器此时不运行，输出等均关闭。梯形图下载完成后点击运行按钮切换至运行状态。运行状态下控制器运行，执行梯形图指令，输出各种信号。

2.5 停电保持

MC系列PLC通过设置可开启停电保持功能。

2.5.1停电保持

设置 D8020=K2222，开启MC系列PLC的停电保持功能。停电保持功能设置后，停电情况下，系统会保存停电保持用的软元件状态等，再次上电时，恢复上次掉电时的状态(即上次掉电时的软元件数值)。

2.5.2软元件停电保持动作

控制器的程序内存以及软元件内存在不同状态下的动作如下区分。下表中动作均是基于 D8020=K2222，即停电保持功能开启而言。

1) 程序内存

项目	电源 OFF	电源 OFF→ON	STOP→RUN	RUN→STOP
程序参数	保持			
程序	保持			

2) 软元件内存

项目		电源 OFF→ON	STOP→RUN
数据寄存器 D	D0-D199	清零	清零
	D200-D7999	保持	保持
扩展寄存器 R	R0-R7999	保持	保持
变址寄存器 VZ	V0-V7 Z0-Z7	保持	保持
定时器 T 寄存器数值	T0-T199	清零	清零
	T200-T245	清零	清零
	T246-T249	保持	保持
	T250-T255	保持	保持
定时器T线圈、触点	T0-T255	清零	清零
计数器C 寄存器数值	C0-C99	清零	清零
	C100-C199	保持	保持
	C200-C234	保持	保持
	C235-C255	保持	保持
计数器C线圈、触点	C0-C255	清零	清零
输入继电器 X	X0-X377	复位	复位
输出继电器 Y	Y0-Y377	复位	复位
辅助继电器M	M0-M499	复位	复位
	M500-M3071	保持	保持
状态继电器 S	S0-S499	复位	复位
	S500-S1023	保持	保持

停电不保持软元件，上电、停止切换至运行状态时均为复位状态。停电保持软元件，上电、停止切换至运行状态时均会掉电保持，可利用初始化脉冲 M8002 将其清除。

2.6 梯形图运行时序

梯形图顺控程序是按照【自上而下】到【自左而右】的顺序循环处理。

2.6.1扫描周期

MC系列PLC在 RUN 状态时，执行一次扫描操作所需的时间称为扫描周期。MC系列PLC工作过程就是循环地执行此扫描操作。

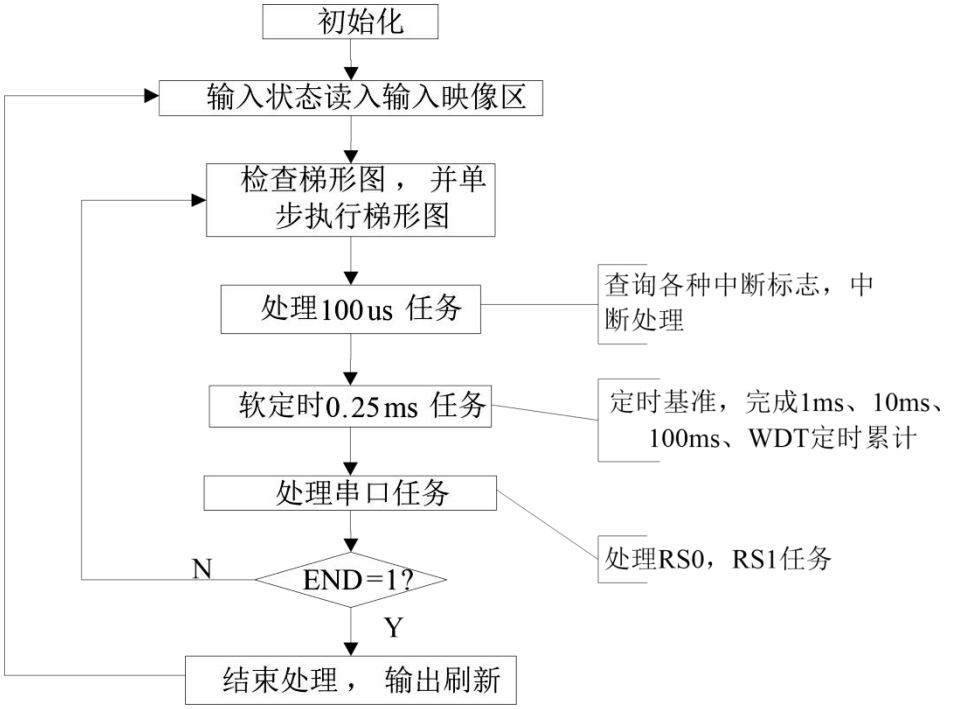
一个完整的扫描周期过程如下：控制器在扫描梯形图程序之前，将所有输入端子的ON/OFF 状态，读入输入映像区。控制器依次从 0 步开始逐条扫描梯形图主程序，至 END 结束。扫描每一条梯形图时，根据逻辑运算的结果，刷新软元件状态或者执行应用指令。梯形图主程序扫描结束后，将输出映像存储区的 ON/OFF 状态传输至输出锁存存储区，从而输出至外部电路。

此过程中，有中断处理任务时，控制器要中断上述扫描过程，处理中断任务，因此中断处理的时间也含括在扫描周期中。

扫描周期的长短取决于指令执行速度、用户程序的长度、各种中断执行时间等。

2.6.2梯形图运行时序图

一个完整的梯形图运行时序图如下：



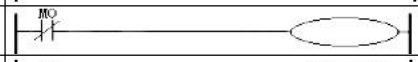
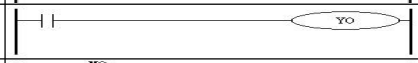
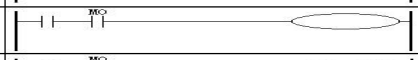
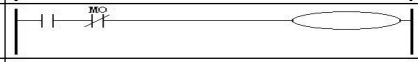
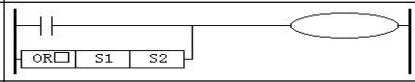
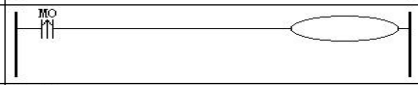
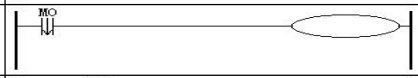
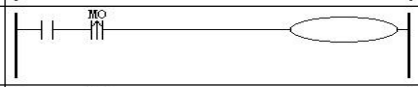
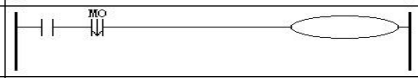
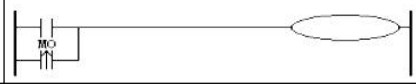
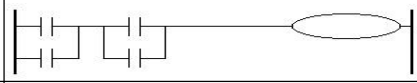
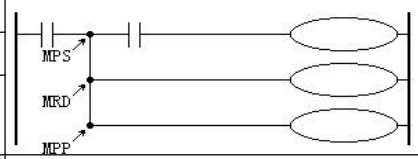
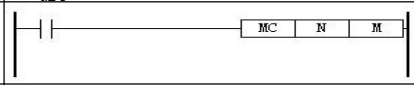
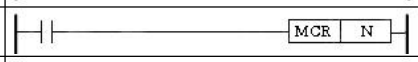
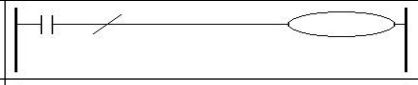
在编写程序时，要合理安排任务。

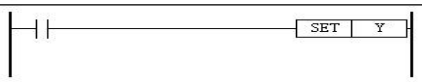
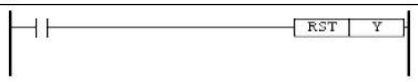
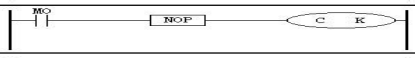
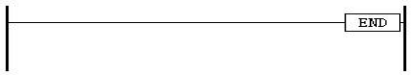
3、指令一览

在本章中，介绍了编程中可以使用的指令的一览。

3.1 基本指令

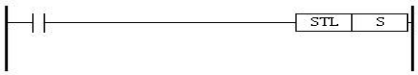
MC系列PLC有如下 28 条基本逻辑指令。

助记符、名称	符号	功能	章节
LD 取		运算开始常开触点	7.1
LDI 取反		运算开始常闭触点	7.1
OUT 输出		线圈驱动	7.2
AND 与		常开触点串联连接	7.3
ANI 与反转		常闭触点串联连接	7.3
OR 或		常开触点并联连接	7.4
ORI 或反转		常闭触点并联连接	7.4
LDP 取脉冲上升沿		上升沿检出运算开始	7.5
LDF 取脉冲下降沿		下降沿检出运算开始	7.5
ANDP 与脉冲上升沿		上升沿检出串联连接	7.5
ANDF 与脉冲下降沿		下降沿检出串联连接	7.5
ORP 或脉冲上升沿		上升沿检出并联连接	7.5
ORF 或脉冲下降沿		下降沿检出并联连接	7.5
ORB 回路块或		串联回路块的并联连接	7.6
ANB 回路块与		并联回路块的串联连接	7.7
MPS 存储进栈		压入堆栈	7.8
MRD 存储读栈		读取堆栈	7.8
MPP 存储出栈		弹出堆栈	7.8
MC 主控		公共串联点连接线圈	7.9
MCR 主控复位		公共串联点消除	7.9
INV 反转		运算结果反转	7.10
ALT 交替输出		交替输出	7.11

PLS 上升沿脉冲		上升沿输出	7.12
PLF 下降沿脉冲		下降沿输出	7.12
SET 置位		动作保持	7.13
RST 复位		清除动作保持，寄存器清零	7.13
NOP 空操作		空动作	7.15
END 结束		梯形图程序扫描结束	7.15

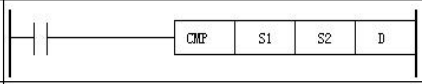
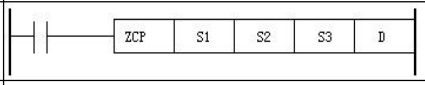
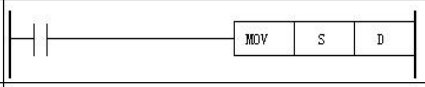
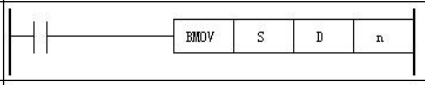
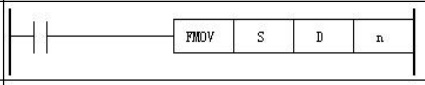
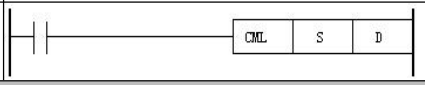
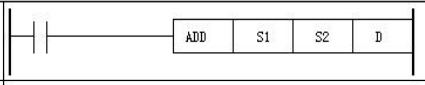
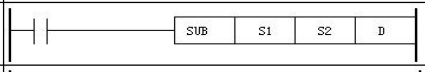
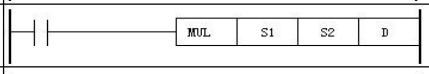
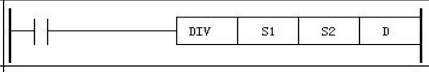
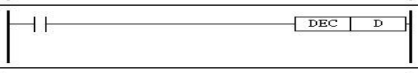
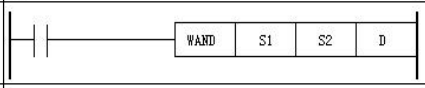
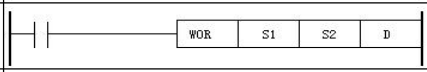
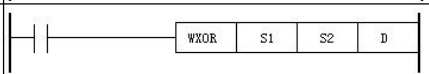
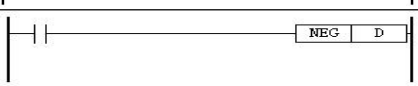
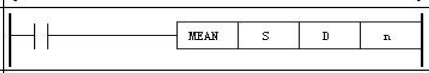
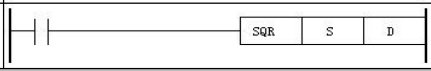
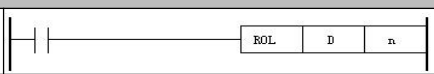
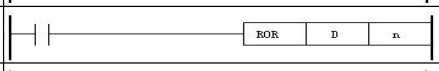
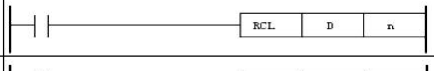
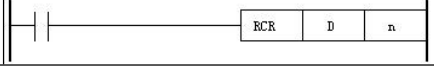
3.2 步进梯形图指令

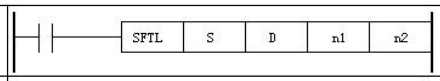
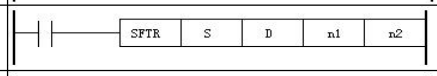
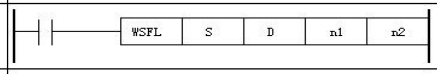
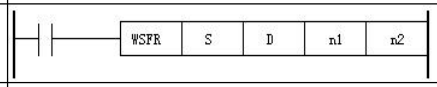
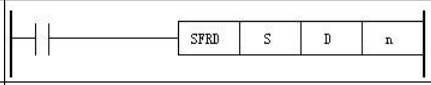
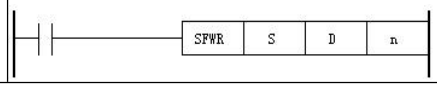
步进梯形图指令可实现工序步进控制。

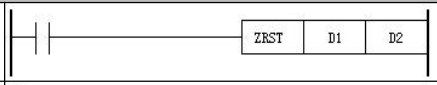
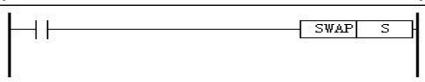
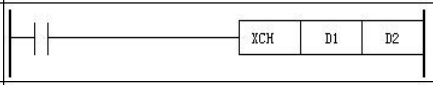
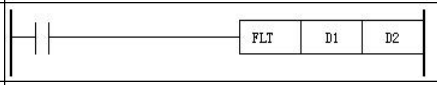
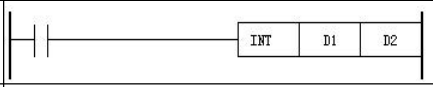
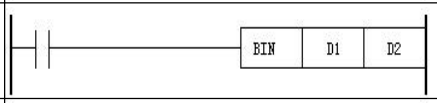
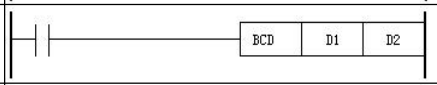
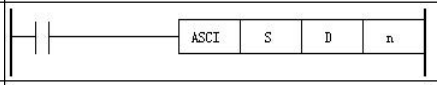
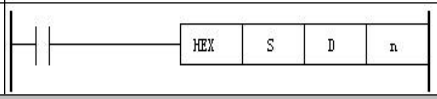
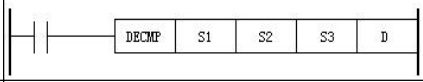
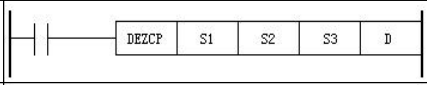
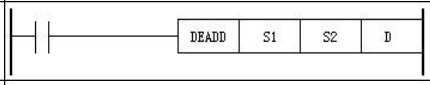
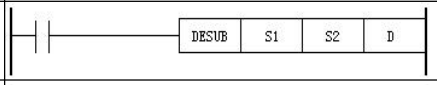
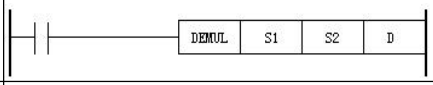
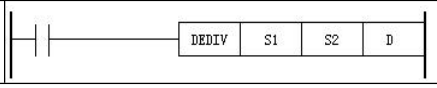
助记符、名称	符号	功能	章节
STL 步进梯形图		步进梯形图开始	流程篇
RET 返回		步进梯形图结束	流程篇

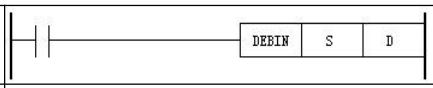
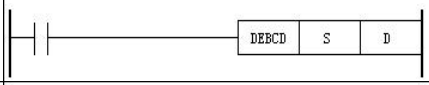
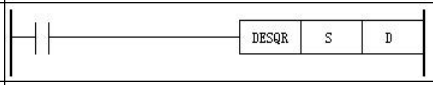
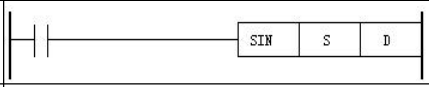
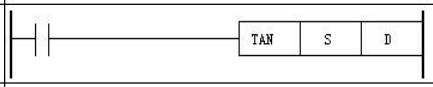
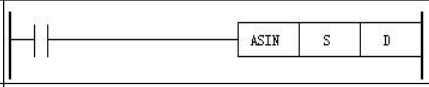
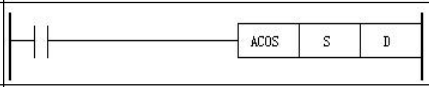
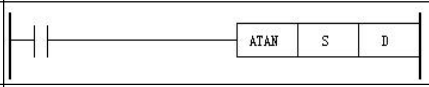
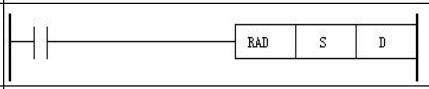
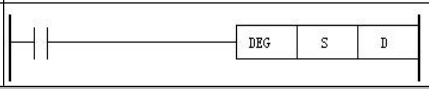
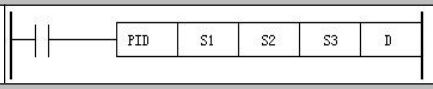
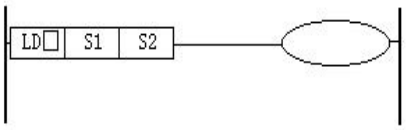
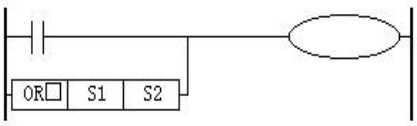
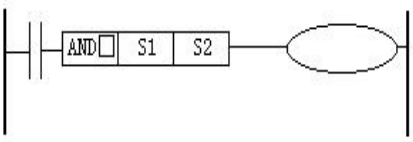
3.3 应用指令

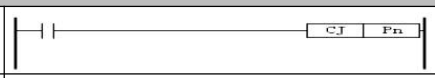
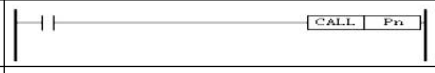
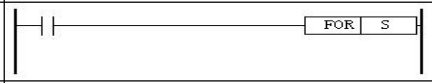
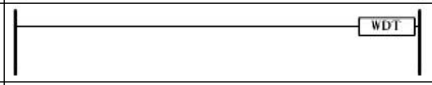
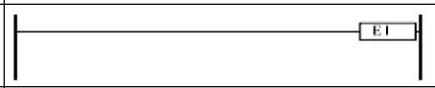
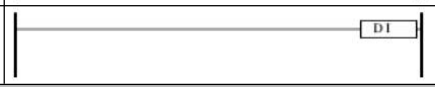
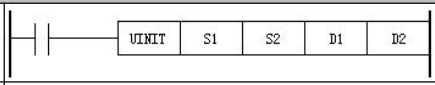
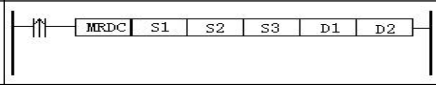
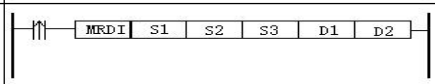
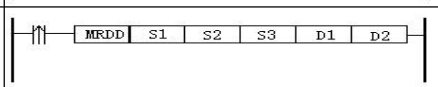
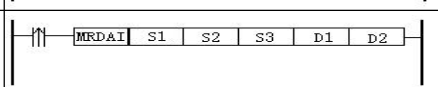
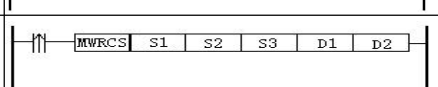
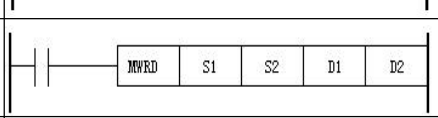
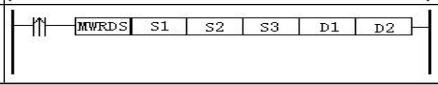
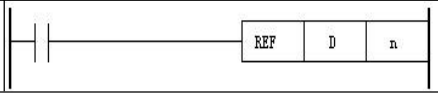
应用指令主要在四则运算、数据处理、高速计数处理、定位输出等时使用。

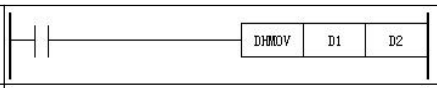
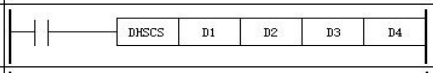
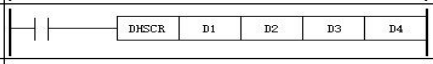
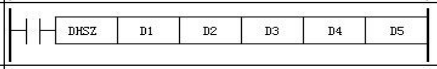
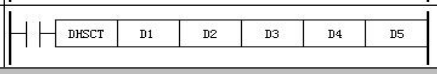
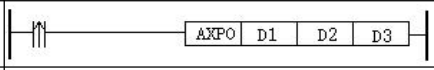
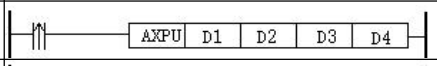
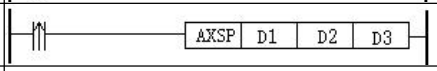
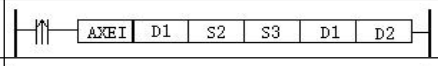
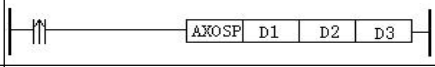
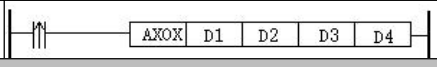
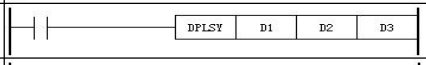
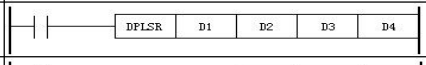
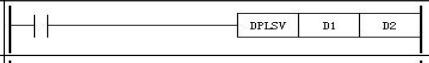
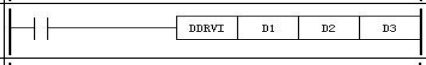
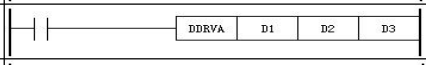
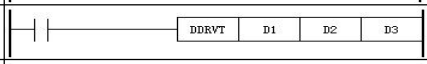
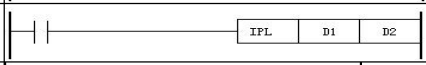
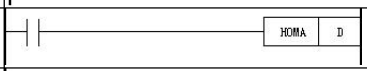
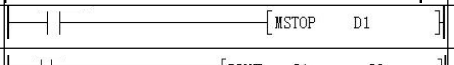
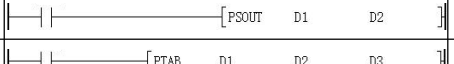
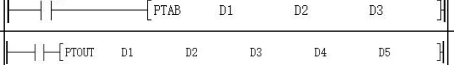
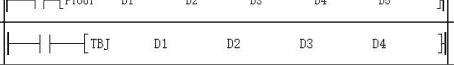
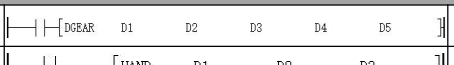
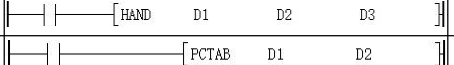
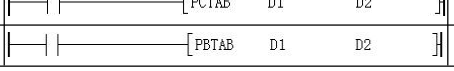
助记符、名称	符号	功能	章节
传送·比较指令			
CMP		数据比较	8.1
ZCP		区间比较	8.2
MOV		传送	8.3
BMOV		成批传送	8.4
FMOV		多点传送	8.5
CML		反向传送	8.6
四则运算指令			
ADD		加法运算	9.1
SUB		减法运算	9.2
MUL		乘法运算	9.3
DIV		除法运算	9.4
INC		加一	9.5
DEC		减一	9.6
WAND		逻辑与	9.7
WOR		逻辑或	9.8
WXOR		逻辑异或	9.9
NEG		求补码	9.10
MEAN		求平均	9.11
SQR		开方	9.12
循环·移位指令			
ROL		循环左移	10.1
ROR		循环右移	10.2
RCL		带进位循环左移	10.3
RCR		带进位循环右移	10.4

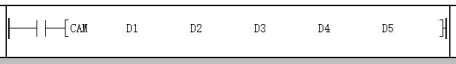
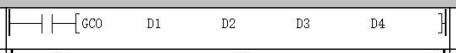
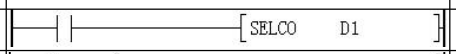
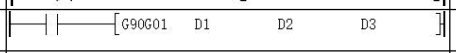
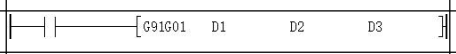
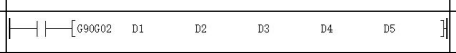
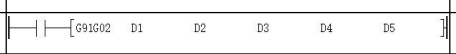
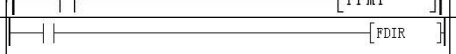
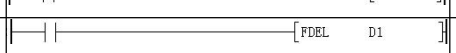
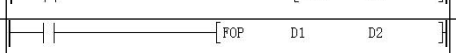
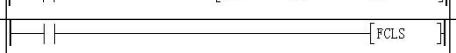
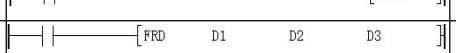
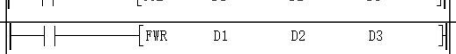
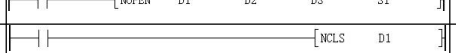
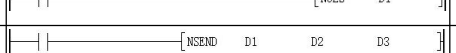
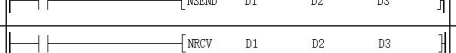
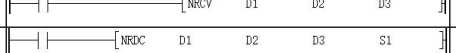
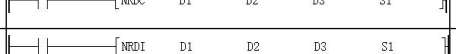
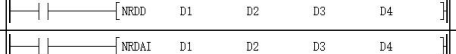
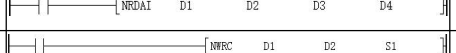
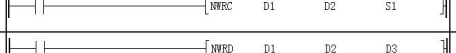
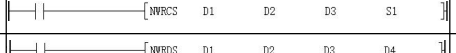
SFTL		位左移	10.5
SFTR		位右移	10.6
WSFL		字左移	10.7
WSFR		字右移	10.8
SFRD		移位读出	10.9
SFWR		移位写入	10.10

数据处理转换指令			
ZRST		成批复位	11.1
SWAP		高低字节交换	11.2
XCH		两个数据交换	11.3
FLT		整数转二进制浮点数	11.4
INT		二进制浮点数转整数	11.5
BIN		BCD 转二进制	11.6
BCD		二进制转 BCD	11.7
ASCI		十六进制转 ASCII	11.8
HEX		ASCII 转十六进制	11.9
浮点数运算指令			
DECMP		浮点数比较	12.1
DEZCP		浮点数区间比较	12.2
DEADD		浮点数加法	12.3
DESUB		浮点数减法	12.4
DEMUL		浮点数乘法	12.5
DEDIV		浮点数除法	12.6

DEBIN		十进制浮点数转二进制浮点数	12.7
DEBCD		二进制浮点数转十进制浮点数	12.8
DESQR		浮点数开方	12.9
SIN		浮点数 SIN 运算	12.10
COS		浮点数 COS 运算	12.11
TAN		浮点数TAN 运算	12.12
ASIN		浮点数反 SIN 运算	12.13
ACOS		浮点数反 COS 运算	12.14
ATAN		浮点数反 TAN 运算	12.15
RAD		浮点数弧度 RAD 运算	12.16
DEG		浮点数DEG 运算	12.17
专用指令			
PID		PID 控制	13.1
触点比较指令			
LD□ LD=		触点比较(S1)=(S2)导通	14.1
LD>		触点比较(S1)>(S2)导通	14.1
LD<		触点比较(S1)<(S2)导通	14.1
LD<>		触点比较(S1)≠(S2)导通	14.1
LD>=		触点比较(S1)≥(S2)导通	14.1
LD<=		触点比较(S1)≤(S2)导通	14.1
OR□ OR=		触点比较并联(S1)=(S2)导通	14.2
OR>		触点比较并联(S1)>(S2)导通	14.2
OR<		触点比较并联(S1)<(S2)导通	14.2
OR<>		触点比较并联(S1)≠(S2)导通	14.2
OR>=		触点比较并联(S1)≥(S2)导通	14.2
OR<=		触点比较并联(S1)≤(S2)导通	14.2
AN□ AND=		触点比较串联(S1)=(S2)导通	14.3
AND>		触点比较串联(S1)>(S2)导通	14.3
AND<		触点比较串联(S1)<(S2)导通	14.3
AND<>		触点比较串联(S1)≠(S2)导通	14.3
AND>=		触点比较串联(S1)≥(S2)导通	14.3
AND<=		触点比较串联(S1)≤(S2)导通	14.3

程序流程指令			
CJ		条件跳转	流程篇
CALL		子程序调用	流程篇
SRET		子程序返回	流程篇
FEND		主程序结束	流程篇
FOR		循环范围开始	流程篇
NEXT		循环范围结束	流程篇
WDT		监视定时器	流程篇
IRET		中断返回	流程篇
EI		允许中断	流程篇
DI		禁止中断	流程篇
MODBUS 通信指令			
UINIT		串口初始化指令	通信篇
MRDC		读线圈的当前状态	通信篇
MRDI		读输入线圈的当前状态	通信篇
MRDD		读寄存器当前值	通信篇
MRDAI		读输入寄存器当前值	通信篇
MWRC		强置一个线圈	通信篇
MWRCS		强置多个线圈	通信篇
MWRD		赋值一个寄存器	通信篇
MWRDS		赋值多个寄存器	通信篇
高速处理指令			
REF		输入输出刷新	运动篇

DHMOV		高速读出	运动篇
DHSCS		高速比较置位	运动篇
DHSCR		高速比较复位	运动篇
DHSZ		高速区间比较	运动篇
DHSCT		高速表格比较输出	运动篇
轴参数指令			
AXPO		轴端口指定	运动篇
AXPU		轴当量指定	运动篇
AXSP		轴速指定	运动篇
AXAS		轴加减速指定	运动篇
AXEI		轴事件输入点指定	运动篇
AXOSP		回原点速度指定	运动篇
AXOX		回原点参数指定	运动篇
单轴基本指令			
HOME		回原点	运动篇
DPLSY		定速定长输出	运动篇
DPLSR		加减速定长输出	运动篇
DPLSV		变速输出	运动篇
DDRVI		相对定位	运动篇
DDRVA		绝对定位	运动篇
DDVIT		中断定位	运动篇
IPL		插补输出	运动篇
HOMA		简易回原点	运动篇
DPTO		多段速输出	运动篇
MSTOP		停止脉冲输出	运动篇
PSOUT		定位输出	运动篇
PTAB		普通制表	运动篇
PTOUT		制表输出	运动篇
TBJ		贴标机	运动篇
单轴关联指令			
DGEAR		电子齿轮	运动篇
HAND		手摇轮	运动篇
PCTAB		追剪正程/滚切制表	运动篇
PBTAB		追剪返程制表	运动篇

CAM		电子凸轮输出	运动篇
两轴插补指令			
GCO		设置坐标系	运动篇
SELCO		选择坐标系	运动篇
G90G01		两轴直线绝对位置插补	运动篇
G91G01		两轴直线相对位置插补	运动篇
G90G02		两轴顺圆弧绝对位置插补	运动篇
G91G02		两轴顺圆弧相对位置插补	运动篇
G90G03		两轴逆圆弧绝对位置插补	运动篇
G91G03		两轴逆圆弧相对位置插补	运动篇
FLASH指令			
FFMT		FLASH格式化	运动篇
FDIR		FLASH状态查看	运动篇
FDEL		FLASH删除文件	运动篇
FOP		FLASH打开文件	运动篇
FCLS		FLASH关闭文件	运动篇
FRD		FLASH读文件	运动篇
FWR		FLASH写文件	运动篇
以太网指令			
NOPEN		通信任务创建	通信篇
NCLS		通信任务关闭	通信篇
NSEND			通信篇
NRCV		自由格式通讯接收	通信篇
NRDC		读多个线圈	通信篇
NRDI		读多个输入线圈	通信篇
NRDD		读多个出寄存器	通信篇
NRDAI		读多个输入寄存器	通信篇
NWRC		写单个线圈	通信篇
NWRD		写单个寄存器	通信篇
NWRCS		写多个线圈	通信篇
NWRDS		写多个寄存器	通信篇

4、软元件的作用和功能

在本章中，主要介绍输入输出继电器、辅助继电器、状态继电器、定时器、计数器、数据寄存器等各种软元件的功能和作用。

4.1 软元件一览表

简称	代号及点数范围	说明
输入 X	X000～X377(8 进制)	输入继电器，单机点数由MC系列PLC具体型号而定
输出 Y	Y000～Y377(8 进制)	输出继电器，单机点数由MC系列PLC具体型号而定
辅助 M	M0～M3071	辅助继电器
	M8000～M8255	特殊辅助继电器
状态 S	S0～S1023	状态继电器
时间 T	T0～T255	T0～T199，100ms 时基定时器
		T200～T245，10ms 时基定时器
		T246～T249，1ms 时基累计定时器
		T250～T255，100ms 时基累计定时器
计数 C	C0～C255	C0～C199，16 位增计数器
		C200～C234，32 位可逆计数器，计数方向由 M8200～M8234 控制
		C235～C255，32 位高速计数器(与高速计数指令配合使用)
寄存器 D	D0～D7999	D0～D7999，用户使用区
	D8000～D8255	系统寄存器区
扩展寄存器R	R0～R15999	R0～R59，内部保留区
		R60～R83，输出轴脉冲显示区
		R84～R99，内部保留区
		R100～R259，软件虚拟输入轴参数区
		R260～R299，内部保留区
		R300～R899，软件虚拟输出轴参数区
		R900～R999，内部保留区
		R1000～R1199，内部保留区
		R1200~R1899，内部保留区
		R1900~R1999，内部保留区
		R2000~R2049，FLASH参数区
		R2050~R2099，内部保留区
		R2100~R2259，KLink总线资源配置参数区
		R2260~R15999，内部保留区
变址 V，Z	V0～V7，Z0～Z7	与其他软元件配合使用，如 V0=10，D0V0 的地址为 D10
常数 K，H	带符号整数型范围	K 为 10 进制表达数，H 为 16 进制表达数，如 H10=K16
程序指针 P	P0～P127	做程序跳转或子程序调用使用
中断指针 I	I000～I999	适用于各类型中断的指示标签
程序容量	2K，4K，8K，12K，16K	MC系列PLC所适用的梯形图步长控制

MC系列PLC在编写梯形图时，请严格按照说明书规则进行编程，否则可能造成 PLC 梯形图运行故障。当MC系列PLC出现运行故障时，按以下顺序进行 PLC 的复位。

- 断电后重新上电
- 在上电后的 1~3 秒的时间内，点击编程视窗的“PLC 停止”按钮
- 出现 PLC 已停止运行的对话框后，下载正确的梯形图以恢复MC系列PLC的正常运行
- 后述所有掉电保持的软元件是指使能了此功能，即将 D8020 赋值K2222。

4.2 输入继电器［X］

● 输入继电器 X 作用是接收外部开关量信息，将接收到的外部信息送到 CPU，经过用户程序的处理由输出继电器 Y 控制外部设备，完成控制任务。因此在 PLC 的完整程序中，X、Y 是相互依存的。

● X 的基本功能是作位元件使用，也可将其组合作为字元件来处理（如 MOV K2X0 D1000，此语句意思为将 X0-X7 的状态，按X0 为低位，X7 为高位顺序排列，不够的位补零，组合成一个字，将值赋给D1000）。

● 上述 K2X0 中的 2 根据需要可以是 1～8 中的任意值，设“1～8”为 n，KnX0 表达的意思是以 X0 为最低位，将 n*4 个以 X0 为起始的连续的 X 状态组合成字的值来处理。16 位操作时 n<=4，n>4 时要按 32 位数据操作。这样组合操作时注意 X 的起始地址编号，避免超出 X 的最大编号范围。

● 输入继电器根据接口硬件电路的不同，分为普通型和高速型（普通型与高速型具体指标参考硬件说明）。普通型无法做高速型使用，但高速型可做普通型使用。

4.2.1 输入继电器编号

输入继电器的编号是控制器输入端口对应的映射标识。这些编号使用 8 进制表达，因此不存在如 8、9 这样的数值。

MC系列PLC的 X 输入继电器编号如下：

控制器型号	X 编号	接口类型	硬件点数
MC-16M16R/T	X000～X017	漏型	16

4.2.2 功能与动作说明

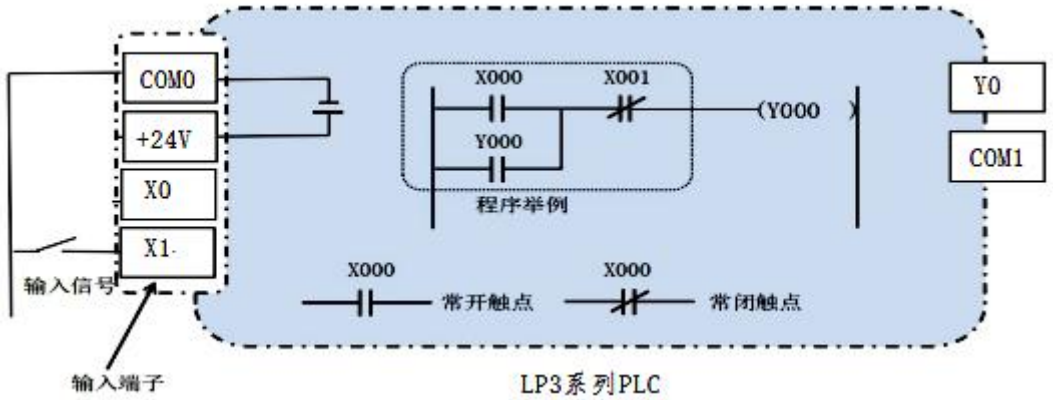
● 输入端子是MC系列PLC接收外部开关信号的窗口。

● 从输入端子接入的信号，在 PLC 内部经滤波、隔离电路处理，开关状态才被中央处理器读入输入映像区（关联到软元件输入继电器 X 的对应编号），赋予它继电器属性，具有常开触点与常闭触点。

● 这些常开、常闭触点可在梯形图程序内反复无数次使用。

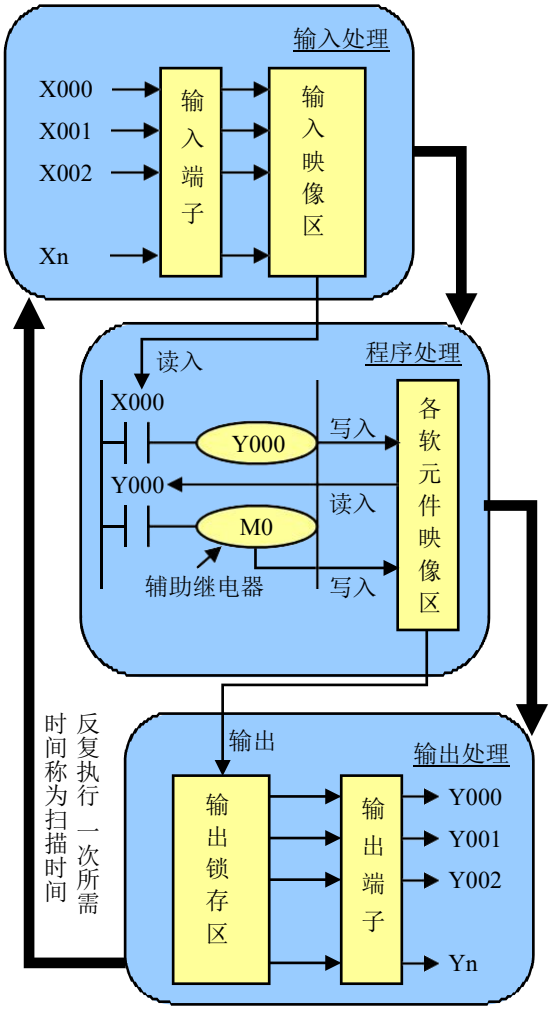
● 特定编号的输入继电器X 通过一些专用指令或用法可以实现一些复杂功能（高速计数、外部中断输入等），见相关指令说明。

● 输入继电器 X 不能通过程序来驱动的（例如：[SET X0]、[RST X0]、[OUT X0]等属于错误用法）。



4.2.3 动作时序

PLC 通过反复执行下面所示的处理，进行顺序控制。采用这种成批输入输出方式时，如图所示， 输入滤波器和输出元件的驱动时间及运算周期也会出现响应滞后的情况。



●输入处理
PLC 在执行程序之前，将所有输入端子的 ON/OFF
状态，读入输入映像区。
在执行程序的过程中，即使输入变化，输入映像区的内容也不变，而在下一扫描周期的输入处理时， 读入该变化。
此外，输入端子出现 ON→OFF、OFF→ON 的变化时，在判定 ON/OFF 之前，输入滤波器会造成短时间的响应滞后（毫秒级）。

●程序处理
PLC 根据程序存储器的指令内容，从输入映像区或其他软元件的映像区中读出各软元件的ON/OFF 的状态，从 0 步开始依次进行运算，然后将结果写入映像区。因此，各软元件的映像区存储区随着程序的执行逐步改变其内容。输出继电器的内部触点根据输出映像存储区的内容执行动作。

●输出处理
一旦所有指令执行结束，将输出 Y 的映像存储区的 ON/OFF 状态传输至输出锁存存储区，此为 PLC 的实际输出。PLC 内的外部输出触点，按照输出软元件的响应滞后实践动作。

4.2.4 输入处理便利功能

- 高速输入单向、双向、AB 相[高速计数器]功能
  - ◆ 由于高速计数器是以中断方式对从特定的输入继电器读入的高速脉冲信号进行处理，不受运算时间长短影响。
  - ◆ 计数的结果或是使用高速计数器专用比较指令立即输出，或是对高速计数器的计数值用高速计数中断方式执行特定的处理程序。
  - ◆ 相关指令：DHSCS(高速比较置位)、DHSCR(高速比较复位)、DHSZ(高速区间比较输出)、DHSCT(高速表格比较输出)、REF（立即刷新输入输出）
- 读取最新输入信息[输入输出刷新]功能
  - ◆ 在成批刷新方式下运行的输入信号，只能在 0 步梯形图运算之前被成批读入输入映像区。在执行 END 指令时成批输出经运算处理的输出信息。
  - ◆ 输入输出刷新指令，可以在程序运算过程中获取输入的最新信息，立即输出运算结果。相关指令：REF(输入输出刷新)

- X0~X7 上升沿计数功能

在成批刷新方式下运行的输入信号，对输入信号的响应、处理速度受硬件接口、运算时间等影响，在梯形图运算过程中，监控X对应的计数器可查看输入信号上升沿个数，地址对应关系见下表。

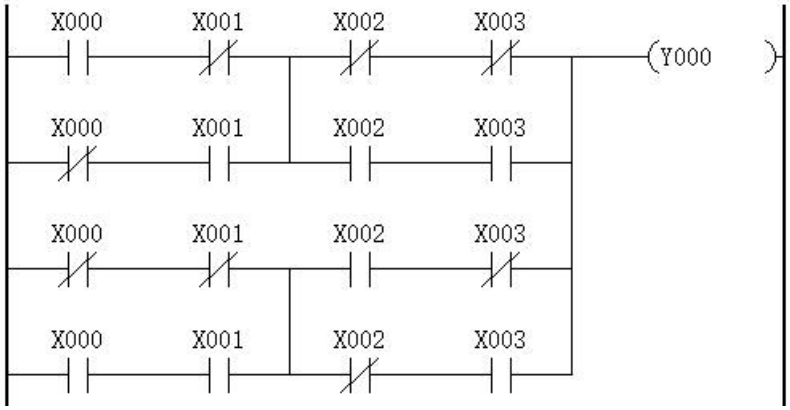
上升沿计数器响应速度：<2KHz

输入端子标号	上升沿计数器地址
X0	D8160
X1	D8161
X2	D8162
X3	D8163

4.2.5 程序应用

利用 4 个开关 X0-X3 共同控制一个灯 Y0，要求任何一个按钮按下灯亮，松开灯灭，且任何两个按钮按下灯不亮。

程序如下：



4.3 输出继电器 [Y]

● 输出继电器 Y 是MC系列PLC控制外部设备，完成控制任务的执行元件。因此在 PLC 的完整程序中，X、Y 是相互依存的。

● Y 的基本功能是作位元件使用，也可将其组合作为字元件来处理（如MOV D1000 K2Y0，此语句意为将D1000的值按二进制方式将低 8 位，从最低位开始分别赋给 Y0～Y7）。

● 上述K2Y0 中的2 根据需要可以是 1～8 中的任意值，设“1～8”为n，KnY0 表达的意思是以 Y0 为最低位，将 n*4 个以 Y0 为起始的连续的 Y 状态组合成字的值来处理，16 位操作时 n<=4，n>4 时要按 32 位数据操作。这样组合操作时注意 Y 的起始地址编号，避免超出 Y 的最大编号范围。

● 输出继电器根据接口硬件电路的不同，分为继电器型和晶体管型，晶体管型分为普通型和高速型（普通型与高速型具体指标参考硬件说明）。普通型无法做高速型使用，但高速型可做普通型使用。

4.3.1 输出继电器编号

输出继电器的编号是输出端口对应的映射标识。这些编号使用 8 进制表达，因此不存在如 8、9 这样的数值。MC系列PLC输出继电器 Y 编号如下：

控制器型号	Y 编号	接口类型	硬件点数
MC-16M16R/T	Y000～Y017	继电器/集电极开路	16

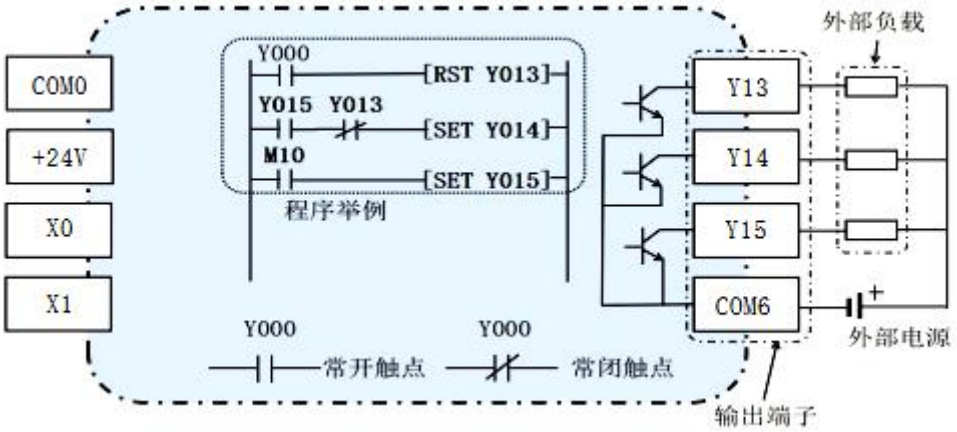
4.3.2 功能与动作说明

● 输出端子是MC系列PLC控制外部设备、完成控制任务的直接接口。

● 中央处理器读入输入映像区的相关输入状态和信息，经过程序运算、时序处理，生成输出信息写入输出锁存区（Y 软元件编号关联到硬件输出端子），最终通过输出端子输出控制信号。

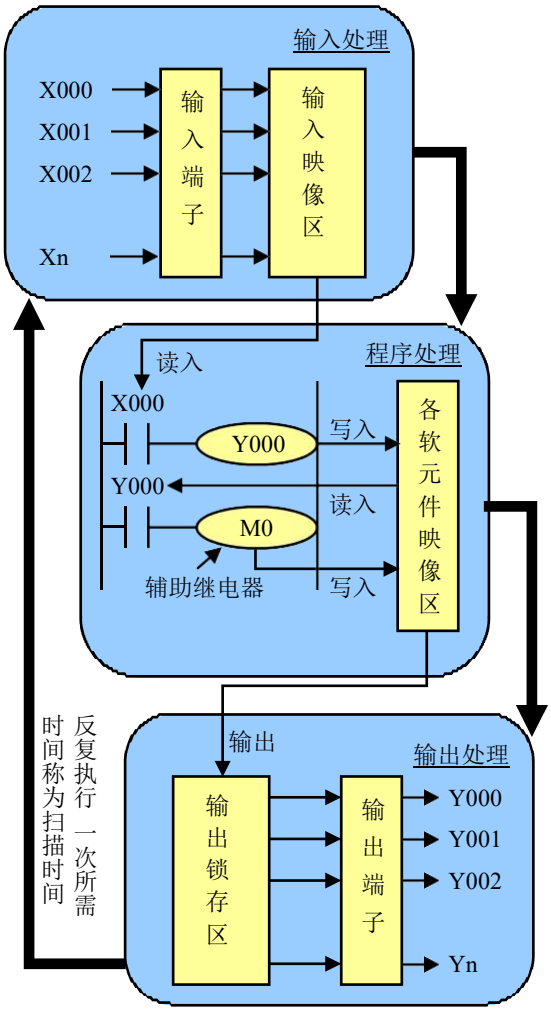
● 这些常开、常闭触点可在梯形图程序内作为输入条件可以反复无数次使用。

● 输出继电器 Y 可以通过程序来驱动（例如：[SET Y0]、[RST Y0]、[OUT Y0]等都是可以操作的）， 程序中支持对同一 Y 编号在不同的步号处进行相应的操作（双线圈输出），由于梯形图程序是成批刷新方式下运行的，最终输出到端子的状态是最后一条操作语句决定的，在编写程序时尽量避免双线圈输出的操作，减小查找问题的难度。



4.3.3 动作时序

PLC 通过反复执行下面所示的处理，进行顺序控制。采用这种成批输入输出方式时，如图所示， 输入滤波器和输出元件的驱动时间及运算周期也会出现响应滞后的情况。



●输入处理
PLC 在执行程序之前，将所有输入端子的 ON/OFF
状态，读入输入映像区。
在执行程序的过程中，即使输入变化，输入映像区的内容也不变，而在下一扫描周期的输入处理时， 读入该变化。
此外，输入端子出现 ON→OFF、OFF→ON 的变化时，在判定 ON/OFF 之前，输入滤波器会造成短时间的响应滞后（毫秒级）。

●程序处理
PLC 根据程序存储器的指令内容，从输入映像区或其他软元件的映像区中读出各软元件的ON/OFF 的状态，从 0 步开始依次进行运算，然后将结果写入映像区。因此，各软元件的映像区存储区随着程序的执行逐步改变其内容。输出继电器的内部触点根据输出映像存储区的内容执行动作。

●输出处理
一旦所有指令执行结束，将输出 Y 的映像存储区的 ON/OFF 状态传输至输出锁存存储区，此为 PLC 的实际输出。PLC 内的外部输出触点，按照输出软元件的响应滞后实践动作。

4.3.4 Y 输出处理便利功能

● 读取最新输入信息[输入输出刷新]功能

在成批刷新方式下运行的输入信号，只能在 0 步梯形图运算之前被成批读入输入映像区。在执行 END 指令时成批输出经运算处理的输出信息。

输入输出刷新指令，可以在程序运算过程中获取输入的最新信息，立即输出运算结果。相关指令：REF(输入输出刷新)

● 脉冲串输出控制的[脉冲输出]功能

相关指令：DPLSY(定速定长脉冲输出)、DPLSR(加减速定长脉冲输出)、DPTO（多段速输出）。

● 定位控制的[定位]功能

相关指令：HOME(回原点)、DDRVI(相对定位)、DDRVA(绝对定位)、DDVIT(中断定位)、DPLSV(变速输出)、IPL(插补)、HOMA(简易回原点)。

4.3.5 程序应用

实现功能：

按下启动按钮，电机 1 启动，间隔 5 秒电机 2 启动，再间隔 5 秒电机 3 启动，按下停止按钮，3 台电机同时停止。

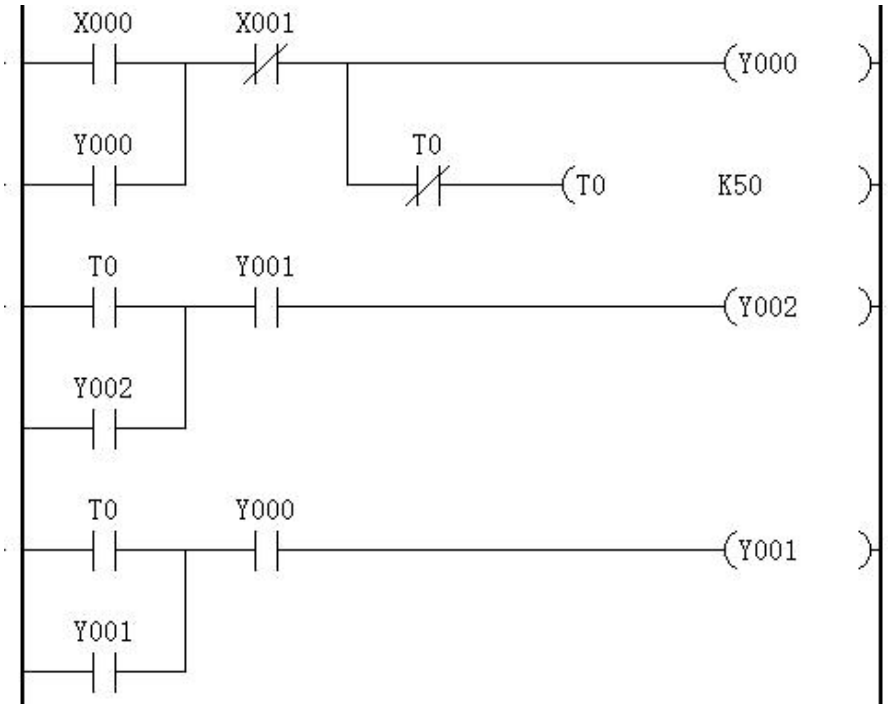
接口定义：

X0、X1 分别为外部接入的启动、停止按钮，Y0、Y1、Y2 分别为电机 1～3。

控制原理：

启动按钮按下，X0 接点闭合，Y0 得电并自锁，第一台电机启动，T0 线圈得电启动计时，5 秒后，T0 输出触点接通（接通一个扫描周期）使得 Y1 得电自锁，第二台电机启动（由于 T0 触点动作时，Y1 触点未接通，所以 Y2 不得电，在第二个扫描周期 Y1 触点接通，但 T0 触点断开，所以 Y2 仍不得电），再过 5 秒， T0 输出触点接通，使得Y2 得电并自锁，第 3 台电机启动。

按下停止按钮，X1 接点闭合，使得 Y0 失电，Y0 触点断开又使 Y1 失电，Y1 触点断开又使 Y2 失电，3 台电机全部停止。



4.4 辅助继电器[M]

辅助继电器 M 同输出继电器 Y 一样，可以被 PLC 内各种软元件的触点驱动。但 M 不能直接与外部端子相连，仅起辅助逻辑运算的作用。它的常开常闭触点在梯形图程序中可以无数次反复使用。M 在使用时既可作位元件也可作字元件（必须拼成字的形式，如 K2M0）。

上述 K2M0 中的 2 根据需要可以是 1～8 中的任意值。设“1～8”为 n，则 KnM0 表达的意思是以 M0 为最低位，将 n*4 个以 M0 为起始的连续的 M 状态组合成字的值来处理,这样操作时注意 M 的起始地址编号， 避免超出 M 的编号范围。

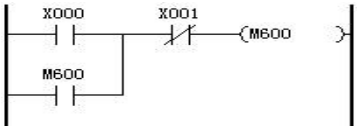
4.4.1 辅助继电器编号

属性	编号范围	掉电保持
普通辅助继电器	M0～M499	掉电清零
普通辅助继电器	M500～M3071	掉电保持
特殊辅助继电器	M8000～M8255	系统使用

- M0～M3071 为普通辅助继电器区，可做触点判断及逻辑运算，无使用限制。
- M8000～M8255 为特殊辅助继电器区，此区域继电器的状态往往是被系统赋予了特殊的含义，比如M8002为上电第一个扫描周期被接通。特殊辅助继电器在一般情况下，只做触点作导通判断使用，应尽量避免修改此区域辅助继电器的状态，以免影响系统正常运行。

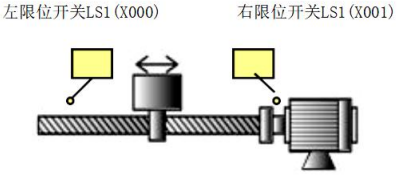
4.4.2 功能与动作说明

MC系列PLC辅助继电器的线圈与输出继电器一样，可由MC系列PLC的各种软元件的触点驱动。它的常开常闭触点在梯形图程序中可以无数次反复使用。但是，该触点不能直接驱动外部负载，外部负载只能输出继电器 Y 驱动。

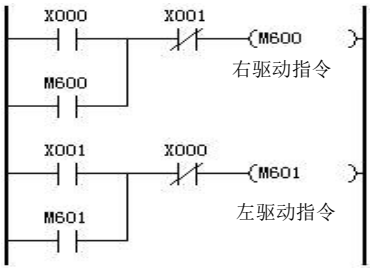


辅助继电器自锁示例

- 左图表示了 M600 动作的自锁示例。在该电路中，如果 X000 接通，M600 动作，即使 X000 断开，由于并联触点导通，M600 也会自己保持动作。但是，如果 X001 的常闭触点断开，则 M600 不会动作。



往复式平台 制动电机

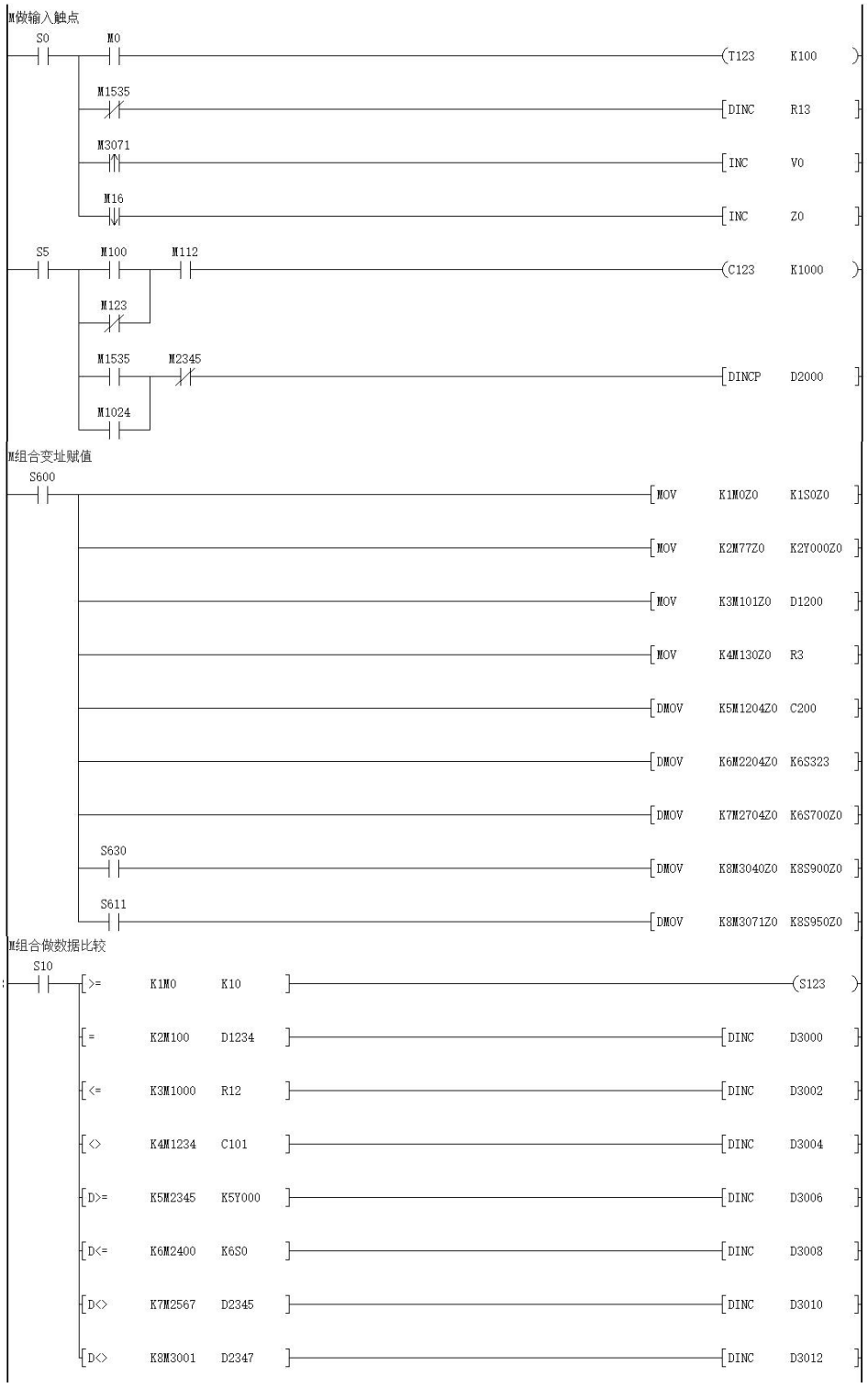


梯形图的功能：

- 可通过 M600/M601 的当前状态，控制左图往复式平台的移动方向。

- X000 = ON（左限位开关）→ M600 = ON → 右驱动 →
  X001 = ON（正转限位开关）→ M600 = OFF， M601=ON → 左驱动。

4.4.3 程序应用


M做输入触点
S0
M0
T123 K100
M1535
DINC R13
M3071
INC V0
M16
INC Z0
S5
M100
M112
C123 K1000
M123
M1535
M2345
DINCP D2000
M1024
M组合变址赋值
S600
MOV K1M0Z0 K1S0Z0
MOV K2M77Z0 K2Y000Z0
MOV K3M101Z0 D1200
MOV K4M130Z0 R3
DMOV K5M1204Z0 C200
DMOV K6M2204Z0 K6S323
DMOV K7M2704Z0 K6S700Z0
S630
DMOV K8M3040Z0 K8S900Z0
S611
DMOV K8M3071Z0 K8S950Z0
M组合做数据比较
S10
>= K1M0 K10
S123
= K2M100 D1234
DINC D3000
<= K3M1000 R12
DINC D3002
<> K4M1234 C101
DINC D3004
D>= K5M2345 K5Y000
DINC D3006
D<= K6M2400 K6S0
DINC D3008
D<> K7M2567 D2345
DINC D3010
D<> K8M3001 D2347
DINC D3012

4.5 状态继电器[S]

状态继电器 S 的主要作用是在步进顺控(STL)指令中表示工艺状态，以流程的方式，可以使程序变得结构清晰易懂，并且容易修改。当程序中没有步进顺控(STL)指令或步进顺控(STL)指令中用到的的范围外的S，可以做辅助继电器 M 一样使用。

4.5.1 状态继电器编号

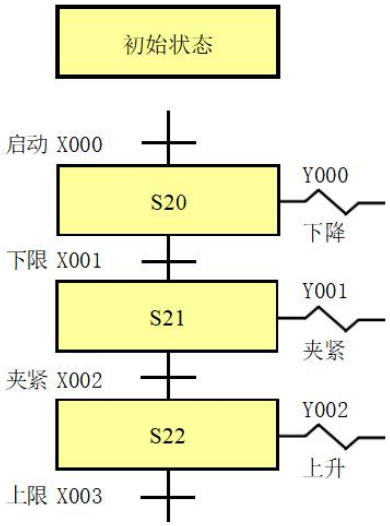
属性	编号范围	掉电保持
通用状态继电器	S0～S499	掉电清零
通用状态继电器	S500～S1023	掉电保持

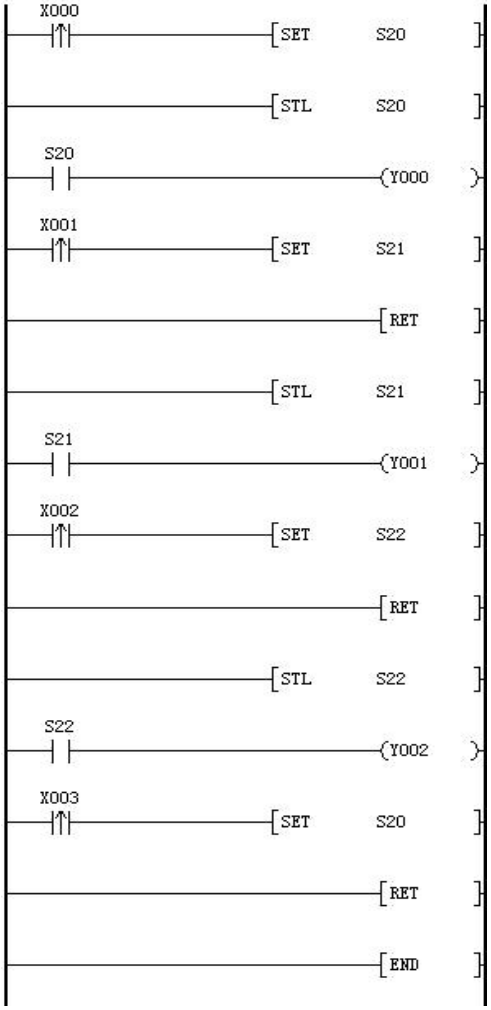
4.5.2 功能与动作说明

状态继电器 S 与辅助继电器 M 一样，有无数的常开/常闭触点，在顺序程序内可随意使用。在不用于步进顺控(STL)指令时，状态继电器 S 可在触点判断和逻辑运算中使用。

状态继电器 S 的编号为 S0～S1023，一共有 1024 个点位可供使用。

4.5.3 步序状态 S



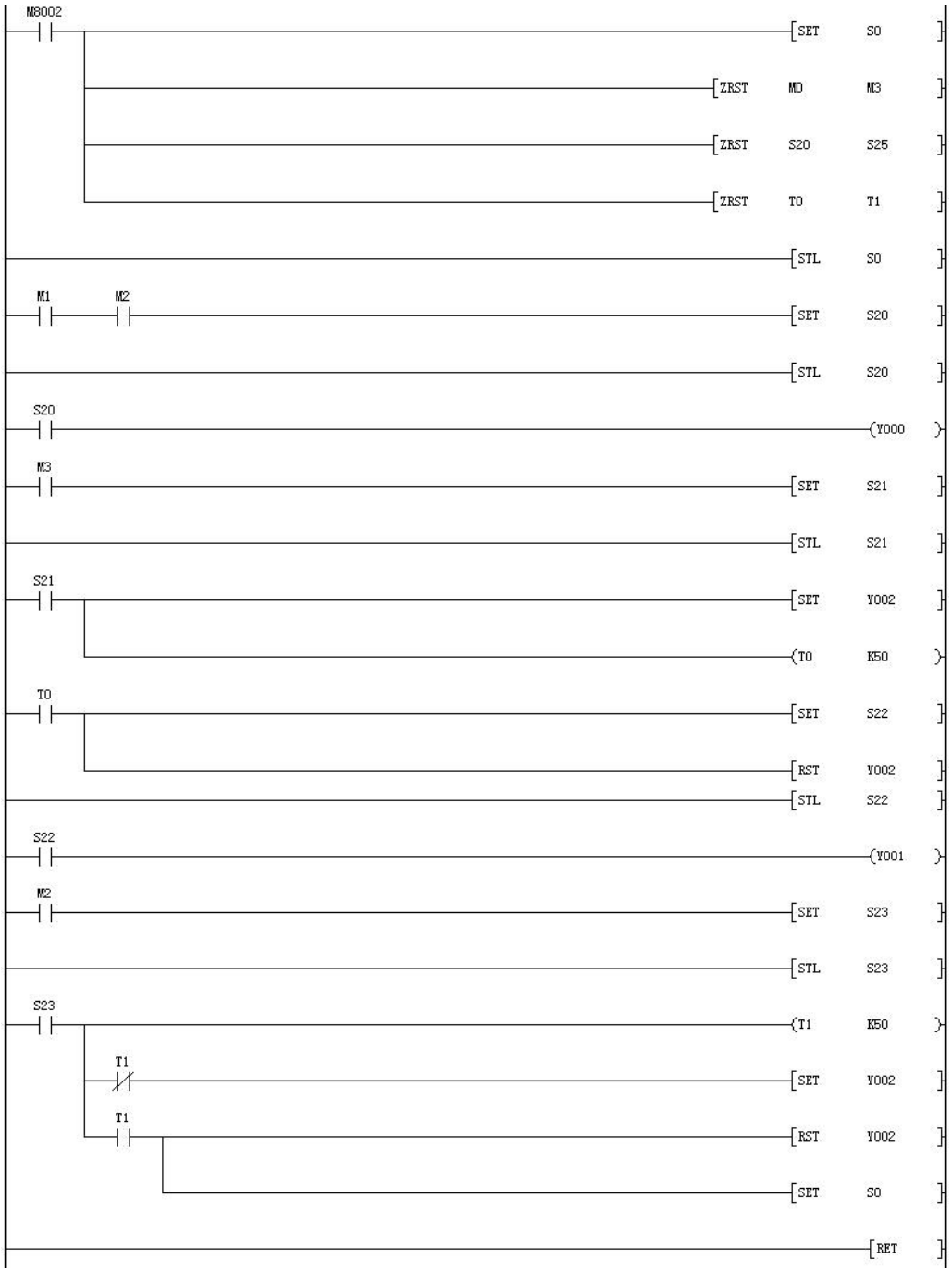


- 状态 S 是工序步进控制简易编程的重要软元件，经常与步进梯形图指令 STL 结合使用。
- 如上图所示的工序步进控制中，如果启动信号 X000 为 ON 则状态 S20 置位（ON），下降用的电磁阀 Y000 开始动作。其结果是，若下限限位开关 X001 为 ON，则状态 S21 置位（ON），夹紧用的电磁阀 Y001 动作。
- 如果夹紧动作确认的限位开关 X002 为 ON，则状态 S22 置位（ON）。
- 随着状态动作的转移，状态自动返回原状态。
- 右图为示例完整梯形图。

4.5.4 程序应用

实现功能：启动前，电机在原点位置，按下启动按钮电机正转，碰到限位开关进行制动，制动 5 秒后，电机反转，回到原点位置，制动 5 秒后，机器停止，等待下一次启动信号。

资源定义：M1、M2、M3 分别为启动标志、原点标志、限位标志，Y0、Y1、Y2 分别为电机正转、反转、制动。



4.6 定时器[T]

控制系统中控制对象是否动作是在某个条件满足多长时间后才执行，或者动作多长时间后关闭等等，这些实际应用中需要等待时间的场合都可用定时器实现。

4.6.1 定时器编号（10 进制表示）

定时器编号	资源个数	定时时基	定时特性	掉电保持
T0～T199	200	100 ms	不累计	掉电清零
T200～T245	46	10 ms	不累计	掉电清零
T246～T249	4	1 ms	累计	掉电保持
T250～T255	6	100 ms	累计	掉电保持

4.6.2 功能和动作说明

● 定时器 T（亦称时间继电器）的主要作用是定时控制。

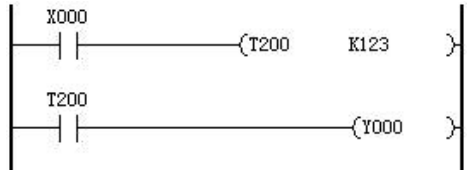
● 定时器 T 可以作为数据寄存器使用，进行赋值、取值、数据比较等操作。

● 使用 OUT 指令开启定时器，当开启定时器的条件满足时，定时器开始计时，当计时值达到指定值时，其触点闭合，若开启定时器的条件不断开，则计时值继续增加，至最大值 K32767 并保持计时值为 K32767。开启定时器的条件断开后，定时器的计时值清零、触点断开。

● 如果是累计定时器，需使用 RST 指令复位、清零。

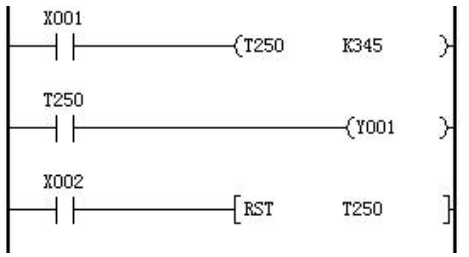
● 当启动了掉电保持，定时器运行时PLC发生断电，再次上电运行，启动触点即使是断开状态（定时器没有被启动运行），掉电保持定时器的值不会被清零，为断电时的计时值，线圈和触点是复位状态。启动定时器运行然后再断开启动触点，掉电保持定时器的值不会被清零，且计时值是断开时的计时值基础上累加的。使用定时器时注意，断电后再上电掉电保持定时器的值要为 0，请在梯形图第一扫描周期使用 RST 指令清零。

示例：



● 非累计型

当 X000 为ON，T200 从0 开始计时，每 10mS 计时值加 1。当计时值等于K123 时，即 10mS*123=1.23S 时定时器 T200 的常开触点闭合，Y0 输出为ON；当 X000 为OFF 时，定时器 T200 计时值清零，常闭触点闭合，Y0 输出为 OFF。

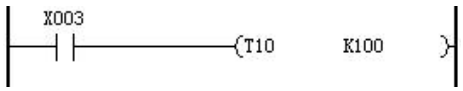


● 累计型（设T250 初始值为 0）

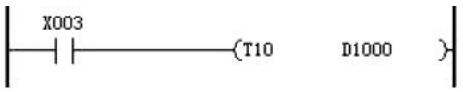
当X001 为 ON，则 T250 从0 开始计时，每 100mS 计时值加 1。如果该值达到设定值 K345 时，定时器 T250 的常开触点闭合。若在计时过程中，X001 由 ON 变为 OFF，则 T250 的计时值不会清零，为 X001 由ON 变为 OFF 时的计时值，当 X001 再变为 ON 时，T250 计时值将在前一次计时值的基础上累加。当 X002 为ON 时，定时器 T250 计时值清零复位，常闭触点闭合。

4.6.3 设定值指定方法

可通过赋值改变定时器当前的值。定时器的定时值可以用常数 K 直接指定，也可以用数据寄存器间接指定。



- T10 是以 100ms（0.1s）为单位的定时器。其定时设定值指定为常数 K100，则定时时间为 0.1s*100=10s，X3 接通启动定时。



- T10 是以 100ms（0.1s）为单位的定时器。其定时设定值指定为 D1000 的值，当需要根据控制对象来修改定时时长时， 可将 D1000 作为可设置对象放置在 HMI 上，用户自行设定。

4.6.4 定时器精度

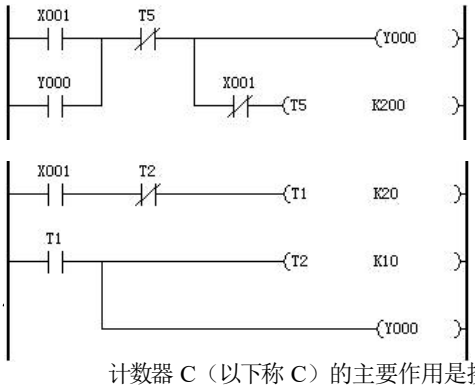
- 定时器分 1mS、10mS、100mS 三种时基，不同时基定时器定时最大误差为一个时基单位，但执行动作的时间会受到程序扫描周期的影响。
- 由于不同时基定时器的定时误差不同，可根据应用对象对时间精度要求的不同而选择不同时基的定时器。

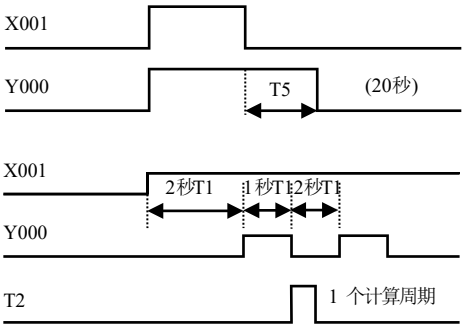
4.6.5 定时器做数据软元件使用

- 定时器 T 可以作为数据寄存器使用，能进行赋值、取值、数据比较等操作。

4.6.6 程序应用

使用下列程序可实现输出延时关断定时器、单脉冲输出定时器和闪烁定时器。





计数器 C（以下称 C）的主要作用是接收脉冲信号并计数，到达设定值时其触点动作，起控制作用。此外， 可作定时器和数据寄存器使用。

4.7 计数器[C]

4.7.1 计数器编号（10 进制表示）

计数器编号	计数器性质	计数器类型	计数方向	掉电保持
C0～C99	普通	16 位	增	掉电清零
C100～C199	普通	16 位	增	掉电保持
C200～C234	普通	32 位	增、减	掉电保持
C235～C255	高速	32 位	增、减	掉电保持

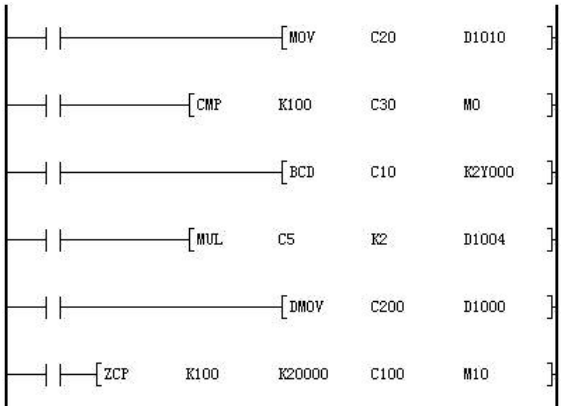
4.7.2 计数器特征

- 根据编号的不同，C 分为16 位和32 位两类，它们都由RST 指令清零、复位。
- 16 位C 是增计数器，计数设定值可用K直接设定，亦可用数据寄存器间接设定，设定范围为 1～ 32767。
- 32 位C 是环形可逆计数器，可进行增、减计数。计数值到达临界值的变化特点：K-2147483648 减1 变为K2147483647；K2147483647 加1 变为K-2147483648。
- 32 位 C 的设定值与 16 位类似，既可用K 直接设定，亦可用数据寄存器间接设定，但必须占用编号连续的两个数据寄存器，例如（D1001、D1000）。

16 位计数器与 32 位计数器的特点如下表所示。可按计数方向的切换与计数范围的使用条件来分开使用。

项目	16 位计数器	32 位计数器
计数方向	增计数	增、减计数可切换使用
设定值范围	1～32767	-2147483648～+2147483647
设定值的指定方式	常数 K 或数据寄存器	同左，但是将占用连续 2 个数据寄存器
当前值的变化	到达设定值后不变化	到达设定值后仍然变化（环形计数）
输出接点	到达设定值后保持动作	根据条件可变
复位动作	执行 RST 命令时，计数器的当前值清零，输出线圈/触点 OFF。	
当前值寄存器	16 位	32 位

4.7.3 程序应用



- C20(当前值)→D1010 传送
- 将 10 进制整数 100 和 C30(当前值)比较，结果输出到 M0～M2。
- 将 C10(当前值)作 BCD 变换，输出到 Y000～Y007。
- 将 C5(当前值)翻倍，向(D1005，D1004)传送。
- 将 C200(当前值)→(D1001，D1000)传送。
- 将 C200(当前值)与 10 进制整数 100～20000 的区间作比较，结果输出到 M10～M12。

4.7.4 计数器计数方向

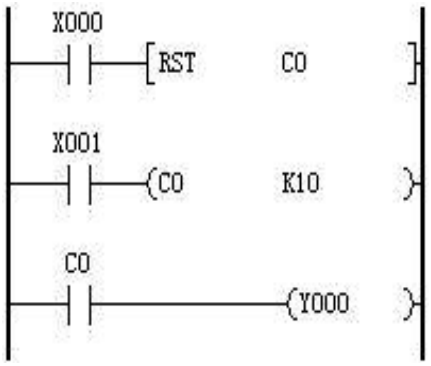
32 位计数器都可以实现增、减计数，可通过设置方向位来确定计数方向，高速计数器还可以通过对应的方向端子 X 来设定计数方向。

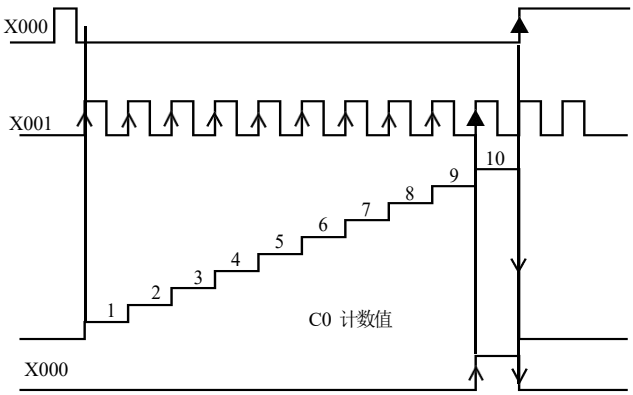
下表为 C200～C255 的方向位对应关系。即 M8200 为 C200 的计数方向设置位，方向位与计数方向的关系：M8200 为 OFF 时，C200 增计数；M8200 为 ON 时，C200 减计数，其余计数器类推。

计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换
C200	M8200	C201	M8201	C202	M8202	C203	M8203
C204	M8204	C205	M8205	C206	M8206	C207	M8207
C208	M8208	C209	M8209	C210	M8210	C211	M8211
C212	M8212	C213	M8213	C214	M8214	C215	M8215
C216	M8216	C217	M8217	C218	M8218	C219	M8219
C220	M8220	C221	M8221	C222	M8222	C223	M8223
C224	M8224	C225	M8225	C226	M8226	C227	M8227
C228	M8228	C229	M8229	C230	M8230	C231	M8231
C232	M8232	C233	M8233	C234	M8234	C235	M8235
C236	M8236	C237	M8237	C238	M8238	C239	M8239
C240	M8240	C241	M8241	C242	M8242	C243	-
C244	-	C245	-	C246	-	C247	-
C248	-	C249	-	C250	-	C251	-
C252	-	C253	-	C254	-	C255	-

4.7.5 功能与动作说明

16 位2 进制增计数器，其有效设定值为K1～K32767（10 进制常数）。设定值 K0 和K1 具有相同的含义，即在第一次计数开始时输出触点就动作。

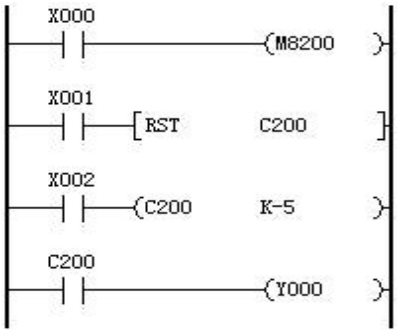




- 计数输入 X001 每接通一次，计数器 C0 的当前值就增加 1，当当前值累加到 10 时，输出触点动作。以后即使计数输入 X001 再接通，计数器的当前值不变。
- 如果复位输入 X000 接通，则执行 RST 指令，计数器的当前值为 0，输出触点复位。
- 计数器的设定值，除用上述常数 K 设定外，还可由数据寄存器指定。例如，指定 D1000，如果 D1000 的值为 K10，计数结果和用常数 K10 设定是一样的。
- 没有计数输入信号时，以MOV 等指令将大于设定值的数据写入计数器当前值寄存器 C，当有下一个计数输入信

号时，输出触点闭合，当前值寄存器数据和设定值相同。

32 位的 2 进制增计数/减计数的设定值有效范围为-2147483648～2147483647（十进制常数）。利用特殊的辅助继电器 M 指定计数的方向。

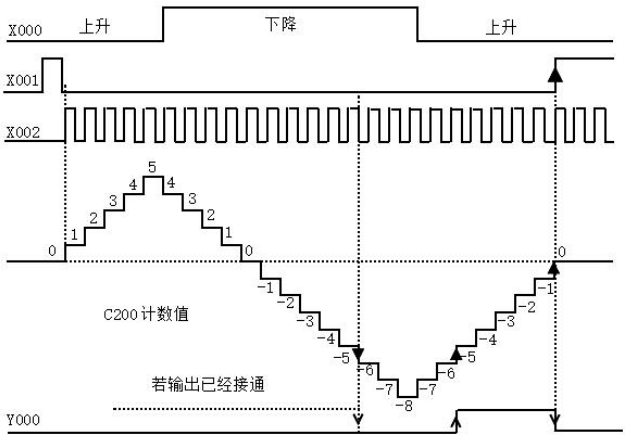


● 左图程序执行结果。

1、X0 为0FF，M8200 为OFF，C200 为增计数；X0 为ON，M8200 为O，C200 为减计数。

2、X1 为 ON，C200 计数值清零，输出触点为 OFF。

3、X1 为OFF，X2 为ON 时，C200 根据X0 的OFF、ON 状态执行增、减计数，X2 每接通一次，计数值加、减 1，当计数值 ≧-5 时，C200 触点为 ON，Y0 输出为 ON

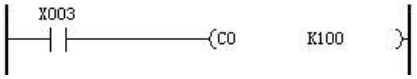


利用 X000 控制C200 计数方向，X001 清零、复位C200，X002 驱动 C200 计数。

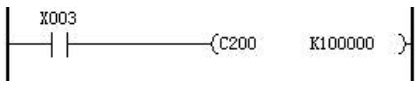
在计数器的当前值由-6→-5 增加时，输出触点置位。在由-5→-6 减少时，输出触点复位。

- 输出触点的置位、复位不影响计数当前值的增减。如果 32 位计数器从 2147483647 开始增计数，则成为 -2147483648；如果从-2147483648 开始减计数，则成为 2147483647。（这类动作被称为环形计数）
- 如果复位输入 X001 为ON，则执行 RST 指令，计数器C200 的当前值变为 0，输出触点也复位。
- 32 位计数器可作为 32 位数据寄存器使用。
- 没有计数信号输入时，以 DMOV等 指令将大于设定值的数据写入当前计数器 C，当有下一个计数输入时，计数器的计数值会在当前写入值的基础上继续计数，计数器触点闭合。

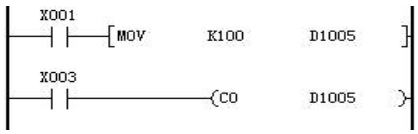
4.7.6 设定值指定方法



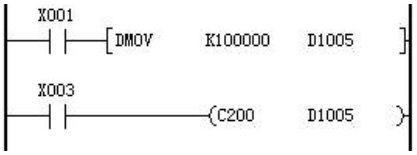
● 16 位常数（10 进制）指定，由于 C0 是 16 位计数 器，因此设定范围为 1～32767， 左图表示计数器 C0设定为 100 次计数。



● 32 位常数（10 进制）指定，由于 C200 是 32 位计数器，因此设定范围为-2147483648～2147483647，左图表 示计数器 C200 设定为 100000 次计数。



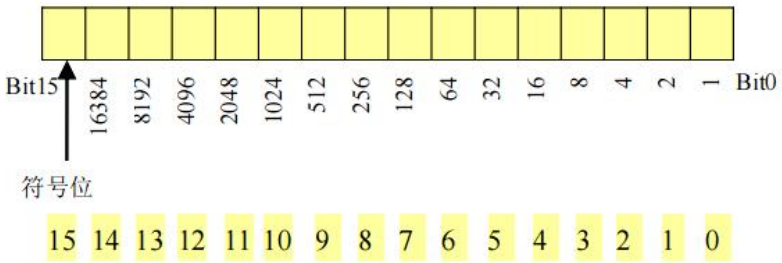
● 16 位间接指定：计数器的设定值为指定数据寄存器的数据（必须是整数且在对应计数器的设定范围）， 左图的计数器的设定值为指定数据寄存器 D1005，运行结果和直接常数指定 K100 是一样的。



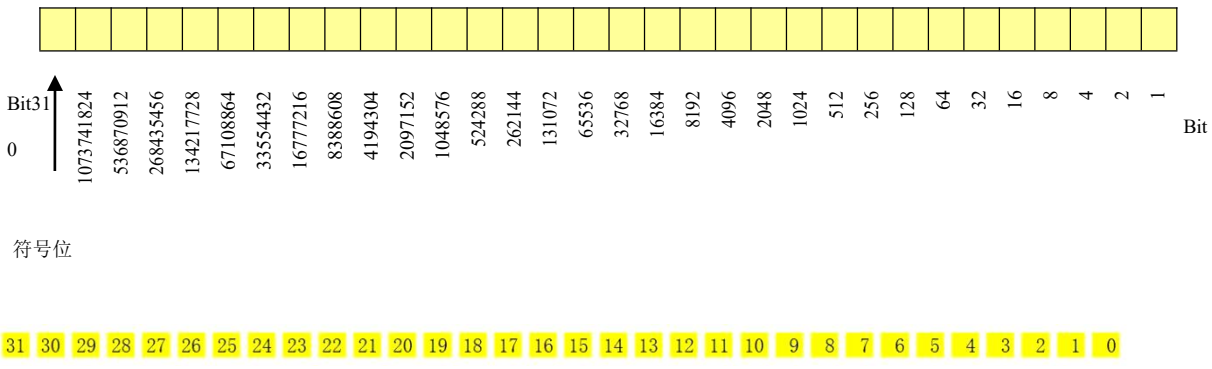
● 32 位间接指定：计数器的设定值为指定数据寄存器的数据（必须是整数且在对应计数器的设定范围）， 由于C200 是 32 位计数器，故设定值必须是 32 位格式，自动占用数据寄存器 D1006、D1005。D1006 为32 位的高字，D1005 为 32 位的低字。在使用 32 位命令写入设定值时，请注意数据寄存器占用情况。

4.7.7 作为数据软元件使用

- 计数器按使用方法分为：根据设定值动作输出触点；将计数值（当前值）作为数值数据使用。
- 计数器的当前值寄存器的结构如下图所示，如果在应用指令的操作数中指定计数器编号时，与数据寄存器相同，可以作为 16 位或 32 位的数据软元件处理。
- 32 位计数器是作为 1 个 32 位数据处理，无法写成两个 16 位数据的形式，故读取、赋值、比较等只能使用32 位指令。
- 16 位计数器，处理数值的范围（10 进制）：0～32767



- 32 位计数器，处理数值的范围（10 进制）：-2147483648～+2147483647



4.8 高速计数器[C]

4.7 章所述计数器为普通计数器，计数响应速度受硬件接口和梯形图扫描周期影响，一般只能接受 50Hz 左右频率变化的信号，在实际应用中常常需要计数速度满足 KHz 级的计数要求，高速计数器就是为满足这类对象所设计的。

4.8.1 高速计数器的类型和编号

高速计数器类型	编号
单向单计数	C235～C242
	C243～C246
双向双计数（AB 相）	C247～C255

4.8.2 高速计数器的外部输入分配

C235～C255 作为32 位高速计数器，可以与高速计数指令(如DHSCS、DHSCR 等)一起配合使用。对应的 C 计数器序号与对应的 X 输入端子的联动关系如下所示：

	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246
X000	U/D				U/D				U			
X001									DIR			
X002		U/D				U/D				U		
X003										DIR		
X004			U/D				U/D				U	
X005											DIR	
X006				U/D				U/D				U
X007												DIR
X010					R				R			
X011					S				S			
X012						R				R		
X013						S				S		
X014							R				R	
X015							S				S	
X016								R				R
X017								S				S

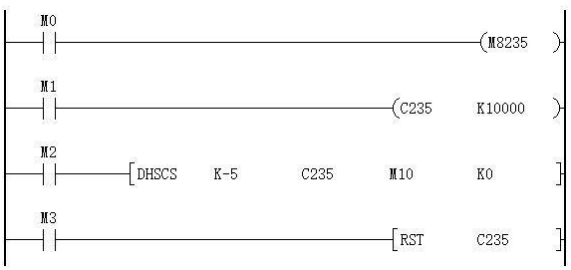
0

	C247	C248	C249	C250	C251	C252	C253	C254
X000	A				A			
X001	B				B			
X002		A				A		
X003		B				B		
X004			A				A	
X005			B				B	
X006				A				A
X007				B				B
X010					R			
X011					S			
X012						R		
X013						S		
X014							R	
X015							S	
X016								R
X017								S

- U/D：计数器类型为增、减计数，可通过对应序号的M82□□进行计数方向的设置
- DIR：增、减计数器的外部方向端口
- R：计数器复位输入端口
- S：计数器启动输入端口
- A：AB 相计数的 A 相接入端口
- B：AB 相计数的 B 相接入端口
- R、S功能使用时，需通过置位特殊M寄存器使对应的外部端子生效。X10-X17对用M8192-M8199

4.8.3 高速计数器的使用

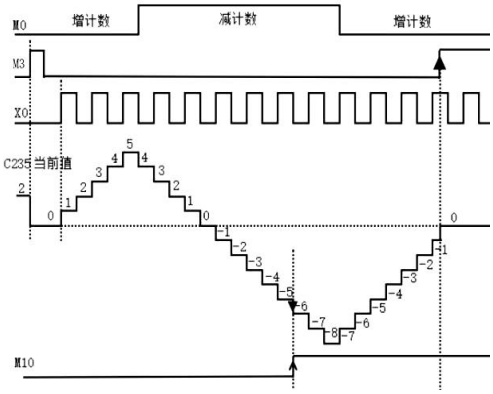
- 要使高速计数器实现计数功能，接入计数信号的外部端子标号和高速计数器编号必须要和高速计数器与外部输入端子的分配表一致，否则无法实现计数功能。
- 要使高速计数器实现计数功能，从外部端子接入的信号形式、信号类型及电平幅值要符合要求（具体要求和接法见硬件说明），否则无法实现计数功能或正确计数。
- 除了上述两项，高速计数器实现计数，还需要梯形图 OUT 指令配合（如[OUT C235]）。
- 高速计数器自己没有触点输出，只能依靠高速计数比较指令（DHSCS、DHSCR、DHST 等）实现比较输出功能。



- M8235 是 C235 计数器的方向设置位。见左图，M0 为 ON 时，C235 为减计数；M0 为 OFF 时，C235 为增计数。
- C235 在 M1 为 ON 时开始，对从 X000 接入的信号进行计数，C235 后带的 K10000，可以是任意整数。
- M2 为 ON，开始对 C235 的计数值同-5 进行比较。

4.8.4 当前值更新时序及比较

- 高速计数器的当前值更新是在外部输入计数信号输入时发生的。
- 执行高速比较输出指令（DHSCS、DHSCR）时，当比较值为正数时，高速计数器的当前值大于等于比较值时输出位被置位或被复位；当比较值为负数时，高速计数器的当前值小于比较值时输出位被置位或被复位。时序见下图



- 见左图，当 C235 的当前值<-5 时，M10 被置位为 ON。
- 见左图，M10 被置位为 ON 后，如果 M3 为ON，只是清零 C235 的当前值，M10 的状态并没有改变，M10 可根据需要在程序中单独复位。

4.8.5 相关软元件

单向单计数的计数器编号与方向控制软元件对应关系。

计数器编号	方向控制元件编号	增计数	减计数
C235	M8235	OFF	ON
C236	M8236		
C237	M8237		
C238	M8238		
C239	M8239		
C240	M8240		
C241	M8241		
C242	M8242		
C243	X1(外部端子)		
C244	X3(外部端子)		
C245	X5(外部端子)		
C246	X7(外部端子)		

4.8.6 关于外部复位输入信号的逻辑变更

支持外部端子复位的高速计数器编号见下表（不支持外部复位输入信号的逻辑变更，固定为复位信号ON有效）：

计数器编号	外部复位端子	复位计数器时复位端子标号状态
C239	X10	ON
C240	X12	
C241	X14	
C242	X16	
C243	X10	
C244	X12	
C245	X14	
C246	X16	
C251	X10	
C252	X12	
C253	X14	
C254	X16	

4.8.7 倍频

- 双向双计数（AB 相）高速计数器的当前值是在外部输入计数信号的 A、B 相信号的上升沿或下降沿时发生的，故计数值是脉冲数的 4 倍。

4.8.8 使用注意事项

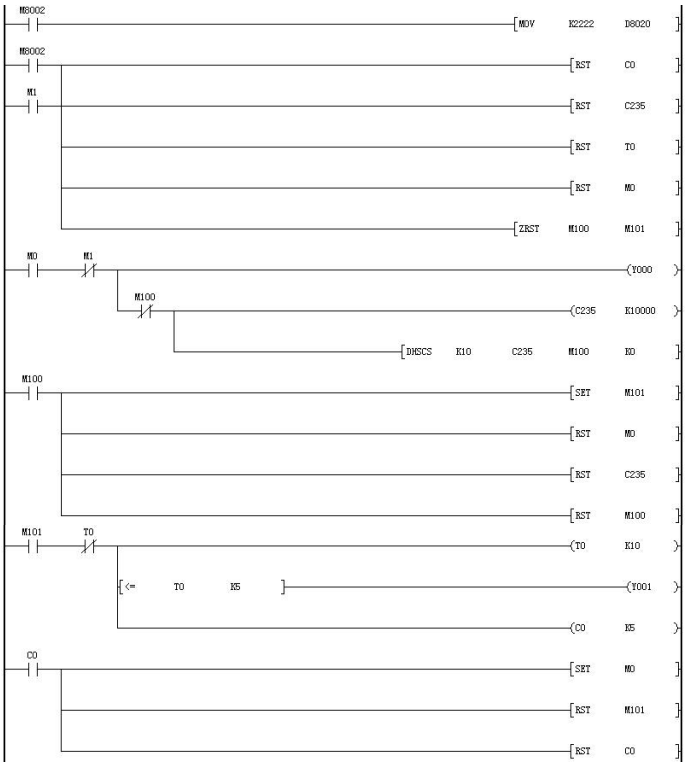
- 要使高速计数器实现计数功能，接入计数信号的外部端子标号和高速计数器编号必须要和高速计数器与外部输入端子的分配表一致，否则无法实现计数功能。
- 要使高速计数器实现计数功能，从外部端子接入的信号形式、信号类型及电平幅值要符合要求（具体要求和接法见硬件说明），否则无法实现计数功能或正确计数。
- 双向双计数（AB 相）高速计数器的当前值是在外部输入计数信号的 A、B 相信号的上升沿或下降沿时发生的，故计数值是脉冲数的 4 倍。
- 高速计数器自身没有输出触点，OUT 指令只是启动高速计数器，不会产生触点输出。
- 高速计数器比较时（DHSCS、DHSCR、DHSZ 等），当比较值为正数时，高速计数器的当前值大于等于比较值时输出位被置位或被复位；当比较值为负数时，高速计数器的当前值小于比较值时输出位被置位或被复位。
- 高速计数器只能使用 RST 指令清零计数值，不能使用批量复位 ZRST 指令或赋值清零。

4.8.9 高速计数器的响应频率

- 单向单计数响应频率<=200KHz
- 双向双计数（AB 相计数）响应频率<=200KHz

4.8.10 程序应用

功能：启动按钮（M0）按下，旋转电机(Y0)运行，当旋转圈数到达设置值时停止旋转电机，然后蜂鸣器(Y1) 以0.5 秒得电 0.5 秒失电的节拍鸣叫 5 次，然后再次启动旋转电机运行，如此往复。按下停止按钮，旋转电机、蜂鸣器全部停止运行。



- 使能掉电保持
- 第一扫描周期、停止按钮按下对相关计数器、定时器、状态继电器做复位处理。
- M0 接通，Y0 输出ON，同时启动 C235 计数和比较，当 C235>=10，M100 被置位为 ON
- M100 被置 ON，Y1 按得电 0.5 秒在失电 0.5 秒的节拍动作 5 次。
- Y1 动作 5 次后，再次启动 Y0、启动 C235 计数和比较，如此往复

4.9 数据寄存器[D]

数据寄存器 D 的作用是存储数据，在数据运算、网络通讯等方面尤显重要。其可作为 16 位或 32 位数据寄存器使用（最高位为正、负号位），作为 32 位寄存器使用时，将连续占用两个相邻地址的 D 寄存器序号。数据寄存器 D 的表现形式多种，可表示整数、二进制浮点数及 BCD 码，作为二进制浮点数使用时将指定为 32 位寄存器使用。

4.9.1 数据寄存器[D]编号

数据寄存器编号	属性	用途	掉电保持
D0～D7999	用户使用区	D7000～D7999为485和SPI扩展占用，尽量避免操作	D200～D3999掉电保持，其余无
D8000～D8255	系统参数区	寄存器数值被系统赋予了特殊的含义，尽量避免修改操作	系统赋值，不做一般掉电保持寄存器使用

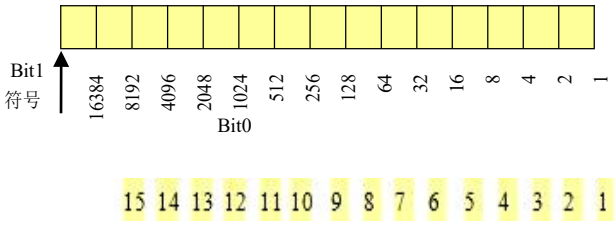
- 掉电保持功能可设置使能或禁止，有掉电保持的前提是使能了掉电保持功能。
- 系统寄存器在一般情况下应尽量避免修改，以免影响系统正常运行。

4.9.2 数据寄存器构造

数据寄存器是存储数值数据的软元件，其种类如下所示。这些寄存器都是 16 位（最高位为正负符号），将两个数据寄存器组合，可存储 32 位（最高位为正负符号）的数值数据。

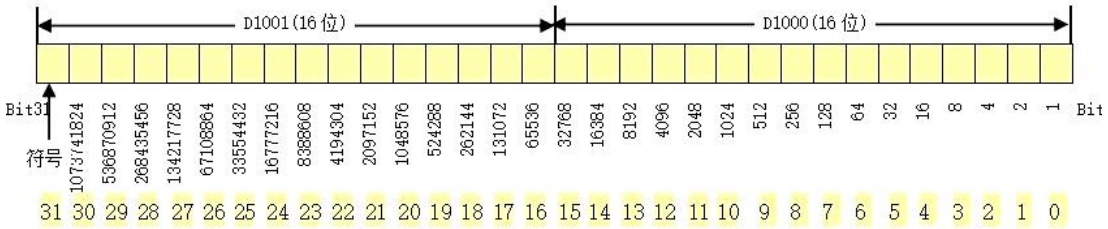
- 处理数值的范围（10 进制）

16 位：-32768～32767



32 位：-2147483648～2147483647

以两个相邻数据寄存器表现 32 位的数据。对于 D1000 作为 32 位寄存器的结构如下（占用 D1001 寄存器）：



数据寄存器 D 在指定 32 位时，如果指定了低位（例如：D1000），则高位为继其之后的编号（例如：D1001） 被自动占有。低位可用奇数或偶数的任意一种软元件编号指定。考虑到外围设备的监视功能，建议低位采用偶数软元件编号。

4.9.3 功能和动作说明

数据寄存器是保存数值数据用的软元件。

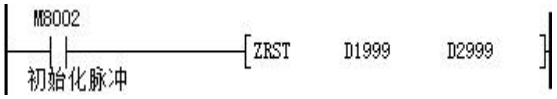
该软元件为 16 位数据(最高位为正负符号)，组合 2 个软元件后可以保存 32 位(最高位为正负符号)的数值数据。

1) 一般用/停电保持用

数据寄存器中的数据一旦被写入，在其他数据未被写入之前都不变化。

在STOP→RUN 时以及停电时，一般用数据寄存器 D 的所有数据都被清除为 0，从 RUN→STOP 时保持运行时的值。

停电保持用数据寄存器，在 RUN/STOP 以及停电时都保持其内容。

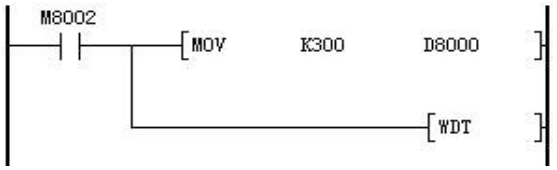


D1999～D2999 的数据清零

2) 特殊用

写入特定目的的数据，预先写入特定的内容的数据寄存器。该内容在每次上电时会被设置为初始值。(一般被清零，带初始值的通过系统写入。)

例如，系统对 D8000 中的 WDT 时间进行初始设定，但如果要更改，使用传送指令 MOV 可以向 D8000 中写入目的时间。

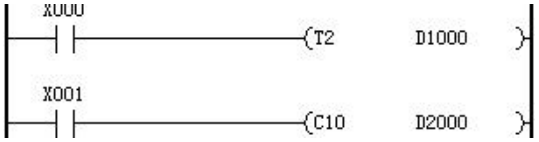


初始演算周期，将监视定时器时间设置为300 毫秒。

数据寄存器可以处理数值数据，用于各种控制。

在本节中，从这些用途中选取了基本指令和应用指令的代表例说明动作。此外，为了能够更有效使用数据寄存器，请阅读后面的应用指令说明。

1) 基本指令中的数据寄存器指定为计数器的设定值。

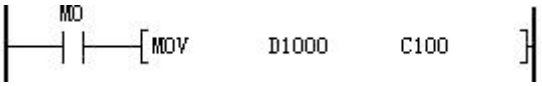


指定的数据寄存器中的内容作为计数器和定时器的设定值进行动作。

2) 应用指令中的数据寄存器

MOV 指令的动作实例

a) 更改计数器的当前值。



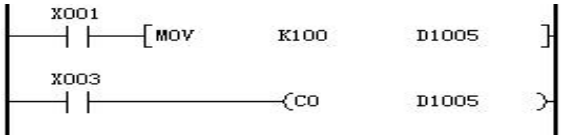
计数器 C100 的当前值改成 D1000 的内容。

b) 将定时器和计数器的当前值读出到数据寄存器中。



计数器 C100 的当前值被传送至 D1200。

c) 数值保存在数据寄存器中。



将 100(10 进制数)传送至D1005。

计数器(C0)的当前值被传送至D1005。

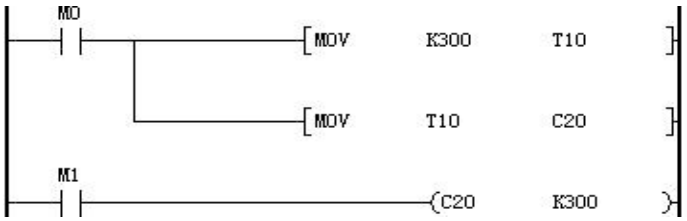


将 D1100 的内容传送至D1200。

3) 将未使用的定时器及计数器作为数据寄存器使用

MOV 指令的动作实例

程序中不使用的定时器和计数器可作为 16 位或是 32 位的数值保存软元件(数据寄存器)使用。



将300 (10 进制数)传送至 T10。

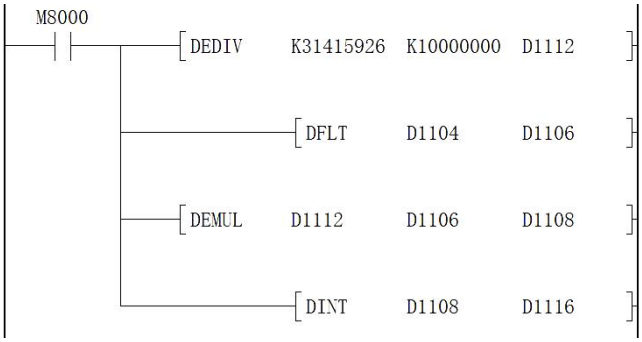
将 T10 的内容传送至 C20 的当前值寄存器中。

此时，T10 不起到定时器的作用，而是作为数据寄存器工作。

作为 32 位使用时，与数据寄存器相同，用 2 个16 位软元件(例如: C1、C0 等)，显示 32 位数值。此外，1 个32 位计数器(例如: C200 等)能够处理 32 位的数值。

4.9.4 程序应用

计算圆的周长，先将圆周率及直径转换为浮点数，然后将浮点数的运算结果转换为整数供后续程序使用。



- 先将将圆周率转成浮点数，由于圆周率是以整数形式赋值的，又需要将其处理成小数形式。
- 将直径转换成浮点数。
- 计算周长，为浮点数。
- 将周长转换为整数。

4.10 扩展寄存器[R]

对于复杂系统工艺控制要求，其编程可能需要更多的用户寄存器作为中间单元或缓存区使用。除了传统的数据寄存器 D 以外，提供扩展寄存器 R 软元件供用户使用，使之能更加方便的构建梯形图程序。

扩展寄存器的使用方法与数据寄存器相同，可进行数值数据的读入、写出。被作为 32 位寄存器使用时，也将连续占用两个相邻地址的 R 寄存器序号。

4.10.1 扩展寄存器的编号

- 扩展寄存器 R 编号为：R0～R7999。

数据寄存器编号	属性	用途	掉电保持
R0～R59	系统参数区	保留	系统根据设置需要赋值，不做一般掉电保持寄存器使用
R60～R83		输出轴脉冲显示区	
R84～R99		保留	
R100～R259		软件虚拟输入轴参数区	
R260～R299		保留	
R300～R899		软件虚拟输出轴参数区	
R900～R999		保留	
R1000～R1999		保留	
R2000~R2049		FLASH参数区	R2000~R7999 掉电保持
R2050~R2099		保留	
R2100~R2259		KLink总线资源配置参数区	
R2300~R2323		以太网通信参数区	
R2324~R7999		保留	

具体作用如下：

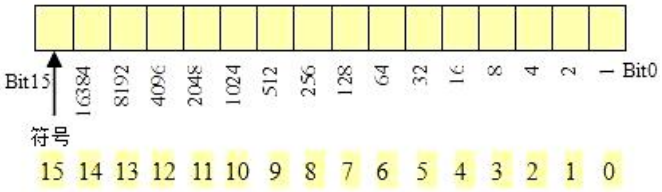
- R0～R599 为系统参数区，一般不允许做中间变量和工艺变量使用。其分配如下：
- R0～R79 为系统保留区。
- R80～R97 为输出轴脉冲显示区域。
- R98～R199 为系统保留区。
- R200～R299 为软件虚拟输入轴参数区域。
- R300～R599 为软件虚拟输出轴参数区域。
- R600～R999 为系统保留区。
- R2000～R2049 为FLASH参数区。
- R2100～R2259 为KLink总线资源配置参数区。
- R2300~R2323为以太网通信参数区
- 掉电保持功能可设置使能或禁止，有掉电保持的前提是使能了掉电保持功能。

4.10.2 扩展寄存器的构造

扩展寄存器是存储数值数据的软元件，其种类如下所示。这些寄存器都是 16 位（最高位为正负符号），将两个数据寄存器组合，可存储 32 位（最高位为正负符号）的数值数据。

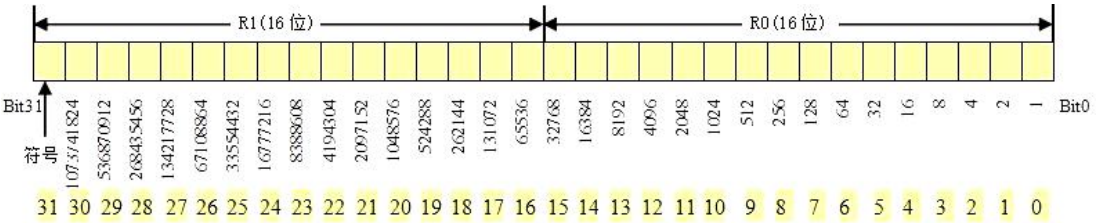
- 处理数值的范围（10 进制）

16 位：-32768～32767



32 位：-2147483648～2147483647

以两个相邻扩展寄存器表现 32 位的数据。对于将 R0 作为 32 位寄存器的结构如下（占用 R1 寄存器）：



扩展寄存器R 在指定 32 位时，指定低位侧(例如: R0)后高位侧会被紧接(例如: R1)的号码自动占用。低位侧既可指定奇数，也可指定偶数的软元件编号，但是考虑到人机界面、显示模块、编程工具的监控功能等，建议低位侧使用偶数的软元件编号。

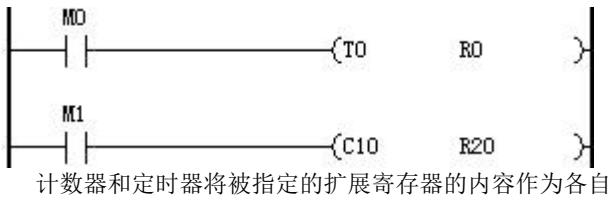
4.10.3 功能和动作说明

- R 寄存器无固定初始值，在分配和使用之前应做好其数值的初始化，以免错误的初值导致梯形图工作异常。
- R 寄存器为特有扩展寄存器，在与第三方设备进行连接时(如 HMI)，不能使用编程口协议进行连接，但可以使用 MODBUS 协议进行 R 寄存器的通信。

扩展寄存器 R 可以处理各种数值数据，利用它进行各种运算和控制，功能和动作参见寄存器[D]。

1) 基本指令中的扩展寄存器

- **指定为定时器和计数器的设定值**。



计数器和定时器将被指定的扩展寄存器的内容作为各自的设定值而动作。

2) 应用指令中的扩展寄存器

MOV 指令的动作实例

- **更改计数器的当前值。**



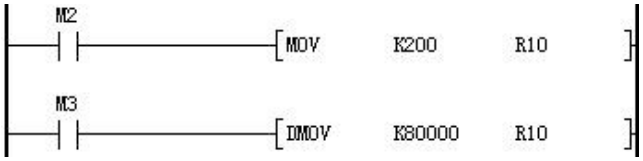
3) 计数器(C2)的当前值更改为 R5 的内容。

- **将计数器的当前值读入扩展寄存器中。**



计数器(C10)的当前值传送给 R4。

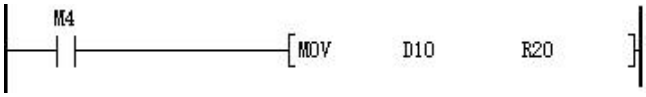
- **将数值保存到扩展寄存器中。**



200(10 进制数)传送到 R10。

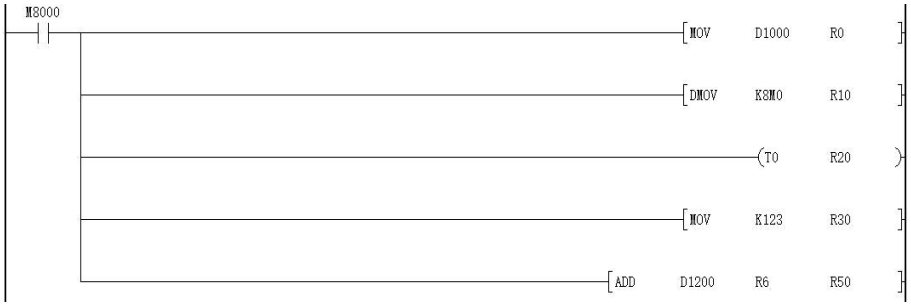
80,000(10 进制数)传送到 R10(R11)。由于数据超出 32,767，所以使用 32 位 指令(32 位运算)。指定了扩展寄存器的低位侧 (R10)后，高位侧 (R11)会被自动占用。

- **将数据寄存器的内容传送至扩展寄存器中。**



D10 的内容传送给 R20。

4.10.4 程序应用



4.11 变址寄存器[V、Z]

变址寄存器 V 与 Z 两者功能完全一样。

同普通的数据寄存器一样，可进行数值数据的读入、写出。

在应用指令的操作数中，变址寄存器还可以同其他的软元件编号或数值组合使用，效果为在程序中改变元件编号或数值内容，是一个特殊寄存器。

注意 LD，AND，OUT 等PLC 的基本顺控指令或步进梯形图指令的软元件编号不能同变址寄存器组合使用。变址寄存器 V 和 Z 均可做 16 位或 32 位寄存器使用，根据其调用指令的位数决定它的数据类型。当作为 32 位寄存器使用时，将连续占用 2 个相邻地址的变址寄存器。例如 V2(L)，占用 V2/V3 两个 16 位变址寄存器。

4.11.1 变址寄存器的编号

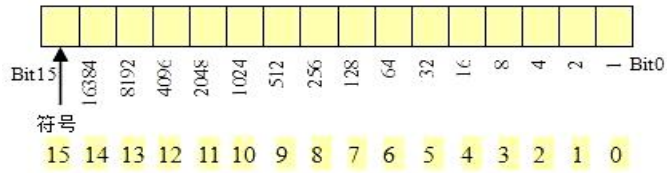
变址寄存器[V、Z]的编号如下所示。(编号以 10 进制数分配)

变址寄存器类别	变址寄存器编号	掉电保持
V	V0～V7	有
Z	Z0～Z7	

4.11.2 变址寄存器的构造

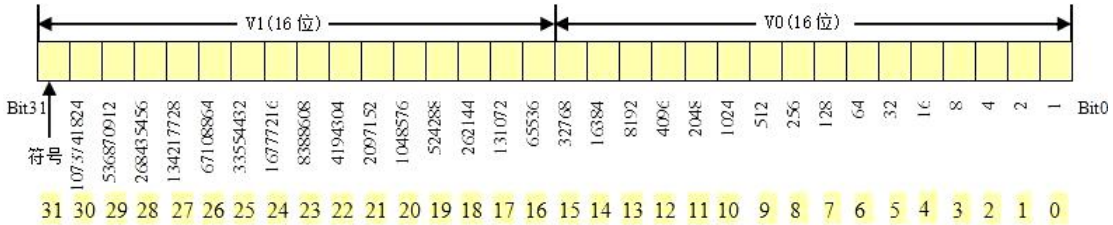
- 处理数值的范围（10 进制）

16 位：-32768～32767



32 位：-2147483648～2147483647

以两个相邻变址寄存器表现 32 位的数据。对于将 V0 作为 32 位寄存器的结构如下（占用V1 寄存器）：



变址寄存器 V、Z 在指定 32 位时，如果指定了低位，则高位为继其之后的编号被自动占用（例如：指定了低位V0，则V1 自动被占用）。低位可用奇数或偶数的任意一种软元件编号指定，建议低位采用偶数编号。

4.11.3 功能和动作说明

可能变址的软元件，其变址内容为：

- 10 进制数的软元件、数值：M，S，T，C，D，KnM，KnS，P，K

例如，V0=K5，执行 D1020V0 时，被执行的软元件的编号为 D1025（D1020+5）。此外，也可变更常数值。例如，指定 K30V0 时，被执行的是作为 10 进制数的数值 K35（K30+5）。

- 8 进制数的软元件：X，Y，KnX，KnY

例如，Z1=K8，执行 X000Z1 时，被执行的软元件编号为X010（X0+8：8 进制数加法）。对于软元件编号为8 进制数的软元件变址来说，V，Z 的内容要被换算成 8 进制数，然后做加法运算。因此，假定 Z1=K10， X000Z1 被指定为 X012，请务必注意此数不是X010。

- 16 进制的数值：H

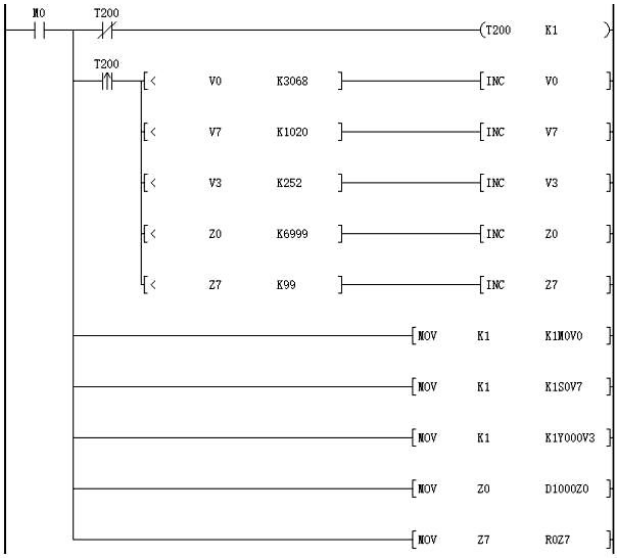
例如：以V5=K30 指定常数 H30V5，则被认为是 H4E（30H+K30）。此外，以 V5=H30 指定常数 H30V5，则被认为是 H60（30H+30H）。

注意：

1) 在使用变址修饰软元件的时候，最后得到的修正编号不能超过软元件的编号范围。如 C0V0，若 V0=-1 则系统将会出错。
2) 常数变址为环形的加减法运算：
   执行 16 位指令时，K32767V0，V0=1，结果为 K-32768；K-32768V0，V0=-1，结果为 K32767。执行 32 位指令时，K32767V0，V0=1，结果为 K32768；K-32768V0，V0=-1，结果为 K32767。
   执行 32 位指令时自动占用连续的两个变址寄存器 V0、V1，V1 为高 16 位，V0 为低 16 位。
3) 执行 32 位指令时，对应的操作数地址使用变址修饰，仍然以 16 位软元件地址进行变址。比如 D1000V0， 当V0 为1 时，该操作数指向 D1001(D1001～D1002 这2 个16 位软元件的组合成 32 位数据，D1001 为低16 位，D1002 为高 16 位)。
4) 当软元件本身属性为 32 位时(比如 C200)，变址寄存器的值必须为双数，如果为单数，它并不会指向下一个软元件编号，故避免使用变址值为单数修饰本身属性为 32 位的软元件。

4.11.4 程序应用

利用变址功能对不同资源批量赋值



- M0 为 ON 时，变址寄存器每10mS 加 1， 到达设置最大值停止自加。
- 将以 M0 为起始的 M 继电器置位， 直至M3068。
- 将以 S0 为起始的 S 继电器置位， 直至S1021。
- 将以Y0 为起始的Y 继电器置位，直至 Y374。
- 将Z0 的当前值传送给D(1000+Z0)的D 寄存器。
- 将Z7 的当前值传送给R(0+Z7)的R 寄存器。

4.12 指针[P]、[I]

标记指针 P 用于用户调用子程序或程序跳转。

标记指针 I 用于中断程序标号，其对应的中断类型有定时中断，外部事件中断，系统软中断及高级中断。

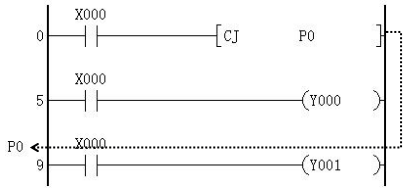
4.12.1 指针的编号

指针编号	指针类型	点数	说明
P0～P127	程序分支	128	用于子程序调用或跳转
I6△△～I8△△	定时中断	3	定时时间作为中断源，时间到进中断
I010	系统软中断	6	1mS 定时系统软中断，无需设置申请位
I020～I050			由中断申请位 M8043～M8047 申请
I000	外部事件中断	8	X10 下降沿中断
I001			X10 上升沿中断
I100			X11 下降沿中断
I101			X11上升沿中断
I200			X12下降沿中断
I201			X12 上升沿中断
I300			X13 下降沿中断
I301			X13上升沿中断
I900	高级中断	8	由系统 1mS 定时申请
I901～I907			由进程申请位 M8089～M8095 申请

4.12.2 分支用指针的功能和动作实例

- 在控制程序流程时，作为分支标志或子程序名称，在 CJ、CALL 指令中使用。
- P 作指令操作数时可变址修饰。
- P 标识程序位置时（紧靠左母线）不能变址修饰。
- 作为子程序调用时，子程序结束时必须使用 SRET 指令返回到主程序，而且子程序必须处于 FEND 与 END 语句之间。

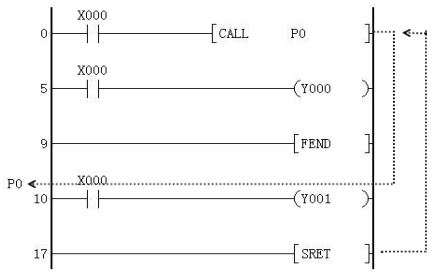
CJ 条件跳转



- CJ 指令操作标记指针 P。

如果 X000 为 ON，通过 CJ 指令跳转到指定的 P0 标号位置，执行随后的程序，Y1 输出为 ON 而 Y0 为 OFF。

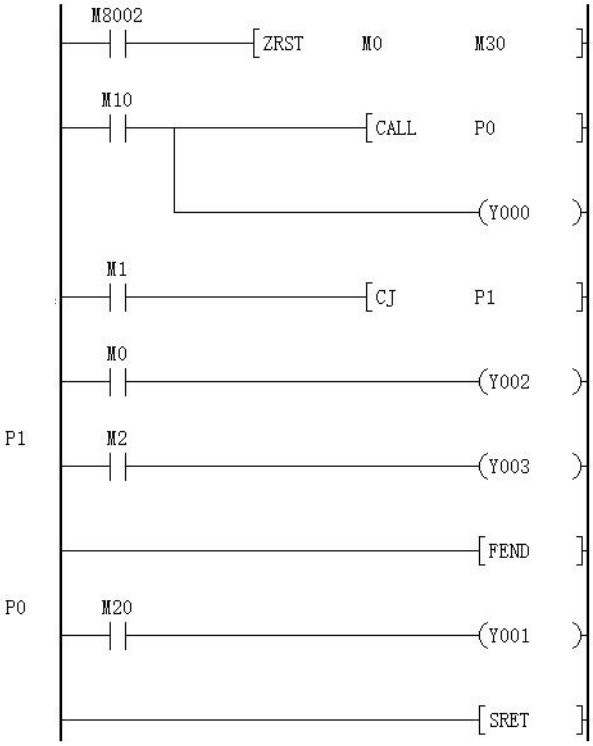
CALL 子程序调用



- CALL 指令操作 P 软元件。

如果 X000 为 ON，通过 CALL 指令跳转到指定的P0 标号位置，执行随后的程序，Y1 输出为 ON， 当执行到 SRET 指令，程序将返回到 CALL 指令位置，并执行下一条指令及随后的程序，Y0 输出为ON。

4.12.3 程序应用



- 置M10 为ON，调用子程序 P0，Y0 输出为ON。

- 置M0 为ON，M1 为OFF，Y2 输出为 ON。

- 置M2 为ON，Y3 输出为 ON。
  Y2 输出为 ON 后，置 M1 为ON，此时改变M0 的状态不会影响 Y2 的输出状态。
  Y1 输出为 ON 后，置 M10 为OFF，此时改变M20 的状态不会影响 Y1 的输出状态。

- 置M20 为ON，Y1 输出为 ON，否则 Y1 为OFF。

所以在使用跳转、子程序调用时要注意程序流 向，避免产生和设计目标不一致的结果。

4.12.4 指针 I 的功能和动作说明

中断用指针 I 有以下四种常用类型，与应用指令IRET（中断返回），EI（开中断）和 DI（关中断）一起使用，而且中断程序必须处于 FEND 与END 语句之间。

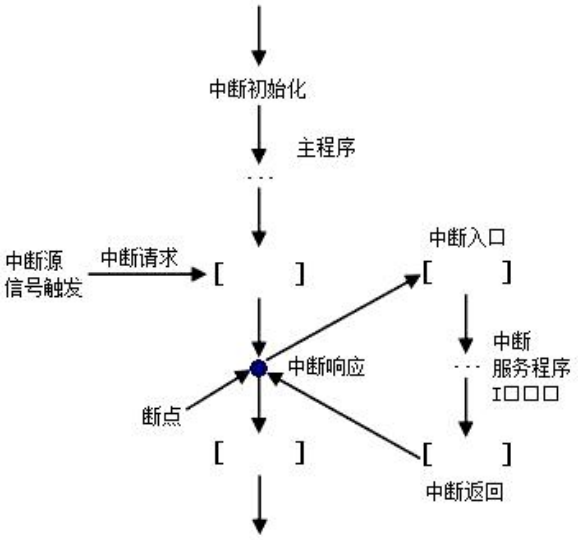
- 外部事件中断：接受来自特定的输入编号的输入信号，而不受 PLC 的扫描周期的影响。触发该输入信号，

执行中断子程序。通过输入中断可处理比扫描周期更短的信号，因而可在顺控过程中作为必要的优先处理或短时脉冲处理控制中使用。

- 定时器中断：在指定的中断循环时间（1ms～100ms）执行中断子程序。在需要有别于 PLC 的运算周期的循环中断处理控制中使用。
- 系统软中断：依据相应中断申请位状态来执行中断程序。
- 高级中断：用于实现高速并发任务的处理以及插补算法的实现。

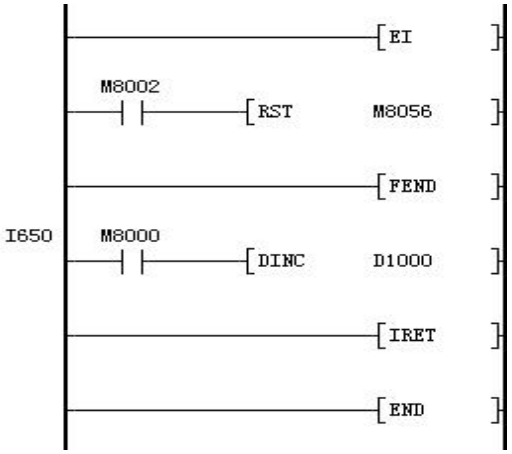
中断的具体应用参照说明书中断系统章节。

中断就是指某事件的发生引起 CPU 暂停当前程序的运行，转入对所发生事件的处理，处理结束又回到程序被打断处接着执行这样的一个过程。



- 梯形图是周期扫描型的处理机制，为了能够响应实时性比较强的任务，引入了梯形图中断的概念。
- 可利用在梯形图母线左端添加I 标号的方式指定一个梯形图中断。
- 系统每扫描完一条指令都会扫描一次所有的中断申请。若中断源满足触发条件且申请有效，则执行相应的中断服务程序。
- 中断执行的流程如左图所示。

4.12.5 程序应用



- 执行 EI 指令后，允许定时器中断。
- 清除 I6□□中断标识禁止位。
- FEND 表示主程序结束，中断程序必须位于FEND 指令之后。
- I650 表示使用的中断标号为 I6□□的定时中断，每隔 50ms 执行中断子程序，用IRET 指令返回主程序。

5. 应用指令构造

在本章中，介绍应用指令的一般构成形式等。

5.1 指令格式

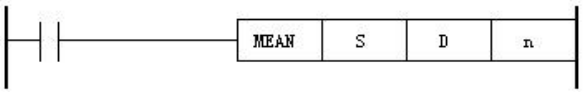
MC系列PLC的应用指令由指令符号（助记符）和操作数组成。

5.1.1 指令符号

大部分应用指令的指令符号都能表示其内容。如 CMP 取自英文字母比较的简写，该指令为比较指令。应用指令都占程序步，不同的应用指令，操作数不同，所占程序步也不同，具体指令讲解中都说明了指令所占的步数。

5.1.2 指令操作数

不同的应用指令，操作数个数也不相同，有的指令没有任何操作数。指令执行后，内容不会发生变化的操作数称为源操作数，用 **S** 符号表示。指令执行后，内容会发生变化的操作数称为目标操作数，用 **D** 符号表示。不符合源操作数也不符合目标操作数的操作数用 n1、n2 表示。如：



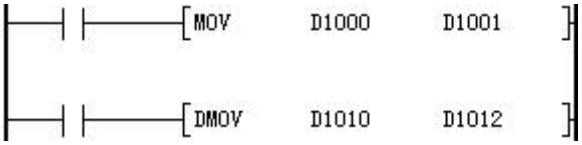
S 为源操作数，D 为目标操作数，n 为软元件长度，将n 点的源数据S 的平均值存入目标地址D 内。

5.2 指令形式和执行形式

根据应用指令处理的数值大小，可以分为 16 位指令和 32 位指令。根据指令的执行形式不同，分为连续执行型和脉冲执行型两种类型。

1） 16 位/32 位指令

处理数值的应用指令中，根据数值数据的位长可分为 16 位和 32 位。



将D1000 的内容传送到 D1001。

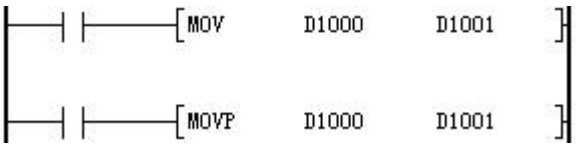
将D1011、D1010 的内容传送到（D1013、1012）

32 位指令在 16 位指令前添加[D]符号来区分表示（如MOV 与DMOV）。16 位指令每个操作数占用 1 个字，32 位指令每个操作数占用 2 个字。但操作数为位软元件时，如 KnX 等，只占 n 位软元件长度。16 位指令中不能写 n=4 以上的位软元件，如 K8Y0。32 位计数器（C200-C255）可以作为 16 位指令的操作数使用， 但只有低 16 位有效。软元件可以使用偶数或是奇数。

2） 连续执行/脉冲执行型

应用指令根据执行一次还是连续执行可分为连续执行型和脉冲执行型。

脉冲执行型指令在连续执行型指令后添加[P]符号来区分表示（如MOV 与MOVP）。连续执行型指令在触点导通的情况下，每个扫描周期都会执行，脉冲执行型指令在触点导通的情况下，只执行一次。



将D1000 的内容传送到 D1001。

将D1000 的内容传送到 D1001。

上图中，MOVP 只执行一次，而 MOV 每个扫描周期都执行。不需要一直执行的指令建议使用脉冲执行型指令。

部分应用指令可能只有 16 位指令，也可能只有 32 位指令，也可能没有脉冲执行型指令。

5.3 操作数数值类型

在 MC系列PLC中，根据用途和目的不同，有 5 种数值可供使用，具体如下。

5.3.1 数值种类

1) **10 进制数（DEC）**

10 进制数是以 0～9 为基数表示的，辅助继电器 M、状态继电器 S、定时器 T、计数器 C、数据寄存器 D 等均是用十进制数表示编号的。可以作为定时器和计数器的设定值，也可以作为应用指令操作数中的数值指定（K 常数）。

2) 16 进制数（HEX）

16 进制数是以 0～9 A B C D E F 为基数表示的，可以作为定时器和计数器的设定值，也可以作为应用指令操作数中的数值指定（H 常数）。

3) 2 进制数（BIN）

2 进制数是以 0～1 为基数表示的。梯形图中设定的 10 进制数和 16 进制数，在MC系列PLC的内部都是以 2 进制数进行处理的。在梯形图监控画面可以选择 2 进制数显示。负数是以 2 的补码来表示的。

4) 8 进制数（OCT）

8 进制数是以 0～7 为基数表示的。控制器中的输入继电器、输出继电器的软元件编号是以 8 进制数表示的。在8 进制数中，不存在 8 和9，编号按 X0～X7，X10～X17...上升排列。

5) BCD（BCD）

BCD 就是 10 进制数的各位上的 0～9 的数值以四位 BIN 来表示的形式。一般应用于BCD 输出型的数字式开关和 7 段码显示器控制。

6) 实数（浮点数）

MC系列PLC具有高精度运算的浮点数运算功能。采用 2 进制浮点数进行浮点运算，10 进制浮点数进行监控。

5.3.2 数值间转换

MC系列PLC中各种数值可以相互转换，转换关系如下表。

10 进制数	8 进制数	16 进制数	2 进制数		BCD	
0	0	00	0000	0000	0000	0000
1	1	01	0000	0001	0000	0001
2	2	02	0000	0010	0000	0010
3	3	03	0000	0011	0000	0011
4	4	04	0000	0100	0000	0100
5	5	05	0000	0101	0000	0101
6	6	06	0000	0110	0000	0110
7	7	07	0000	0111	0000	0111
8	10	08	0000	1000	0000	1000
9	11	09	0000	1001	0000	1001
10	12	0A	0000	1010	0001	0000
11	13	0B	0000	1011	0001	0001
12	14	0C	0000	1100	0001	0010
13	15	0D	0000	1101	0001	0011
14	16	0E	0000	1110	0001	0100
15	17	0F	0000	1111	0001	0101
16	20	10	0001	0000	0001	0110
…	…	…	…	…	…	…

5.4 操作数对象软元件

MC系列PLC应用指令中的操作数可以设定为不同的软元件，可设定常数、位软元件、位软元件组合、字软元件等。X、Y、M、S 处理 ON/OFF 信息的软元件为位软元件，T、C、D、R 等处理数值的软元件为字软元件。位软元件组合使用也可以处理数值，以位数 Kn 和起始软元件编号组合来表示，如K4X0。

5.4.1 常数 K，H，E 指定

应用指令中处理常数用常数 K（10 进制数）、常数 H（16 进制数）或 E（浮点数）。

1) 常数 K（10 进制数）

K 是十进制数的符号，应用时在数值前加 K，就表示十进制数，主要用于设定定时器和计数器的值，或应用指令的操作数的值（例如：K1234）。16 位10 进制常数范围为 K-32768～K32767，32 位10 进制常数范围为K-2147483648～K2147483647。

2) 常数 H（16 进制数）

H 是十六进制数的符号，应用时在数值前加 H，就表示十六进制数，主要用于设定定时器和计数器的值，或应用指令的操作数的值（例如：H1234）。16 位16 进制常数范围为 H0～HFFFF，32 位16 进制常数范围为H0～HFFFFFFFF。

数值范围在 0～9范围内时，16 进制数和BCD 代码相同，16 进制数可以指定BCD 数据，BCD 数值范围为H0～

H9999，H0～H99999999。

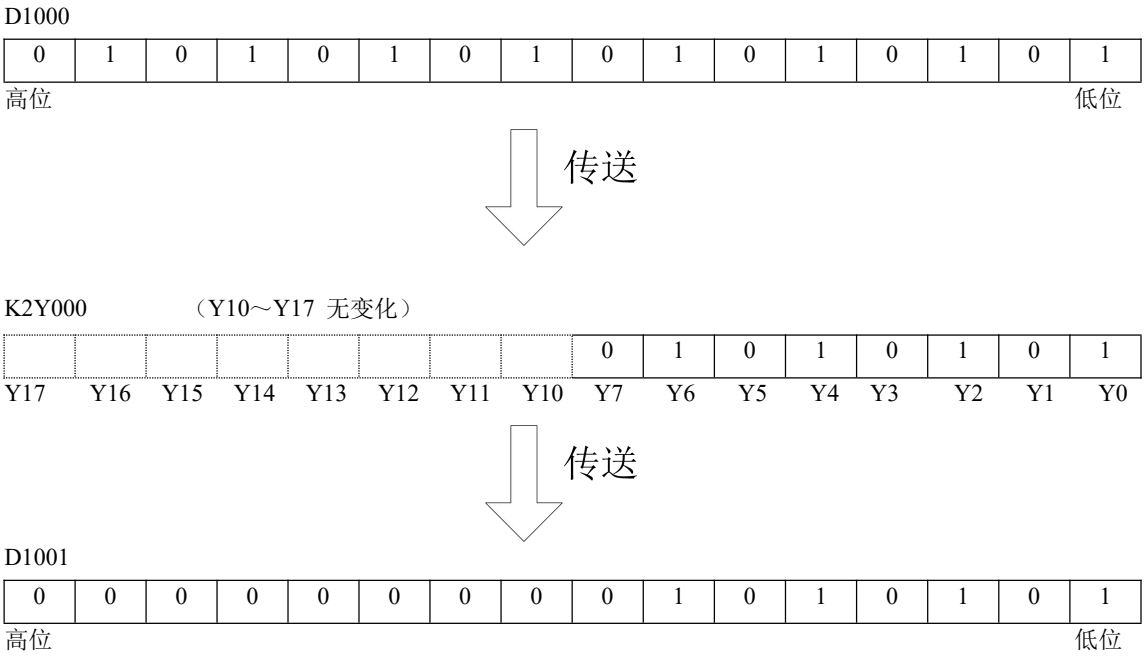
3) 常数 E（实数）

E 是浮点实数的符号，应用时在数值前加 E，就表示浮点实数，主要用于浮点数指令中。浮点实数的范围为 H0～HFFFF。浮点实数有普通表示和指数表示两种表示。

5.4.2 位软元件指定

应用指令中的操作数可以是位软元件，位软元件只能处理 ON/OFF 信息，需要处理数值可以将位软元件组合。位软元件组合是以位数 Kn 和起始软元件的编号组合来表示。位数为 4 位单位的 K1～K4（16 位数据）、K5～K8（32 位数据）。例如，K2Y0， 是 Y0～Y7 的组合，是 2 位数的数据。下面举例说明位组合的传送。

传送结果如下：



向K1Y0～K3Y0 传送 16 位数据，数据长度不足的高位 Y 不会被传送，32 位数据情况相同。将 K1Y0～K3Y0 传送至数据寄存器 D 等时，不足的高位补 0，32 位数据情况相同。

没有特别的限制时，被指定的起始位软元件的编号可以是任意的，为了避免混乱，X，Y 建议起始位软元件的最低位为 0（0，10，20...），M，S 建议起始位软元件为 8 的倍数（0，8，16...）。

位软元件组合时，不能超过位软元件的范围，例如 K8Y320，不能超过Y377，超过则控制器报错。

5.4.3 字软元件指定

应用指令中的操作数可以是字软元件。字软元件有数据寄存器 D，扩展寄存器 R，定时器 T 和计数器 C， 变址寄存器 V，Z。

位软元件组合也可以作为字软元件使用。例如 K4M0 可以作为 16 位数据使用，在32 位指令中使用了K4M0， 高 16 位数据被视为 0，需要 32 位数据时可使用 K8M0。

C200～C235 可作为 16 位数据使用，低 16 位数据有效。

字软元件组合成 32 位数据时，以连续两个字软元件为 1 个 32 位数据，低位在前，高位在后，例如（D1001，D1000），D1000，D1001 组成 32 位数据，D1000 为低位，D1001 为高位。

5.4.4 变址修饰

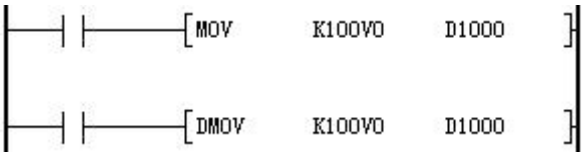
应用指令的操作数中，变址寄存器可以同其他的软元件编号或数值组合使用，效果为改变元件编号或数值内容，称为变址修饰。变址修饰是通过变址寄存器 V0～V7，Z0～Z7 来实现，有关变址寄存器的介绍，已在4.11 中详细介绍。

LD、LDI、AND、ANI、OR、ORI、OUT、SET、RST、PLS、PLF 等基本顺控指令和步进梯形图指令 STL不能使用变址修饰。不能对 V，Z 本身变址修饰。位软元件组合中 Kn 的 n 不能使用变址修饰。

大部分应用指令的操作数支持变址修饰，小部分不支持变址修饰，具体参考每条指令的对象软元件表。

1) 常数变址修饰

应用指令中的常数操作数允许变址修饰。常数变址修饰为环形的，K32767V0，V0=1，K 实际为-32768。K-32768V0，V0=-1，K 实际为 32767。



V0=-1，V1=0，上述两条指令导通时执行结果 D1000 是不相同的，第一条D1000 为99，第二条（D1001，D1000）为65635。16 位常数变址修饰以 16 位常数进行变址，变址修饰占用 V0 寄存器，32 位常数变址修饰以 32 位常数进行变址，变址修饰占用 V0（L），V1（H）两个寄存器，因此第二条指令 100 加上 65535。

2) 位软元件变址修饰

应用指令中的位软元件操作数允许变址修饰。例如用变址寄存器 Z0 对 CMP 的比较结果进行修饰。

Z0=2，CMP 指令导通时，Y000Z0 为Y002，比较结果输出至 Y2～Y4 中。位软元件组合操作数允许变址修饰。



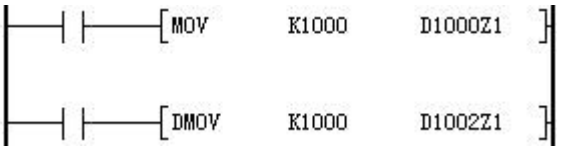
V0=20，MOV 指令导通时，K4M0V0 为M20～M27，将其组合值传送给 D1000。



8 进制数的软元件编号，X，Y，KnX，KnY，变址修饰时 V，Z 的内容要被换算成 8 进制数，然后做加法运算。因此，假定 Z1=K10，X000Z1 被指定为 X012，请务必注意此数不是 X010。

3) 字软元件变址修饰

应用指令中的字软元件操作数允许变址修饰。



16 位指令和 32 位指令的字软元件操作数变址修饰均是以 16 位软元件地址进行变址的。如上例中，Z1=1，MOV 和DMOV 指令导通时，1000 的值分别赋给 D1001 和（D1004，D1003）。变址寄存器只占用 16 位Z1。

当软元件本身属性为 32 位时(比如 C200～C255)，不建议使用变址修饰，因为它并不会指向下一个软元件序号，而是指向本软元件的高 16 位。

在使用变址修饰软元件时，最后指向的软元件编号不能超过软元件的编号范围。如 C0V0，若 V0=-1 则控制器报错。

5.5 指令说明

应用指令的指令说明里讲述具体指令的功能和动作说明，注意要点，列举实例讲解应用指令，还有涉及到一些标志位的使用，指令使用次数限制等。

6. 编程前须知

在本章中，说明了编写程序的一些方法及注意事项。

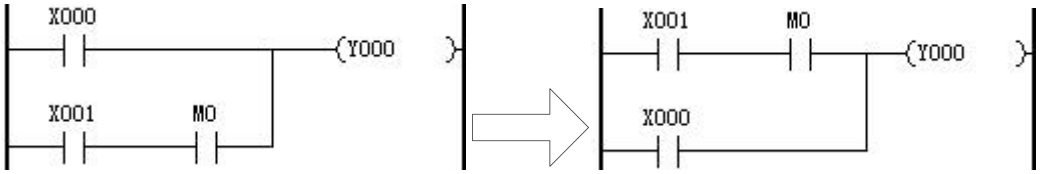
6.1 编程方面基本注意事项

说明了编程时需要注意的内容。

6.1.1 程序步骤及执行顺序

1) 触点的结构和步数

即使是执行相同动作的顺控梯形图，触点的构成方法不同，程序步数也不同。编程时注意可以简化程序和节省程序步数。



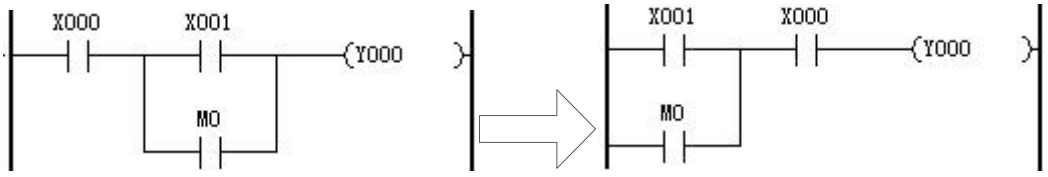
■ 串联触点较多的梯形图写在上方比较好

对应指令表：

0	LD	X000
2	LD	X001
4	AND	M0
6	ORB	
7	OUT	Y000

0	LD	X001
2	AND	M0
4	OR	X000
6	OUT	Y000

串联触点较多的梯形图写在上方程序步数更少。



■ 并联触点多的梯形图写在左方比较好。

对应指令表：

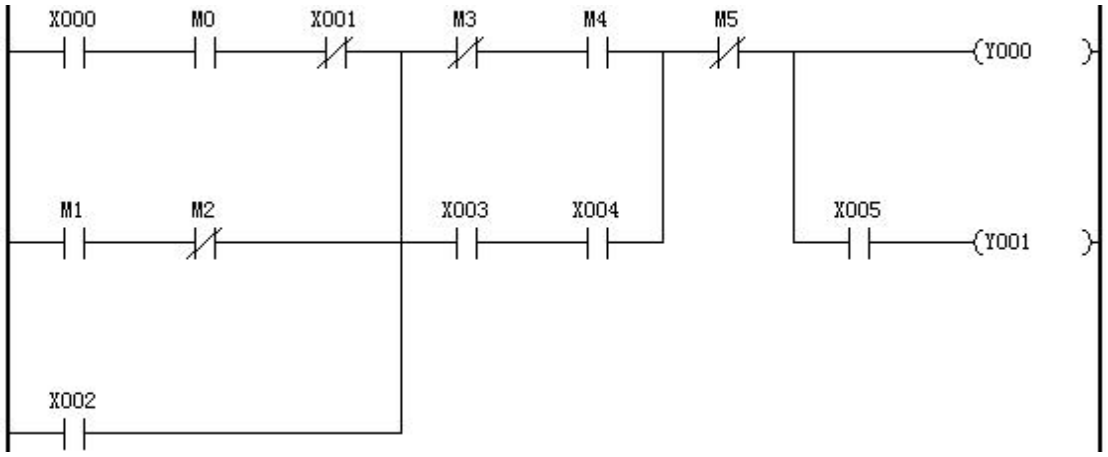
0	LD	X000
2	LD	X001
4	OR	M0
6	ANB	
7	OUT	Y000

0	LD	X001
2	OR	M0
4	AND	X000
6	OUT	Y0

并联触点较多的梯形图写在左方程序步数更少。

2) 程序执行顺序

顺控程序是按照【自上而下】到【自左而右】的顺序处理程序。顺控指令表也按照此顺序编码。



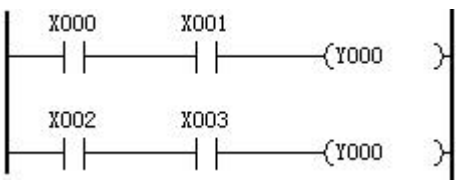
上图梯形图按下列标号顺序执行：

对应的指令表如下，也是按照【自上而下】到【自左而右】的顺序编码的。

```
0    LD      X000
2    AND     M0
4    ANI     X001
6    LD      M1
8    ANI     M2
10   ORB
11   OR      X002
13   LDI     M3
15   AND     M4
17   LD      X003
19   AND     X004
21   ORB
22   ANB
23   ANI     M5
25   OUT     Y000
27   AND     X005
29   OUT     Y001
```

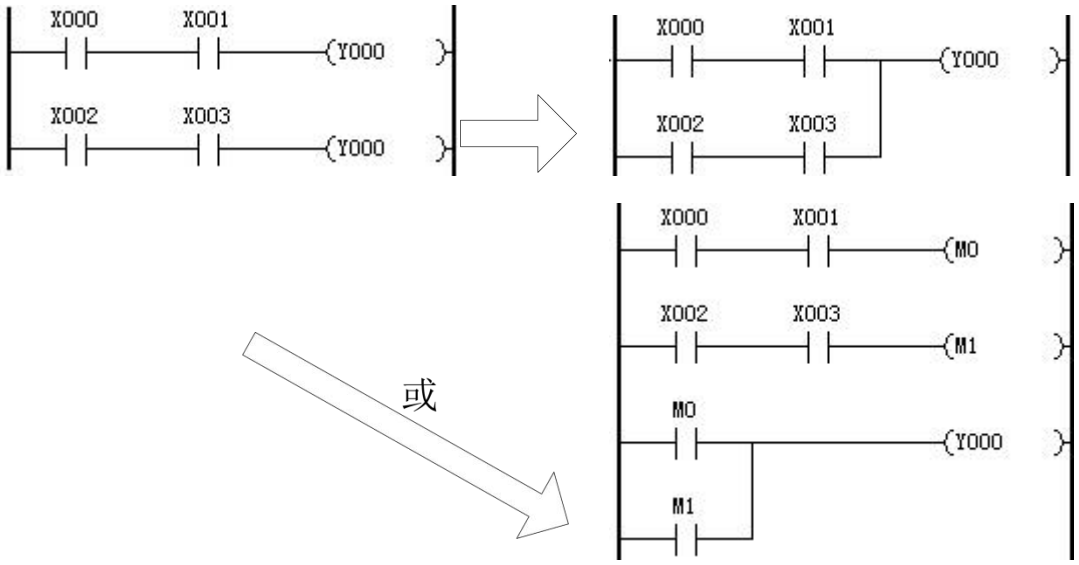
6.1.2 双线圈输出

同一个顺控程序中同一线圈有两个输出，称为双线圈输出。如下所示双线圈输出。



双线圈输出时，后面的线圈动作优先执行。

双线圈输出在程序方面并不违反输出规则，但是会使上述动作变得复杂，因此请按以下示例改变程序。



用上述两种方法可以避免双线圈输出，还有其它编程方法，如采用SET、RST 指令、跳转指令、流程指令， 不同状态控制同一输出线圈。使用步进梯形图指令（STL）时，主程序中存在的输出线圈，在状态中也存在时，也视为双线圈输出。

6.2 输入输出响应延时

MC系列PLC中重复执行下述输入输出处理过程。除了输入元器件和输出元器件的硬件延时，还受扫描周期影响，输入输出元器件响应会有延迟。需要获取输入的最新信息或输出结果立即输出，可以使用输入输出刷新指令REF。

MC系列PLC的输入 ON 时间或 OFF 时间，必须比控制器的扫描周期时间更长。假设 10ms 扫描周期，则输入ON、OFF 的时间各需 10ms。因此，不可以处理 1000/(10+10)=50Hz 以上的普通输入脉冲。要处理更高速的信号，可以应用高速计数器功能，输入中断功能，输入上升沿计数等。

6.3 程序流程控制指令之间的相互关系

程序流程控制指令之间的相互关系如下表，编程时要遵循此表规则，否则可能造成运行故障。

流程之间的相互/调用关系									
说明 1：左列调用上列：√表示可以，×表示不可以，◎表示谨慎使用									
说明 2：其中 CJ 的和其它指令的重叠包括 CJ 指针 P 直接指向其它指令的开始指令，比如 CJ 的指针 P 指向 CJ 或 FOR。									
上列 / 左列	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET	P-SRET	I-IRET /PRET	FEND-END	MODBUS 自由协议
MC-MCR	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
CJ-P	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 ◎	包含 ◎	包 含 ×	包 含 √
	重叠 ◎	重叠◎	重叠 √	重叠 ◎	重叠◎	重叠 ◎	重叠 ◎	重 叠 √	重 叠 ×
						调用 √			
EI-DI	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 √	包含 √	包 含 √	包 含 √
	重 叠 √	重叠√	重叠 √	重 叠 √	重叠 √	重叠 √	重叠 √	重 叠 √	重 叠 ×
FOR-NEXT	包 含 ×	包含√	包含 √	包 含 √	包含 ×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 √	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
STL-RET	包 含 ×	包含◎	包含 √	包 含 √	包含 ×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
P-SRET	包 含 ×	包含√	包含 √	包 含 √	包含 2×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
I-IRET/PRET	包 含 ×	包含√	包含 √	包 含 √	包含 2×	包含 ×	包含 ◎	包 含 ×	包 含 √
	重 叠 ×	重叠 3×	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
FEND-END	包含 1√	包含√	包含 √	包 含 √	包含 1◎	包含 √	包含 √	包含 √	
	重 叠 ×	重叠×	重叠√	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重叠 √	

1：需配合 CJ 指令使用，否则主控执行不了。

2：可以使用，但不建议使用。

3：I 中跳转到 I 外部时必须要有跳回到 I 中的 CJ-P 语句。

定义 1：两个发生关系的流程指令的语句为 A1 和 B1、A2 和 B2。

定义 2：包含关系为，A1→A2→B2→B1，例，CJ（A1）→FOR（A2）→NEXT（B2）→P（B1）。

定义 3：重叠关系为，A1→A2（B1）→B2，例，CJ（A1）→FOR（A2）→P（B1）→NEXT（B2）。

定义 4：调用的意思是在左列的指令中调用上列的指令，一般使用 CALL 指令调用上列指令。

定义 5：MODBUS 和自由协议表示的是两种通信协议的通信指令，只存在被其它流程指令包含的情况。

在编写梯形图时，请严格按照说明书规则进行编程，否则可能造成运行故障。当MC系列PLC出现运行故障时，按以下顺序进行控制器的复位。

- 断电后重新上电
- 在上电后的 1～3 秒的时间内，点击编程视窗的“PLC 停止”按钮
- 出现 PLC 已停止运行的对话框后，下载正确的梯形图以恢复MC系列PLC的正常运行

7. 基本逻辑指令

在本章中，描述了基本指令系统的种类及其功能。 基本指令系统分为两部分：一部分是基本逻辑运算及输出指令，包括取、与、或及它们的反运算、置位、复位和输出指令。这些指令是 PLC 的基本逻辑指令，加上定时器和计数器的综合应用，基本上可以实现继电器控制系统的程序编程。在程序中，这部分指令用触点、线及连线可以很方便的在梯形图中表示。另一部分是逻辑处理指令。这些指令在程序中并不表示一定的逻辑运算，而是对复杂逻辑运算的处理，它包括电路块、堆栈、主控操作、边沿处理指令等。

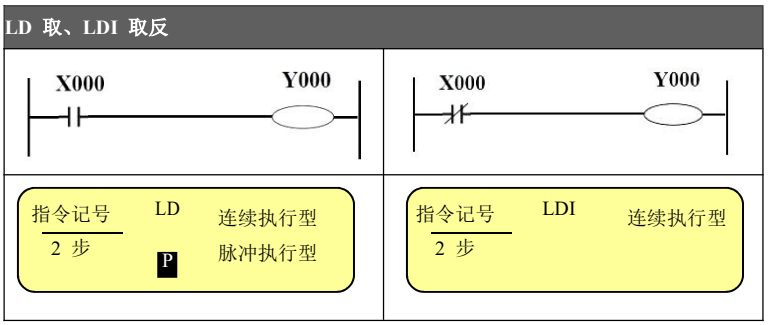
助记符、名称	功能	章节
LD/LDI	运算开始常开/常闭触点	7.1
OUT	线圈驱动	7.2
AND/ANI	常开/常闭触点串联连接	7.3
OR/ORI	常开/常闭触点并联连接	7.4
LDP/LDF	上升/下降沿检出运算开始	7.5
ANDP/ANDF	上升/下降沿检出串联连接	
ORP/ORF	上升/下降沿检出并联连接	
ORB	串联回路块的并联连接	7.6
ANB	并联回路块的串联连接	7.7
MPS	运算存储	7.8
MRD	存储读出	
MPP	存储读出与复位	
INV	取反	7.9
ALT	位取反	7.10
PLS/PLF	上升/下降沿输出	7.11
SET	动作保持	7.12
RST	清除动作保持，寄存器清 0	
OUT/RST	对应计数器软元件的[OUT],[RST]指令	7.13
NOP	空动作	7.14
END	扫描周期结束	

7.1 [LD]取,[LDI]取反指令

7.1.1 概要

LD、LDI 指令是连接在母线上的触点。和后述的 ANB 指令组合后，也可用在分支起点处。

1) 指令格式



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
LD	•	•	•	•	•	•														
LDI	•	•	•	•	•	•														

7.1.2 功能和动作说明

1) LD 指令(a 触点的逻辑运算开始)

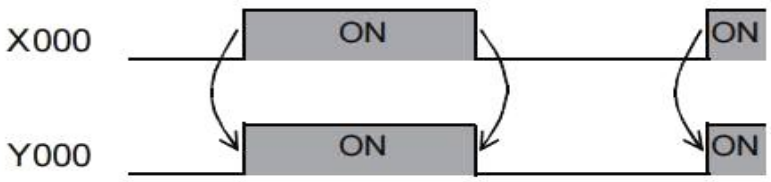
梯形图程序



指令表程序

```
LD   X000
OUT  Y000
```

时序图



2) LDI 指令(b 触点的逻辑运算开始)

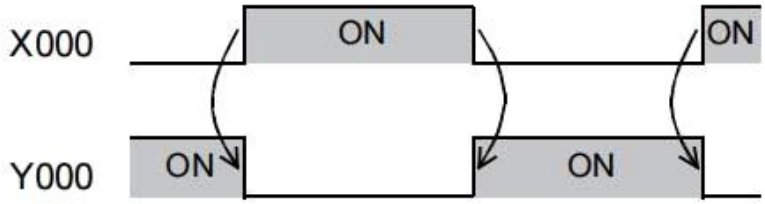
梯形图程序



指令表程序

LDI	**X000**
OUT	Y000

时序图



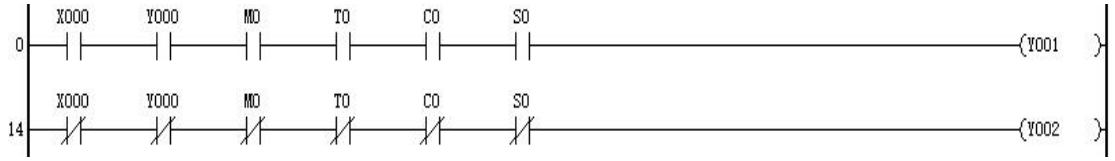
7.1.3 注意要点

1) 变址修饰

LD 和 LDI 指令中使用的软元件，不能直接用变址寄存器(V、Z)进行修饰。

2) 编程规则

梯形图中，每一梯级的第一个触点必须用取指令 LD(常开)或取反指令 LDI (常闭)，并与左母线相连。LD,LDI 指令也可用在电路块的第一个触点上，也可用在主控指令的子母线相连的触点上，如何用法，这在指令语句表程序设计中是必须要熟练掌握的。如果用的不恰当，编译时会出错。但在使用编程软件编辑梯形图程序时，由于可以用快捷键或快捷图标输入，可以根本不考虑在什么情况下用取指令 LD，LDI，当梯形图被切换成指令语句表程序时，会自动安排取指令的使用。类似这样的情况还有电路块指令和堆栈指令等。



7.1.4 程序应用

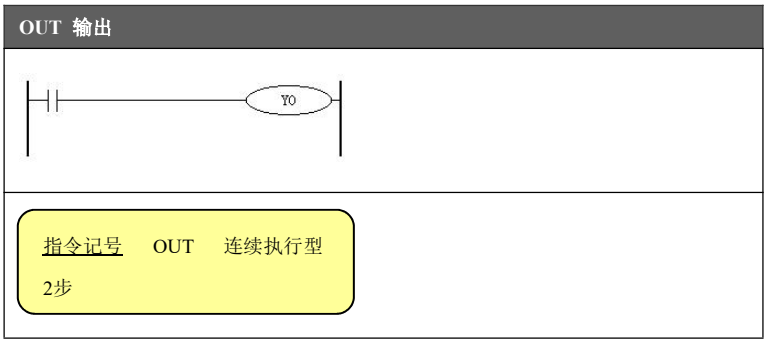
注意：LD、LDI 分别可以用软元件 X、Y、M、T、C、S 做触点。

7.2 [OUT]输出指令

7.2.1 概要

OUT 指令是对输出继电器(Y)、辅助继电器(M)、状态继电器(S)、定时器(T)、计数器(C)进行线圈驱动的指令。对输入继电器不能使用。

1) 指令格式



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
OUT		•	•	•	•	•														

7.2.2 功能和动作说明

1) 使用位软元件时

用 OUT 指令编写的软元件，根据驱动触点的状态执行 ON/OFF。

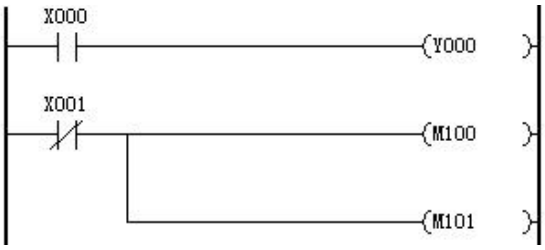
OUT 指令为继电器线圈驱动指令。将线圈前的逻辑运算结果输出到指定的继电器，使其触点产生相应的动作。逻辑运算结果为 1，继电器闭合；结果为 0，继电器断开。

并联的 OUT 命令能够多次连续使用。

下面的程序举例中，接着 OUT M100 的 OUT M101 就是这样。

但是，对同一软元件编号，使用多个OUT 指令时，会变成双重输出(双线圈)，请注意。

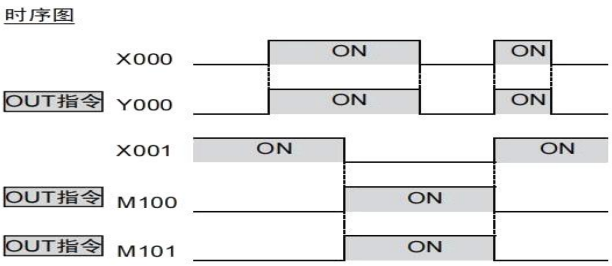
梯形图程序



指令表程序

```
LD    X000
OUT   Y000
LDI   X001
OUT   M100
OUT   M101
```

时序图



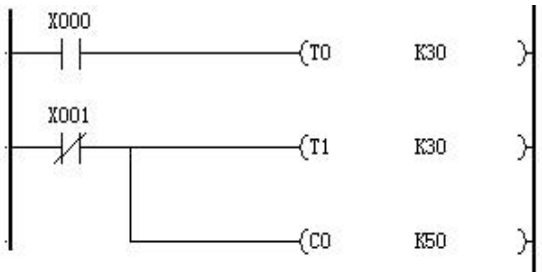
2）使用定时器和计数器时

在针对定时器的计时线圈和计数器的计数线圈的 OUT 指令后需要加上设定值。

设定值可以使用 10 进制数(K)直接指定，也可以使用数据寄存器(D)或扩展寄存器(R)间接指定。

①直接指定

梯形图程序



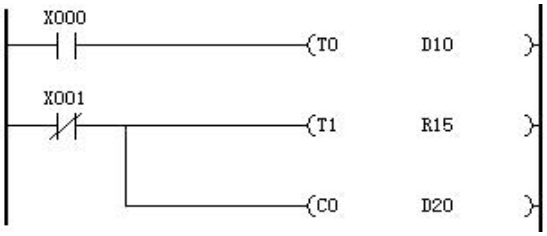
指令表程序

```
LD      X000
OUT T0  K30
LDI     X001
OUT T1  K30
OUT C0  K50
```

以 10 进制数(K)设定定时器、计数器的设定值。

②间接指定

梯形图程序



指令表程序

```
LD      X000
OUT T0  D10
LDI     X001
OUT T1  R15
OUT C0  D20
```

可以在数据寄存器 (D)和扩展寄存器 (R)中设定定时器和计数器的设定值。此时，数据寄存器(D)和扩展寄存器(R)的当前值即为定时器的设定值。在驱动定时器或计数器之前，需要事先通过赋值指令、人机界面等将设定值写入作为设定值使用的数据寄存器 (D)和扩展寄存器(R)中，且要关注所使用的定时器或计数器的设定值范围，超出设定范围可能会造成运行异常或得不到预设的结果。

3) 定时器、计数器的设定范围

定时器和计数器的设定值的设定范围以及实际的定时器常数、OUT 指令的程序步数(包含设定值)都参照下表所示

定时器,计数器	常数的设定范围	实际的设定值	步数
1ms 定时器	1～32,767	0.001～32.767 秒	5
10ms 定时器		0.01～327.67 秒	5
100ms 定时器		0.1～3276.7 秒	5
16 位计数器		1～32,767	5
32 位计数器	-2,147,483,648 ～ +2,147,483,648	同左	7

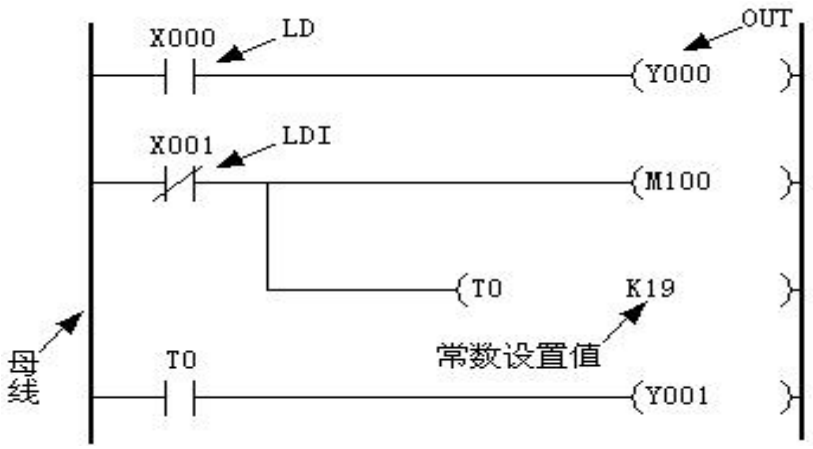
7.2.3 注意要点

1) OUT 指令中使用的软元件，不能直接用变址寄存器(V、Z)进行修饰。

2) 使用特殊内部继电器 (M)、定时器（T）、计数器(C)时，程序步按照上述的“定时器、计数器的设定范围”中记载的内容增加。

3) 32 位计数器的设定值中，请勿使用数据寄存器(D)和扩展寄存器(R)的末尾编号。

7.2.4 程序应用



并列的 OUT 命令能多次连续使用，上图中 OUT M100 后紧跟着OUT T0。

X000 导通（ON）Y000 输出，X001 闭合（OFF）M100 输出，T0 定时 1.9S 后 Y1 输出。

7.3 [AND]与,[ANI]与反指令

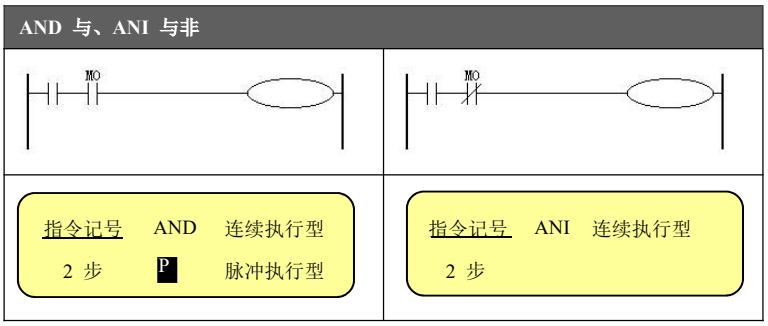
7.3.1 概要

AND、ANI 指令是执行串联连接 1 个触点。

串联触点的数量没有限制，该指令可以连续多次使用。

OUT 指令后，触点对其他的线圈使用 OUT 指令，称为纵接输出。只要顺序不错，这样的纵接输出可以重复使用多次。

1) 指令格式



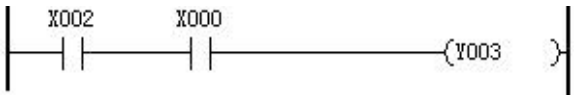
2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
AND	•	•	•	•	•	•														
ANI	•	•	•	•	•	•														

7.3.2 功能和动作说明

1) AND 指令(串联 a 触点)

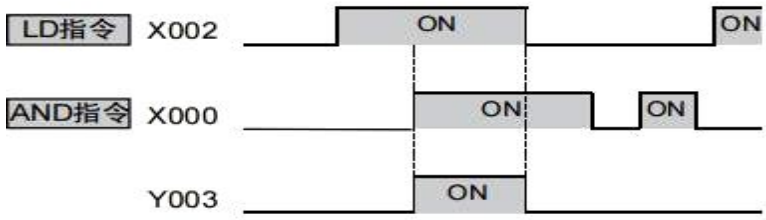
梯形图程序



指令表程序

```
LD    X002
AND   X000
OUT   Y003
```

时序图



2) ANI 指令(串联 b 触点)

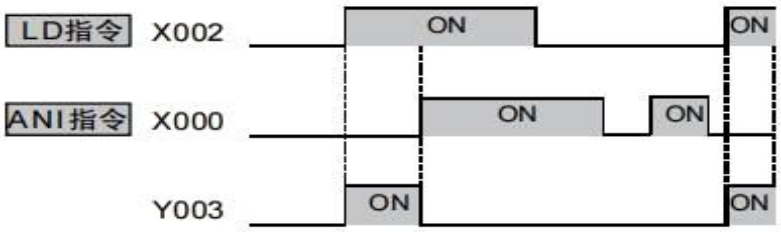
梯形图程序



指令表程序

```
LD    X002
ANI   X000
OUT   Y003
```

时序图

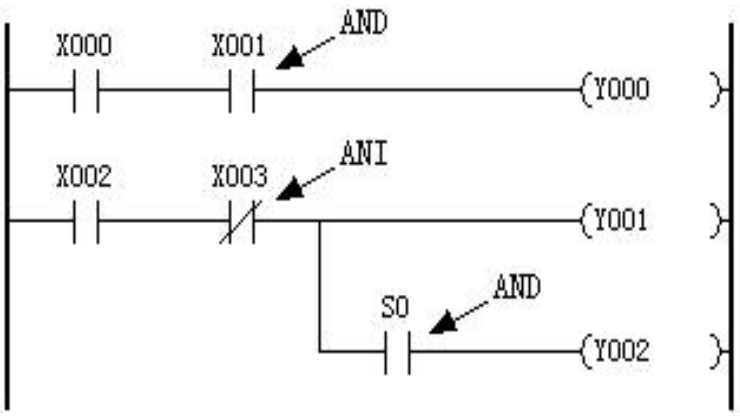


7.3.3 注意要点

- **变址修饰**

AND 和 ANI 指令中使用的软元件，不能直接用变址寄存器(V、Z)进行修饰。

7.3.4 程序应用



OUT 指令后，通过触点对其他线圈使用OUT 指令，称之为纵接输出。(上图的 OUT Y001 与 OUT Y002)。这样的纵接输出如果顺序不错，可重复多次。

X000、X001 常开串联（AND）导通 Y000 输出，X002、X003 常闭串联（ANI）导通 Y001 输出，S0 导通Y002 输出。

7.4 [OR]或、[ORI]或反指令

7.4.1 概要

OR、ORI 指令可以作为并联连接 1 个触点的指令使用。

串联连接了 2 个以上的触点时，要将这样的串联回路块与其他回路并联的时候，采用后述的 ORB 指令。

OR、ORI 是从这个指令的步开始，与前面的LD、LDI 指令的步进行并联连接。并联连接的次数不受限制。

1) 指令格式

OR 或、ORI 或反	
	
指令记号 OR 连续执行型 2 步 P 脉冲执行型	指令记号 ORI 连续执行型 2 步

2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
OR	•	•	•	•	•	•														
ORI	•	•	•	•	•	•														

7.4.2 功能和动作说明

1) OR 指令(并联 a 触点)

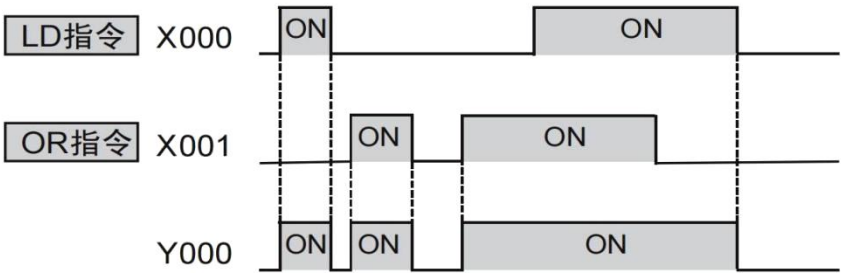
梯形图程序



指令表程序

```
LD    X000
OR    X001
OUT   Y000
```

时序图



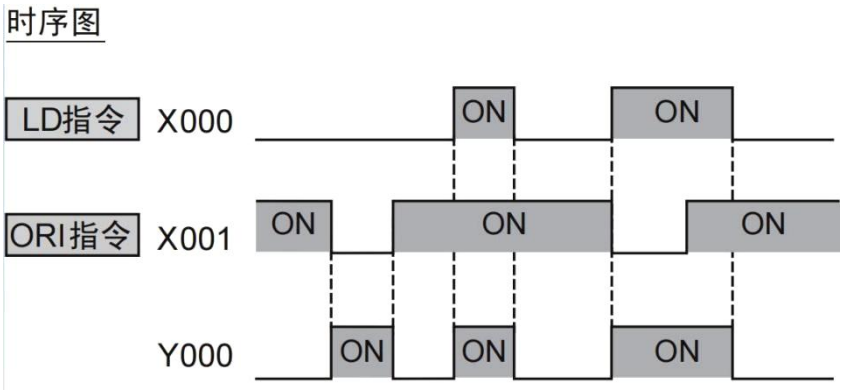
2) ORI 指令(并联 b 触点)

梯形图程序

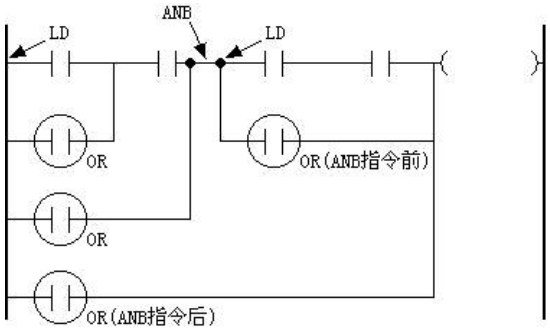


指令表程序

```
LD    X000
ORI   X001
OUT   Y000
```



3) ANB 指令的关系



用 OR、ORI 指令并联连接，原则上是与前面的 LD、LDI 点连接，但是在后述的 ANB 指令后，则是与前一个的 LD、LDI 点连接。

7.4.3 注意要点

1) 变址修饰

OR 和 ORI 指令中使用的软元件，不能直接用变址寄存器(V、Z)进行修饰。

2) 编程规则

触点串、并联指令仅是用来描述单个触点与其他触点的电路连接关系。如果所串联的是一个并联电路块或并联的是一个串联电路块, 则不能使用串、并联指令，要用后述的电路块指令 ANB 和 ORB。

7.4.4 程序应用

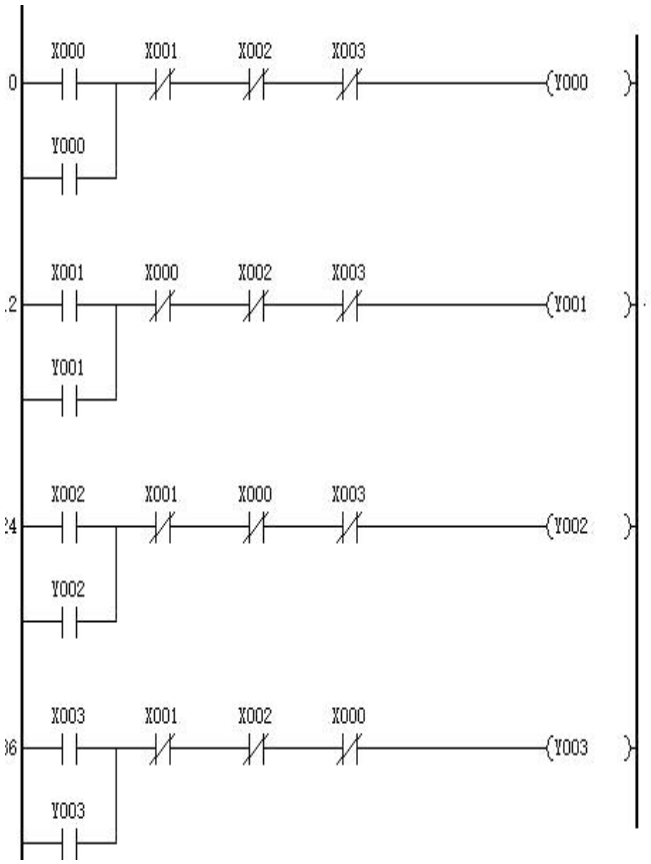
1) 控制要求：

有四个自复位按钮 S1、S2、S3、S4 和四个指示灯 L1、L2、L3、L4，要求按下 S1 则L1 亮，其它灯熄，按下 S2 则 L2 亮，其它灯熄， 按下 S3 则 L3 亮，其它灯熄， 按下 S4 则 L4 亮， 其它灯熄。

2) IO 分配

输入			输出		
动作演示区	S1	X0	动作演示区	L1	Y0
	S2	X1		L2	Y1
	S3	X2		L3	Y2
	S4	X3		L4	Y3

3) 梯形图程序



● 按下 S1（X000）,停止 L2、L3、L4 灯，亮 L1 灯（Y000)；

● 按下 S2（X001）,停止 L1、L3、L4 灯，亮 L2 灯（Y001)；

● 按下 S3（X002）,停止 L1、L2、L4 灯，亮L3 灯（Y002)；

● 按下 S4（X003）,停止 L1、L2、L3 灯，亮L4 灯（Y003)；

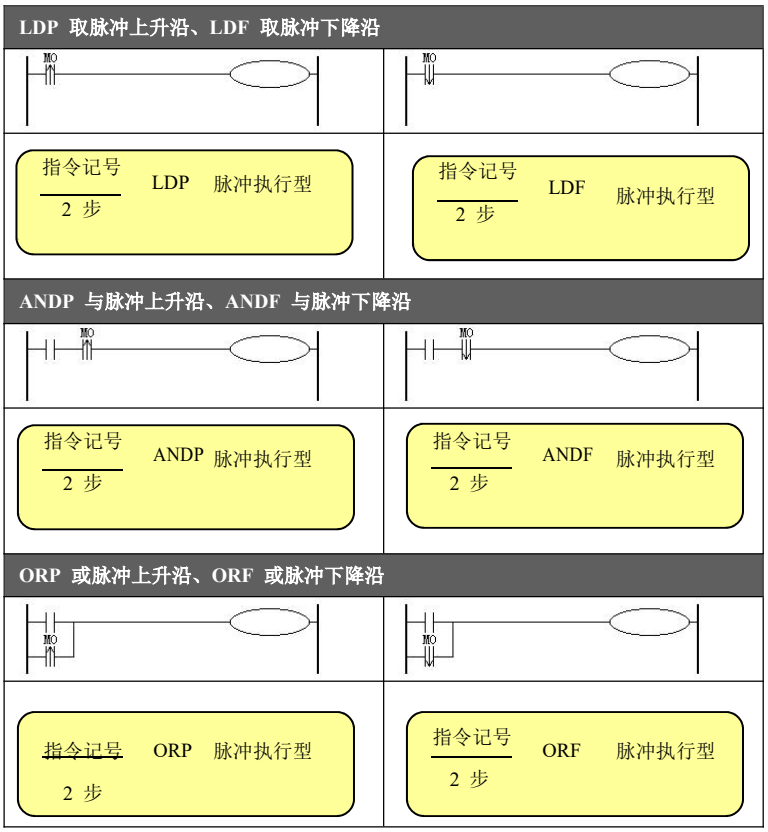
7.5 [LDP]、[LDF]、[ANDP]、[ANDF]、[ORP]、[ORF]指令

7.5.1 概要

LDP、ANDP、ORP 指令是检测上升沿的触点指令，仅在指定位软元件的上升沿(从OFF 改变到 ON 的时候) 时接通 1 个运算周期。

LDF、ANDF、ORF 指令是检测下降沿的触点指令，仅在指定位软元件的下降沿(从ON 改变到 OFF)时，接通1个运算周期。

1) 指令格式



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
LDP	•	•	•	•	•	•														
LDF	•	•	•	•	•	•														
ANDP	•	•	•	•	•	•														
ANDF	•	•	•	•	•	•														
ORP	•	•	•	•	•	•														
ORF	•	•	•	•	•	•														

7.5.2 功能和动作说明

1) LDP、ANDP、ORP 指令(检测到上升沿时运算开始、串联连接、并联连接)

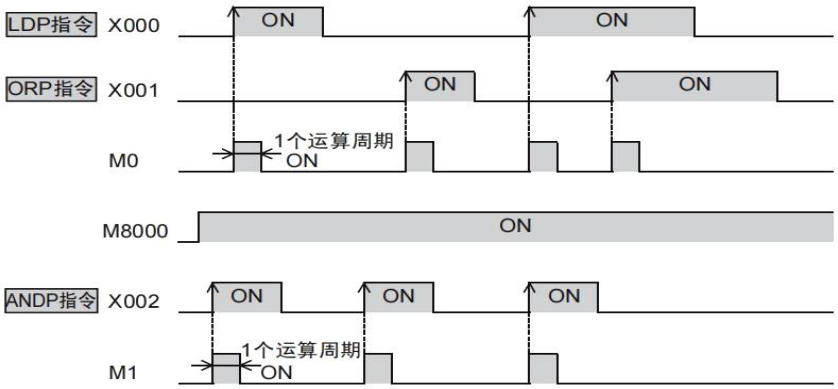
梯形图程序



指令表程序

```
LDP   X000
ORP   X001
OUT   M0
LD    M8000
ANDP  X002
OUT   M1
```

时序图



在上图中，X000～X002 从OFF 变成 ON 时，M0 或 M1 只维持 1 个运算周期为 ON。

2) LDF、ANDF、ORF 指令(检测到下降沿时运算开始、串联连接、并联连接)

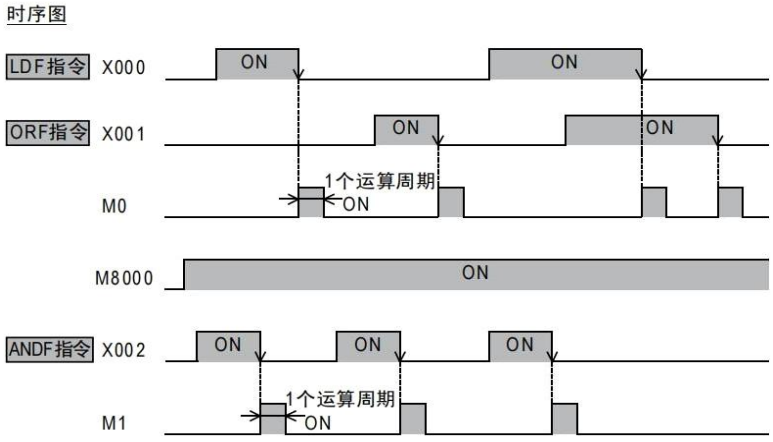
梯形图程序



指令表程序

```
LDF   X000
ORF   X001
OUT   M0
LD    M8000
ANDF  X002
OUT   M1
```

时序图



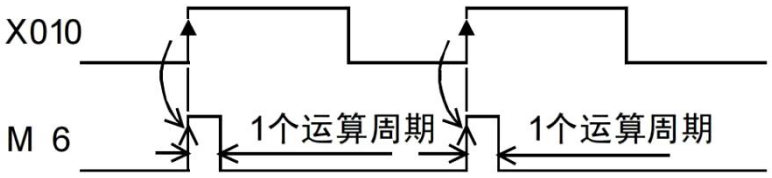
在上图中，X000～X002 从ON 变成 OFF 时，M0 或 M1 只维持 1 个运算周期为 ON。

3) 输出驱动实例

下面的左边梯形图与右边梯形图执行结果相同。

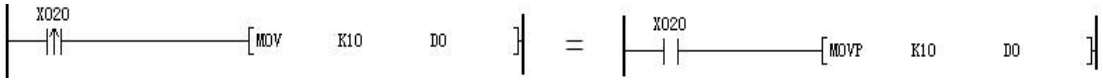
《OUT 指令》 《脉冲指令》





无论何种情况，X010 从 OFF 变为 ON 的时候，M6 只维持 1 个运算周期为 ON。

《上升沿检出》 《应用指令的脉冲指令》



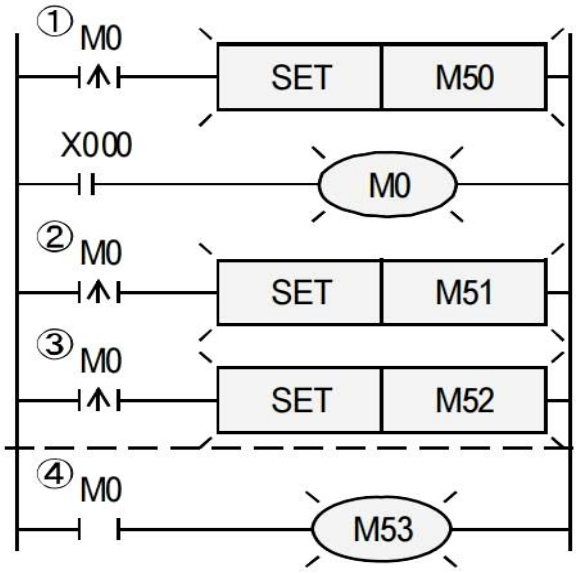
无论何种情况，X020 从 OFF 变为 ON 时，只执行 1 次 MOV 指令。

4) 辅助继电器(M)编号决定的不同动作

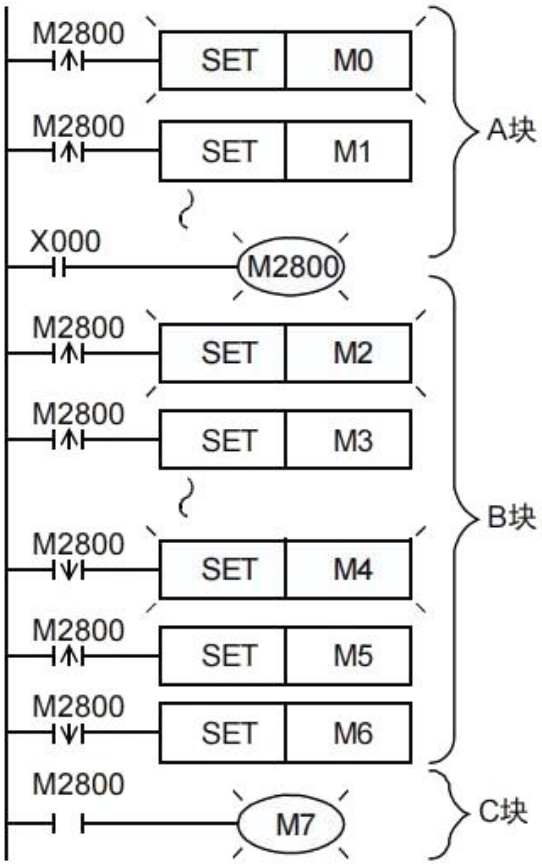
将辅助继电器(M)指定为 LDP、LDF、ANDP、ANDF、ORP、ORF 指令的软元件时，根据这个软元件的编号范围，会如下图所示，出现不同的动作。

◆ M0～M2799

◆



由 X000 驱动M0 后，M0 对应的 1～4 的所有触点都会动作。1～3，执行 M0 的上升沿检测。4 中由有 LD 指令，所以 M0 为ON 时导通。

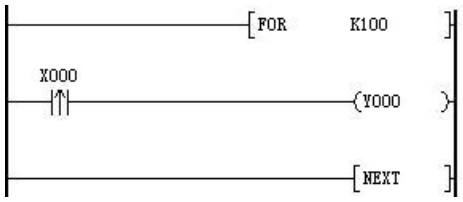


以 X000 驱动的M2800 为中心，分为上下 AB 块。AB 各块内的上升沿或下降沿检测触点，只有最前面 1 个会动作。C 块的触点中由于有LD 指令，在 M2800 为 ON 时导通。使用这个特性，能够更加有效地对步进梯形图回路中的「采用同一信号进行状态转移」进行编程。

7.5.3 注意要点

同一步中编写的LDP、LDF、ANDP、ANDF、ORP、ORF 指令，在 1 个运算周期中被多次执行的情况下的注意事项。

被执行多次的程序



- FOR-NEXT 指令之间的程序
- 在1 个运算周期中，多个CALL 指令调用执行相同的子程序
- 使 用 CJ 指令，跳转到步编号小的标签(P)的程序

程序动作

1) 软元件从OFF 变化到 ON 的情况

第 1 次 : LDP、ANDP、ORP 指令置 ON。

第 2 次以后 :和执行上一次指令时的软元件状态相同时，指令置 OFF。

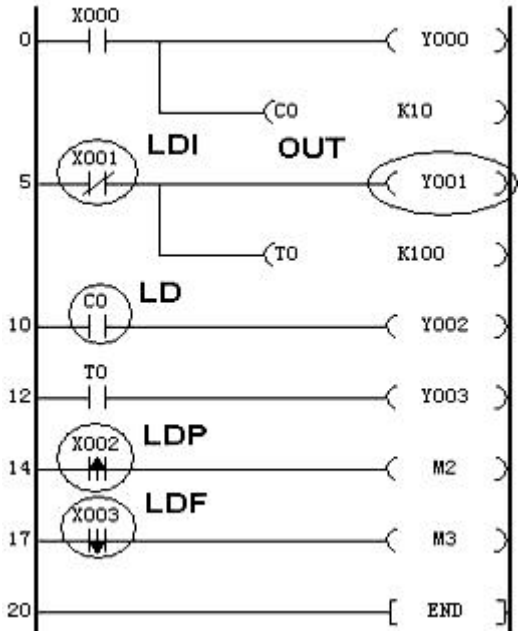
2) 软元件从ON 变化到 OFF 的情况

第 1 次 : LDF、ANDF、ORF 指令置 ON。

第 2 次以后 :和执行上一次指令时的软元件状态相同时，指令置 OFF。

7.5.4 程序应用

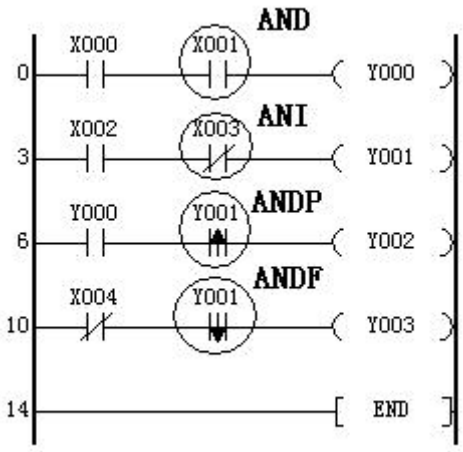
1） LDP、LDF 编程实例




```
0   LD    X000
1   OUT   Y000
2   OUT   C0 K10
5   LDI   X001
6   OUT   Y001
7   OUT   T0 K100
10  LD    C0
11  OUT   Y002
12  LD    T0
13  OUT   Y003
14  LDP   X002
16  OUT   M2
17  LDF   X003
19  OUT   M3
20  END
```

- 用 LD,LDI,LDP,LDF 指令与母线连接。输出使用OUT 指令驱动线圈。
- 使 用 OUT 指令驱动定时器的计时线圈或者计数器的计数线圈时，必须设定定时和计数的时间和计数值。
- 可 以 是 常 数K,或者由数据寄存器间接指定数值。每个程序结束必须要有 END 指令，关于 END 指令详见后面的 END 指令介绍。

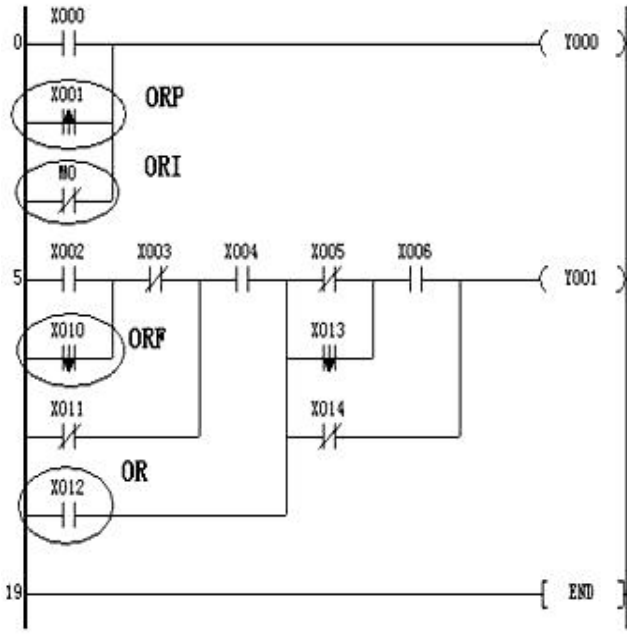
2） ANDP、ANDF 编程实例




```
0   LD    X000
1   AND   X001
2   OUT   Y000
3   LD    X002
4   ANI   X003
5   OUT   Y001
6   LD    Y000
7   ANDP  Y001
9   OUT   Y002
10  LDI   X004
11  ANDF  Y001
13  OUT   Y003
14  END
```

- 实例中X001，X003，Y001 作为串联触点与前面的触点相连。

3） ORP、ORF 编程实例




```
0    LD      X000
1    ORP     X001
3    ORI     M0
4    OUT     Y000
5    LD      X002
6    ORF     X010
8    ANI     X003
9    ORI     X011
10   AND     X004
11   OR      X012
12   LDI     X005
13   ORF     X013
15   AND     X006
16   ORI     X014
17   ANB
18   OUT     Y001
19   END
```

使用 OR，ORI，ORP，ORF 与前面的 LD，LDI，LDP，LDF 并联连接，在程序步 12 到16 中，由于是两个并联回路块的串联，所以使用 ANB 指令，关于 ANB 指令详见后面说明。

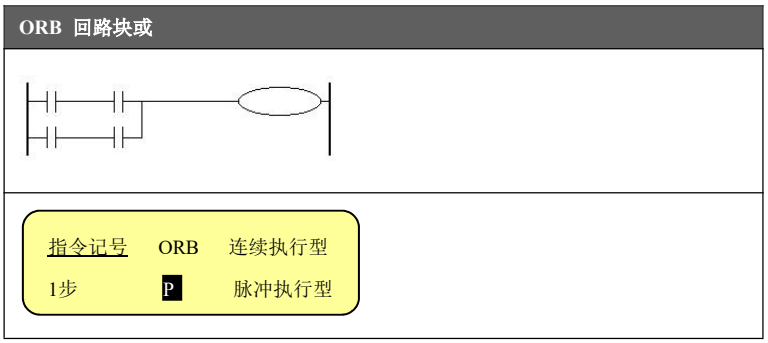
7.6 [ORB]指令

7.6.1 概要

ORB 指令是执行回路块并联连接的指令。

由 2 个以上的触点串联连接的回路称为串联回路块。

1) 指令格式



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
ORB	无对象软元件																			

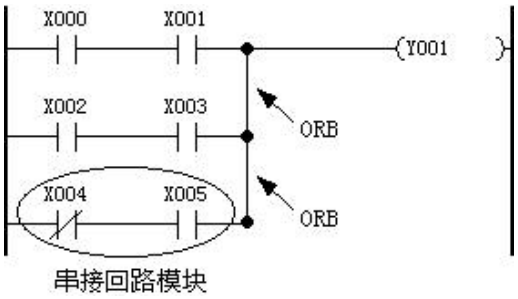
7.6.2 功能和动作说明

ORB 指令(回路块的并联连接)

并联连接串联回路块时，分支的起点使用 LD、LDI 指令，分支的结束使用 ORB 指令。

ORB 指令，与后述的 ANB 指令等相同，都是不带软元件编号的独立指令。有多个并联回路时，在每个回路块中使用 ORB 指令，从而形成连接。

梯形图程序



指令表程序

正确的程序

0	LD	X000
2	AND	X001
4	LD	X002
6	AND	X003
8	**ORB**	
9	LDI	X004
11	AND	X005
13	**ORB**	
14	OUT	Y001

不佳的程序

0	LD	X000
2	AND	X001
4	LD	X002
6	AND	X003
8	LDI	X004
10	AND	X005
12	**ORB**	
13	**ORB**	
14	OUT	Y001

7.6.3 注意要点

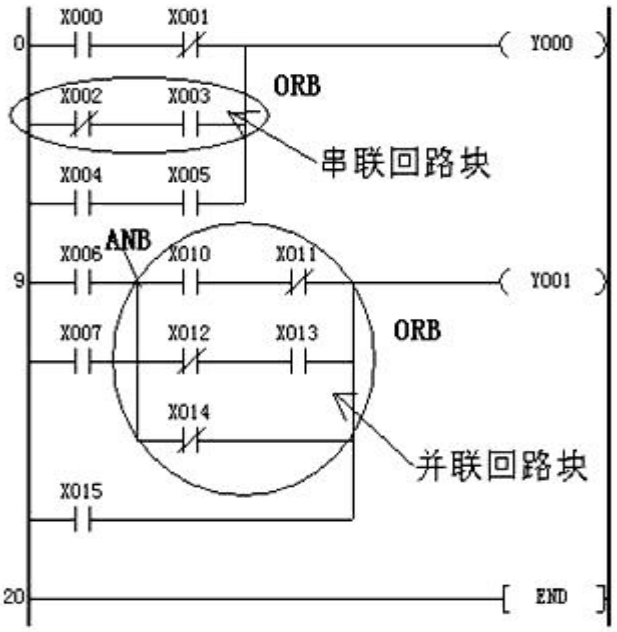
ORB 指令的使用方法有两种：一种是在要并联的每个串联回路块后加 ORB 指令，详见上图 推荐程序语句表；另一种是集中使用 ORB 指令，详见上图不推荐程序所示语句表。

对于上图推荐程序分散使用 ORB 指令时，并联电路的个数没有限制，但对于不推荐程序集中使用 ORB

指令时，这种电路块并联的个数不能超过 5 个（即重复使用ORB 指令的次数限制在 5 次以下）

7.6.4 程序应用

ORB 编程实例




```
0   LD      X000
1   ANI     X001
2   LDI     X002
3   AND     X003
4   ORB
5   LD      X004
6   AND     X005
9   ORB
10  OUT     Y000
11  LD      X006
12  OR      X007
13  LD      X010
15  ANI     X011
16  LDI     X012
17  AND     X013
18  ORB
19  ORI     X014
17  ANB
18  OR      X015
19  OUT     Y001
20  END
```

在每个分支的最后使用 ORB 指令，不要在所有的分支后面使用ORB 指令，如程序步 4 和 17 所示。

ORB 和ANB 指令只对块的连接，如果不是对块就不能使用，如程序步 16 和18 不是块就不能使用。如图所示串联回路块和并联回路块的示例

7.7 [ANB]指令

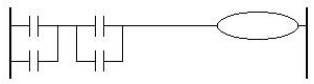
7.7.1 概要

当分支回路(并联回路块)与前面的回路串联连接时，使用ANB 指令。

分支的起点使用 LD、LDI 指令，并联回路块结束后，可以使用 ANB 指令和前面的回路串联连接。有多个并联回路的时候，对每个回路块使用 ANB 指令，从而连接。

1) 指令格式

ANB 回路块与



指令记号	ANB	连续执行型
1步	P	脉冲执行型

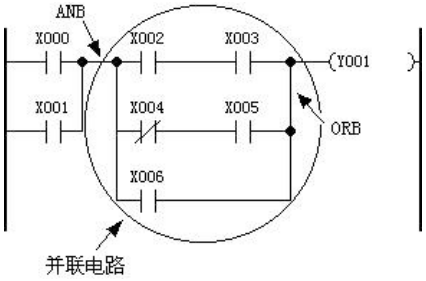
2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
ANB	无对象软元件																			

7.7.2 功能和动作说明

ANB 指令(回路块的串联连接)

梯形图程序



指令表程序

0	LD	X000	
2	OR	X001	
4	LD	X002	分支起点
6	AND	X003	
8	LDI	X004	分支起点
10	AND	X005	
12	ORB		并联电路块结束
13	OR	X006	
15	**ANB**		与前面的电路串联
16	OUT	Y001	

7.7.3 注意要点

ANB 指令的使用次数没有限制。

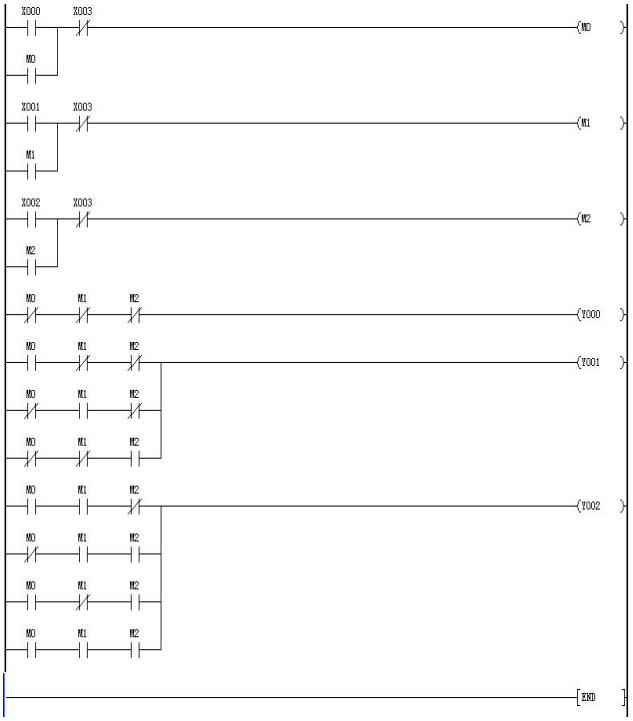
7.7.4 程序应用

1) **控制要求：** 有三个按钮 S1、S2、S3，用此三个按钮模拟三个电动机工作状态检测传感器，按下相应电动机对应的按钮，表示该电动机工作不正常了。有三个指示灯 L1、L2、L3。要求当三个电动机都工作正常时 L1 亮；当三个电动机中只有一台工作不正常则 L2 亮；当三台电动机中有两台或两台以上工作不正常时L3 亮。S4 为复位按钮，在三台电动机均工作正常后按此按钮系统复位。

2) IO 分配

输入			输出		
	S1	X0		L1	Y0
	S2	X1		L2	Y1
	S3	X2		L3	Y2
	S4	X3			

3) 梯形图程序



- 按下 S1（X000）,第一台电动机有故障，启动标志为 M0,按S4(X003)复位。
- 按下 S2（X001）,第二台电动机有故障，启动标志为 M1,按S4(X003)复位。
- 按下 S3（X002）,第三台电动机有故障，启动标志为 M2,按S4(X003)复位。
- 三台电机均正常，L1亮
- 三台电机中有一台工作不正常，L2亮
- 三台电机中有两台或两台以上工作不正常时，L3亮

7.8 [MPS]、[MRD]、[MPP]指令

7.8.1 概要

在可编程控制器中，有 11 个被称为堆栈的内存，用于记忆运算的中间结果(ON 或OFF)。

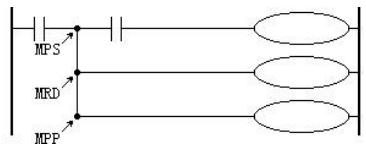
MPS 指令：把分支点的状态（1 或者 0）送到栈存储器的第一位并保存起来。第一次使用 MPS 指令，该时刻的运算结果就压入栈的第一个栈单元中（称之为栈项）。再次使用 MPS 指令时，当时的运算结果压入栈顶，而原先压入的数据依次向栈的下一个栈单元推移。

MPP 指令：读出栈存储器第一位保持的数据（1 或者 0），并移除栈存储器第一位保存的数据。栈存储器中各数据依次向上一个栈单元传送。栈顶数据在弹出后就从栈内消失。

MRD 指令：是栈顶数据的读出专用指令，但栈内的数据不发生下压或上托的传送。

1) **指令格式**

MPS 进栈、MRD 读栈、MPP 出栈



指令记号 MPS、MRD、MPP 连续执行型

1步

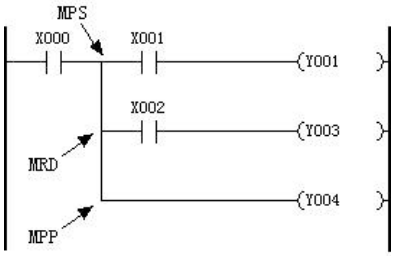
2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
MPS	无对象软元件																			
MRD	无对象软元件																			
MPP	无对象软元件																			

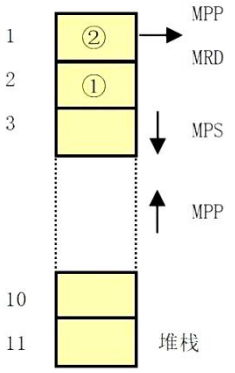
7.8.2 功能和动作说明

该指令是用于编写多重分支输出回路的便捷指令。

1. MPS、MRD、MPP 指令(压入堆栈、读取堆栈、弹出堆栈)




```
0   LD    X000
2   MPS
3   AND   X001
5   OUT   Y001
7   MRD
8   AND   X002
10  OUT   Y003
12  MPP
13  OUT   Y004
```



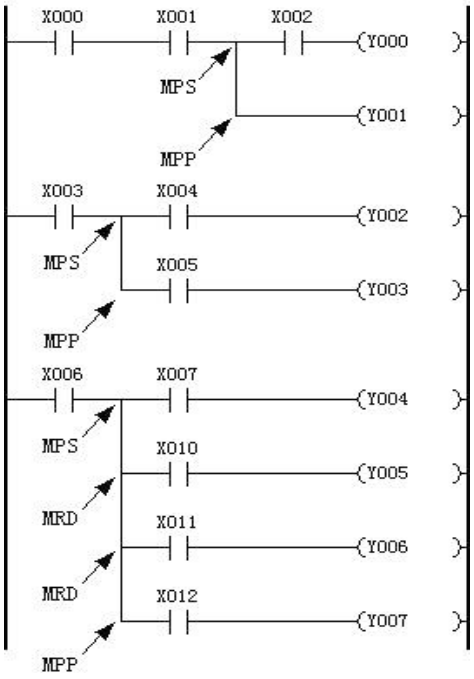
- 使 用 MPS 指令存储运算的中间结果后，驱动输出 Y001。
- 使 用 MRD 指令读取该存储内容后，驱动输出 Y003。MRD 指令可以多次编程。
- 在最终输出回路中使用MPP 指令替代 MRD 指令。从而在读出上述存储内容的同时将其复位。

程序举例 1

在这个实例中，只使用了一段堆栈。

梯形图程序

一段堆栈



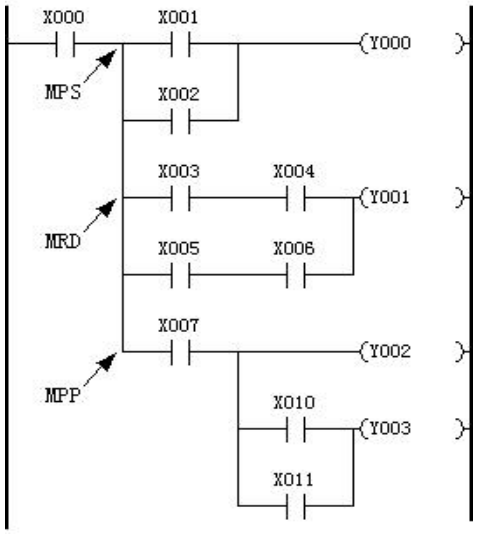
指令表程序

0	LD	X000	36	MRD	
2	AND	X001	37	AND	X011
4	**MPS**		39	OUT	Y006
5	AND	X002	41	**MPP**	
7	OUT	Y000	42	AND	X012
9	**MPP**		44	OUT	Y007
10	OUT	Y001			
12	LD	X003			
14	**MPS**				
15	AND	X004			
17	OUT	Y002			
19	**MPP**				
20	AND	X005			
22	OUT	Y003			
24	LD	X006			
26	**MPS**				
27	AND	X007			
29	OUT	Y004			
31	**MRD**				
32	AND	X010			
34	OUT	Y005			

程序举例 2

梯形图程序

一段堆栈，ANB，ORB指令并用

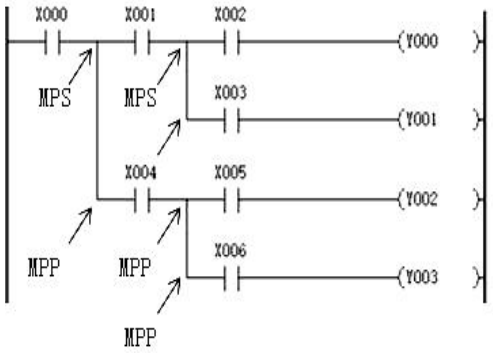


指令表程序

0	LD	X000	19	ORB	
2	**MPS**		20	ANB	
3	LD	X001	21	OUT	Y001
5	OR	X002	23	**MPP**	
7	ANB		24	AND	X007
8	OUT	Y000	26	OUT	Y002
10	**MRD**		28	LD	X010
12	LD	X003	30	OR	X011
13	AND	X004	32	ANB	
15	LD	X005	33	OUT	Y003
17	AND	X006			

程序举例3 (2 段堆栈)

梯形图程序



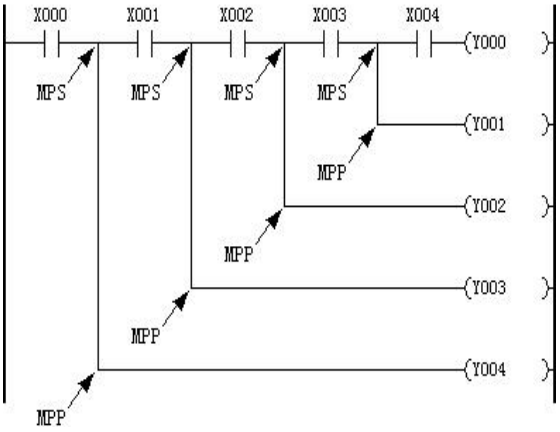
指令表程序

0	LD X000	14	**MPP**
2	**MPS**	16	AND X004
3	AND X001	17	**MPS**
5	**MPS**	19	AND X005
6	AND X002	20	OUT Y002
8	OUT Y000	22	**MPP**
9	**MPP**	23	AND X006
11	AND X003	25	OUT Y003
12	OUT Y001		

程序举例 4

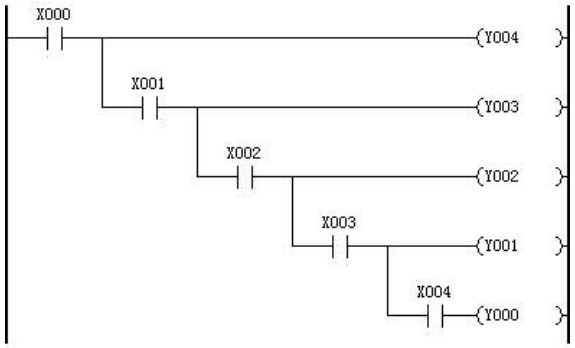
梯形图程序

多段堆栈(4段)



指令表程序

0	LD X000	17	OUT Y001
2	**MPS**	19	**MPP**
3	AND X001	20	OUT Y002
5	**MPS**	22	**MPP**
6	AND X002	23	OUT Y003
8	**MPS**	25	**MPP**
9	AND X003	26	OUT Y004
11	**MPS**	28	END
12	AND X004		
14	OUT Y000		
16	**MPP**		



0	LD X000	12	AND X003
2	OUT Y004	14	OUT Y001
4	AND X001	16	AND X004
6	OUT Y003	18	OUT Y000
8	AND X002	20	END
10	OUT Y002		

在图上方的梯形图中，需要使用三重的 MPS 指令编程。

但是，如果改用图中下方的回路，不使用 MPS 指令也能轻松编程

7.8.3 注意要点

MPS 指令也可以重复使用，但是MPS 指令和 MPP 指令的数量差小于 11，最终两者的指令数量需要一致。

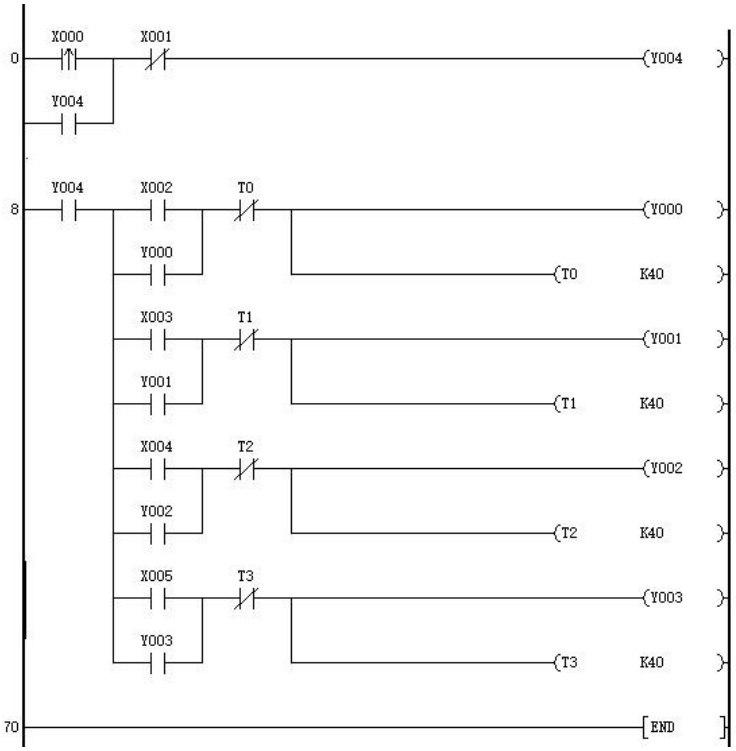
7.8.4 程序应用

1) **控制要求：**有L1、L2、L3、L4、L5 五个灯和 S1、S2、S3、S4 四个按钮，按钮区有 SB0、SB1 两个按钮。要求按下按钮 SB0 后，L5 指示灯亮，此时轮流按 S1、S2、S3、S4 四个按钮，相对应的 L1、L2、L3、L4 四个指示灯分别亮 4 秒后熄灭。按下复位按钮 SB1 后L5 熄灭停止上述过程的运行。

2) IO 分配

输入			输出		
动作演示区	S1	X2	动作演示区	L1	Y0
	S2	X3		L2	Y1
	S3	X4		L3	Y2
	S4	X5		L4	Y3
按钮区	SB0	X0		L5	Y4
	SB1	X1			

3)



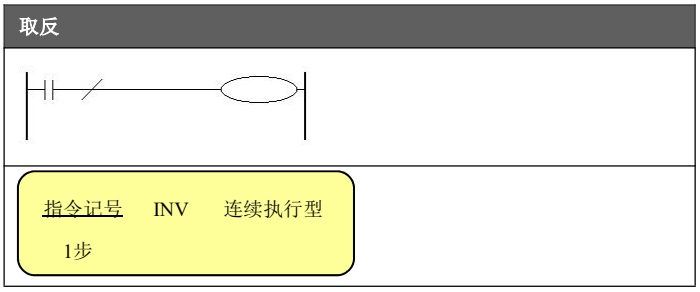
- 按下 SB0, 启动 L5 指示灯， 按下 SB1,L5 熄灭。
- L5 亮后，分别按下 S1,S2,S3,S4, L1L2L3L4 分别亮 4S 后熄灭。

7.9 [INV]指令

7.9.1 概要

INV 指令是将INV 指令执行之前的运算结果反转的指令。不需要指定软元件号。

1) 指令格式



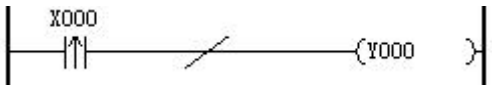
2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
INV	无对象软元件																			

7.9.2 功能和动作说明

1) INV 指令(运算结果的反转)

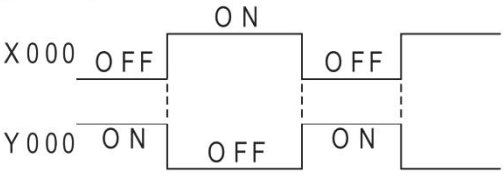
梯形图程序



指令表程序

```
0   LDP   X000
2   INV
3   OUT   Y000
```

时序图



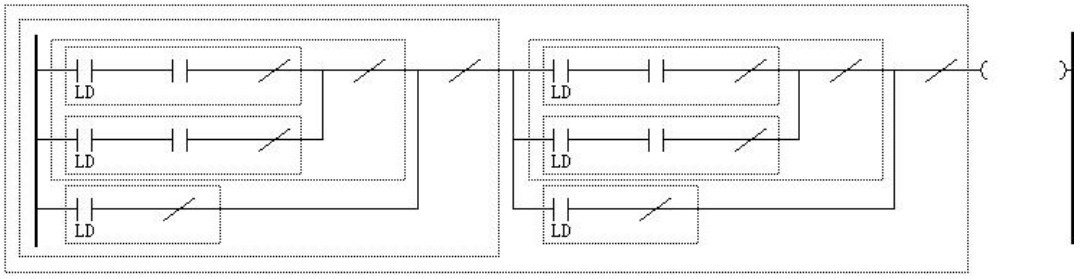
执行 INV 指令前的运算结果	执行 INV 指令后的运算结果
OFF	ON
ON	OFF

上图中，X000 为 OFF 时，Y000 为ON，如果 X000 为ON 时， 则 Y000 为 OFF。INV 指令可以在与串联触点指令(AND、ANI、ANDP、ANDF 指令)相同的位置处编程。

不能像指令表上的 LD、LDI、LDP、LDF 那样与母线连接，也不能像 OR、ORI、ORP、ORF 指令那样独立地与触点指令并联使用。

2) INV 指令的动作范围

在包含 ORB 指令、ANB 指令的复杂的回路中编写INV 指令时，INV 指令的动作范围如下图所示。



INV 指令的功能，是将INV 指令执行前存在的 LD、LDI、LDP、LDF 指令以后的运算结果反转。

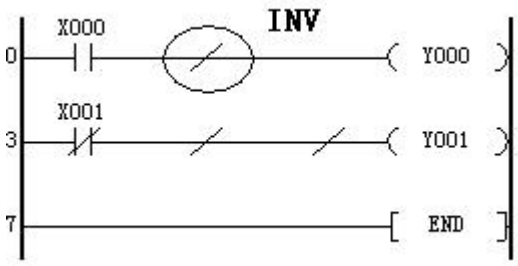
因此，如上图所示，在 ORB 指令、ANB 指令中编程时，从各自的 INV 指令的位置上见到的 LD、LDI、LDP、LDF 以后的块作为 INV 运算的对象。

7.9.3 注意要点

1. 该指令不带参数，前边必须跟随触点，运行取反操作。

7.9.4 程序应用

梯形图程序



指令表程序

```
0  LD   X000
1  INV
2  OUT Y000
3  LDI X001
4  INV
5  INV
6  OUT Y001
7  END
```

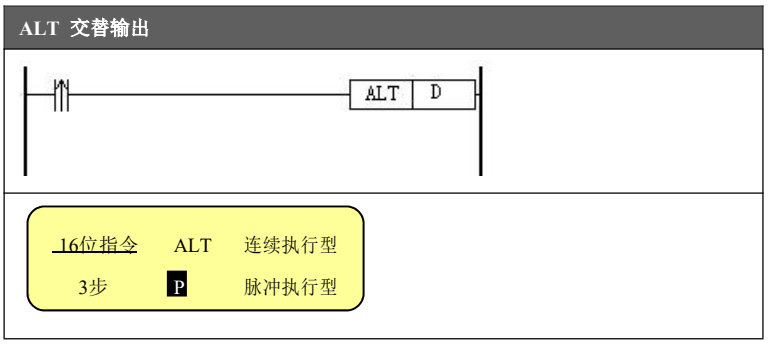
当 X000 为OFF 时 Y0 导通，当 X001 为 OFF 时 Y1 导通

7.10 [ALT]交替输出指令

7.10.1 概要

输入为 ON 时，使位软元件反转(ON-OFF)用的指令。

1) 指令格式



2) 设定数据

操作数	作用	类型
D	交替输出的位软元件编号	位

3) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D		•	•			•														

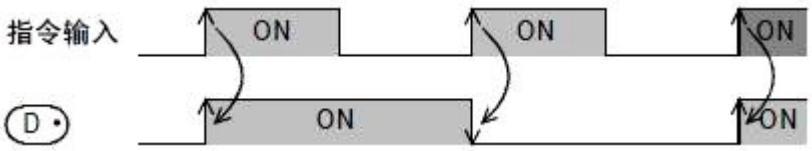
7.10.2 功能和动作说明

1) 16 位运算(ALT、ALTP）

- **交替输出(1 段)**

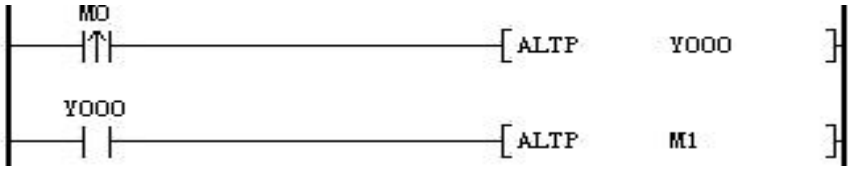


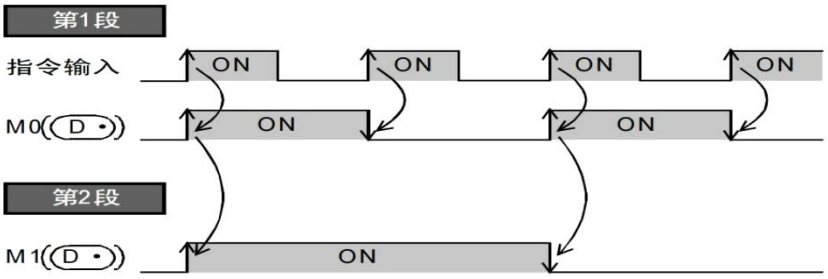
指令输入每次从 OFF 变为 ON 时，D 中指定的位软元件就执行ON-OFF 的反转。



- **分频输出(采用交替输出(2 段)**

组合多个 ALTP 指令后，可以实现使用多个段的分频输出。





组合多个 ALTP 指令后，可以实现使用多个段的分频输出。

7.10.3 注意要点

1. 使用 ALT 指令(连续执行型)时交替输出(1 段)

使 用 ALT 指令编程时，每个运算周期都执行反转动作。

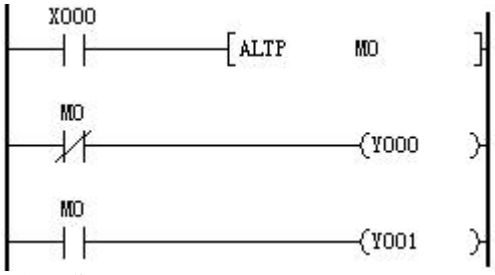
希望通过指令的ON/OFF 使其反转动作时，请使用 ALTP 指令(脉冲执行型)、或是 LDP 指令触点等(脉冲执行型)。

7.10.4 程序应用

1) 通过 1 个输入启动/停止

① 按下按键X000 后，启动输出Y001 的动作。

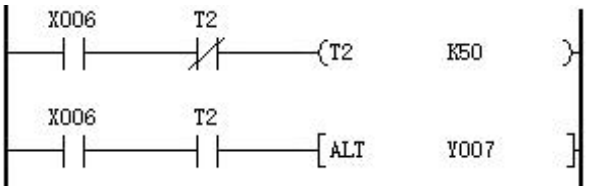
② 再次按下按键X000 后，停止输出Y000 的动作。

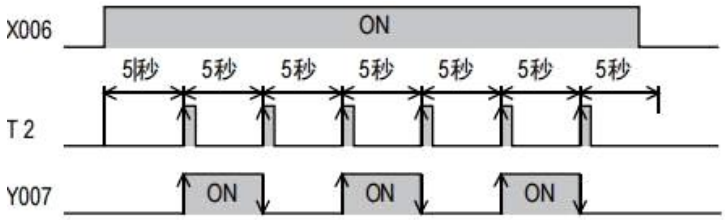


2) 闪烁动作

① 输入 X006 为 ON 时，定时器T2 的触点每隔 5 秒瞬间动作一次。

② T2 的触点，每次ON 时都使输出 Y007 交替 ON/OFF。



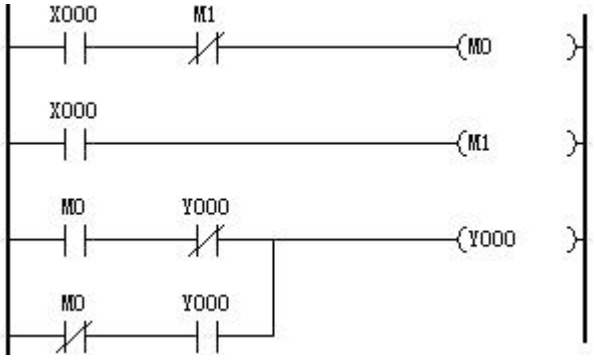


3) 使用辅助继电器(M)的交替输出动作(和ALT 指令相同的动作)

在下面的梯形图中，是使用了基本指令及辅助继电器(M)，实现与 ALT 指令相同交替动作的例子。

① X000 为 ON 时，M0 仅维持一次运算周期为ON。

② 当 M0 初次为 ON 时，Y000 自保持，当第 2 次为 ON 时，解除自保持。



7.11 [PLS]、[PLF]指令

7.11.1 概要

使用 PLS 指令后，仅在驱动输入ON 以后的 1 个运算周期内，对象软元件动作。使用 PLF 指令后，仅在驱动输入OFF 以后的 1 个运算周期内，对象软元件动作。

1) 指令格式

PLS 脉冲、PLF 下降沿脉冲	
	
指令记号 PLS 连续执行型 2 步	指令记号 PLF 连续执行型 2 步

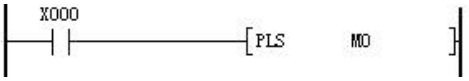
2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
PLS		•	•																	
PLF		•	•																	

7.11.2 功能和动作说明

1) PLS 指令(上升沿输出)

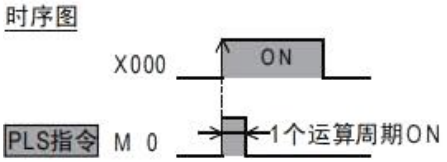
梯形图程序



指令表程序

```
LD    X000
PLS   M0
```

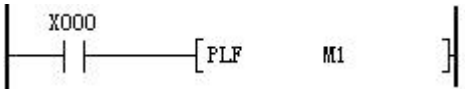
时序图



在上图中，X000 从 OFF 变为 ON 时，只有一个运算周期的 M0 为 ON。

2) PLF 指令(下降沿输出)

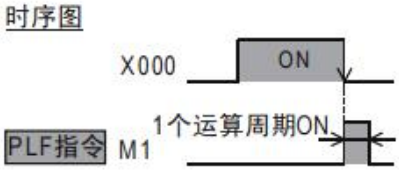
梯形图程序



指令表程序

```
LD    X000
PLF   M1
```

时序图

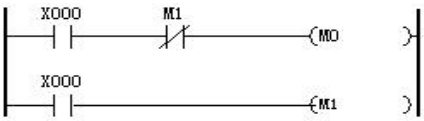


在上图中，X000 从 ON 变为 OFF 时，只有一个运算周期的 M1 为 ON。

3) 输出驱动实例

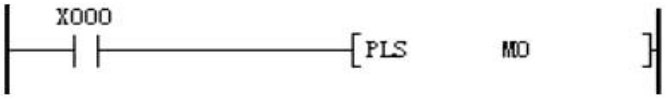
与下面的回路的动作相同。

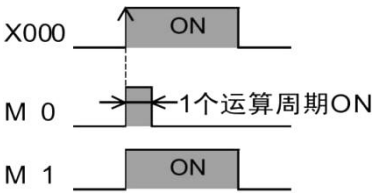
《OUT 指令》



=

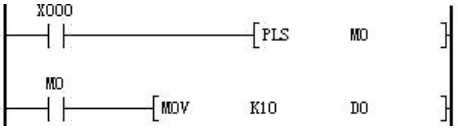
《脉冲指令》



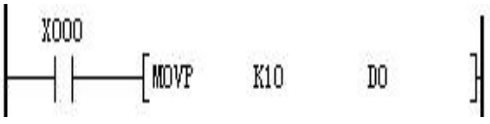


无论哪种情况，当 X000 从OFF 变为 ON 时，都只有一个运算周期的 M0 为ON。

《脉冲指令》



《应用指令的脉冲指令》



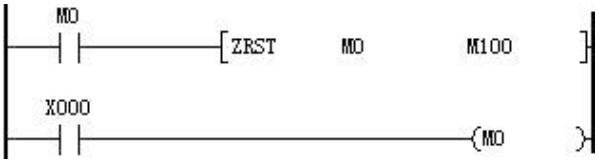
无论哪种情况，当 X000 从OFF 变为 ON 时，MOV 指令只执行一次。

7.11.3 注意要点

同时驱动 ZRST 指令和 PLS 指令时的注意事项

ZRST 指令会将对象软元件的 PLS、PLF 指令用的上一次状态以及 T、C 复位状态也进行复位。只启动 1 次 M0 时，请按以下方法编程。

梯形图程序



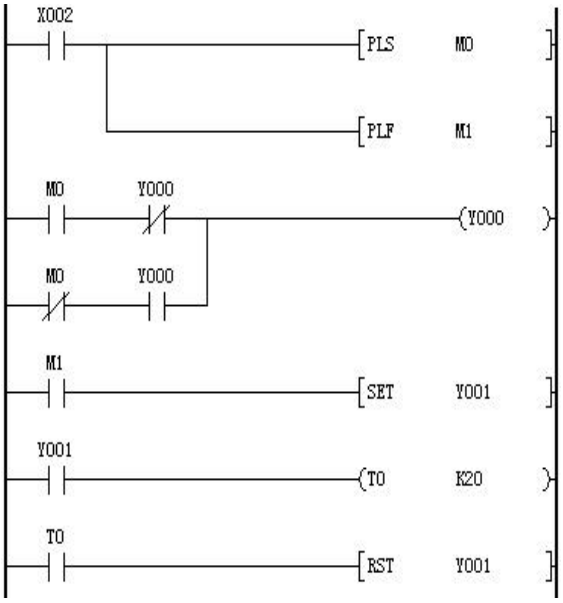
7.11.4 程序应用

1) **控制要求：**第一次按下按钮 S1，指示灯 L1 立即亮。放开按钮 S1，指示灯 L2 立即亮，2 秒后， L2 熄灭。再按一次 S1，指示灯 L1 立即熄，放开按钮 S1，指示灯 L2 立即亮，2 秒后，L2 熄灭。再按一次 S1，指示灯又亮，重复上述过程。

2) IO 分配

输入			输出		
动作演示区	S1	X2	动作演示区	L1	Y0
				L2	Y1

3) 梯形图程序



- 按下按钮 S1，按下时 M0 输出一个扫描周期的脉冲，放开时M1 输出一个扫描周期的脉冲。
- 第一扫描周期，M0 为 ON，接通 Y0 ；第二扫描周期，Y0 为 ON 自锁；再次按下 S1，M0为ON，Y0 断开。
- M1 脉冲启动 Y1。
- 定时 2S。
- 2S 时间到，Y1 断开。

7.12 [SET]、[RST]指令

7.12.1 概要

- SET 位软元件的置位(动作保持)

SET 指令是当指令输入为 ON 时，对输出继电器(Y)、辅助继电器(M)、状态继电器(S)置 ON 的指令。此外，即使指令输入为 OFF，通过 SET 指令置 ON 的软元件保持 ON 动作。

- RST 位软元件的复位(RST 指令[解除动作保持])

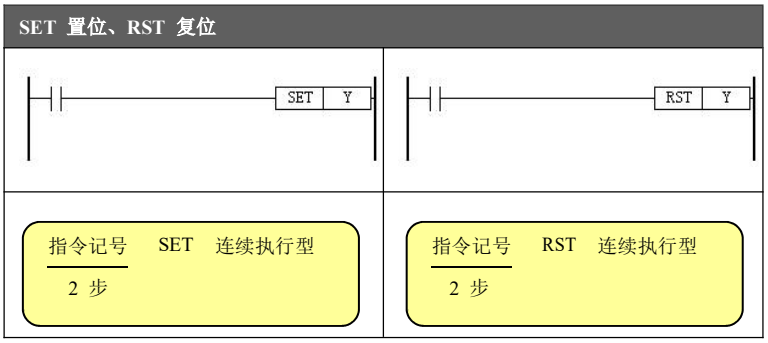
RST 指令是对输出继电器(Y)、辅助继电器(M)、状态继电器(S)、定时器(T)触点、计数器(C)触点进行复位的指令。可以对用 SET 指令置 ON的软元件进行复位(OFF)处理。

- 字软元件的当前值清除(RST 指令[当前值及寄存器的清除]）

RST 指令是清除定时器(T)、计数器(C)的当前值数据及复位输出触点的指令。

- 此外，要将数据寄存器(D)、扩展寄存器(R)和变址寄存器(V)、(Z)的内容清零时，也可使用 RST 指令。
  (使用常数为K0 的MOV 传送指令也可以得到相同效果。)

1) **指令格式**



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
SET		•	•			•														
RST		•	•	•	•	•					•	•	•	•	•	•				

7.12.2 功能和动作说明

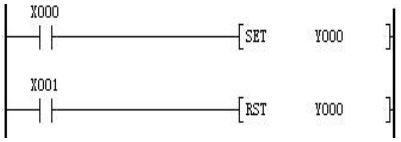
SET 指令是对输出继电器(Y)、辅助继电器(M)、状态继电器(S)进行置位的指令。RST 指令是对输出继电器(Y)、辅助继电器(M)、状态继电器(S)、定时器(T)触点、计数器(C)触点、定时器(T)、计数器(C)的当前值数据、数据寄存器(D)、扩展寄存器(R)和变址寄存器(V)、(Z)的内容进行复位的指令。

1) 使用位元软件时

并联的 SET 指令，可以连续使用多次。

下面的程序举例中，SET Y000 后接着RST Y000 的程序就是这个情况。

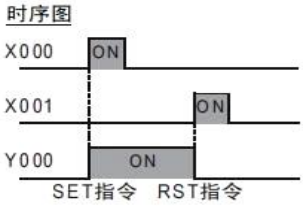
梯形图程序



指令表程序

```
LD   X000
SET Y000
LD   X001
RST Y000
```

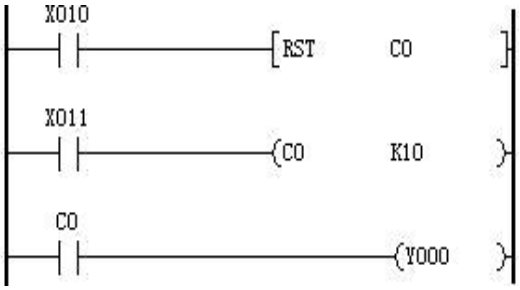
时序图



2) 使用字软元件(定时器、计数器)时

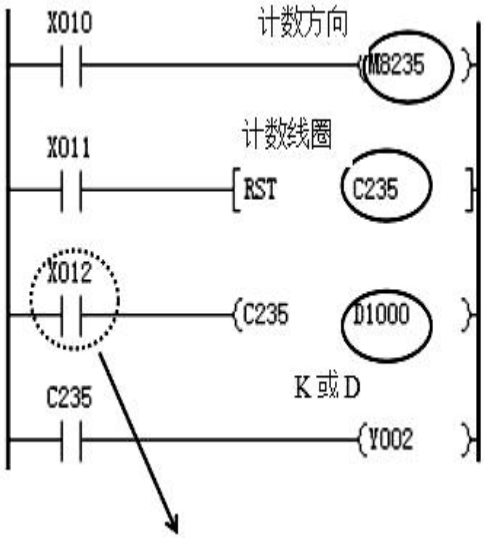
使用 RST 指令复位计数器和定时器的当前值和输出触点。

① 内部计数器的编程



由 C0 对 X011 的 OFF→ON 的次数增计数，计数结果达到设定值 K10 的时候，输出触点 C0 动作。此后， 即使 X011 从 OFF 变为 ON，计数器的当前值也不改变，输出触点也保持动作。 为了清除这些，恢复输出触点，要使 X010 为 ON。OUT C 指令后面，需要指定常数 K 或间接设定用的数据寄存器编号。停电保持(保持)用计数器的场合，即使停电也能保持当前值以及 输出触点的动作状态和复位状态

② 高速计数器的编程



C235～ C245 的单相单输入计数器中，使用特殊辅助继电器 M8235 ～ M8245 来指定计数方向。
X010: ON 时递减 X010: OFF 时递增
X011 为 ON 时,计数器 C235 的输出触点恢复，计数器的当前值也变为 0。
对于带复位输入的计数器(C241、C242、…)，其相应的复位输入为 ON 时，通过中断动作可实现同上相同的运行，所以不需要为此编程。
X012 为 ON 时， 对由计数器编号决定的计数输入
X000～X005 的ON/ OFF 次数进行计数。
对于带开始输入的计数器(C244、C245、…)，其相应的开始输入若不为 ON，则无法开始进行计数。
计数器的当前值增加，经过设定值(K 或 D 的内容)的时候，输出触点被置位，在减少的方向中经过的时候被复位。

对于驱动高速计数器计数线圈用的触点，
请使用高速计数执行时一直为 ON 的触点编程。

如果使用了分配为高速计数器用输入编号的输入继电器(X000～X003)将其作为计数线圈驱动时，有可能不能正确计数。

7.12.3 注意要点

1) 在同一运算周期内，对同一软元件执行 SET 和RST 指令时，会输出距 END 指令(程序的结束)近的那条指令的结果。

2) 使用 SET、 RST 指令时，应避免与 OUT 指令同时作用于相同的软元件。

3） 可以对于同一软元件，多次使用 SET、RST 指令，而且顺序也可随意，但软元件最终的输出状态受最后执行的指令决定。

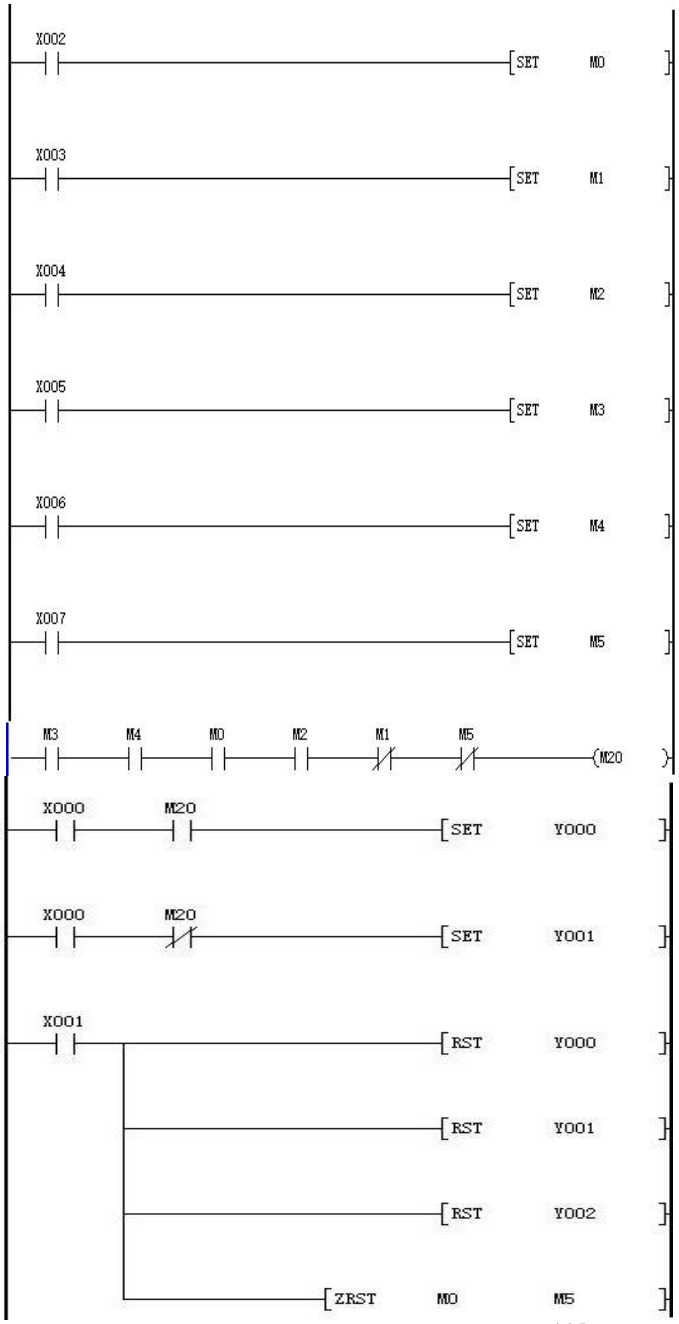
7.12.4 程序应用

1) **控制要求：**有SB0-SB5 共6 个按钮，分别输入数字 0、1、2、3、4、5。当按下数字按钮 3402 四个数字后（可以不要求按的顺序）再按下按钮区开锁按钮 SB6，板上动作演示区 L1 灯亮，表示密码输入正确，锁开。多按、少按或按错在按下开锁键SB6 后，L2 灯按 1HZ 的频率闪烁报警，表示开锁失败。按下复位按钮SB7 可重复开锁动作。

2) I/O 分配

输入	SB0	SB1	SB2	SB3	SB4	SB5	SB6	SB7
	X2	X3	X4	X5	X6	X7	X0	X1
输出	L1	L2						
	Y0	Y1						

3) 梯形图程序



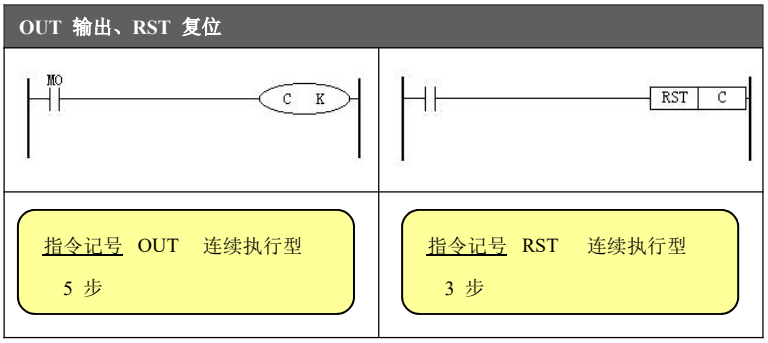
- 按按下 X2(S1),M0 置位为1。
- 按按下 X3(S2),M1 置位为1。
- 按按下 X4(S3),M2 置位为1。
- 按按下 X5(S4),M3 置位为1。
- 按按下 X6(S5),M4 置位为1。
- 按按下 X7(S6),M5 置位为1。
- 输写入 3042,则 M20 为ON。
- 输写入 正确，锁开。
- 写写入 错误，报警。
- 按按下停止按钮，复位。

7.13 对应计数器软元件C的[OUT],[RST]指令

7.13.1 概要

OUT 输出：计数线圈的驱动；RST 复位：输出触点的复位及当前值清零；

1) 指令格式

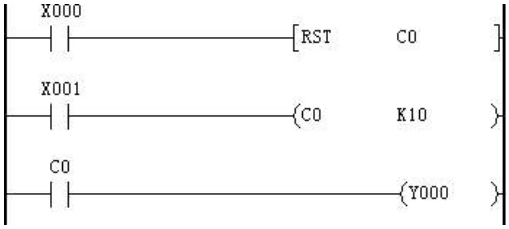


2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
OUT												•								
RST												•								

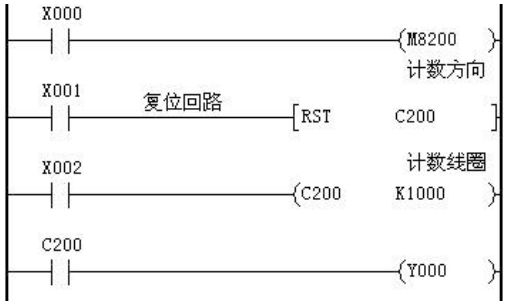
7.13.2 功能和动作说明

1) 利用[OUT],[RST]指令对计数器软元件 C 进行增计数



- 令 X000 为接通状态，可清除计数器当前值并使触点复位。
- 此后，当 X001 继续由 OFF→ON 变化时，计数器的当前值会继续累加，输出触点仍保持动作。
- 计数器 C0 对 X001 的 OFF→ON 次数进行增计数，当 C0 达到设定值 K10 时输出触点 C0 动作，即 Y000 动作。

2) 利用[OUT],[RST]指令对计数器软元件 C 进行增减计数



- 在 C200～C234 的单相单输入的计数器中， 采用特殊辅助继电器 M8200～M8234 指定计数的方向。
  X000:ON 时,C200 为减计数；X000:OFF 时，C200 为增计数。
- 令 X001 为接通状态，C200 输出触点复位, 当前值清零。
- 通过 X002 触点的关断/闭合进行对 C200 计数器的计数，从而得到 X002 端子上的脉冲序列计数值。
- 当 C200 当前数值≥1000 时，C200 触点得电 (Y000 = ON)，反之触点失电(Y000 = OFF)。

7.13.3 注意要点

1. C0-C199 是 16 位增计数，当使用 32 位单相单输入的计数器 C200～C234 增减计数时，对应的特殊辅助继电器 M8200～M8234 指定计数的方向。

7.13.4 程序应用

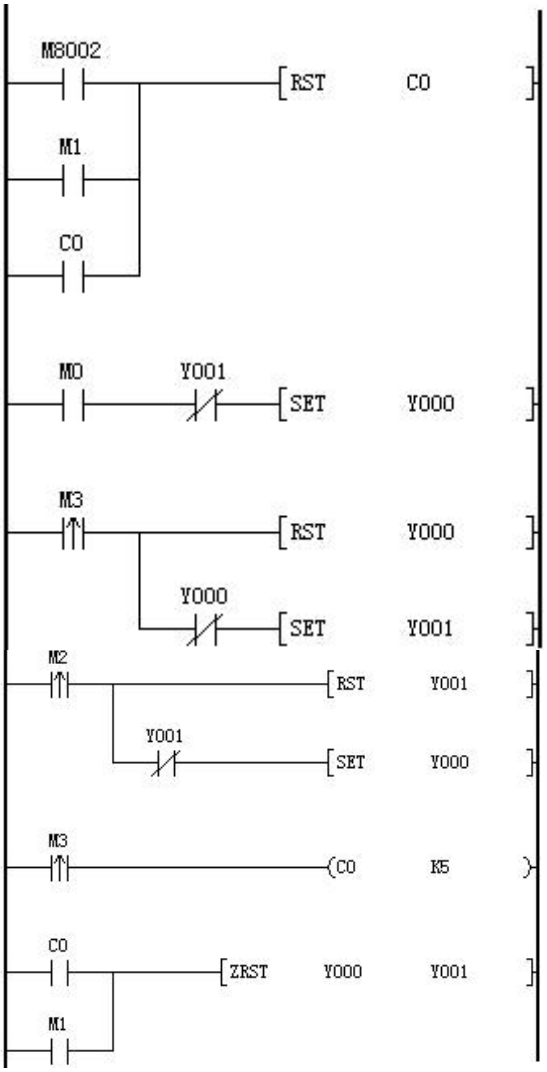
1) 控制要求:

按下启动按钮S1，动作演示板上的 LED 条开始上行流转，碰到 SQ4 后动作演示板上的 LED 条立即下行流转，碰到SQ1 后动作演示板上的 LED 条又立即上行，当动作演示板上的 LED 条上下移动 5 次后自动停止。系统运行过程中，按下 S2 可立即停止运行。下次启动时，计数器重新开始计数。

2) I/O 分配

输入		输出	
S1	M0	升	Y0
S2	M1	降	Y1
SQ1	M2		
SQ4	M3		

3) 梯形图程序



- PLC启动时或按下停止按钮或有5次C0复位
- 按下启动按钮，上升。
- 停止上升，下降。
- 停止下降，上升。
- 统计上下升的次数。
- 有 5 次或按下停止按钮，电机停止工作。

7.14 [NOP],[END]指令

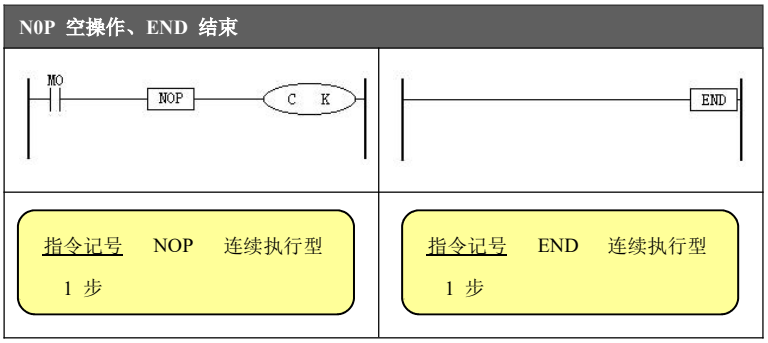
7.14.1 概要

- NOP 指令为空操作的指令。

将程序全部清除时，所有指令都成为 NOP。在一般的指令和指令之间加入 NOP 时，可编程控制器会无视其存在而继续运行。如在程序的中间加入 NOP，当需要更改、增加程序的时候，只需要很小地变动步编号就能实现，但是要求程序有余量。此外，若将已经写入的指令换成 NOP 指令，则回路会发生变化，请务必注意。

- END 指令为表示程序结束的指令。

1) 指令格式



2) 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
NOP	无对象软元件																			
END	无对象软元件																			

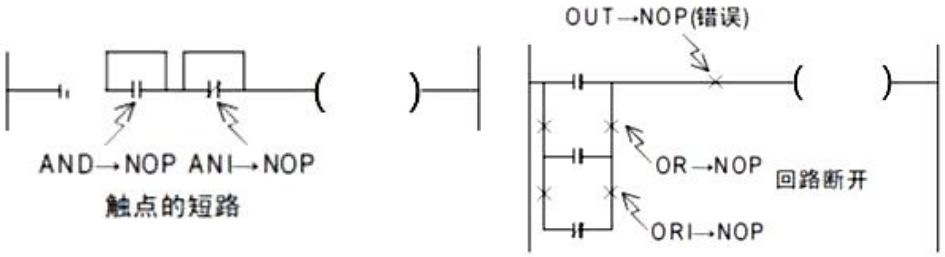
7.14.2 功能和动作说明

1) NOP 指令(空操作)

NOP 指令为空操作的指令。

写在程序中时，可编程控制器会无视其存在而继续运行。

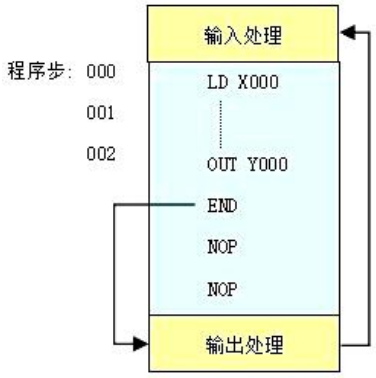
更改现有的程序，改写成 NOP 指令时，等同于执行删除指令的操作。



2) END 指令(程序结束以及输入输出处理和返回 0 步)

可编程控制器重复执行[输入处理]→[执行程序]→[输出处理]，若在程序的中间写入 END 指令，则不执行后面剩余的程序步，而直接进行输出处理。

在程序的最后写 END 指令的时候，可编程控制器会执行到程序的最后一步，然后才执行输出处理。此外，第一次执行开始 RUN 时，是从 END 指令开始执行的。执行 END 指令时，也刷新看门狗定时器(检查运算周期是否过长的定时器)。



PLC 反复进行输入处理、程序执行和输出处理。程序的最后必须以 END 指令结束，即使 END 以后仍有程序，也不再执行，而直接进行输出处理。

此外，RUN 开始时首次执行，从执行 END 指令开始。

执 行 END 指令时，也刷新监视定时器（检查扫描周期是否过长）。

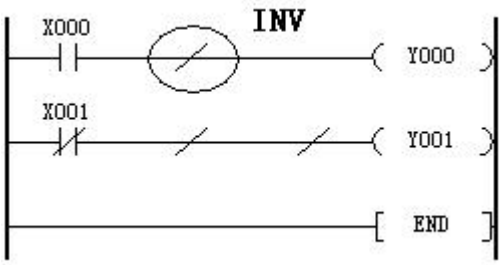
7.14.3 注意要点

1. 程序中间请勿写入END 指令。

2.通过编程工具传送时，END 指令以后都成为 NOP 指令(空操作)。

7.14.4 程序应用

梯形图程序



指令表程序

```
0    LD    X000
1    INV
2    OUT   Y000
3    LDI   X001
4    INV
5    INV
6    OUT   Y001
7    END
```

8. 传送·比较指令

在本章中，详细介绍了数据传送和数据比较等基本应用指令。

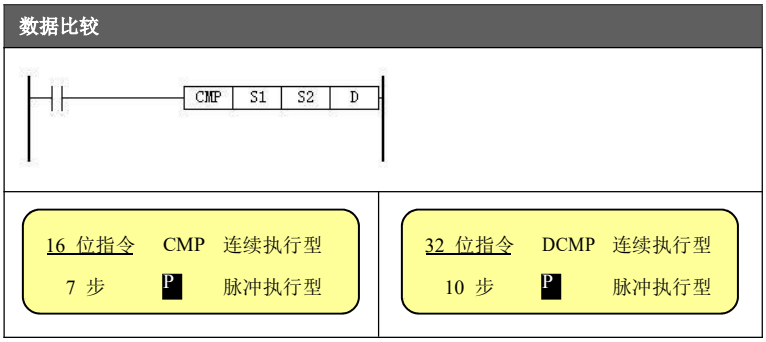
助记符、名称	功能	章节
CMP 数据比较	将指定的两个数据进行大小比较，并输出比较结果	8.1
ZCP 区间比较	将当前数据与 2 点指定数据进行大小比较，并输出比较结果	8.2
MOV 传送	将指定数据或软元件的值传送到其他软元件中	8.3
BMOV 成批传送	将指定区域软元件的值传送到其他软元件区域中	8.4
FMOV 多点传送	将指定数据或软元件内容传送到其他软元件区域中	8.5
CML 反向传送	以位为单位反转数据后进行传送(复制)的指令	8.6

8.1 [CMP]数据比较

8.1.1 概要

比较 2 个值，将其结果（大、一致、小）输出到位软元件中（3 点）。

1) 指令格式



2) 设定数据

操作数	作用	类型
S1	指定被比较的数据或软元件地址编号	16/32 位 BIN
S2	指定比较源的数据或软元件地址编号	16/32 位 BIN
D	指定比较输出结果的软元件地址编号	位

3) 对象软元件

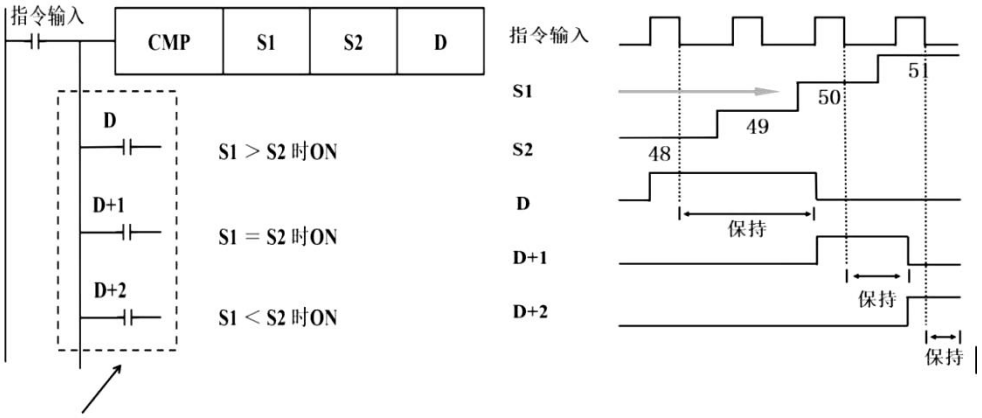
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D		•	•			•														

8.1.2 功能和动作说明

1) 16 位运算(CMP、CMPP)

对比较值 S1 和比较源 S2 的内容进行比较，根据其结果 (大、一致、小)，使D、D +1、D +2 其中一个为ON。

- S1 、S2 内的数据，作为 BIN(二进制)的值进行处理。
- 按照代数形式进行比较。例如：-10 < 1



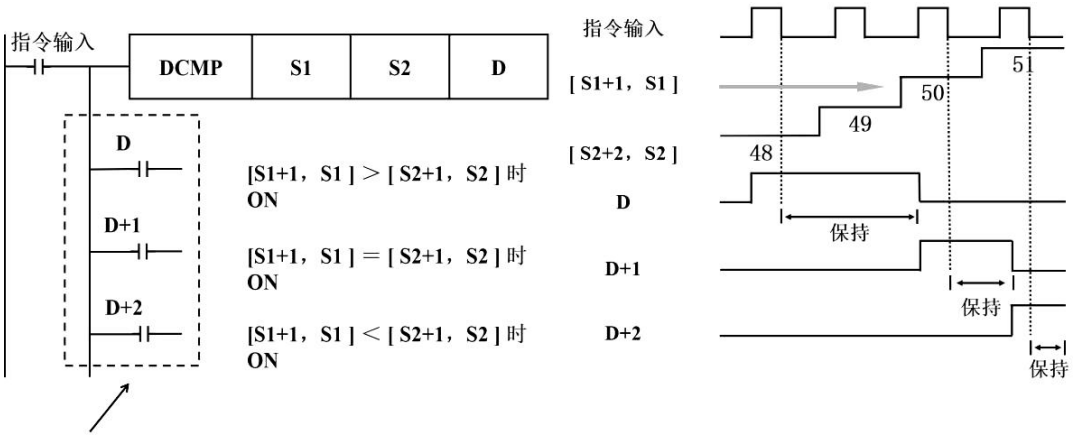
即使指令输入为OFF，CMP 指令不执行时，D ～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状

态。

2) 32 位运算（DCMP、DCMPP）

对比较值 [S1+1，S1]和比较源 [S2+1，S2]的内容进行比较，根据其结果(大，一致，小)，使 D、D +1、D+2 其中一个为 ON。

- [S1+1，S1]与 [S2+1，S2] 中的数据作为 BIN(二进制)的值进行处理。
- 按代数形式进行大小的比较，例如：-32767＜1



即使指令输入为OFF，DCMP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

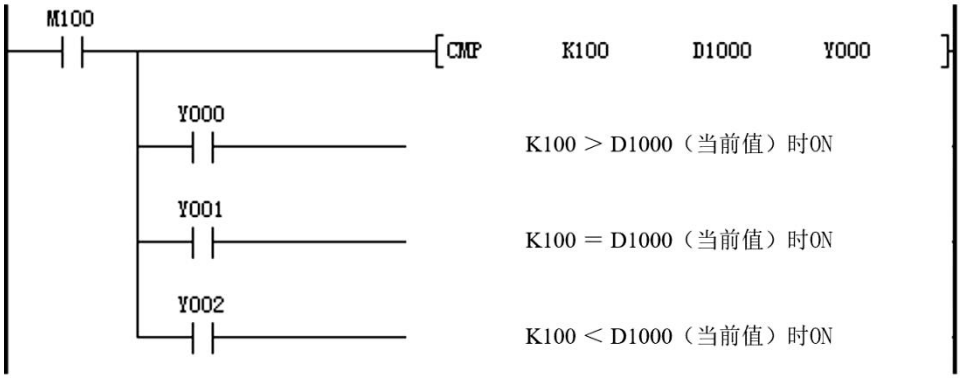
8.1.3 注意要点

1、软元件的占用数

占用以 D 中设置的软元件为起始的三个软元件。注意不要与其他控制中使用的软元件重复。

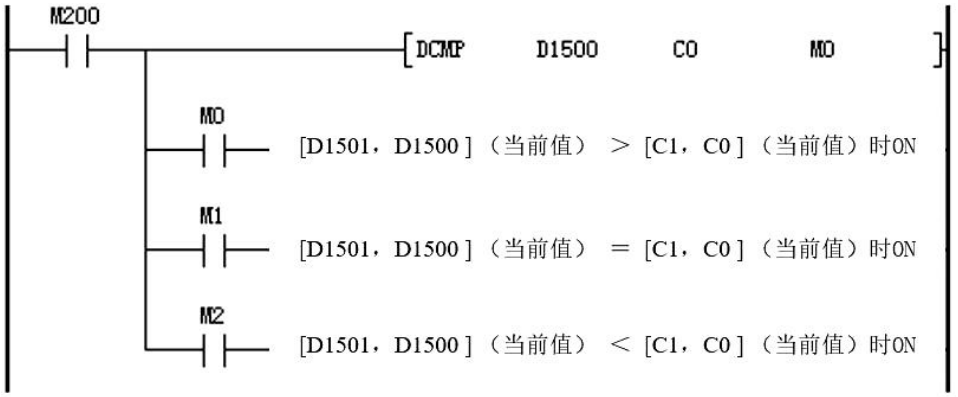
8.1.4 程序应用

1) 16 位运算(CMP、CMPP)



注意：指令中占用 Y0、Y1、Y2 三个元件

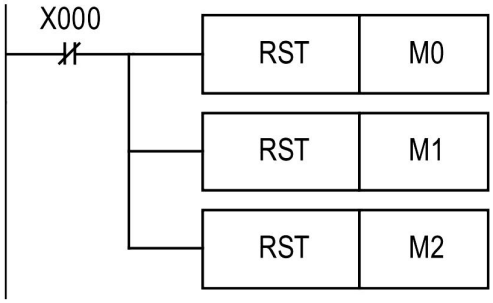
2) 32 位运算（DCMP、DCMPP）



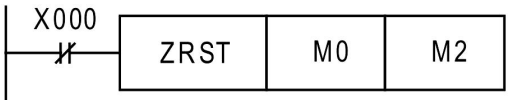
注意：指令中占用 M0、M1、M2 三个元件

想在不执行指令时清除比较结果，可以使用以下指令：

① RST 指令



② **ZRST 指令**



对 M0~M2 进行复位

8.2[ZCP]区间比较

8.2.1 概要

针对 2 个值(区间)，将与比较源的值比较得出的结果(小于、等于（区域内）、大于)输出到位软元件(3 点) 中。

1. 指令格式

区域比较



16 位指令	ZCP	连续执行型
9 步	P	脉冲执行型

32 位指令	DZCP	连续执行型
13 步	P	脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	16/32 位 BIN
S2	指定比较基准上限的数据或软元件地址编号	16/32 位 BIN
S3	指定比较数据或软元件地址编号	16/32 位 BIN
D	指定比较输出结果的软元件地址编号	位

3. 对象软元件

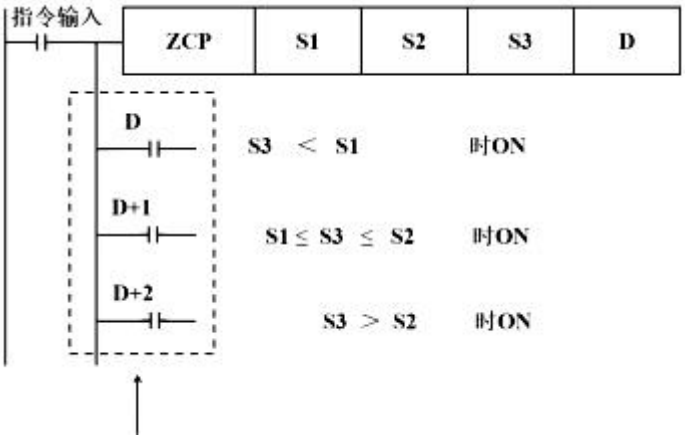
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D		•	•			•														

8.2.2 功能和动作说明

1. 16 位运算（ZCP、ZCPP）

将比较源 S3 的内容与比较基准下限 S1 和比较基准上限 S2 进行比较，根据其结果(小、区域内、大)， 将 D 、D+1、D+2 的其中一个置 ON。

- S1 、S2 内的数据，作为 BIN(二进制)的值进行处理。
- 按照代数形式进行比较。例如：-10 < 1 < 10



即使指令输入为OFF，ZCP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

2. 32 位运算（DZCP、DZCPP）

将比较源 [S3+1，S3] 的内容与比较基准下限 [S1+1，S1] 和比较基准上限 [S2+1，S2] 进行比较，根据其结果(小、区域内、大)，将 D 、D+1、D+2 的其中一个置 ON。

- [S1+1，S1]与 [S2+1，S2] 中的数据作为 BIN(二进制)的值进行处理。
- 按代数形式进行大小的比较，例如：-32767＜1＜102455



即使指令输入为 OFF，DZCP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

8.2.3 注意要点

1. 软元件的占用数

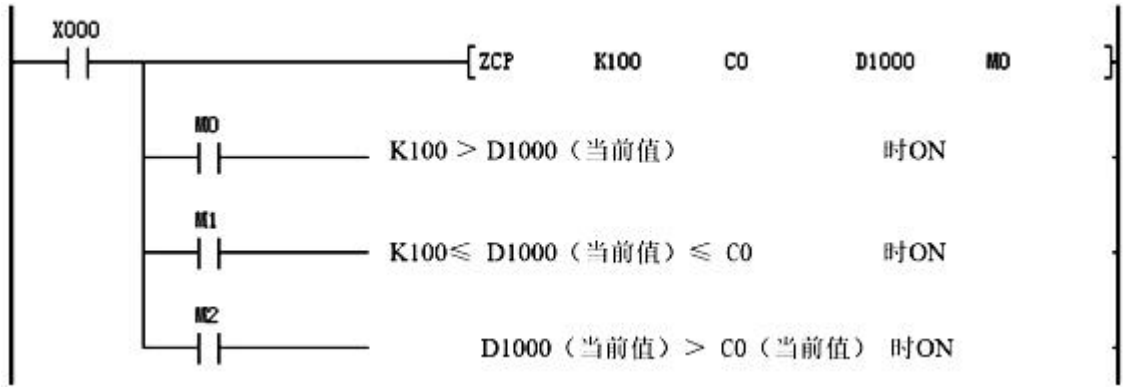
占用以 D 中设置的软元件为起始的三个软元件。注意不要与其他控制中使用的软元件重复。

2. 比较基准上下限

比较基准下限 S1 的值需要比比较基准上限 S2 的值小。

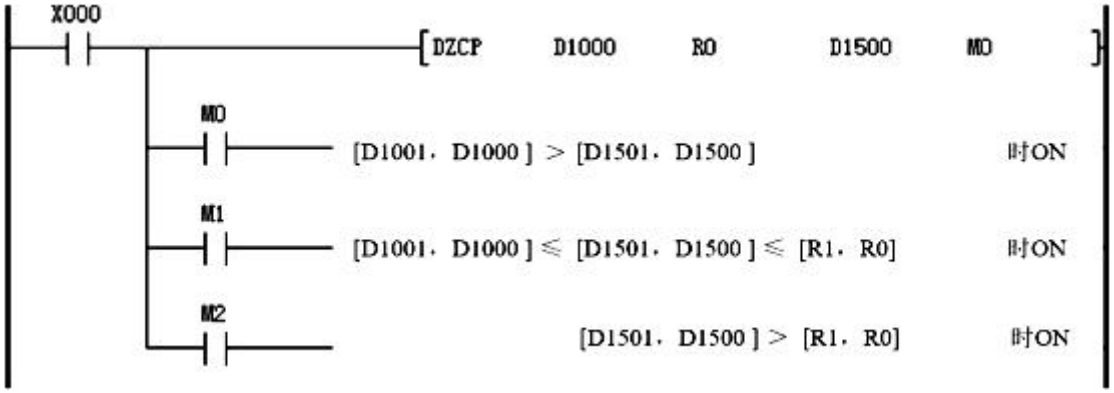
8.2.4 程序应用

1. 16 位运算（ZCP、ZCPP）



注意：指令中占用 M0、M1、M2 三个元件

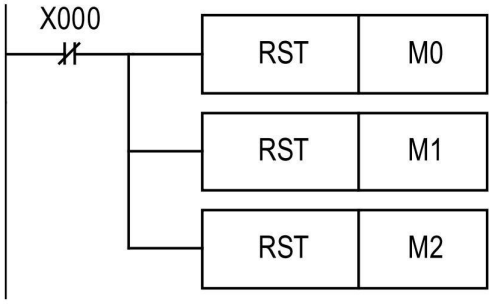
2. 32 位运算（DZCP、DZCPP）



注意：指令中占用 M0、M1、M2 三个元件

想在不执行指令时清除比较结果，可以使用以下指令：

1) RST 指令



2) ZRST 指令



对 M0~M2 进行复位

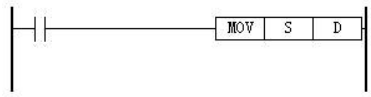
8.3 [MOV]传送

8.3.1 概要

使指定数据或软元件值传送到其他软元件中。

1. 指令格式

传送



16 位指令 MOV 连续执行型	32 位指令 DMOV 连续执行型
5 步 P 脉冲执行型	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定传送源数据或保存数据的软元件地址编号	16/32 位 BIN
D	指定传送目标软元件地址编号	16/32 位 BIN

3. 对象软元件

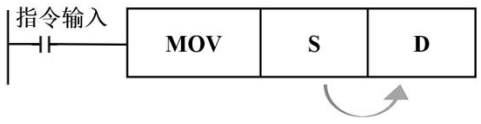
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•	•	•	

8.3.2 功能和动作说明

1. 16 位运算（MOV、MOVP）

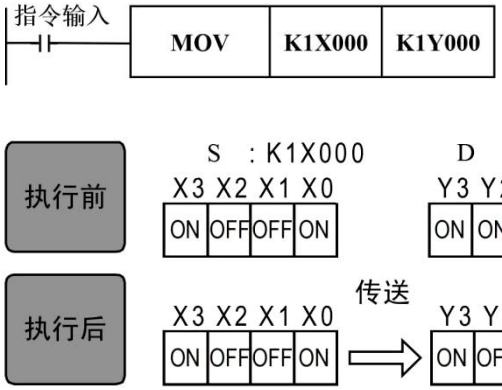
将传送源 S 的内容传送给传送目标 D。

- 指令输入为 OFF 时，传送目标 D 不会变化。
- 传送源 S 中指定了常数（K），会自动执行 BIN 转换



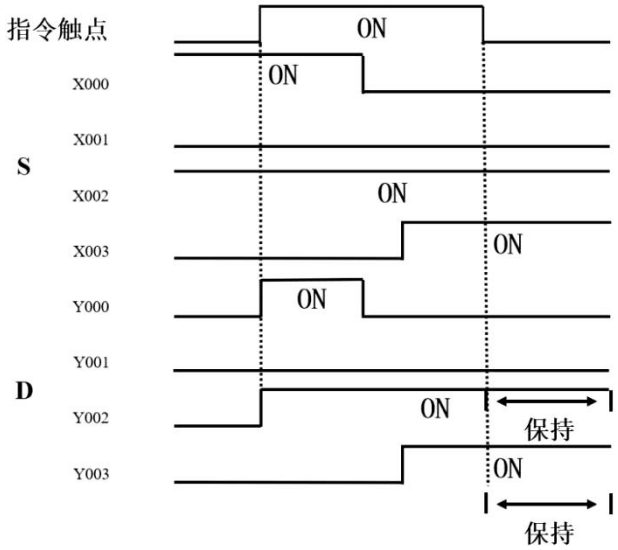
➢ **指定位软元件**

一次最多传送 16 个(4 的倍数)位软元件



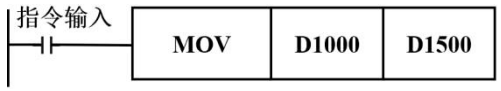
即使指令输入为OFF，MOV 指令不执行时，D 也会保持当指令输入从ON 变为 OFF 之前的状态。

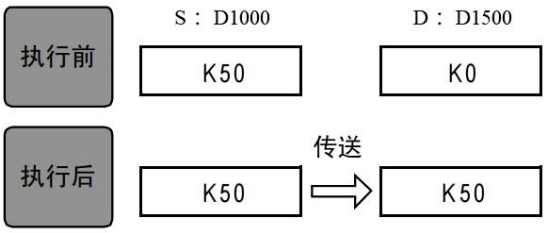
指令执行前后改变 S 中的数据，各个位元件的状态下图：

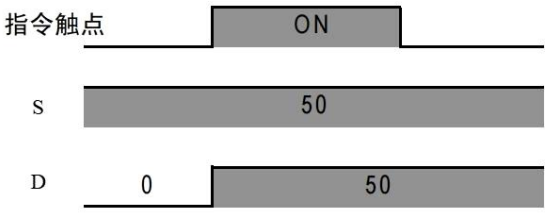


- **指定字软元件**

传送一个字软元件



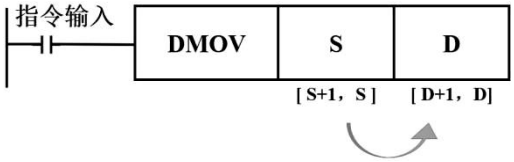




2. 32 位运算（DMOV、DMOVP）

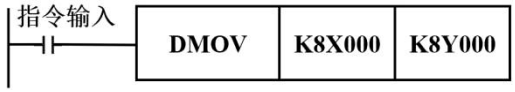
将传送源 [S+1，S] 的内容传送到传送目标[D +1，D]中。(字软元件为 2 点的传送)

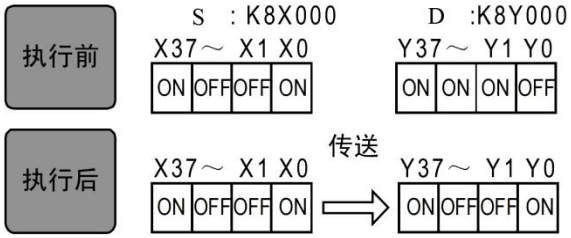
- 指令输入为 OFF 时，传送目标 D 不会变化。
- 传送源[S+1，S]中指定了常数（K），会自动执行 BIN 转换



- **指定位元件**

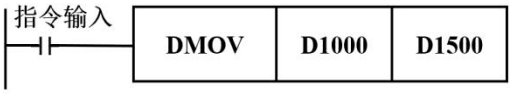
一次最多传送 32 个(4 的倍数)位软元件

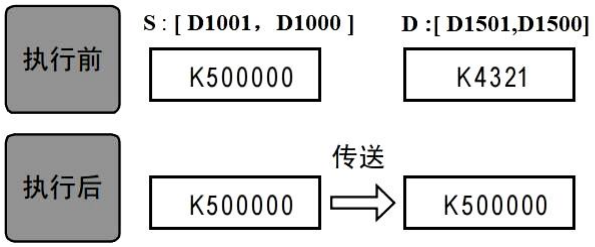


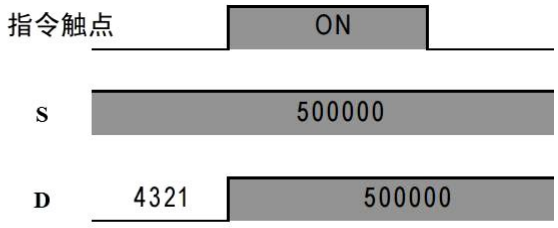


- **指定字元件**

传送两个字软元件。







8.3.3 注意要点

传送位元件时，应注意传送的位元件个数，以免导致数据丢失，例如：

K8X000 表示X0~X37（8 进制），共 32 个位元件；

K7X000 表示X0~X33，共 28 个位元件； K6X000 表示X0~X27，共 24 个位元件； K5X000 表示X0~X23，共 20 个位元件； 以此类推

8.3.4 程序应用

1. 读出计数器、定时器当前值



将C0的当前值传送给D1000

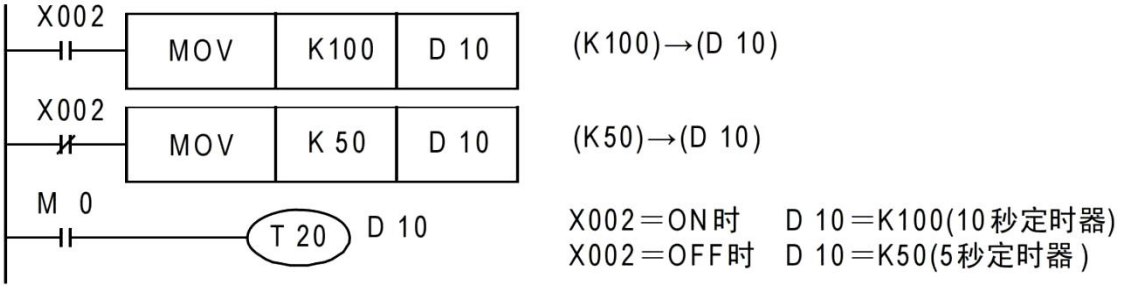


将T0的当前值传送给D1000

2. 间接指定定时器、计数器的值

通过开关(X002)的 ON/OFF 可以对定时器(或计数器)设定 2 个设定值。

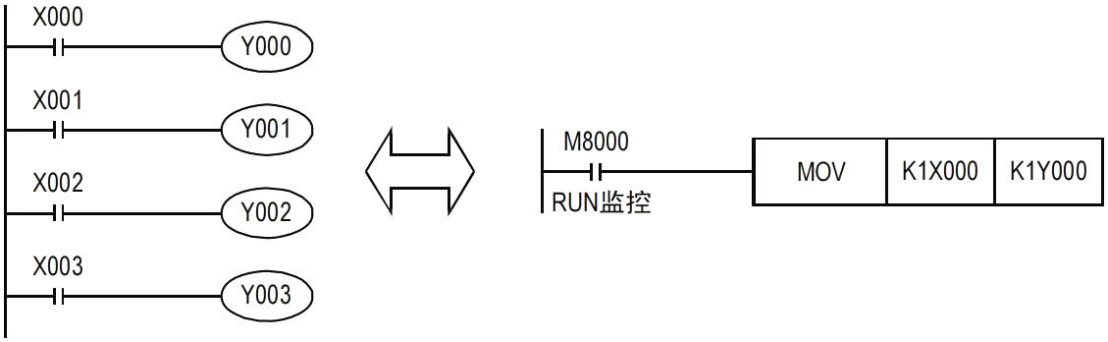
2 个以上时，需要使用多个开关。



计数器同理。

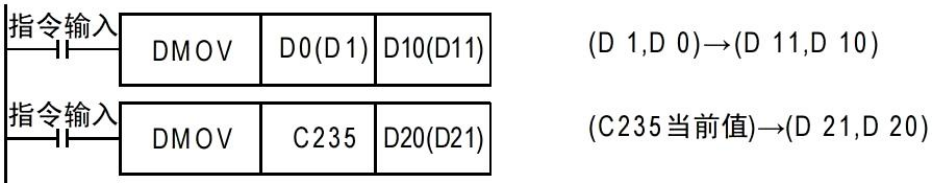
3. 位软元件的传送

可以使用 MOV 指令表现下述的顺控程序。



4. 32 位数据的传送

运算结果作为 32 位被输出的应用指令(MUL 等) 或者用 32 位的数值、或是 32 位的位软元件传送高速计数器当前值 (C235～C255) 时，必须使用 DMOV 指令。

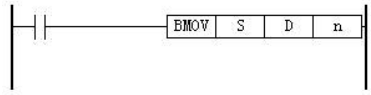


8.4 [BMOV]成批传送

8.4.1 概要

对指定点数的多个数据进行成批传送(复制)。

1. 指令格式

成批传送	
	
16 位指令 BMOV 连续执行型 7 步 P 脉冲执行型	

2. 设定数据

操作数	作用	类型
S	指定传送源软元件地址编号	16 位 BIN
D	指定传送目标软元件地址编号	16 位 BIN
n	指定传送点数的数值	16 位 BIN

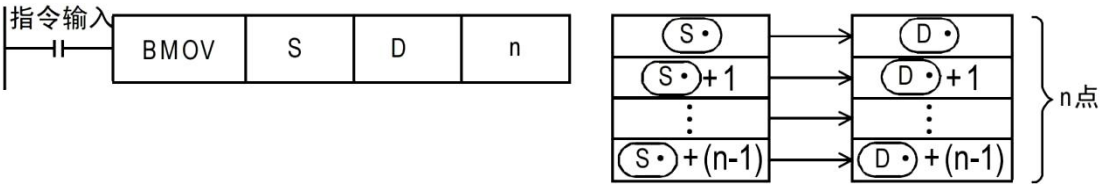
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•			•			
D								•	•	•	•	•	•	•			•			
n																		•	•	

8.4.2 功能和动作说明

将 S 开始的 n 点的数据成批传送到 D 开始的 n 点中。

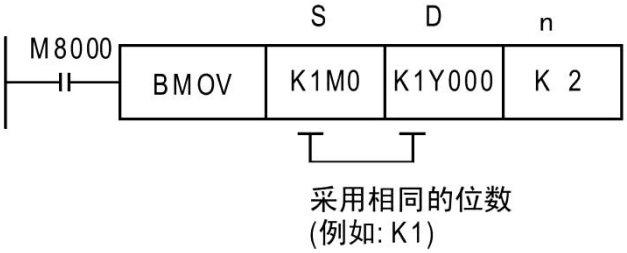
- 超出软元件编号范围时，会出错并报警。



8.4.3 注意要点

1. 传送位软元件

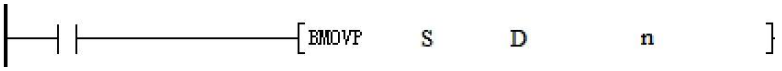
使用位软元件传送时，S 和 D 尽量采用相同的位数，以免造成数据丢失。



2. n 的设置范围

指令执行的 n（传送点数）最少为 1 个点，最多为 256 个点，传送点数设置过大会影响定时器的精确度， 建议将 n 设置小一些，或者使用脉冲执行型指令。

例如：



也可以使用上升沿触点



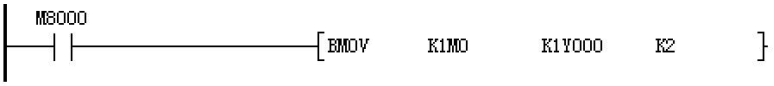
3. 地址的设置

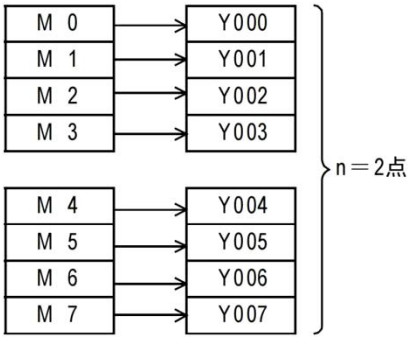
当指定的软元件编号加上传送点数后，所指向的软元件地址必须是连续有效的（C199 到 C200 这类情况除外）。

C 计数器的地址分为两类：C0~C199（16 位）和 C200~C255（32 位），二者地址在使用过程中不连续。

8.4.4 程序应用

传送多点位软元件时





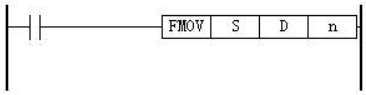
8.5 [FMOV]多点传送

8.5.1 概要

使指定数据或软元件内容传送到其他软元件区域中。

1. 指令格式

多点传送



16 位指令 FMOV 连续执行型	32 位指令 DFMOV 连续执行型
7 步 [P] 脉冲执行型	10 步 [P] 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定传送源数据或保存数据的软元件地址编号	16/32 位 BIN
D	指定传送目标软元件地址编号	16/32 位 BIN
n	指定传送点数的数值	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

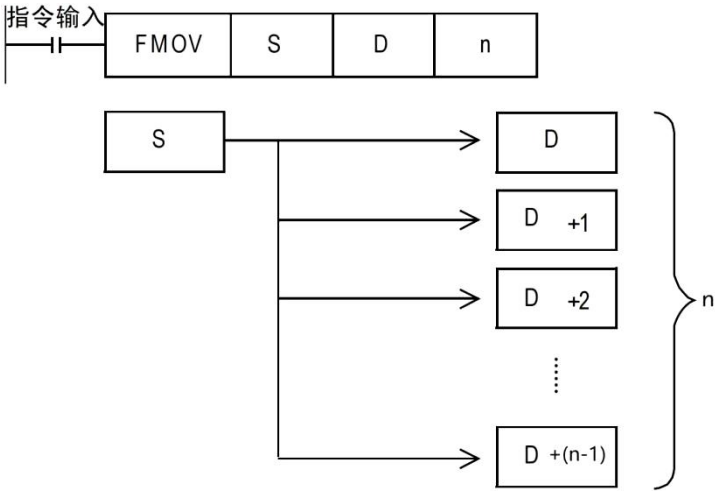
8.5.2 功能和动作说明

1. 16 位运算（FMOV、FMOVP）

将 S 的内容传送到以 D 起始的 n 点的软元件中。

- n 点的软元件内容都相同。
- 使用该指令时超出软元件编号范围，会出错并报警。
- 指令输入为 OFF 时，传送目标 D 不变化。
- 指令输入为 ON 时，传送源 S 的数据不变化。
- 传送源 S 中指定了常数(K)时，会自动执行 BIN 转换。

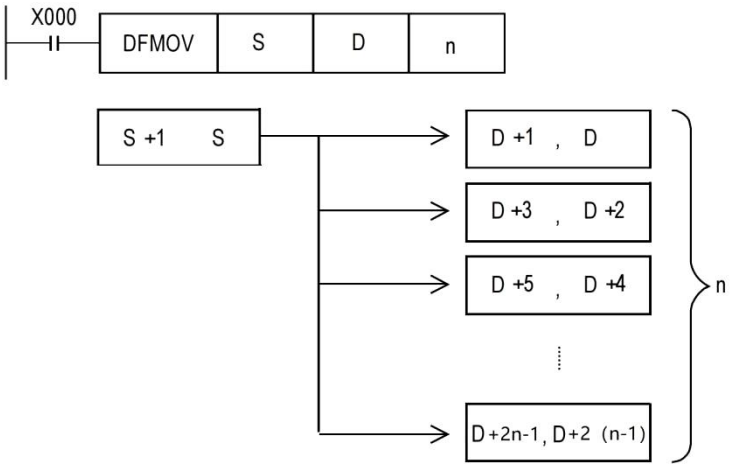
指定数据多次写入



2. 32 位运算（DFMOV、DFMOVP）

将[S+1，S]的内容传送到以[D+1，D]起始的 n 点的软元件中。

- n 点的 32 位软元件内容都相同。
- 使用该指令时超出软元件编号范围，会出错并报警。
- 指令输入为 OFF 时，传送目标[D+1，D]不变化。
- 指令输入为 ON 时，传送源[S+1，S]的数据不变化。
- 传送源[S+1，S]中指定了常数(K)时，会自动执行 BIN 转换。



8.5.3 注意要点

1. 传送位软元件

使用位软元件传送时，S 和 D 尽量采用相同的位数，以免造成数据丢失。

2. n 的设置范围

指令执行的 n（传送点数）最少为 1 个点，最多为 256 个点，传送点数设置过大会影响定时器的精确度， 建议将 n 设置小一些，或者使用脉冲执行型指令，也可以使用上升沿触点。

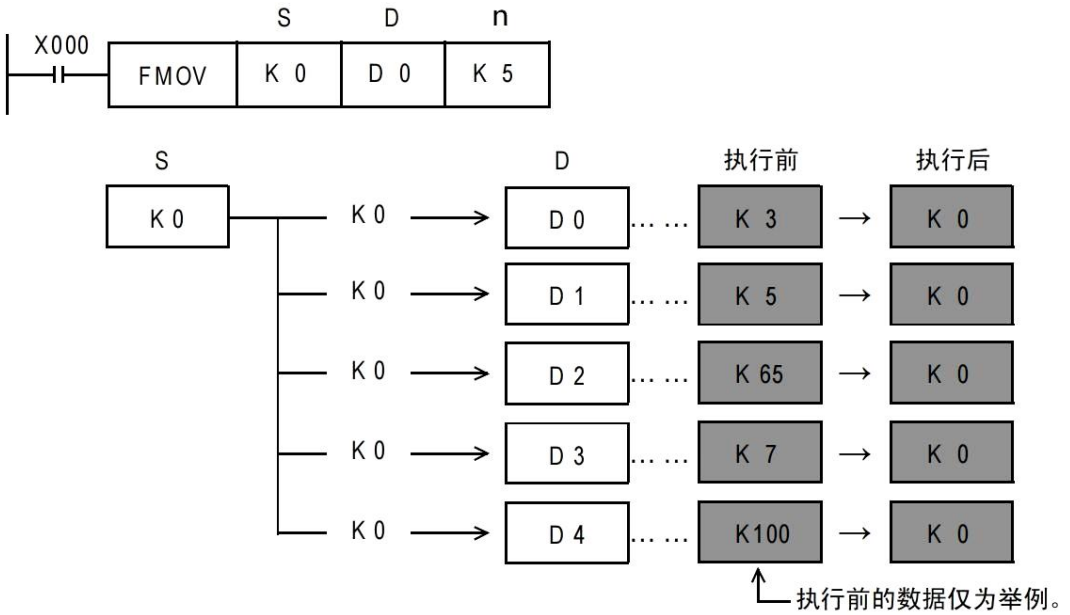
3. 地址的设置

当指定的软元件编号加上传送点数后，所指向的软元件地址必须是连续有效的（C199 到 C200 这类情况除外）。

C 计数器的地址分为两类：C0~C199（16 位）和 C200~C255（32 位），二者地址在使用过程中不连续。

8.5.4 程序应用

1. 指定数据多次写入

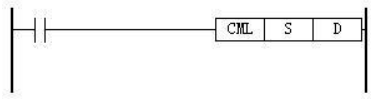


8.6 [CML]反向传送

8.6.1 概要

将指定数据或软元件内容进行反向传送。

1. 指令格式

反向传送	
	
16 位指令 CML 连续执行型 5 步 P 脉冲执行型	32 位指令 DCML 连续执行型 7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定源数据值或软元件地址编号	16/32 位 BIN
D	指定保存结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

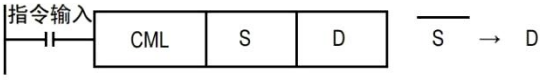
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

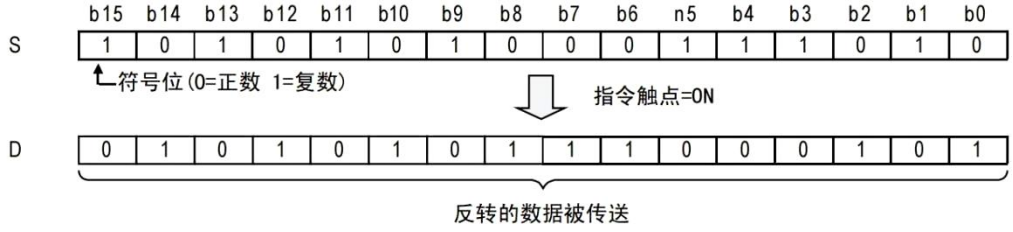
8.6.2 功能和动作说明

1. 16 位运算（CML、CMLP）

将 S 中指定的软元件的各位反转(0→1,1→0)后，传送至 D。

- 在S 中指定常数(K)时，会自动转换为 BIN。
- 希望将输出以逻辑反转输出时可以使用。

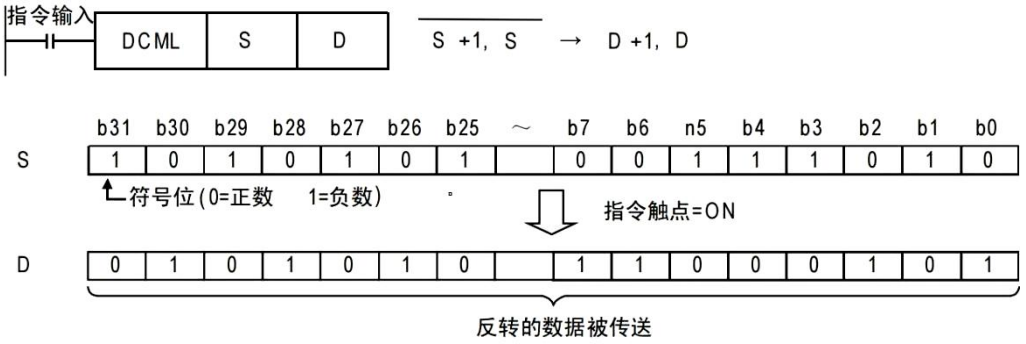




2. 32 位运算（DCML、DCMLP）

将[S+1，S]中指定的软元件的各位反转(0→1, 1→0)后，传送至[D+1，D]。

- 在[S+1，S]中指定常数(K)时，会自动转换为 BIN。
- 希望将输出以逻辑反转输出时可以使用。



8.6.3 注意要点

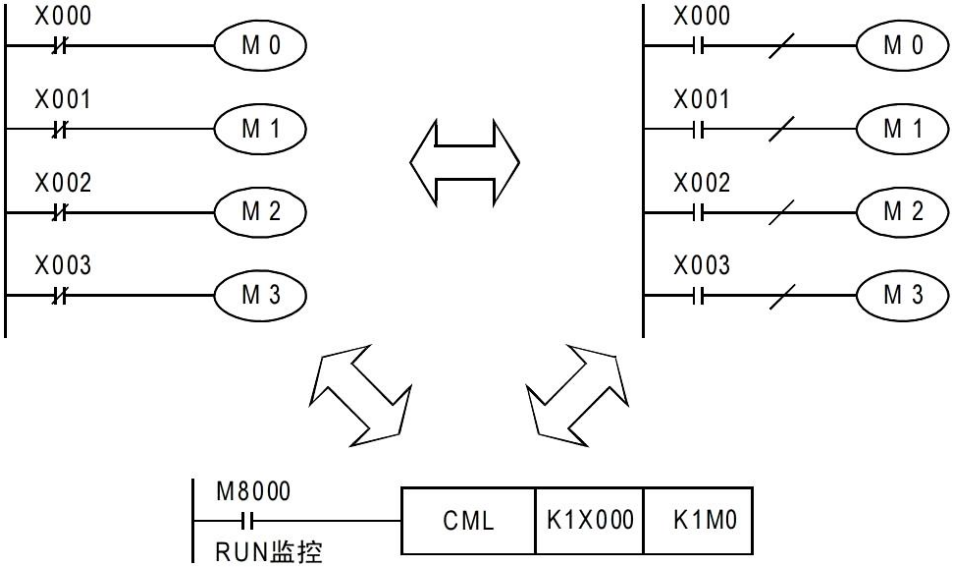
1. 传送位软元件

使用位软元件传送时，S 和 D 尽量采用相同的位数，以免造成数据丢失。

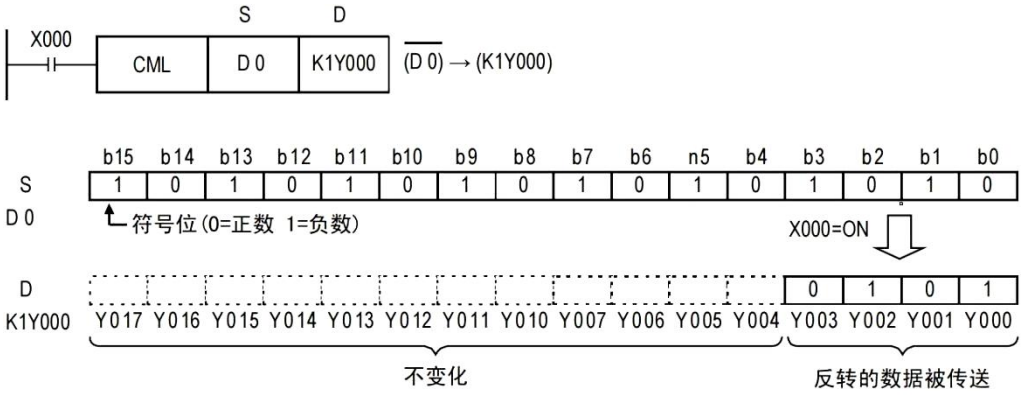
8.6.4 程序应用

1. 反转输入的获取

也可以使用 CML 指令编写下述的顺控程序。



2. S 与 D 中软元件位数不等的情况



9. 四则运算指令

在本章中，主要介绍 BIN 的四则运算指令。有以下四则运算指令：

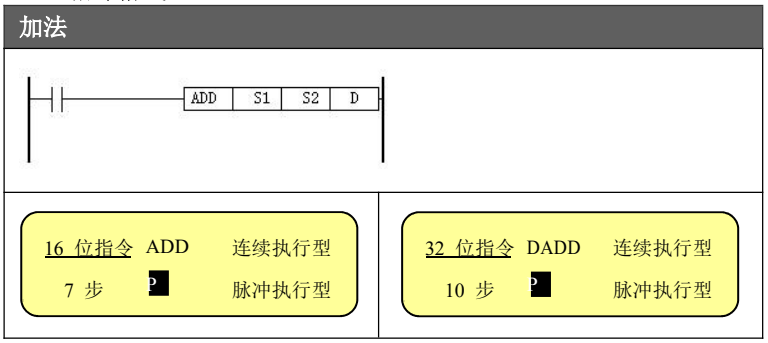
助记符、名称	功能	章节
ADD	加法运算	9.1
SUB	减法运算	9.2
MUL	乘法运算	9.3
DIV	除法运算	9.4
INC	加 1	9.5
DEC	减 1	9.6
WAND	逻辑与	9.7
WOR	逻辑或	9.8
WXOR	逻辑异或	9.9
NEG	求补码	9.10
MEAN	求平均	9.11
SQR	开方	9.12

9.1 [ADD]加法

9.1.1 概要

两个数据或软元件内容进行二进制加法运算并将结果存储到目的地址软元件中。

1. 指令格式



2. 设定数据

操作数	作用	类型
S1	指定进行加法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行加法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存加法运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.1.2 功能与动作说明

16 位指令（ADD，ADDP）是将（S1）和（S2）的内容进行二进制加法运算后传送到（D）中。

32 位指令（DADD，DADDP）是将（S1+1，S1）和（S2+1，S2）的内容进行二进制加法运算后传送到（D+1，D） 中。

各数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行加法运算。



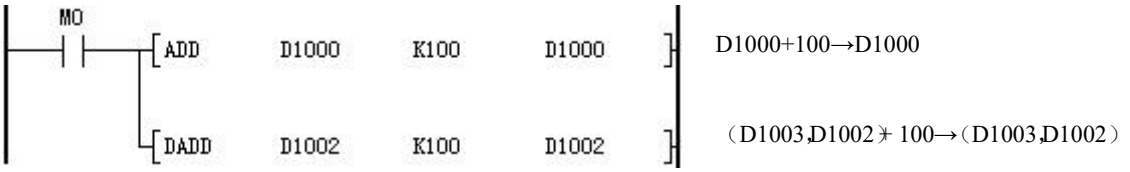
D1000=-3，D1100=11，则运算结果为(-3)+ 11 = 8，送到D1200。

9.1.3 注意要点

- 如运算结果向上溢出，即超过 32767（16 位运算）或2147483647（32 位运算）时，其结果保持为 32767 或 2147483647。
- 如运算结果向下溢出，即超过-32768（16 位运算）或-2147483648（32 位运算）时，其结果保持为-32768 或-2147483648。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的

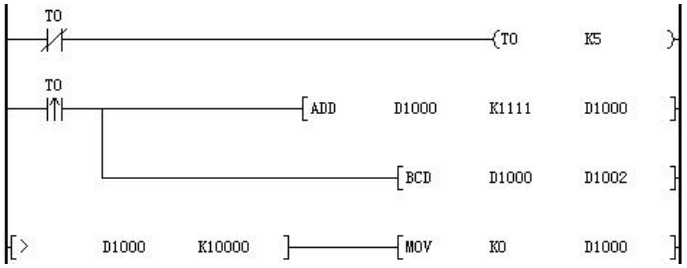
软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。

- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（ADD，DADD），则每个扫描周期的加法运算结果都会变化，请务必注意。



9.1.4 程序应用

D1000 每0.5S 加K1111，结果经 BCD 转换后送至 4 个数码管显示，数码管每 0.5S 按1111→2222→3333→4444→5555→6666→7777→8888→9999→1111 顺序循环显示。



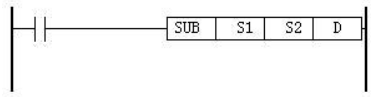
9.2 [SUB]减法

9.2.1 概要

两个数据或软元件内容进行二进制减法运算并将结果存储到目的地址软元件中。

1. 指令格式

减法



16 位指令 SUB	连续执行型	32 位指令 DSUB	连续执行型
7 步	脉冲执行型	10 步	脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定进行减法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行减法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存减法运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.2.2 功能与动作说明

16 位指令（SUB，SUBP）是将（S1）和（S2）的内容进行二进制减法运算后传送到（D）中。

32 位指令（DSUB，DSUBP）是将（S1+1，S1）和（S2+1，S2）的内容进行二进制减法运算后传送到（D+1，D） 中。

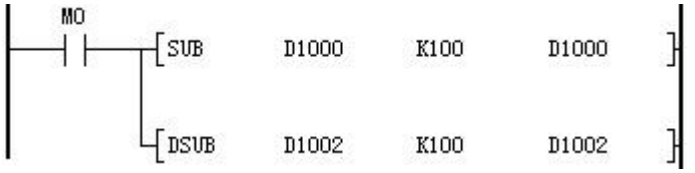
各数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行减法运算。



D1000=-3，D1100=11，则运算结果为(-3)- 11 = -14，送到D1200。

9.2.3 注意要点

- 如运算结果向上溢出，即超过 32767（16 位运算）或2147483647（32 位运算）时，其结果保持为 32767 或 2147483647。
- 如运算结果向下溢出，即超过-32768（16 位运算）或-2147483648（32 位运算）时，其结果保持为-32768 或-2147483648。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（SUB，DSUB），则每个扫描周期的减法运算结果都会变化，请务必注意。

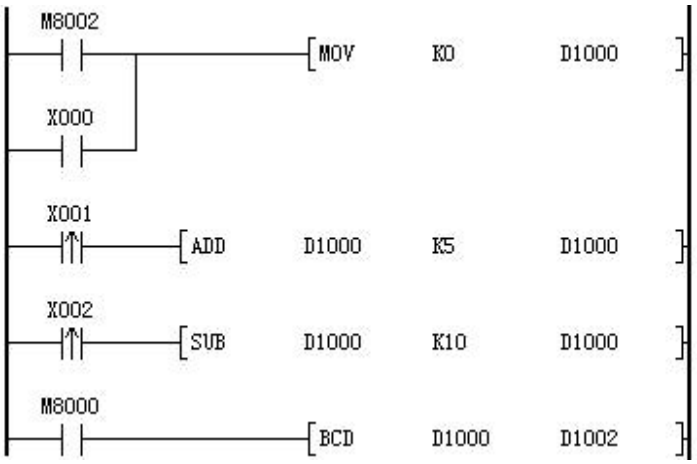


D1000-100→D1000

（D1003，D1002）-100→（D1003，D1002）

9.2.4 程序应用

D1002 值传送数码管显示。通电数码管显示0。每按下一次 X1 按钮，数码管显示值加 5，每按下一次X2按钮，数码管显示值减 10。按下 X0 按钮，数码管显示 0。



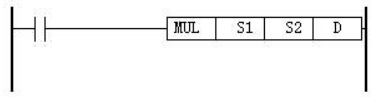
9.3 [MUL]乘法

9.3.1 概要

两个数据或软元件内容进行二进制乘法运算并将结果存储到目的地址软元件中。

1. 指令格式

乘法



16 位指令 MUL	连续执行型	32 位指令 DMUL	连续执行型
7 步	脉冲执行型	10 步	脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定进行乘法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行乘法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存乘法运算结果的软元件地址编号	32/64 位 BIN

3. 对象软元件

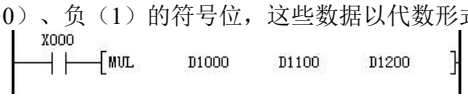
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.3.2 功能与动作说明

16 位指令（MUL，MULP）是将（S1）和（S2）的内容进行二进制乘法运算后传送到（D+1，D）中。

32 位指令（DMUL，DMULP）是将（S1+1，S1）和（S2+1，S2）的内容进行二进制乘法运算后传送到（D+3，D+2，D+1，D）中。各数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行乘法运算。

《16 位乘法运算》



(D1000)×(D1100) → (D1201，D1200)

16 位 × 16 位 → 32 位

《32 位乘法运算》



(D1001，D1000)×(D1101，D1100)→(D1203，D1202，D1201，D1200)

32位 × 32位 → 64位

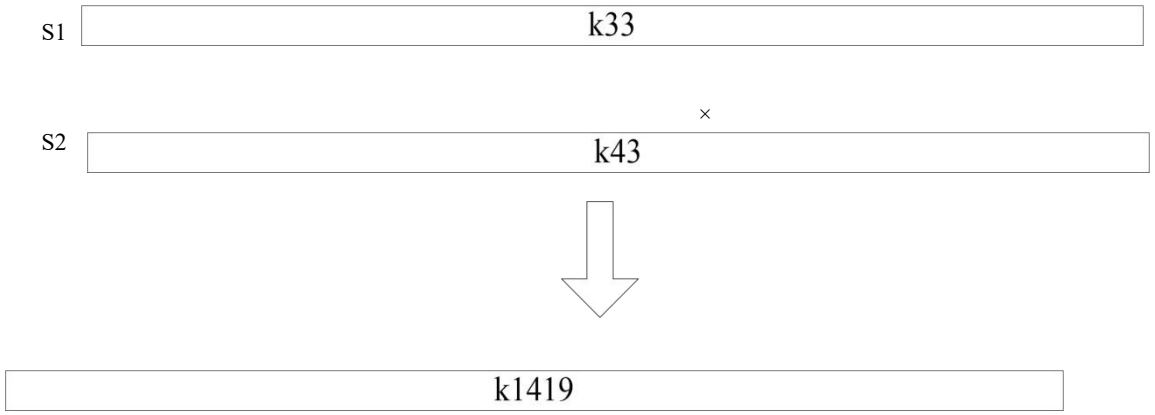
9.3.3 注意要点

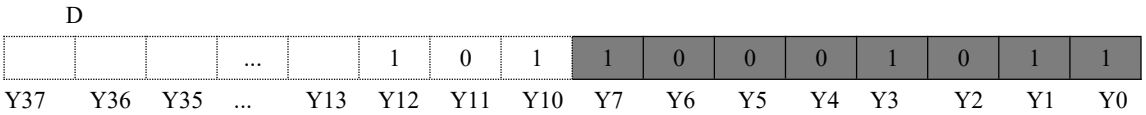
● 存放结果不要指定软元件的最后一个编号（如 D7999），可能导致高位运算结果丢失。

建议指定偶数编号的软元件地址。

- D 是位软元件组合时(如K4M100)，可以进行K1～K8 的位指定。
  - 16 位运算，指定为 Kn（1～4）时，只能得到乘积（32 位）的 n*4 位，如指定为 K4 时，只能得到乘积的低 16 位，不能得到高 16 位。
  - 32 位运算，指定为 Kn（1～8）时，只能得到乘积（64 位）的 n*4 位，如指定为 K8 时，只能得到乘积的低 32 位，不能得到高 32 位。
  - 一般建议将位软元件组合向字软元件传送，使用字软元件进行运算。







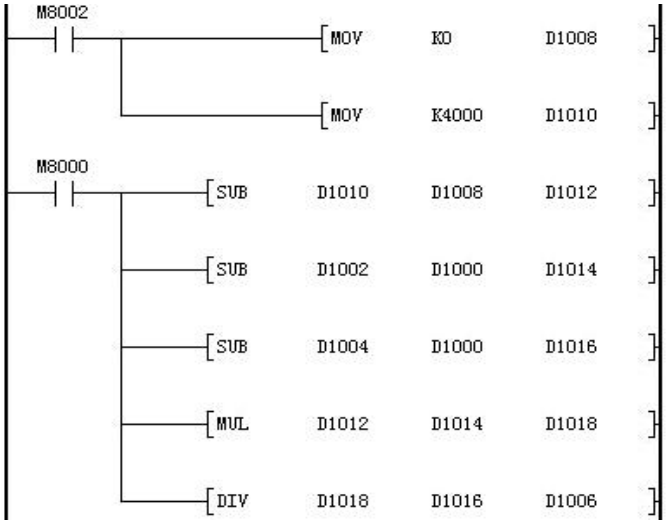
Y0～Y7 输出运算结果，Y10～Y37 不输出。

- 由于无法监控到 64 位字元件，所以使用 32 位乘法指令时，若存在运算结果超出 32 位的情况，建议使用浮点运算。

9.3.4 程序应用

模拟量标准信号使用时，将 0-20mA 模拟量信号线性化为 0-4000。设 D1000 为0-20mA 零点输入AD 值， D1002 为0-20mA 量程输入 AD 值，D1004 为当前输入 AD 值，则当前输入 AD 值对应 0-4000 中的线性化计算关系为：D1006=（4000-0）*（D1002-D1000）/（D1004-D1000）.。

梯形图实现如下：

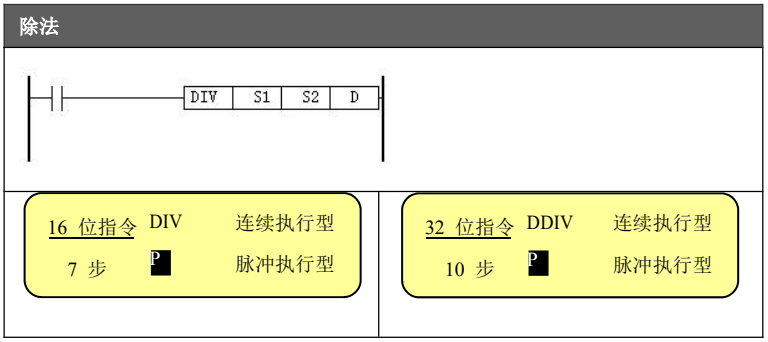


9.4 [DIV]除法

9.4.1 概要

两个数据或软元件内容进行二进制除法运算并将结果存储到目的地址软元件中。

1. 指令格式



2. 设定数据

操作数	作用	类型
S1	指定进行除法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行除法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存除法运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.4.2 功能与动作说明

16 位指令（DIV，DIVP）是将（S1）和（S2）的内容进行二进制除法运算后传送到 D，D+1 中，D 存放商，D+1 存放余数。

32 位指令（DDIV，DDIVP）是将（S1+1，S1）和（S2+1，S2）的内容进行二进制除法运算后传送到（D+1，D），（D+3，D+2）中，（D+1，D）存放商，（D+3，D+2）存放余数。

各数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行除法运算。

《16 位除法运算》



被除数 除数 商 余数

(D1000)÷(D1100) → (D1200) …… (D1201)

16 位 ÷16 位 → 16位 16位

《32 位除法运算》



被除数 除数 商 余数

(D1001，D1000)÷(D1101，D1100)→(D1201，D1200) ……(D1203，D1202)

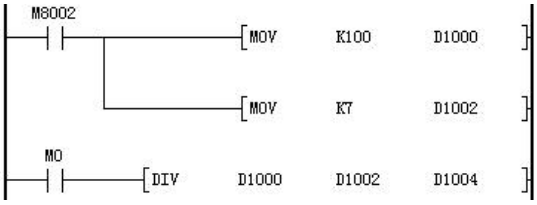
32位 ÷ 32位 → 32位 32位

9.4.3 注意要点

- 存放结果不要指定软元件的最后一个编号（如 D7999），可能导致余数丢失。建议指定偶数编号的软元件地址。
- D 是位软元件组合时(如K4M100)，可以进行K1～K8 的位指定。
  - 16 位运算，指定为 Kn（1～4）时，占用n*4*2位，分别为商和余数，如指定为 K4 时，占用32位，低 16 位为商，高 16 位为余数。
  - 32 位运算，指定为 Kn（1～8）时，占用n*4位，只得到商无余数，如指定为 K8 时，只能得到32 位商，无余数。
  - 一般建议将位软元件组合向字软元件传送，使用字软元件进行运算。

9.4.4 程序应用

100/7=14.....2，D1004 存放商，D1005 存放余数，梯形图实现如下

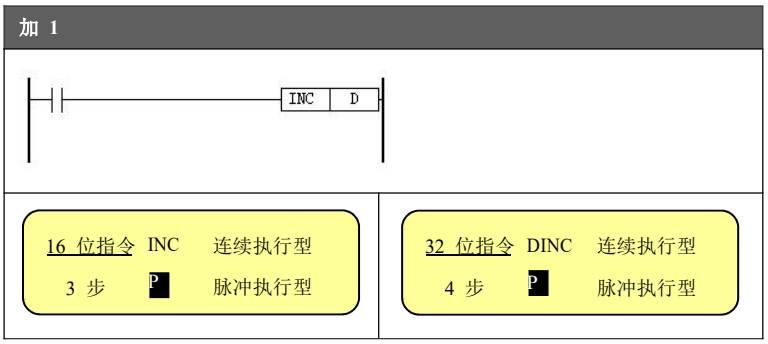


9.5 [INC]加一

9.5.1 概要

指定软元件的数据进行自加 1 运算。

1. 指令格式



2. 设定数据

操作数	作用	类型
D	指定自加 1 的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			

9.5.2 功能与动作说明

16 位指令（INC，INCP）是将（D）的内容进行自加 1 运算后传送到（D）中。

32 位指令（DINC，DINCP）是将（D+1，D）的内容进行自加 1 运算后传送到（D+1，D）中。数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行自加 1 运算。

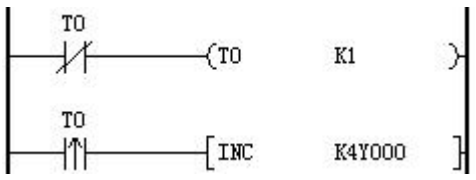


X0 每置 ON 一次，D1000 内容加 1

9.5.3 注意要点

- 源操作数和目的操作数是同一个软元件地址。
- 连续执行型指令触点 ON 时，每个扫描周期都执行加 1 运算。
- 16 位运算时，如果当前值 D1000=32767，X0 再置 ON 结果均保持不再自加 1。
- 32 位运算时，如果当前值 D1000=2147483647，X0 再 置 ON 结果均保持不再自加 1。

9.5.4 程序应用



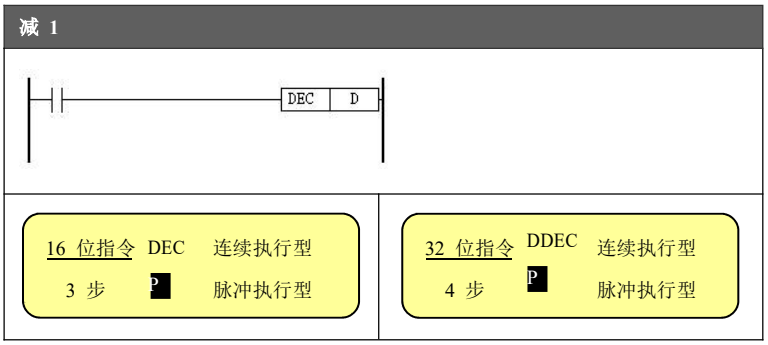
K4Y000 的值每 100ms 加 1，Y0-Y17 以每 100ms 的速度跑马输出。

9.6 [DEC]减一

9.6.1 概要

指定软元件的数据进行自减 1 运算。

1. 指令格式



2. 设定数据

操作数	作用	类型
D	指定自减 1 的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			

9.6.2 功能与动作说明

16 位指令（DEC，DECP）是将（D）的内容进行自减 1 运算后传送到（D）中。

32 位指令（DDEC，DDECP）是将（D+1，D）的内容进行自减 1 运算后传送到（D+1，D）中。数据的最高位是正（0）、负（1）的符号位，这些数据以代数形式进行自减 1 运算。

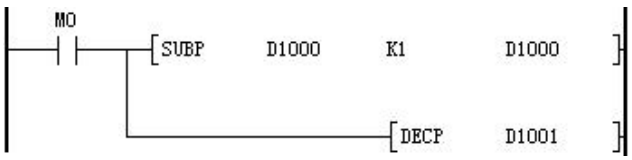


X0 每置 ON 一次，D1000 内容减 1

9.6.3 注意要点

- 源操作数和目的操作数是同一个软元件地址。
- 连续执行型指令触点 ON 时，每个扫描周期都执行减 1 运算。
- 16 位运算时，如果当前值 D1000=-32768，X0 再置 ON 结果均保持不再自减 1。
- 32 位运算时，如果当前值 D1000=-2147483648，X0 再 置 ON 结果均保持不再自减 1。

9.6.4 程序应用



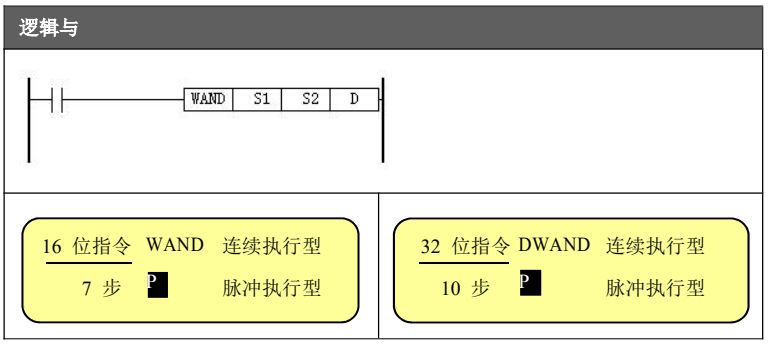
M0 由OFF 变为ON 时，D1000 和D1001 的值自减 1 运算，两条语句功能相同，当D1000 和D1001 的值减为-32768 时，M0 再 置 ON 结果均保持不再自减 1。

9.7 [WAND]逻辑与

9.7.1 概要

两个数据或软元件内容的对应位进行逻辑与运算并将结果存储到目的地址软元件中。

1. 指令格式



2. 设定数据

操作数	作用	类型
S1	指定进行运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

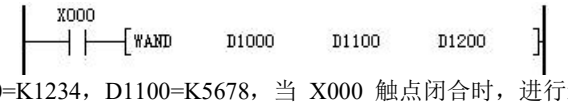
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.7.2 功能与动作说明

16 位指令（WAND，WANDP）是将（S1）和（S2）的内容进行逻辑与运算后传送到（D）中。

32 位指令（DWAND，DWANDP）是将（S1+1，S1）和（S2+1，S2）的内容进行逻辑与运算后传送到（D+1，D） 中。位逻辑与运算状态表如下：

位状态	与(&)			
操作位	0	1	0	1
操作位	0	0	1	1
结果位	0	0	0	1



假设 D1000=K1234，D1100=K5678，当 X000 触点闭合时，进行逻辑与运算。

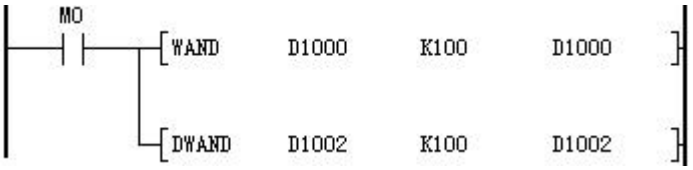
D1000 二进制 =0000 0100 1101 0010，

D1100 二进制 =0001 0110 0010 1110，按照上表进行计算。

D1200 二进制 =0000 0100 0000 0010，十进制表示为 1026。

9.7.3 注意要点

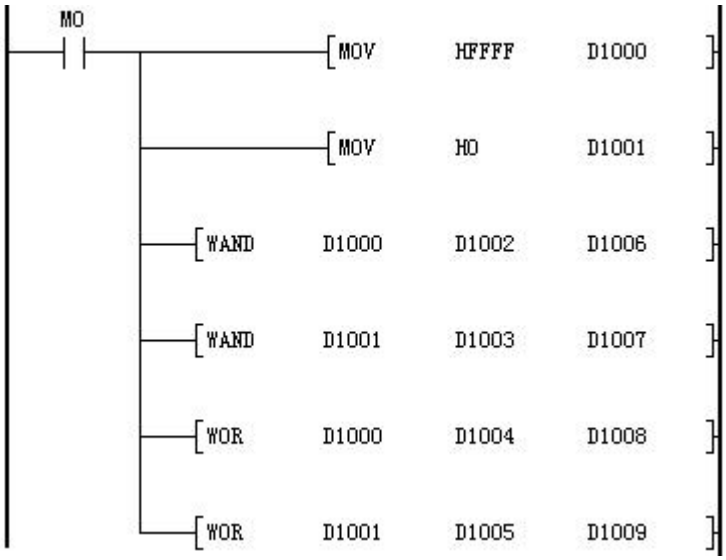
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（WAND，DWAND），则每个扫描周期的逻辑与运算结果都会变化，请务必注意。
- 以位为单位进行逻辑与运算，请务必注意。



D1000&100→D1000

（D1003D1002&100→（D1003D1002）

9.7.4 程序应用



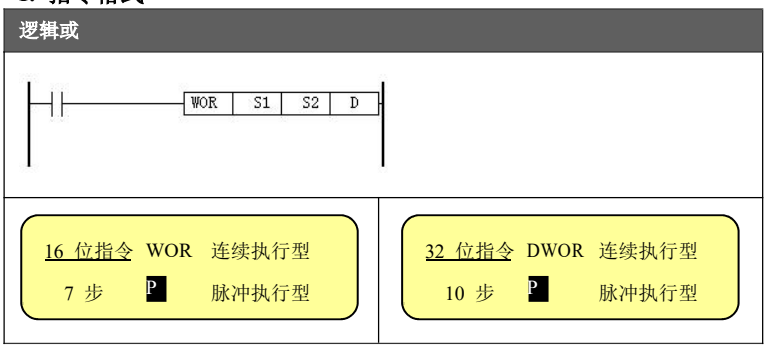
任何数和-1 与结果为本来数，任何数和 0 与结果为0，任何数和-1 或结果为-1，任何数和 0 或结果为本来数。可应用于软元件屏蔽某些位处理。

9.8 [WOR]逻辑或

9.8.1 概要

两个数据或软元件内容的对应位进行逻辑或运算并将结果存储到目的地址软元件中。

1. 指令格式



2. 设定数据

操作数	作用	类型
S1	指定进行运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	
D								•	•	•	•	•	•	•	•	•	•			

9.8.2 功能与动作说明

16 位指令（WOR，WORP）是将（S1）和（S2）的内容进行逻辑或运算后传送到（D）中。

32 位指令（DWOR， DWORP）是将（S1+1，S1）和（S2+1，S2）的内容进行逻辑或运算后传送到（D+1，D）中。位逻辑或运算状态表如下：

位状态	或(|)			
操作位	0	1	0	1
操作位	0	0	1	1
结果位	0	1	1	1



假设 D1000=K1234，D1100=K5678，当 X000 触点闭合时，进行逻辑或运算。

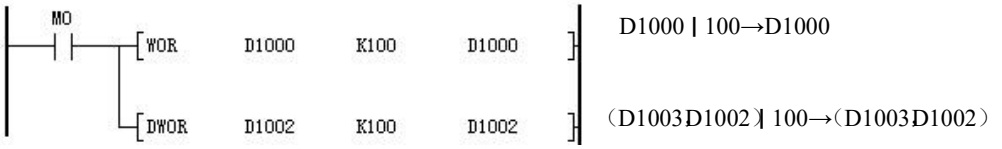
D1000 二进制 =0000 0100 1101 0010，

D1100 二进制 =0001 0110 0010 1110，按照上表进行计算。

D1200 二进制 =0001 0110 1111 1110，十进制表示为 5886。

9.8.3 注意要点

- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（WOR，DWOR），则每个扫描周期的逻辑或运算结果都会变化，请务必注意。
- 以位为单位进行逻辑或运算，请务必注意。

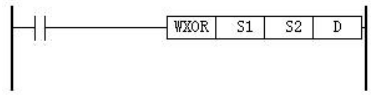


9.9 [WXOR]逻辑异或

9.9.1 概要

两个数据或软元件内容的对应位进行逻辑异或运算并将结果存储到目的地址软元件中。

1. 指令格式

逻辑异或	
	
16 位指令 WXOR 连续执行型 7 步 P 脉冲执行型	32 位指令 DXWOR 连续执行型 10 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定进行运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•			
D								•	•	•	•	•	•	•	•	•	•			

9.9.2 功能与动作说明

16 位指令（WXOR，WXORP）是将（S1）和（S2）的内容进行逻辑异或运算后传送到（D）中。

32 位指令（DWXOR，DWXORP）是将（S1+1，S1）和（S2+1，S2）的内容进行逻辑异或运算后传送到（D+1， D）中。位逻辑异或运算状态表如下：

位状态	异或(^)			
操作位	0	1	0	1
操作位	0	0	1	1
结果位	0	1	1	0



假设 D1000=K1234，D1100=K5678，当 X000 触点闭合时，进行逻辑异或运算。

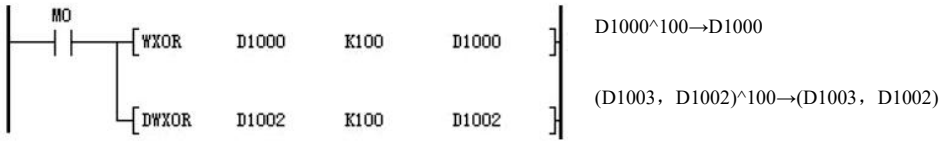
D1000 二进制 =0000 0100 1101 0010，

D1100 二进制 =0001 0110 0010 1110，按照上表进行计算。

D1200 二进制 =0001 0010 1111 1100，十进制表示为 4860。

9.9.3 注意要点

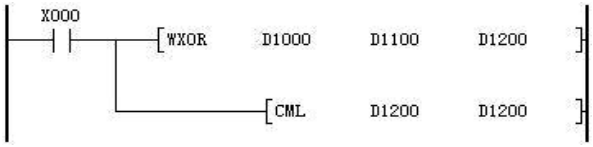
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（WXOR，



DWXOR），则每个扫描周期的逻辑异或运算结果都会变化，请务必注意。

- 以位为单位进行逻辑异或运算，请务必注意。

9.9.4 程序应用



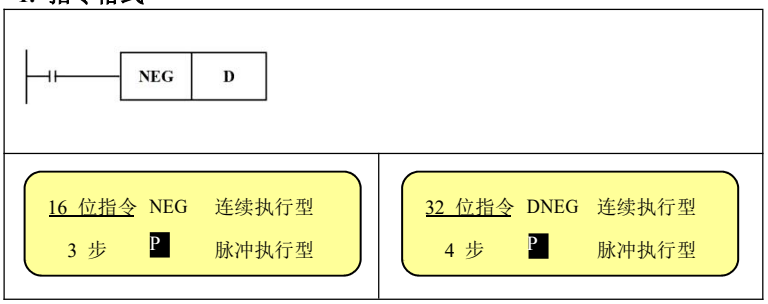
逻辑异或指令(WXOR)与取反传送指令(CML)结合使用，实现逻辑异或非运算。

9.10[NEG]求补

9.10.1概要

将指定软元件的内容按位取反后，再加 1 进行保存。

1. 指令格式



2. 设定数据

操作数	作用	类型
D	指定源数据的软元件地址编号	16/32 位 BIN

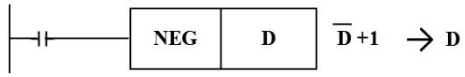
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			

9.10.2功能和动作说明

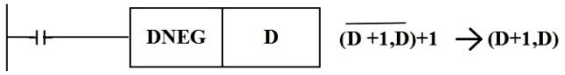
1. 16 位运算(NEG、NEGP)

将 D 内容中的各位反转(0→1、1→0)后加 1 的结果保存到原先的软元件中。



2. 32 位运算(DNEG、DNEGP)

将[D+1，D]内容中的各位反转(0→1、1→0)后加 1 的结果保存到原先的软元件中。



9.10.3注意要点

1. 使用连续执行型(NEG、DNEG)指令时，每个扫描周期(各运算周期)都执行，请务必注意。
2. 当 D 的值为-32768 时，求补值不变。

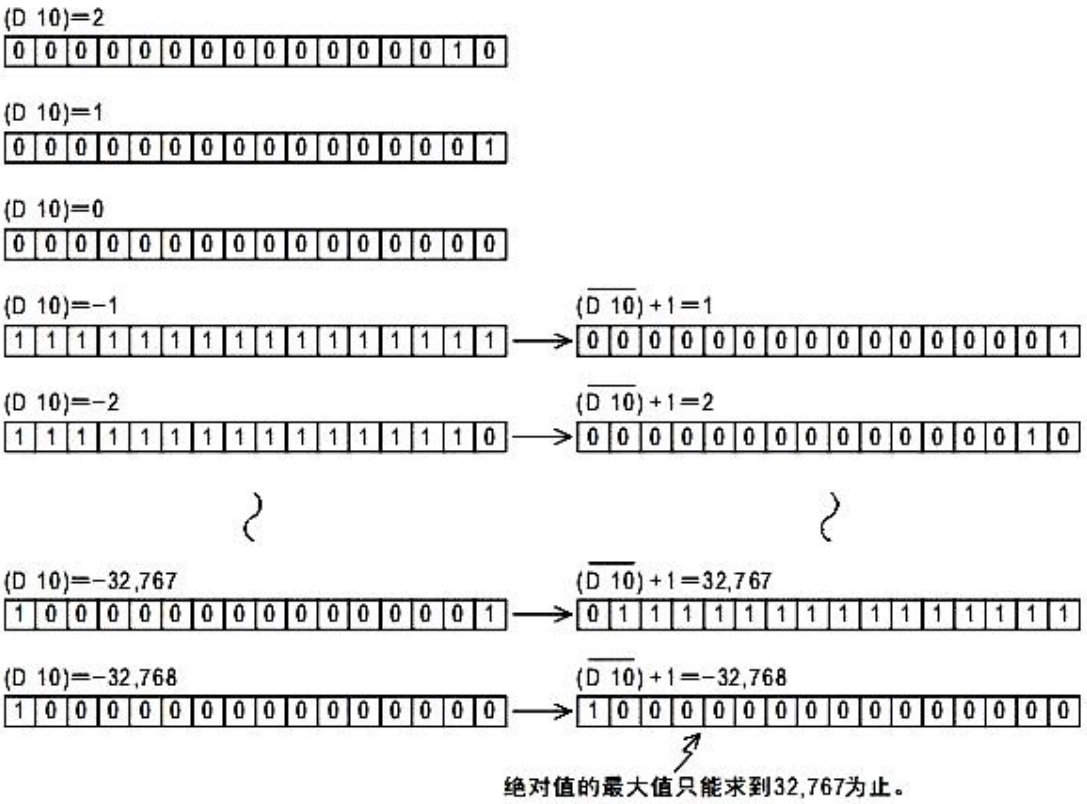
9.10.4案例应用



如上图所示 假设 D1000=1234，当 X000 触点闭合时，进行求负运算。D1000 二进制 =0000 0100 1101 0010，进行取反操作。D 取反二进制 =1111 1011 0010 1101，十进制表示为-1235。将上步结果加 1 送入 D1000，D1000=-1234。

负数的显示及绝对值(参考)

在可编程控制器中，用2的补码显示负数。
最高位为1时为负数，使用补码指令可以求出其绝对值。

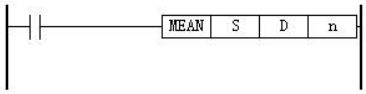


9.11 [MEAN]求平均

9.11.1 概要

将指定数据或软元件进行求平均值的运算。

1、指令格式

求平均值	
	
16 位指令 MEAN 连续执行型 7 步 P 脉冲执行型	32 位指令 DMEAN 连续执行型 10 步 P 脉冲执行型

2、设定数据

操作数	作用	类型
S	指定源数据的软元件首地址编号	16/32 位 BIN
D	指定存储平均运算结果的软元件地址编号	16/32 位 BIN
n	指定源数据个数的数值	16/32 位 BIN

3、对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•			
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

9.11.2 功能和动作说明

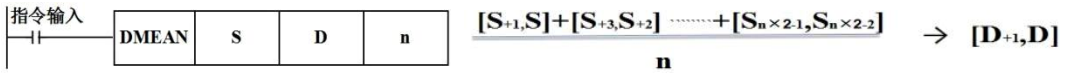
1. 16 位运算(MEAN、MEANP)

将S开始的n个 16 位数据的平均值保存到 D 中,合计是求出代数和后被 n 除,余数舍去。



2. 32 位运算(DMEAN、DMEANP)

将[S+1，S]开始的n 个 32 位数据的平均值保存到[D+1，D]中,合计是求出代数和后被n 除,余数舍去。

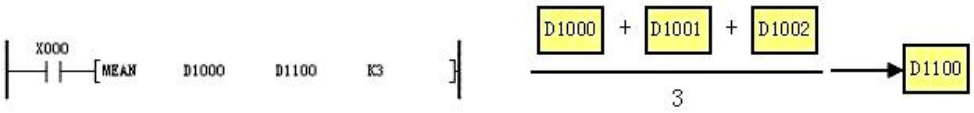


9.11.3 注意要点

1.S 取n 后的地址编号不要超过该软元件的地址范围。对 16 位指令而言，n 取值范围为 1～64。对 32 位指令而言，n 取值范围为 1～64。

2. 如果 S 取的是 K8X0,那 X 占用资源=32*n，计算时不能溢出。

9.11.4 案例应用



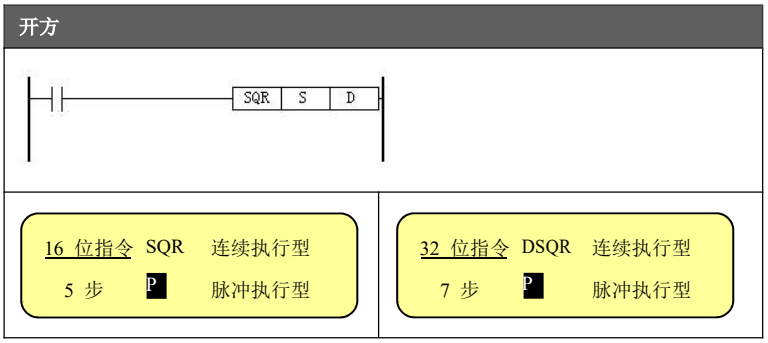
如上图所示，当 X000 触点闭合时，D1100=(D1000+D1001+D1002)÷3

9.12[SQR]开方

9.12.1概要

指定数据或软元件中内容进行开方运算并将结果存储到目的地址软元件中。

1. 指令格式



2. 设定数据

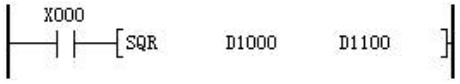
操作数	作用	类型
S	指定进行开方运算的数据或软元件地址编号	16/32 位 BIN
D	指定存储运算结果的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

9.12.2功能与动作说明

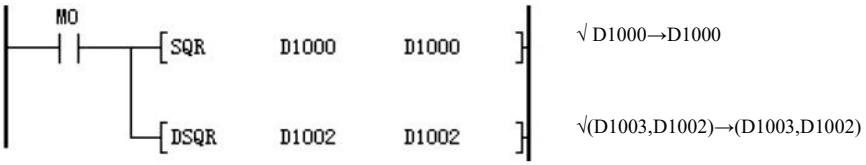
16 位指令（SQR，SQRP）是将（S）的内容进行开方运算后传送到（D）中。32 位指令（DSQR，DSQRP） 是将（S+1，S）的内容进行开方运算后传送到（D+1，D）中。



X000 触点闭合，进行开平方根的运算。D1000=10000， 结 果 D1100=100，开方无余数。D1000=88，结果 D1100=9，开方余数为 7(舍弃)。D1000=-100，结果为 0。

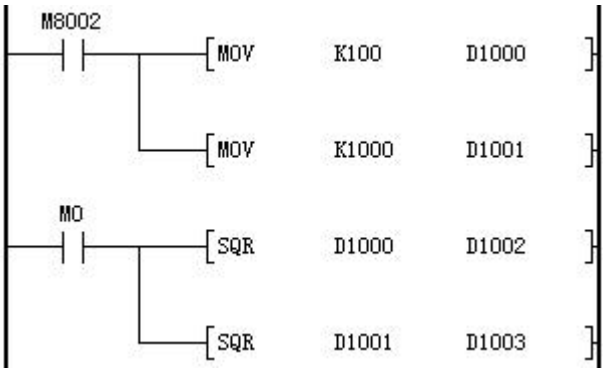
9.12.3注意要点

- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。这种情况下，如使用连续执行型指令（SQR，DSQR），则每个扫描周期的开方运算结果都会变化，请务必注意。



- 开方运算，结果保留整数位，舍弃小数位。运算精度低，需要更高的运算精度，可用浮点数开方运算DESQR。
- S 是负数，结果D为0。

9.12.4程序应用



100 开方结果为 10,1000 开方结果为 31，小数部分舍弃。

10. 循环·移位

在本章中，主要介绍位软元件和字软元件的移位处理指令。

助记符、名称	功能	章节
ROL	循环左移	10.1
ROR	循环右移	10.2
RCL	带进位循环左移	10.3
RCR	带进位循环右移	10.4
SFTL	位左移	10.5
SFTR	位右移	10.6
WSFL	字左移	10.7
WSFR	字右移	10.8
SFRD	移位读出	10.9
SFWR	移位写入	10.10

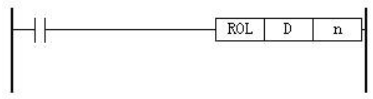
10.1 [ROL]循环左移

10.1.1 概要

使指定软元件进行各位数据的循环左移。

1. 指令格式

循环左移



16 位指令 ROL 连续执行型	32 位指令 DROL 连续执行型
5 步 P 脉冲执行型	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
D	保存循环左移数据的字软元件编号	16/32 位 BIN
n	指定循环左移的位数	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

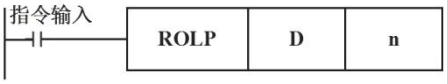
16 位运算中，K4Y000、K4M000、K4S000有效。

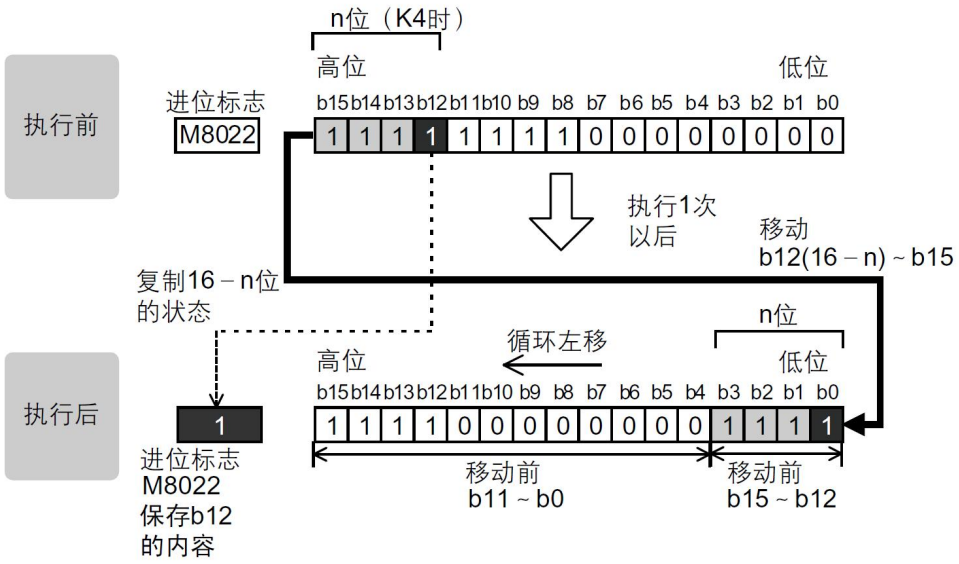
32 位运算中，K8Y000、K8M000、K8S000有效。

10.1.2 功能和动作说明

1. 16 位运算(ROL、RORP)

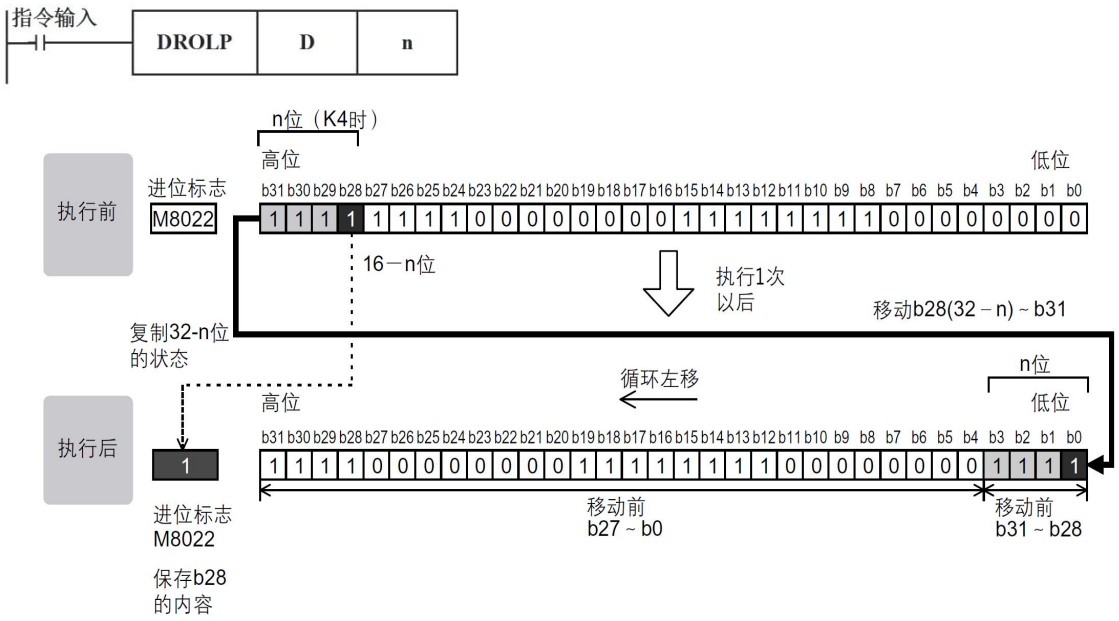
D 的 16 位中的 n 位循环左移。





2. 32 位运算(DROL、DROLP)

D 的 32 位中的 n 位循环左移。



10.1.3 注意要点

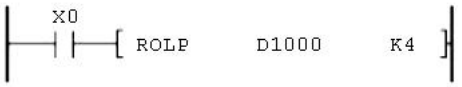
1 在指定位组合元件进行移位时，常数修饰项必须指定 K4(16 位指令)或 K8(32 位指令)，例如 K4S100， K8M64。否则运行将出错。

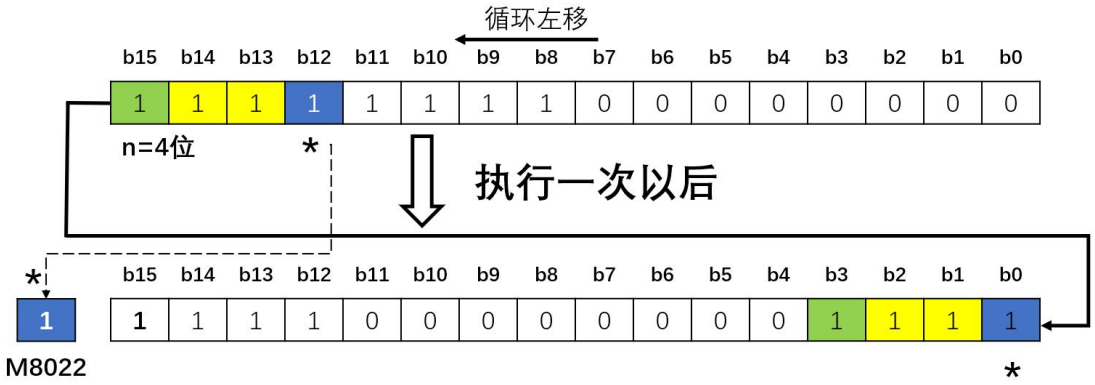
2 16 位指令时指定 1≤n≤16，32 位指令时指定 1≤n≤32，否则运行将出错。

3 连续执行型指令每个扫描周期都会进行回转动作，请务必注意。

4 最后一位数据（16-n）或（32-n）位保存至进位标志M8022，在移位时不占用移位位数。

10.1.4 程序应用





如上图所示，X0 每闭合一次，D1000 的 16 位的二进制值，都会向左移动 4 位，最后一位数据b12保存至进位标志 M8022，在移位时不占用移位位数。

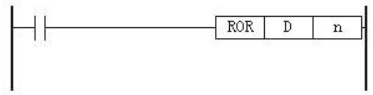
10.2 [ROR]循环右移

10.2.1 概要

使指定软元件进行各位数据的循环右移。

1. 指令格式

循环右移



16 位指令 ROR 连续执行型 5 步ROR P 脉冲执行型	32 位指令 DROR 连续执行型 7 步DROR P 脉冲执行型

2. 设定数据

操作数	作用	类型
D	保存循环右移数据的字软元件编号	16/32 位 BIN
n	指定循环右移的位数	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

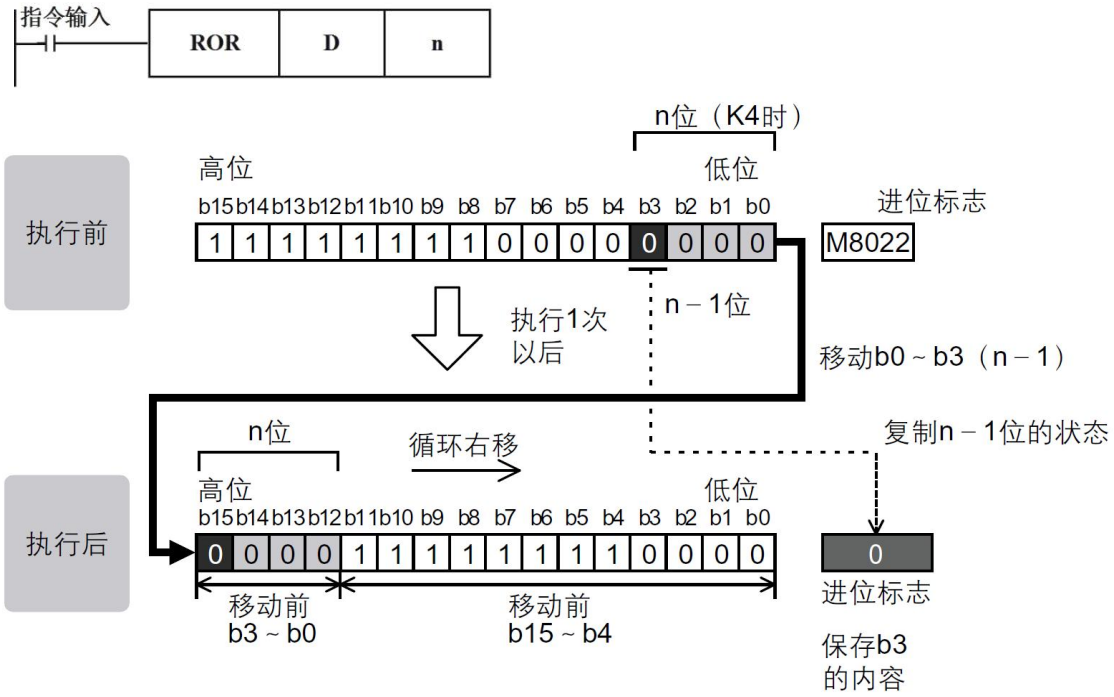
16 位运算中，K4Y000、K4M000、K4S000有效。

32 位运算中，K8Y000、K8M000、K8S000有效。

10.2.2 功能和动作说明

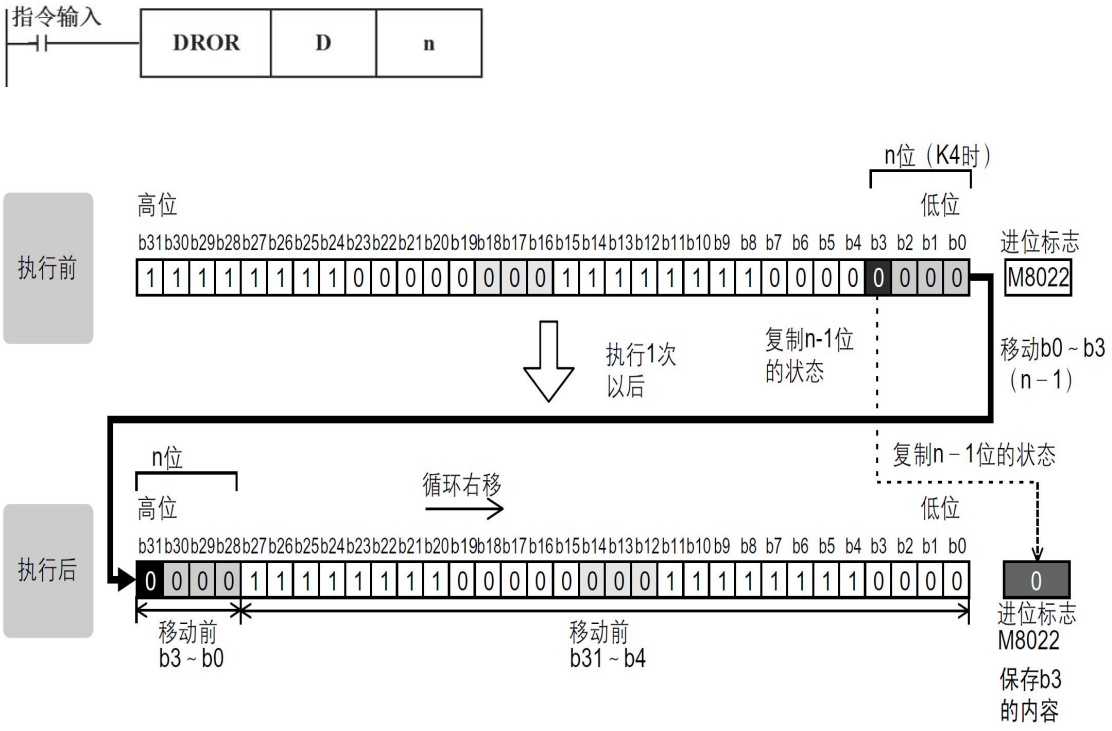
1、16 位运算(ROR、RORP)

D 的 16 位中的 n 位循环右移。



2、32 位运算(DROR、DRORP)

D 的 32 位中的 n 位循环左移。



10.2.3 注意要点

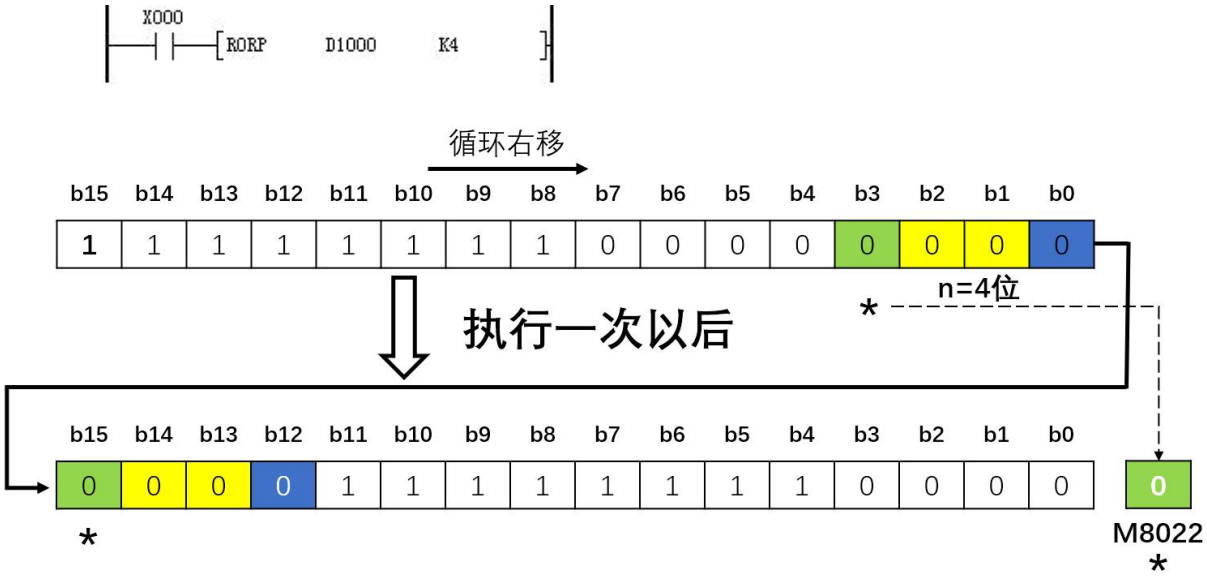
1 在指定位组合元件进行移位时，常数修饰项必须指定 K4(16 位指令)或 K8(32 位指令)，例如 K4S100， K8M64。否则运行将出错。

2 16 位指令时指定 1≤n≤16，32 位指令时指定 1≤n≤32，否则运行将出错。

3 连续执行型指令每个扫描周期都会进行回转动作，请务必注意。

4 最后一位数据（n-1）或（n-1）位保存至进位标志M8022，在移位时不占用移位位数。

10.2.4 程序应用



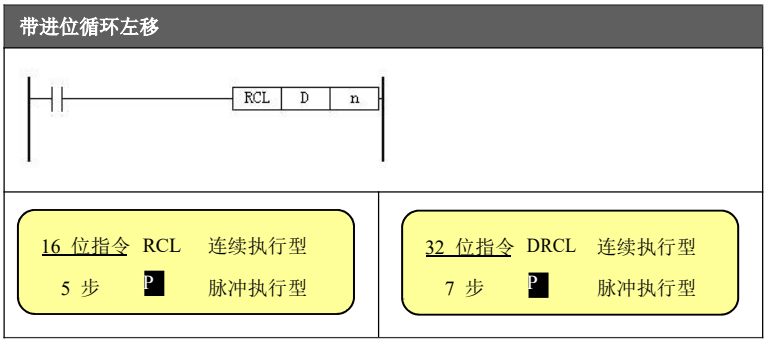
如上图所示，X0 每闭合一次，D1000 的 16 位的二进制值，都会向右移动 4 位，最后一位数据b3保存至进位标志 M8022，在移位时不占用移位位数。

10.3[RCL]带进位循环左移

10.3.1 概要

使指定软元件进行各位数据的带进位循环左移。

1. 指令格式



2、设定数据

操作数	作用	类型
D	保存循环左移数据的字软元件编号	16/32 位 BIN
n	循环移动的位数	16/32 位 BIN

3、对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

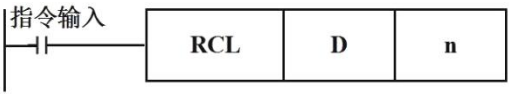
16 位运算中，K4Y000、K4M000、K4S000有效。

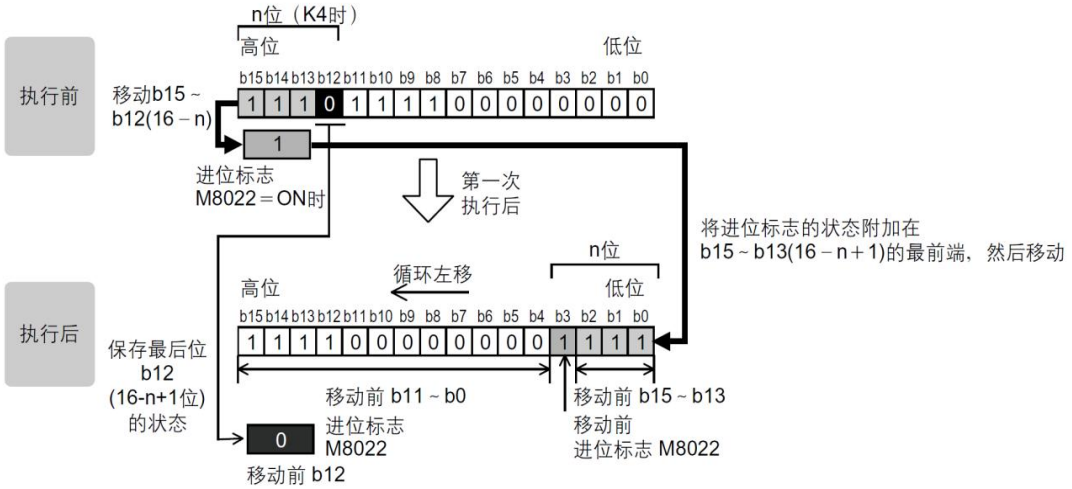
32 位运算中，K8Y000、K8M000、K8S000有效。

10.3.2 功能和动作说明

1. 16 位运算(RCL、RCLP)

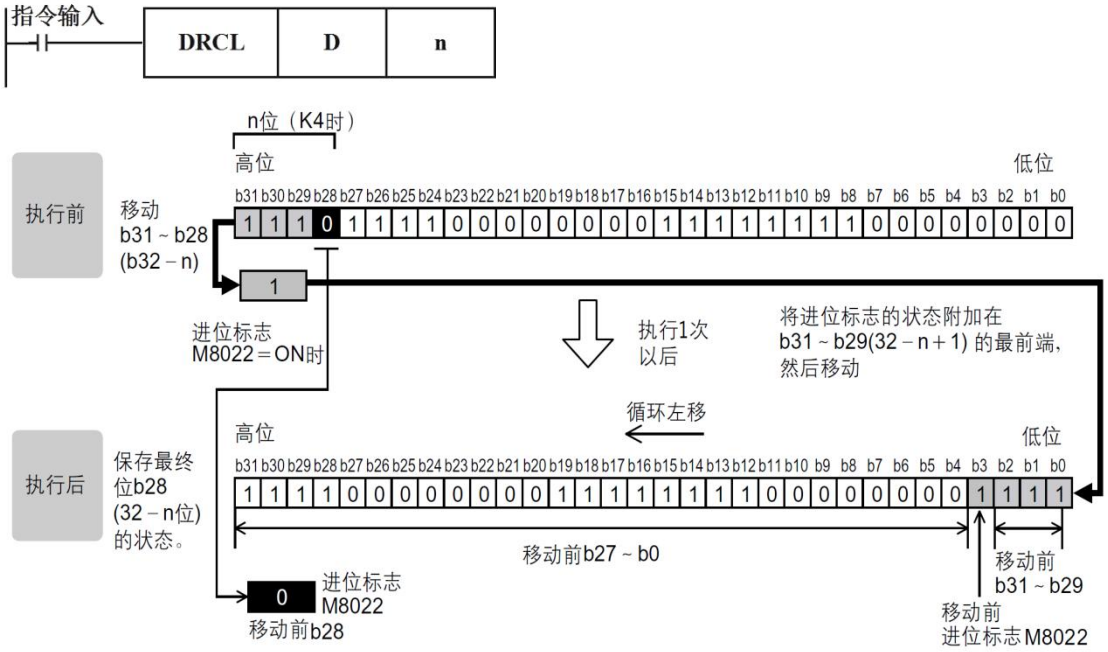
D 的 16 位+1 位(进位标志位M8022)向左移动 n 位。





2. 32 位运算(DRCL、DRCLP)

D 的 32 位+1 位(进位标志位 M8022)向左移动 n 位。



10.3.3 注意要点

1 在指定位组合元件进行移位时，常数修饰项必须指定 K4(16 位指令)或 K8(32 位指令)，例如 K4S100， K8M64。否则运行将出错。

2 16 位指令时指定 1≤n<16，32 位指令时指定 1≤n<32，否则运行将出错。

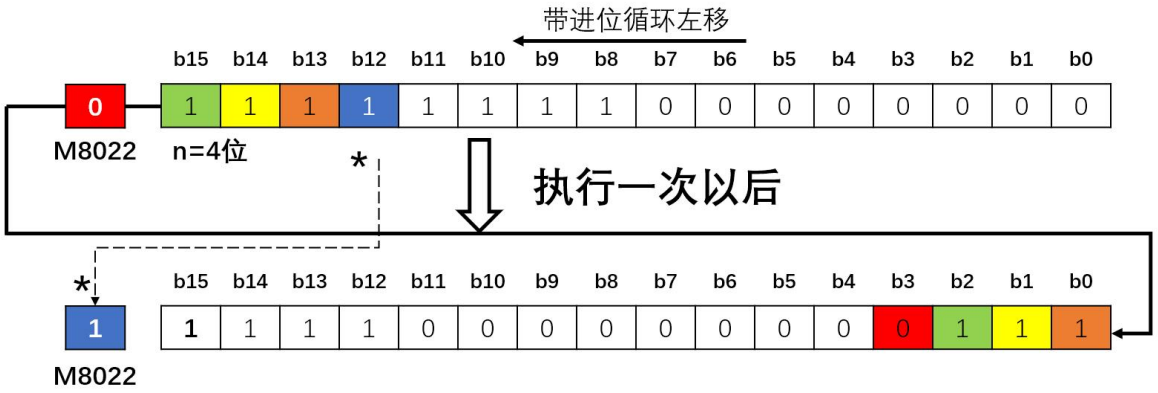
3 执行带进位循环指令前预先设置M8022 状态，可以将其送入目标地址。

4 连续执行型指令每个扫描周期都会进行回转动作，请务必注意。

5 最后一位数据（16-n）或（32-n）位保存至进位标志M8022，在移位时占用一位移位位数。

10.3.4 程序应用





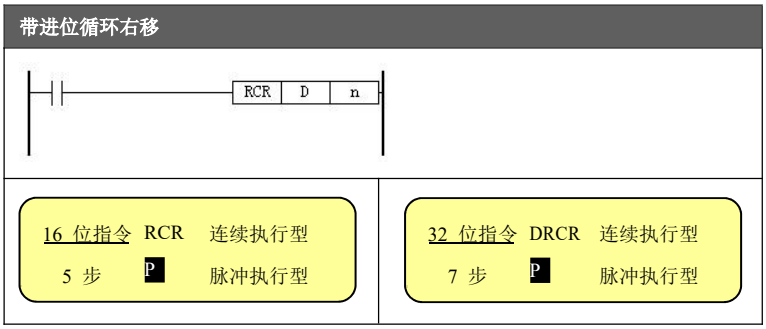
如上图所示，X0 每闭合一次，D1000 的 16 位的二进制值，都会向左移动 4 位，最后一位数据b12保存至进位标志 M8022，在移位时占用一位移位位数。

10.4 [RCR]带进位循环右移

10.4.1 概要

使指定软元件进行各位数据的带进位循环右移。

1. 指令格式



2. 设定数据

操作数	作用	类型
D	保存循环右移数据的字软元件编号	16/32 位 BIN
n	循环移动的位数	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D								•	•	•	•	•	•	•	•	•	•			
n																		•	•	

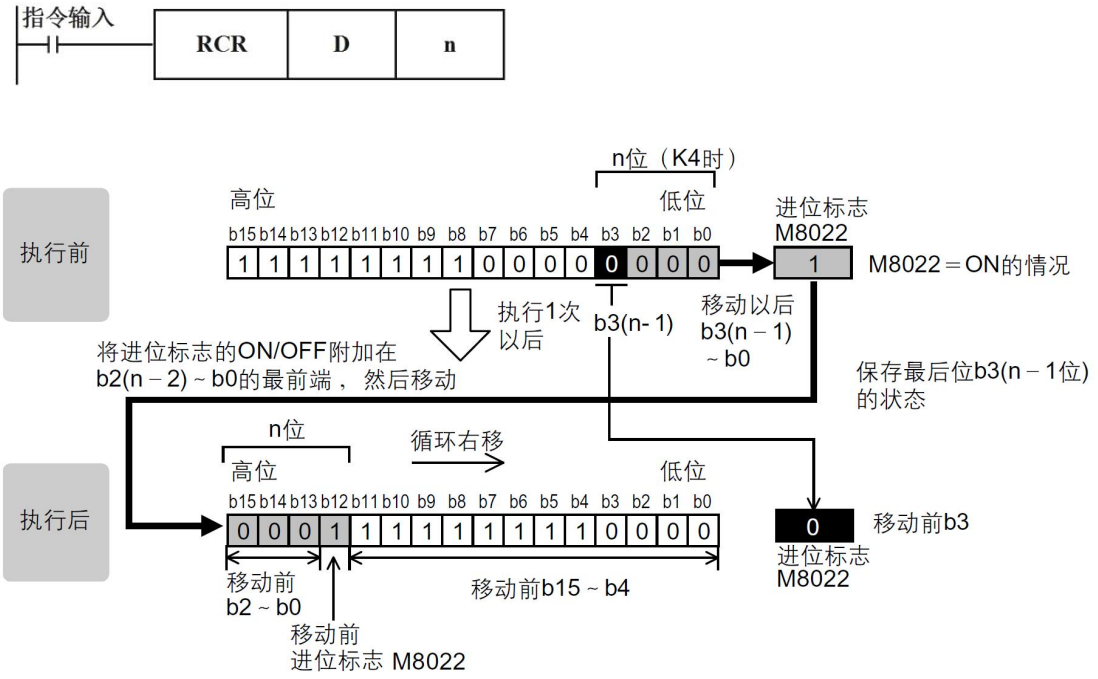
16 位运算中，K4Y000、K4M000、K4S000有效。

32 位运算中，K8Y000、K8M000、K8S000有效。

10.4.2 功能和动作说明

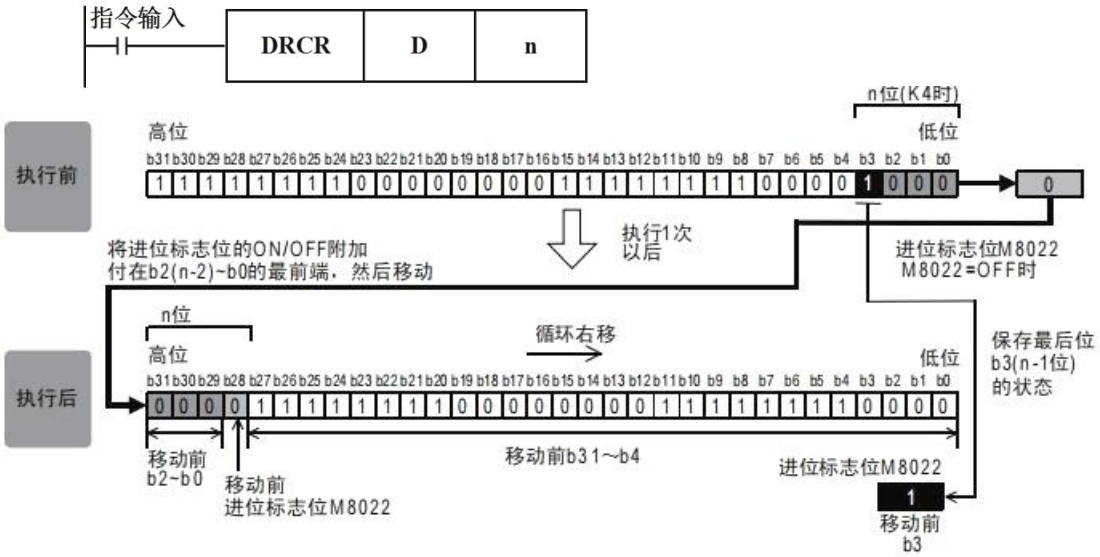
1. 16 位运算(RCR、RCRP)

D 的 16 位+1 位(进位标志位M8022)向右移动 n 位。



2. 32 位运算(DRCR、DRCRP)

D 的 32 位+1 位(进位标志位M8022)向右移动 n 位。



10.4.3 注意要点

1. 在指定位组合元件进行移位时，常数修饰项必须指定 K4(16 位指令)或K8(32 位指令)，例如 K4S100，K8M64。否则运行将出错。

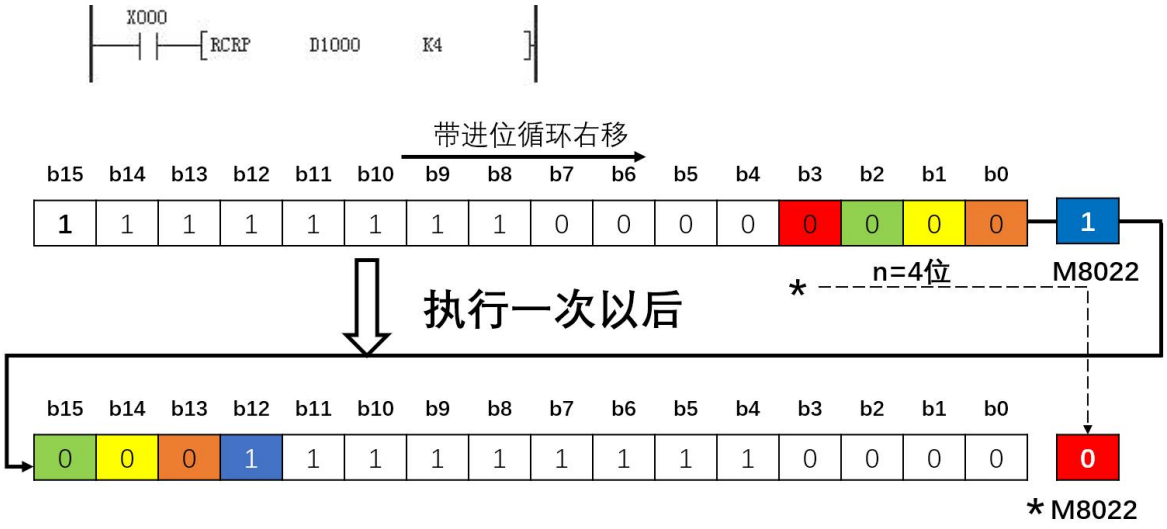
2. 16 位指令时指定 1≤n<16，32 位指令时指定 1≤n<32，否则运行将出错。

3. 执行带进位循环指令前预先设置M8022 状态，可以将其送入目标地址。

4. 连续执行型指令每个扫描周期都会进行回转动作，请务必注意。

5. 最后一位数据（n-1）位保存至进位标志M8022，在移位时占用一位移位位数。

10.4.4 程序应用



如上图所示，X0 每闭合一次，D1000 的 16 位的二进制值，都会向右移动4位，最后一位数据b3保存至进位标志 M8022，在移位时占用一位移位位数。

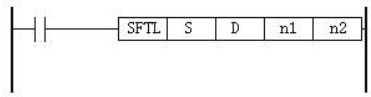
10.5 [SFTL]位左移

10.5.1 概要

使指定位长度的位软元件每次左移指定的位长度的指令。

1. 指令格式

位左移



16 位指令	SFTL	连续执行型
9 步	P	脉冲执行型

32 位指令

无

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件首地址编号	位
D	指定目标软元件首地址编号	位
n1	指定目标软元件的个数	16 位 BIN
n2	指定位左移每次移动的个数	16 位 BIN

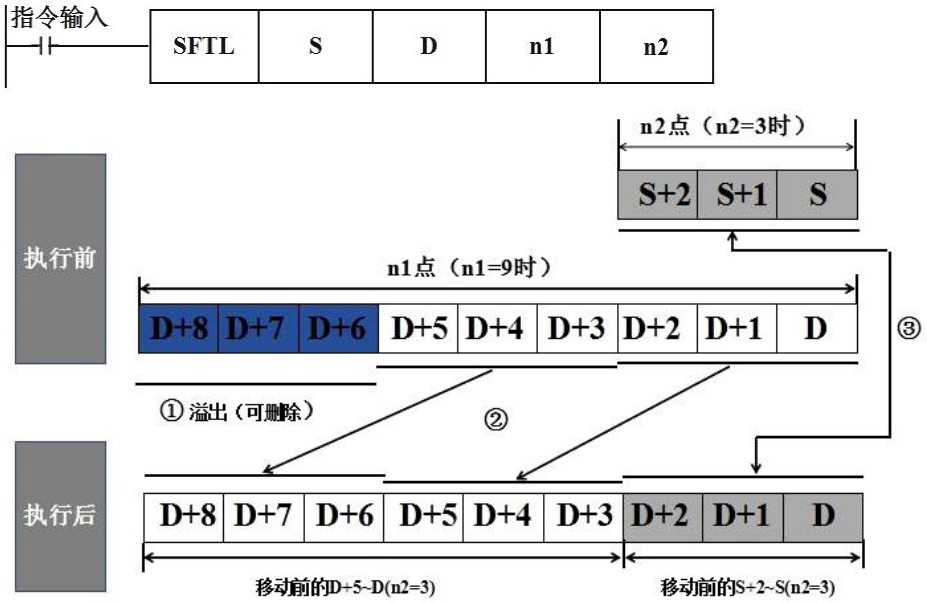
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S	•	•	•			•														
D		•	•			•														
n1																		•	•	
n2																		•	•	

10.5.2 功能和动作说明

1. 16 位运算(SFTL、SFTLP)

对于 n1 位（移动软元件的长度）的位元件进行 n2 的位移动指令(指令执行时执行n2 位的移位)。将S 开始的 n2 位数据传送到(下面的③)从D 开始的 n2 位中。



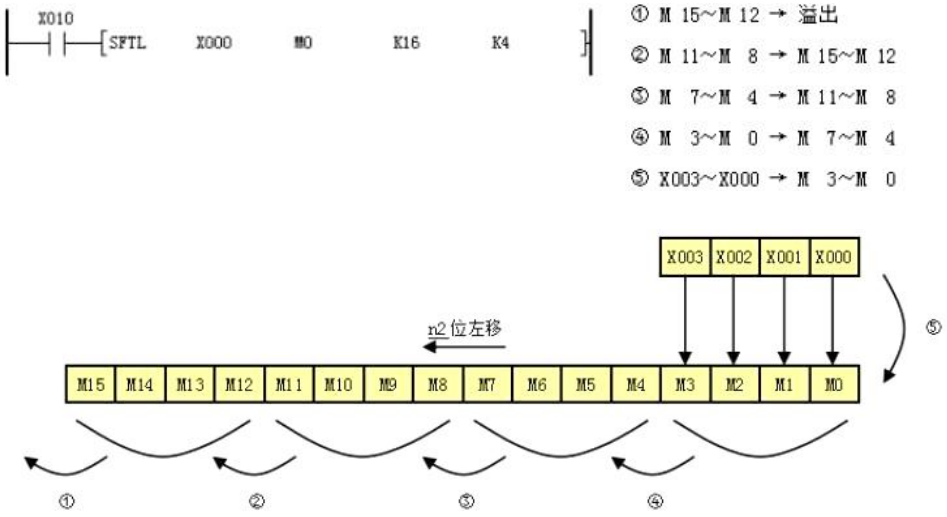
10.5.3 注意要点

1、 设置 1≤n2<n1≤1024 是有效的。

2、连续执行型指令每个扫描周期都会进行位移动操作，请务必注意。

3、传送源S 和移位软元件 D 重复时，发生运算错误。

10.5.4 程序应用



如上图所示，短接 X10 闭合一次，按照①②③④⑤的顺序执行，先把 M0-M11 的值移位到 M4-M15，再把 X0-X3 的值传到 M0-M3，向左移动 4 位。上面 X10 因为不是上升沿信号，每一个扫描周期执行一次， 所以最后结果，是 X0-X3 的值传 M0-M3、M4-M7、M8-M11、M12-M15。

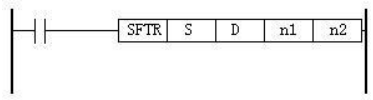
10.6 [SFTR]位右移

10.6.1 概要

使指定位长度的位软元件每次右移指定的位长度的指令。

1. 指令格式

位右移



16 位指令 SFTR	连续执行型	32 位指令
9 步	脉冲执行型	无

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件首地址编号	位
D	指定目标软元件首地址编号	位
n1	指定目标软元件的个数	16 位 BIN
n2	指定位右移每次移动的个数	16 位 BIN

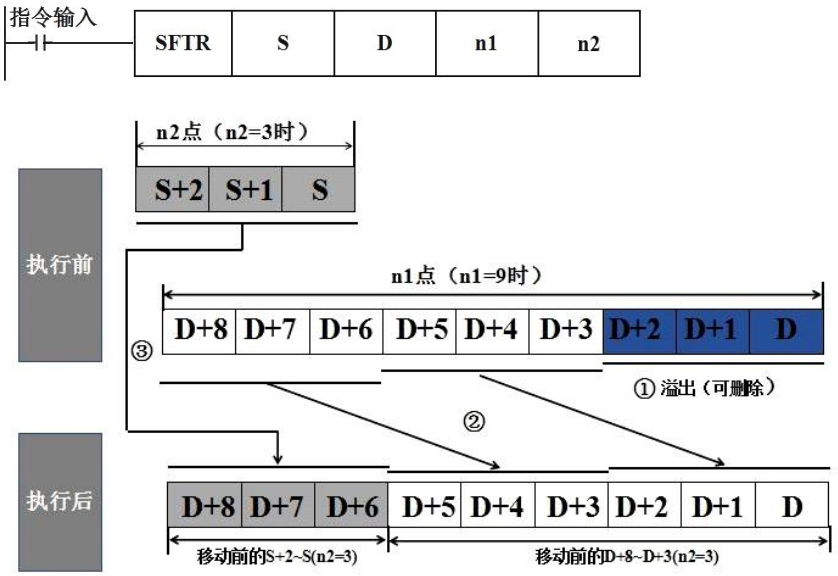
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S	•	•	•			•														
D		•	•			•														
n1																		•	•	
n2																		•	•	

10.6.2 功能和动作说明

1. 16 位运算(SFTR、SFTRP)

对于 n1 位（移动软元件的长度）的位元件进行 n2 的位移动指令(指令执行时执行 n2 位的移位)。移位后，将 S 开始 n2 位数据传送(下记的③)到从+n1-n2 开始的 n2 位中。



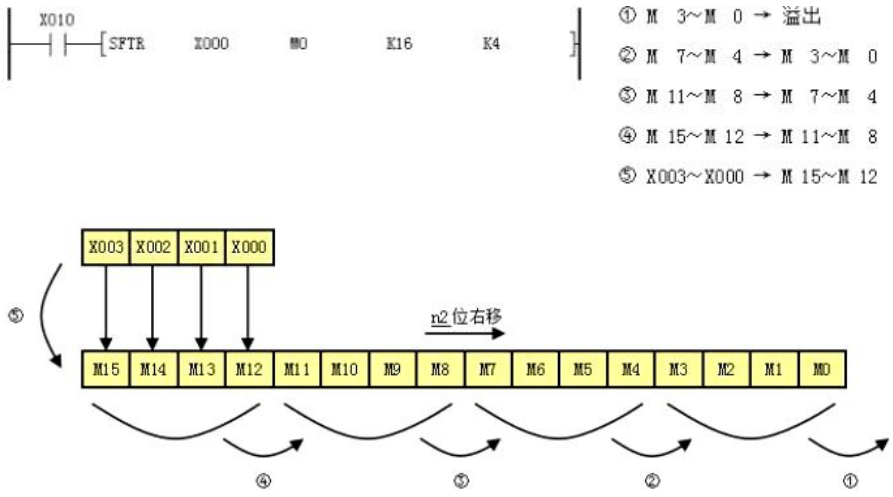
10.6.3 注意要点

1、设置 1≤n2<n1≤1024 是有效的。

2、连续执行型指令每个扫描周期都会进行位移动操作，请务必注意。

3、传送源S 和移位软元件 D 重复时，发生运算错误。

10.6.4 程序应用



如上图所示，短接 X10 闭合一次，按照①②③④⑤的顺序执行，先把 M15-M4 的值移位到 M11-M0, 再把 X0-X3 的值传到 M12-M15，向右移动 4 位。上面 X10 因为不是上升沿信号，每一个扫描周期执行一次，所以最后结果，是 X0-X3 的值传 M15-M12、M11-M8、M7-M4、M3-M0。

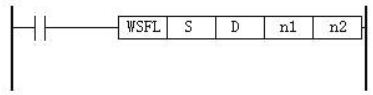
10.7 [WSFL]字左移

10.7.1 概要

将指定软元件中的数据进行字左移操作。

1. 指令格式

字左移



16 位指令 WSFL 连续执行型	32 位指令
9 步 P 脉冲执行型	无

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位 BIN
D	指定目标软元件首地址编号	16 位 BIN
n1	指定目标软元件的个数	16 位 BIN
n2	指定字左移每次移动的个数	16 位 BIN

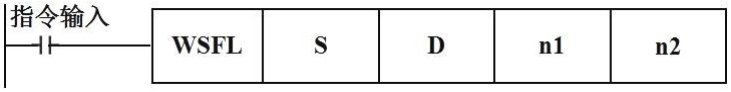
3. 对象软元件

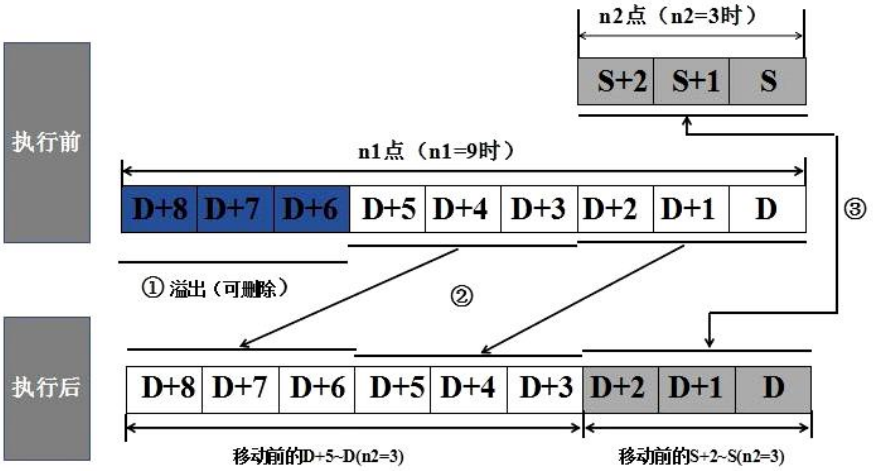
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•						
D								•	•	•	•	•	•	•						
n1																		•	•	
n2																		•	•	

10.7.2 功能和动作说明

1. 16 位运算(WSFL、WSFLP)

对于 n1 字（移动软元件的长度）的字元件进行 n2 的字左移动指令(指令执行时执行 n2 字的移字)。





10.7.3 注意要点

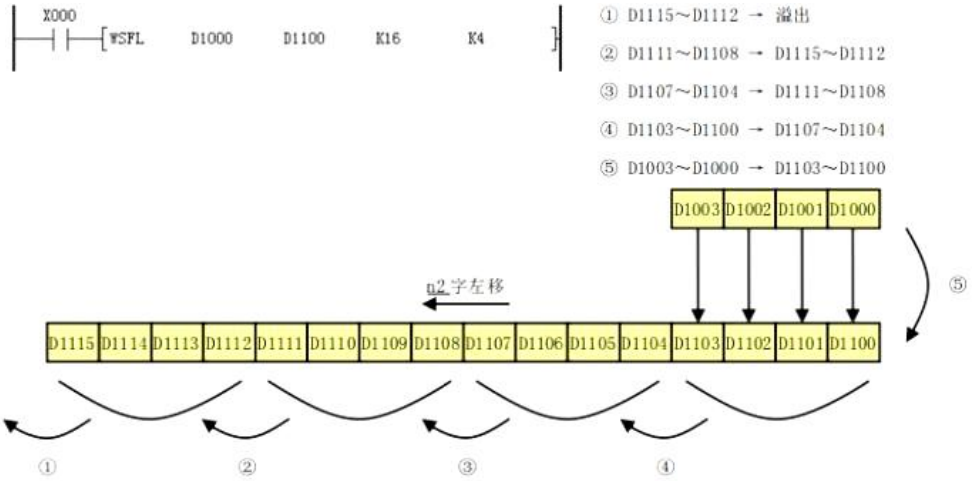
1. n2 为 1 时，每执行一次移字指令，目标软元件移动 1 字。
2. 对于 D，C，T 字软元件，设置 n2≤n1≤512 是有效的。
3. 连续执行型指令每个扫描周期都会进行字移动操作，请务必注意。
4. 传送源S 和移位软元件D 重复的时候，发生运算错误。
5. 当源操作数和目的操作数为位组合软元件，且常数修饰项<4 时，位组合软元件的常数修饰项配置必须一致，比如 S 配置为 K4M0，D 配置为 K4S0。
6. 位组合软元件其移动位数总和必须≤1024，否则运行将错误。

S 为 K2M100，n1 为 4，移动总位数=2×4×4=32，运行正常。

S 为 K2M100，n1 为 128，移动总位数=2×4×128=1024，运行正常。

S 为 K2M100，n1 为 129，移动总位数=2×4×129=1032，运行出错。

10.7.4 程序应用



如上图所示，短接 X0 闭合一次，按照①②③④⑤的顺序执行，①先把 D1115-D1112 的值移位因为溢出不管,② 把D1108-D1111 的值传到 D1112-D1115，③把D1104-D1107 的值传到 D1108-D1111,④ 把D1100-D1103 的值传到 D1104-D1107,⑤把 D1000-D1003 的值传到 D1100-D1103. 如果 X0 一直闭合，一个扫描周期就执行①②③④⑤一

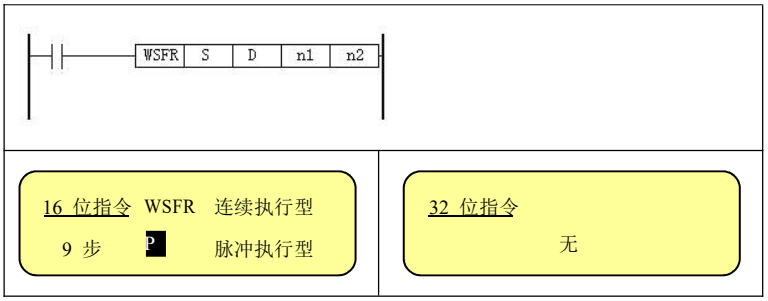
次。

10.8 [WSFR]字右移

10.8.1 概要

将指定软元件中的数据进行字右移操作。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位 BIN
D	指定目标软元件首地址编号	16 位 BIN
n1	指定目标软元件的个数	16 位 BIN
n2	指定字右移每次移动的个数	16 位 BIN

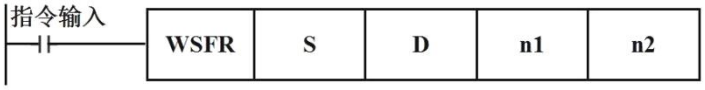
3. 对象软元件

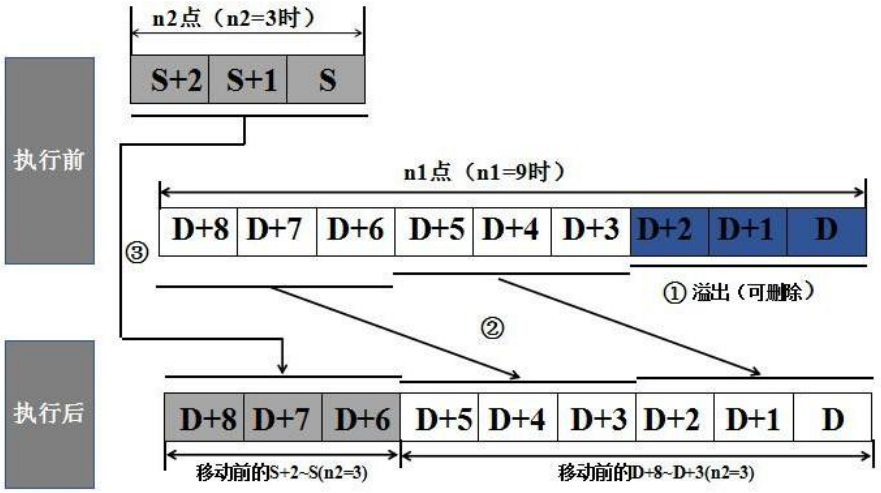
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•						
D								•	•	•	•	•	•	•						
n1																		•	•	
n2																		•	•	

10.8.2 功能和动作说明

1. 16 位运算(WSFR、WSFRP)

对于 n1 字（移动软元件的长度）的字元件进行 n2 的字右移动指令(指令执行时执行 n2 字的移字)。





10.8.3 注意要点

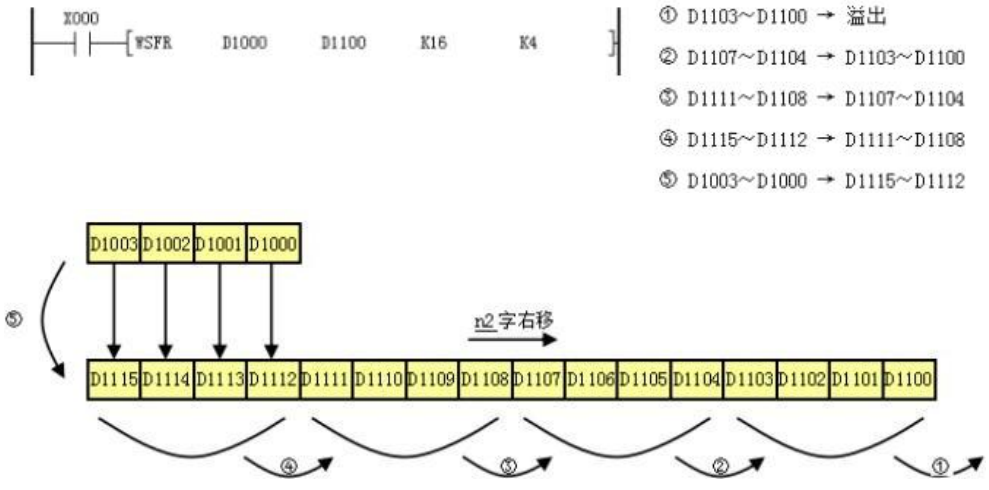
1. n2 为 1 时，每执行一次移字指令，目标软元件移动 1 字。
2. 对于 D，C，T 字软元件，设置 n2≤n1≤512 是有效的。
3. 连续执行型指令每个扫描周期都会进行字移动操作，请务必注意。
4. 传送源S 和移位软元件D 重复的时候，发生运算错误。
5. 当源操作数和目的操作数为位组合软元件，且常数修饰项<4 时，位组合软元件的常数修饰项配置必须一致，比如 S 配置为 K4M0，D 配置为 K4S0。
6. 位组合软元件其移动位数总和必须≤1024，否则运行将错误。

源地址 S 为 K2M100，n1 为 4，移动总位数=2×4×4=32，运行正常。

源地址 S 为 K2M100，n1 为 128，移动总位数=2×4×128=1024，运行正常。

源地址 S 为 K2M100，n1 为 129，移动总位数=2×4×129=1032，运行出错。

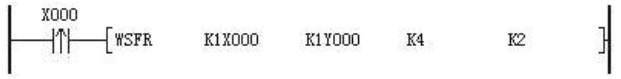
10.8.4 程序应用

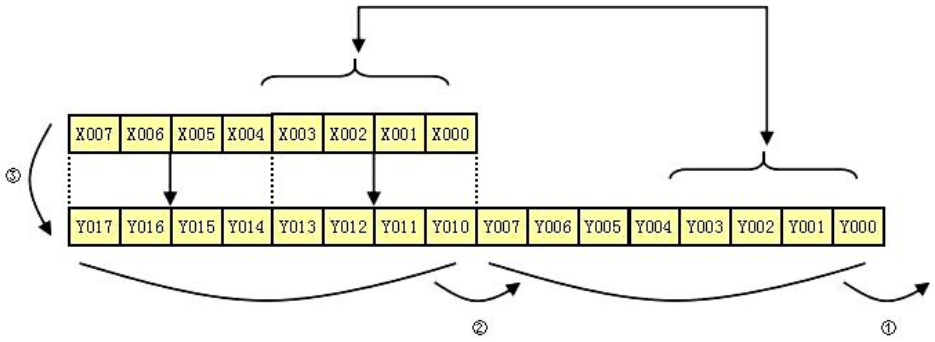


如上图所示，短接 X0 闭合一次，按照①②③④⑤的顺序执行，①先把 D1103-

D1100 的值移位因为溢出 不 管 ,② 把D1107-D1104 的 值 传 到 D1103-D1100 ， ③把 D1111-D1108 的 值 传 到 D1107-D1104,④ 把D1115-D1112 的值传到 D1111-D1108,⑤把 D1003-D1000 的值传到 D1115-D1112. 如果 X0 一直闭合，一个扫描周期就执行①②③④⑤ 一次。

下面例子是位代替字操作案例，按照①②③的顺序执行。



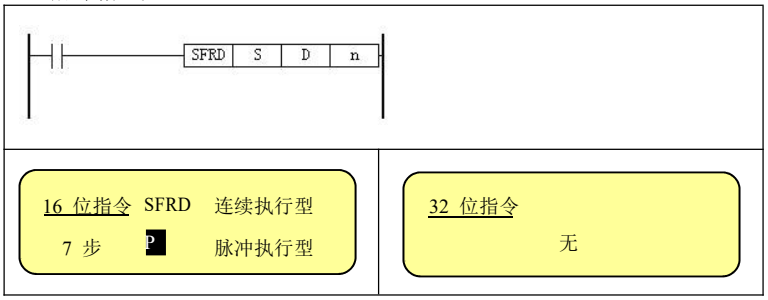


10.9 [SFRD]移位读出

10.9.1 概要

为控制先入先出数据的读出指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定移位指针的软元件地址编号	16 位 BIN
D	指定读出的软元件地址编号	16 位 BIN
n	指定移位读出的点数	16 位 BIN

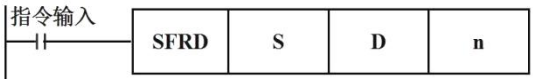
3. 对象软元件

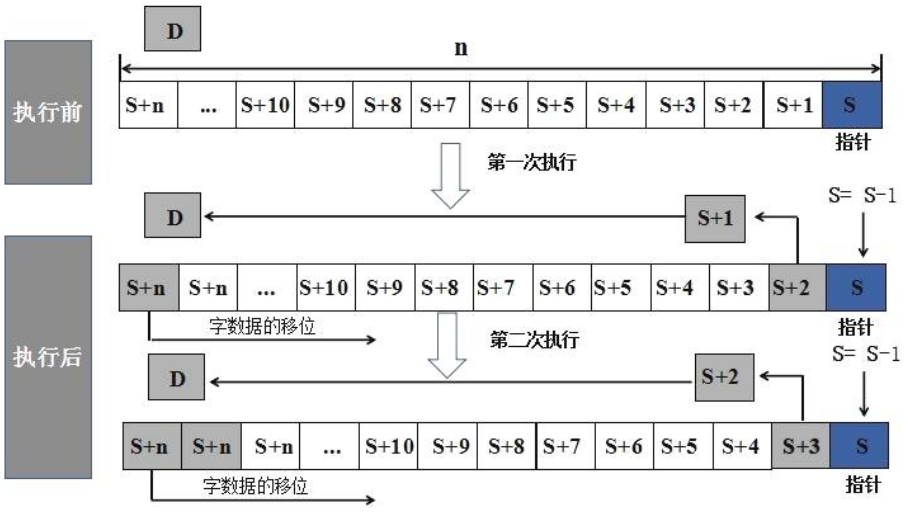
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S								•	•	•	•	•	•	•						
D								•	•	•	•	•	•	•						
n																		•	•	

10.9.2 功能和动作说明

1. 16 位运算(SFRD、SFRDP)

SFRD 指令被依次写入的 S +1 传送 (读出)到 D 中后，从 S +1 开始的 n-1 点逐字右移。S 的值每执行一次会自减 1,当 S 的值为 0 时，指令执行无效。





指令触点为 ON 时，S+1 的内容传送(读出)到 D 中。与此同时，指针 S 的内容减少，左侧的数据逐字右移。

10.9.3 注意要点

1. 对于字元件，2≤n≤512 是有效的；对于位组合元件，2≤n≤64 是有效的。
2. S 首地址内容超过 n 时，不能执行移位操作，M8022 动作。
3. 不足_1 个字的位组合移位，S 和D 必须同时指定位组合配置，且K 值相同。

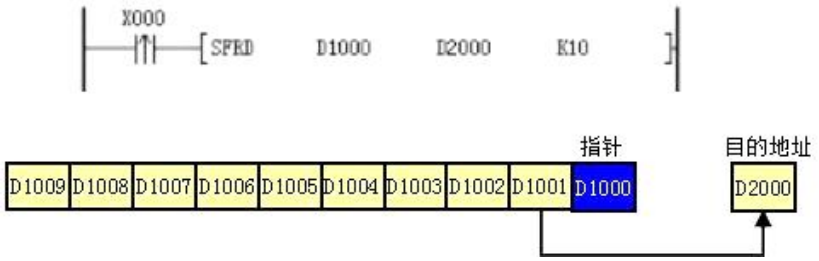
   SFRD K1M0 D2000 K10，不合法。

   SFRD K2M0 K3S0 K10，不合法。

   SFRD K3M0 K3S0 K10，合法。

3. 传送源S 和移位软元件D 重复的时候，发生运算错误。
4. S+n 的值不会因为读出而变化。

10.9.4 程序应用



如上图所示，当 X000 从OFF→ON 变化时，D 首地址编号后面的 1 个软元件地址数值被传送到目的地址，堆栈区当前指针数值个数个地址前移 1 个位置，指针数值自减 1。比如指令执行时 D1000=8，那么先执行 D1001 移动到 D2000，D1008 不变，再把 D1008～D1002 送到 D1007～D1001，然后 D1000 内容变为7。每执行一次 X000 从 OFF→ON 变化时，就按照上面顺序一直执行，当 D1000=0 时，后面执行无效。

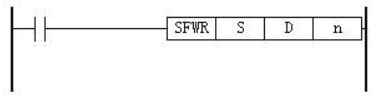
10.10 [SFWR]移位写入

10.10.1 概要

为先入先出和先入后出控制准备的数据写入指令。

1. 指令格式

移位写入



16 位指令	SFWR	连续执行型
7 步	P	脉冲执行型

32 位指令 — 无

2. 设定数据

操作数	作用	类型
S	指定写入的软元件地址编号	16 位 BIN
D	指定移位指针的软元件地址编号	16 位 BIN
n	指定移位写入的点数	16 位 BIN

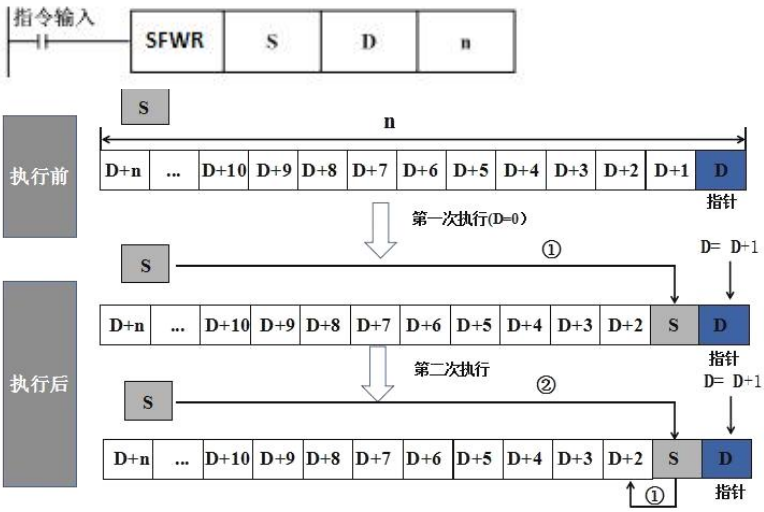
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S								•	•	•	•	•	•	•						
D								•	•	•	•	•	•	•						
n																		•	•	

10.10.2 功能和动作说明

1. 16 位运算(SFWR、SFWRP)

在D +1 开始的 n－1 点中依次写入 S 的内容，并对 D 中保存的数据数+1。



指令输入从OFF 变为ON 时， S 的内容被保存到 D+1 中，D +1 的内容变为 S 的值。S 的内容变化后再次执行输入从 OFF 变为 ON 后， S 的内容被保存到 D +2 中，D+2 的内容变为S。以下过程相同，从右端依次执行，在指针 D 的内容中表示数据的保存点数，每执行一次会加 1。字移先执行 D+1-D(n-1)传到 D+2 到Dn,再执行把S 的值传到 D+1。

10.10.3注意要点

1. 对于字元件，2≤n≤512 是有效的；对于位组合元件，2≤n≤64 是有效的。
2. S 首地址内容超过 n 时，不能执行移位操作，M8022 动作。
3. 不足 1 个字的位组合移位，S 和D 必须同时指定位组合配置，且K 值相同。

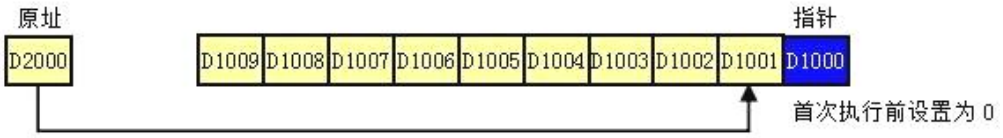
   SFWR K1M0 D2000 K10，不合法。

   SFWR K2M0 K3S0 K10，不合法。

   SFWR K3M0 K3S0 K10，合法。
4. 传送源S 和移位软元件D 重复的时候，发生运算错误。

10.10.4程序应用





如上图所示，当X000 从OFF→ON 变化时，D2000 的内容被存入 D1001，指针 D1000 的内容变加 1。X000 再次从 OFF→ON 时，D1001 的数值传到 D1002，D2000 的内容被存入 D1001，指针 D1000 的内容变为2。X000 再次从 OFF→ON 时，D1002 的数值传到 D1003，D1001 的数值传到 D1002，D2000 的内容被存入D1001，指针 D1000 的内容变为 3。以此类推直到 D1000 变成10 后，结束整个移位写入过程。配置n 值和D 首地址时需要考虑在执行过程中不能超出 D 软元件所支持的地址范围。

11. 数据处理转换指令

在本章中，主要介绍数据交换和数据转换等应用指令。

- 数据的批量复位
- 高低字节的相互交换和数据的相互交换
- 整数与浮点数的相互转换
- BCD 与二进制的相互转换
- 十六进制与 ASCII 码的相互转换

助记符、名称	功能	章节
ZRST	批复位	11.1
SWAP	高低字节交换	11.2
XCH	数据交换	11.3
FLT	整型转浮点数	11.4
INT	浮点数转整型	11.5
BIN	BCD 转二进制	11.6
BCD	二进制转 BCD	11.7
ASCI	十六进制转 ASCII	11.8
HEX	ASCII 转十六进制	11.9

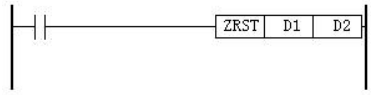
11.1 [ZRST]批次复位

11.1.1 概要

将指定软元件的数据或位清零或复位。

1. 指令格式

批次复位



16 位指令	ZRST	连续执行型	32 位指令	无
5 步	P	脉冲执行型		

2. 设定数据

操作数	作用	类型
D1	指定批次复位的起始软元件地址编号	16 位 BIN；位
D2	指定批次复位的结束软元件地址编号	16 位 BIN；位

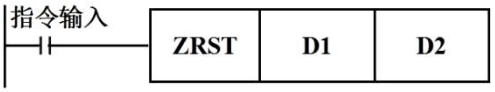
3. 对象软元件

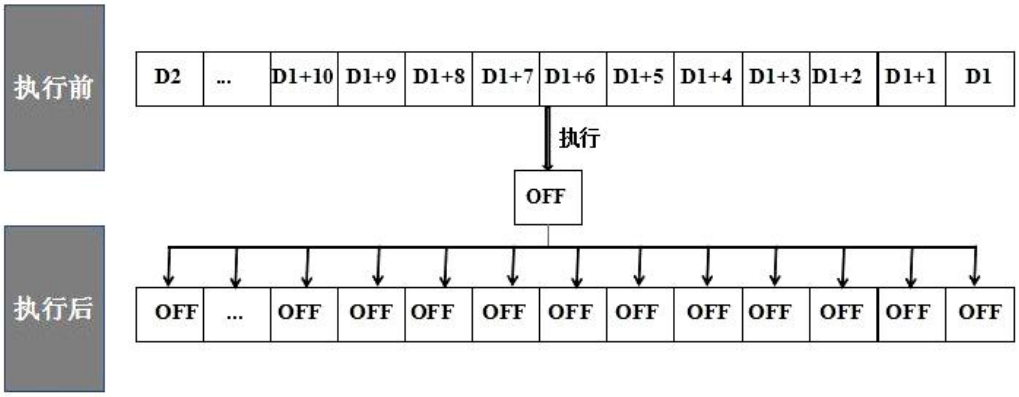
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D1		•	•	•	•	●					•	•	•	•						
D2		•	•	•	•	●					•	•	•	•						

11.1.2 功能和动作说明

1. 16 位运算(ZRST、ZRSTP)

将同一种类的 D1 ～ D2 全部复位。





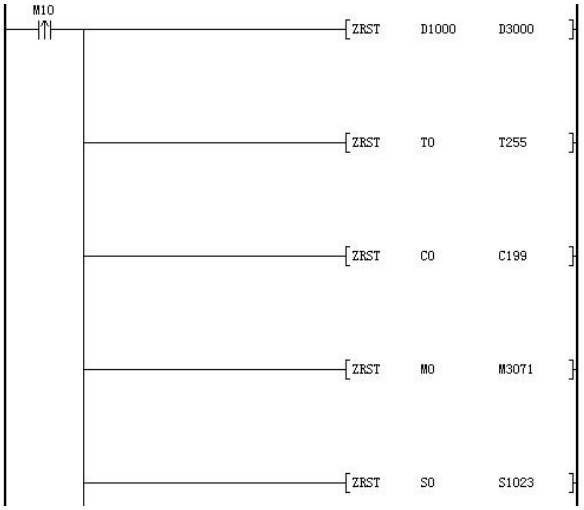
11.1.3 注意要点

1. 指定为同一种类的软元件，且D1 编号<D2 编号。
2. 软元件对象指定为C 或者 T 时，除了清除寄存器数值，还将复位其触点。
3. D1，D2 在指定软元件地址时应注意避免以下设置(非连续物理内存地址)：

   D1：指定≤M3071；D2 指定≥M8000。

   D1：指定≤D7999；D2 指定≥D8000。
4. 变 址 V Z 不能用 ZRST 清零，可用 RST 清零。

11.1.4 程序应用



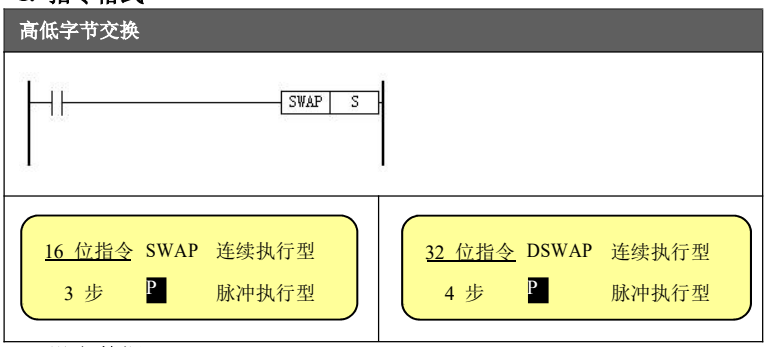
如上图所示，当 M10 从OFF→ON 变化时，D1000-D3000 的内容全部被清零，T0-T255 的值及触点全部被清零，C0-C199 的计数值及触点被清零，M0-M3071 与 S0-S1023 的状态也会变成OFF。

11.2 [SWAP]高低字节交换

11.2.1 概要

将指定软元件的高 8 位字节和低 8 位字节进行交换。

1. 指令格式



2. 设定数据

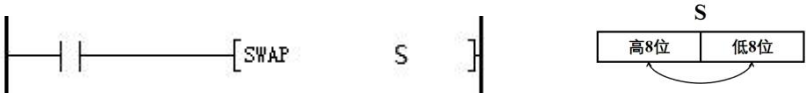
操作数	作用	类型
S	指定高低字节交换的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S								•	•	•	•	•	•	•	•	•	•			

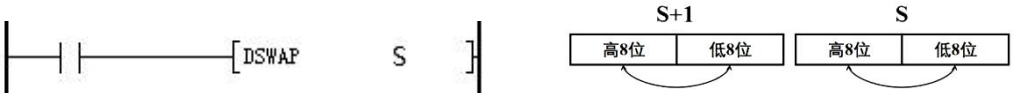
11.2.2 功能与动作说明

1、16 位运算(SWAP/SWAPP) 执行低 8 位和高 8 位的互换。



2、32 位运算(DSWAP/DSWAPP)

即使是 32 位指令，也是执行各自的低 8 位和高 8 位的互换。

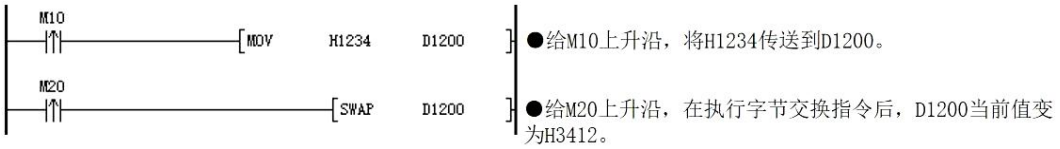


注意要点

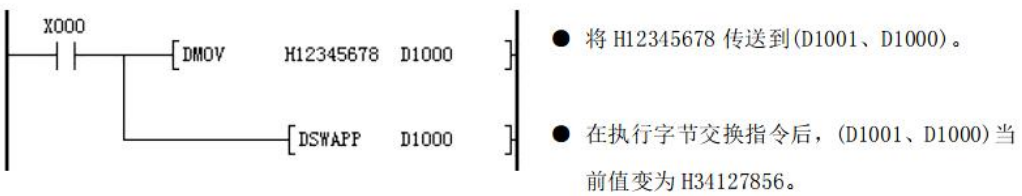
1. 若使用连续型高低字节交换指令，则每个扫描周期均会执行一次高低字节交换。

11.2.3 程序应用

1. 16 位高低字节交换



2. 32 位高低字节交换

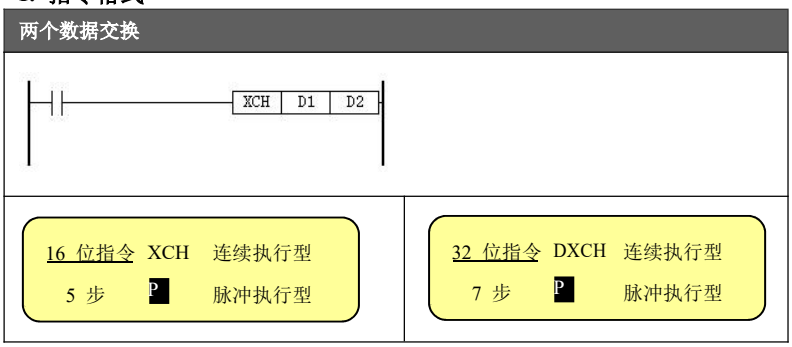


11.3 [XCH]数据交换

11.3.1 概要

将两个指定的软元件中的数据进行交换。

1. 指令格式



2. 设定数据

操作数	作用	类型
D1	指定互换的软元件地址编号	16/32 位 BIN
D2	指定互换的软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
D1								•	•	•	•	•	•	•	•	•	•			
D2								•	•	•	•	•	•	•	•	•	•			

11.3.2 功能与动作说明

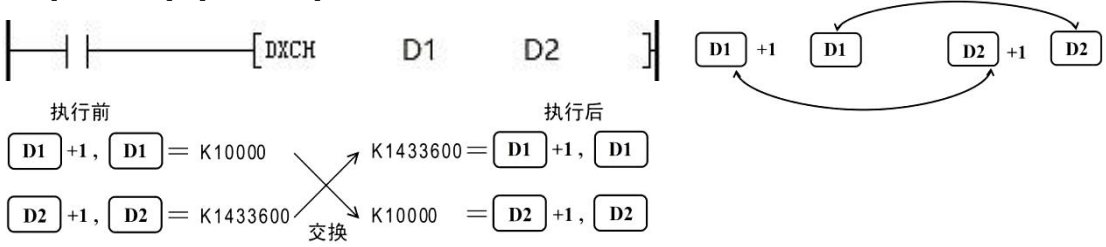
1. 16 位运算(XCH、XCHP)

D1 和 D2 相互之间进行数据交换。



2. 32 位运算(XCH、XCHP)

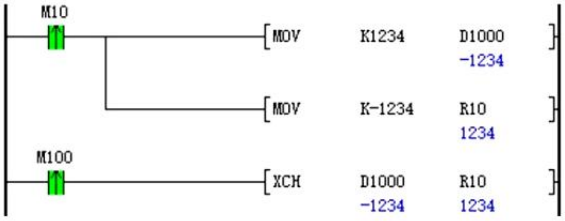
[D1+1、D1]和[D2+1、D2]相互之间进行数据交换。



1. 若使用连续型两个数据交换指令，则每个扫描周期均会执行一次数据交换。

11.3.4 程序应用

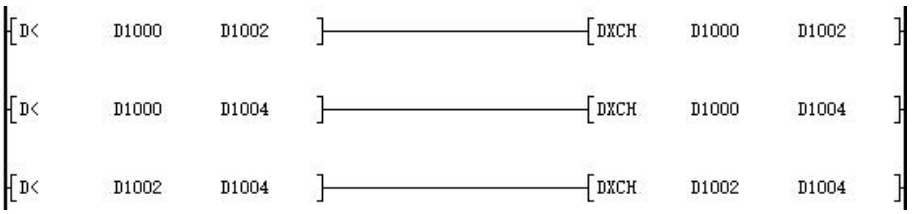
1. 16 位数据交换



- M10上升沿，将K1234赋值给D1000；
- 将K-1234赋值给R10；
- M100上升沿，在执行数据交换指令XCH后，D1000=-1234，R10=1234

2. 将数据按大小顺序排列

现有三个数 D1000，D1002，D1004，需将其按从大到小的顺序排列在 D1000，D1002，D1004 中。程序如下：



首先判断若 D1000＜D1002，执行 D1000 与D1002 的数据交换。即将 D1000 与D1002 的源数据中较大的数放在 D1000 中。

再判断若 D1000＜D1004，执行 D1000 与D1004 的数据交换。即将 D1000 与D1004 的源数据中较大的数放在 D1000 中。即达成了将三个数中最大的放入 D1000 中。

再判断若 D1002＜D1004，执行 D1002 与D1004 的数据交换。即将 D1002 与D1004 的源数据中较大的数放在D1002 中。最终达成了将三个数按从大到小的顺序排列。

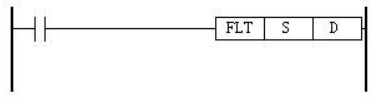
11.4 [FLT]整型转浮点

11.4.1 概要

将指定数据或软元件内容转换为 32 位二进制浮点数的指令。

1. 指令格式

整型转浮点



16 位指令 FLT 连续执行型	32 位指令 DFLT 连续执行型
5 步 P 脉冲执行型	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定源数据或软元件地址编号	16/32 位 BIN
D	指定目标软元件地址编号	32 位二进制浮点

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

11.4.2 功能和动作说明

1. 16 位运算(FLT、FLTP)

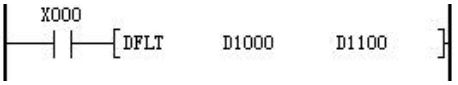
将 S 的 BIN 整数值数据转换成 2 进制浮点数(实数)值后，保存在[D+1，D]中。



D1000 → (D1101,D1100)
BIN整数 2进制浮点数

2. 32 位运算(DFLT、DFLTP)

将[S +1，S]的 BIN 整数值数据转换成 2 进制浮点数(实数)值后，保存在[D +1，D]中。



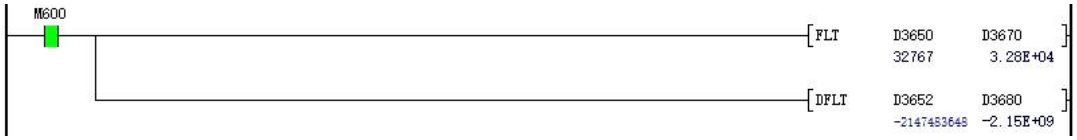
(D1001,D1000) → (D1101,D1100)
BIN整数 2进制浮点数

11.4.3 注意要点

1. 不需要常数(K、H)的浮点数转换

由于在各 2 进制浮点数(实数)运算指令中，指定的 K、H 的值会自动转换成 2 进制浮点数(实数)，所以不需要使用 FLT 指令进行转换。

11.4.4 程序应用



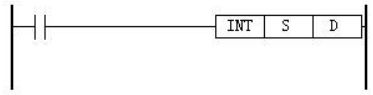
11.5 [INT]浮点转整型

11.5.1 概要

将指定软元件内容(必须为 32 位二进制浮点数)转换为 2 进制整数。

1、指令格式

浮点转整型



16 位指令	INT	连续执行型
5 步	P	脉冲执行型

32 位指令	DINT	连续执行型
7 步	P	脉冲执行型

2、设定数据

操作数	作用	类型
S	指定源数据或软元件地址编号	32 位二进制浮点
D	指定目标软元件地址编号	16/32 位 BIN

3、对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•			
D													•	•			•			

11.5.2 功能和动作说明

1、16 位运算(INT、INTP)

[S+1，S]的 2 进制浮点数转换成 BIN 整数后，传送到 D 中。

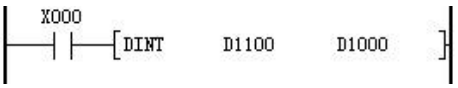


(D1101，D1100) → (D1000)

2进制浮点值 BIN整数(小数位舍去)

2、32 位运算(DINT、DINTP)

[S+1，S]的 2 进制浮点数转换成 BIN 整数后，传送到[D +1，D]中。



(D1101，D1100) → (D1001，D1000)

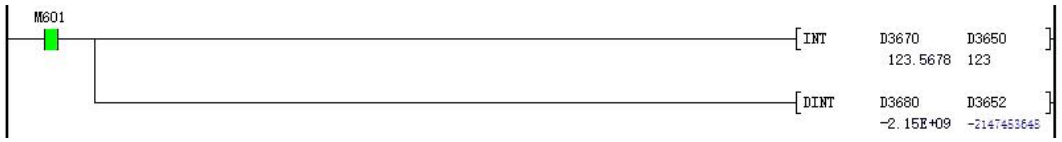
2进制浮点值 BIN整数(小数位舍去)

11.5.3 注意要点

1. 运算时的注意事项

- 小数点以后的值被舍去。

11.5.4 程序应用



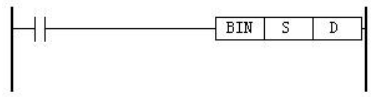
11.6 [BIN]BCD 转二进制

11.6.1 概要

将指定软元件内容(必须为 BCD 码格式)转换为二进制数。

1. 指令格式

BCD转二进制



16 位指令	BIN	连续执行型
5 步	P	脉冲执行型

32 位指令	DBIN	连续执行型
7 步	P	脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件地址编号	BCD
D	指定目标软元件地址编号	16/32 位 BIN

3. 对象软元件

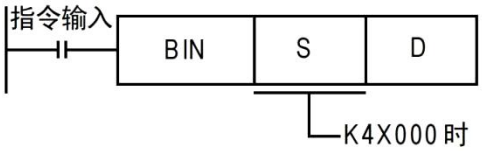
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•			
D								•	•	•	•	•	•	•	•	•	•			

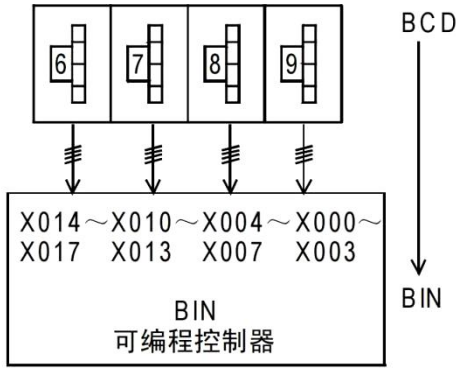
11.6.2 功能和动作说明

1、16 位运算（BIN、BINP）

将S 的BCD(10 进制数)数据转换成 BIN(2 进制数)数据后传送到D 中。

- S 的数据可以转换成 K0～K9999 的 BCD(10 进制数)。
- S 和D 指定位数的时候，参考下表。



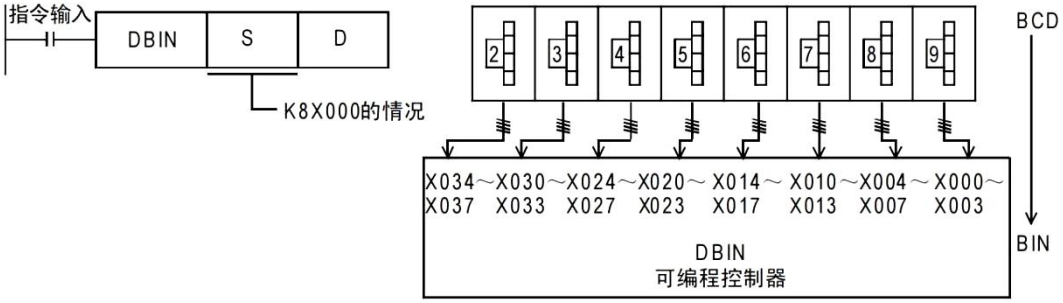


S	位数	数据范围
K1X000	1位数	0～9
K2X000	2位数	00～99
K3X000	3位数	000～999
K4X000	4位数	0000～9999

2、32 位运算（DBIN）

将[S+1，S]的BCD(10 进制数)数据转换成BIN(二进制数)数据后传送到[D+1，D]中。

- [S+1，S]的数据，可以转换成 K0～K99999999 的BCD(10 进制数)。
- [S+1，S]和[D+1，D]指定位数的时候，参考下表。



S +1， S	位数	数据范围
K1X000	1位数	0～9
K2X000	2位数	00～99
K3X000	3位数	000～999
K4X000	4位数	0000～9999
K5X000	5位数	00000～99999
K6X000	6位数	000000～999999
K7X000	7位数	0000000～9999999
K8X000	8位数	00000000～99999999

11.6.3 注意要点

1. S 的内容必须为BCD 码格式，否则不能得到正确的结果。BCD 是用 4 位二进制数来表示 1 位十进制数中的 0~9 这10 个数的方法。

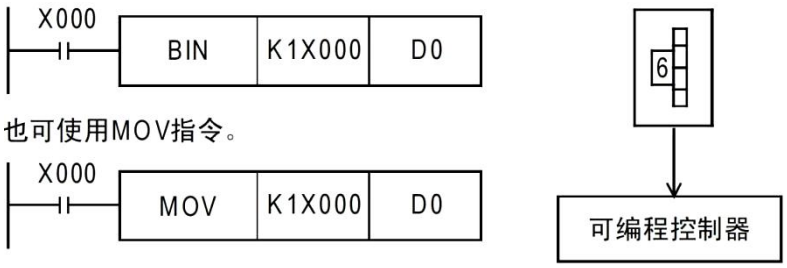
2. 为简便理解，可以视作将十六进制数值(不能有 A，B，C，D，E，F)直接拿出来变为十进制数值(如H1234 转换后变为 K1234)。

3. 四则运算(+-×÷) 和加一，减一指令等在可编程控制器内的运算都以BIN(二进制数)执行。

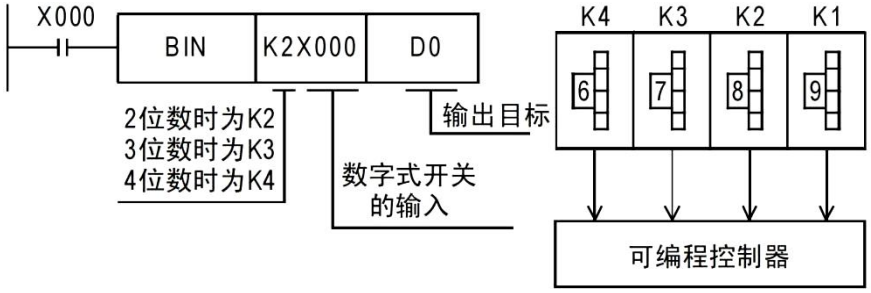
- 在将BCD(10 进制数)的数字式开关信息读入可编程控制器中时，使用 BIN(BCD→BIN)转换传送指令。
- 在向BCD(10 进制数)的7 段数码管显示进行输出时，使用 BCD(BIN→BCD)的转换传送指令。

11.6.4 程序应用

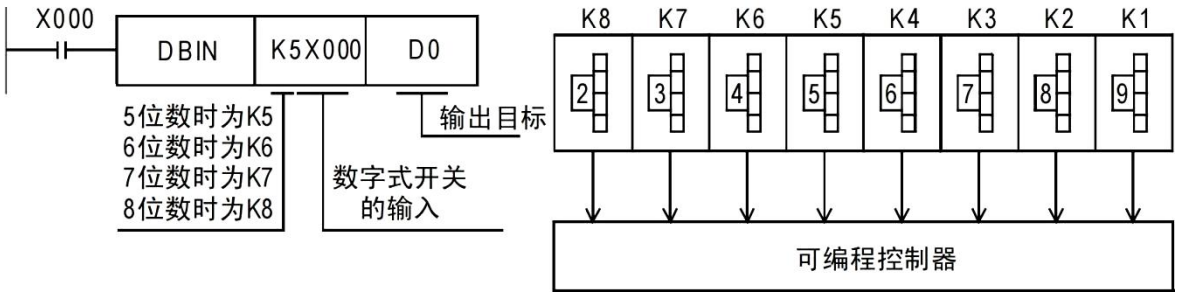
1. 1 位数字开关



2. 2-4 位数字开关



3. 5-8 位数字开关



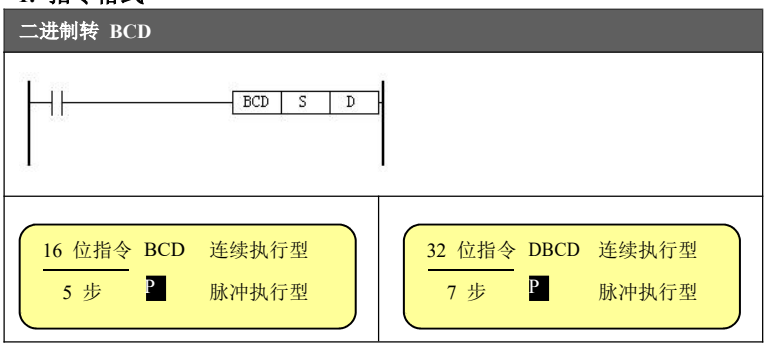
11.7 [BCD]二进制转 BCD

11.7.1 概要

将 BIN(2 进制数)转换成BCD(10 进制数)后传送的指令。

可编程控制器的运算按照 BIN(2 进制数)数据进行处理，在带 BCD 译码的 7 段码显示器中显示数值时,可使用本指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定源数据的软元件地址编号	16/32 位 BIN
D	指定目标软元件地址编号	BCD

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S							•	•	•	•	•	•	•	•	•	•	•			
D								•	•	•	•	•	•	•	•	•	•			

11.7.2 功能与动作说明

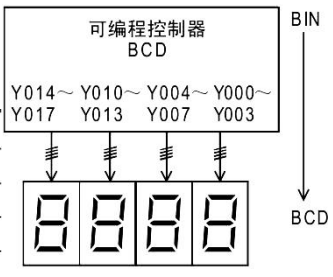
1. 16 位运算(BCD、BCDP)

将S 的BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到D 中。

- S 的数据可以转换成 K0～K9999 的 BCD(10 进制数)。
- S 和D 指定位数的时候，参考下表。



如右图，当指定目标软元件地址 D 为 K4Y000 的情况。



D	位数	数据范围
K1Y000	1位数	0～9
K2Y000	2位数	00～99
K3Y000	3位数	000～999
K4Y000	4位数	0000～9999

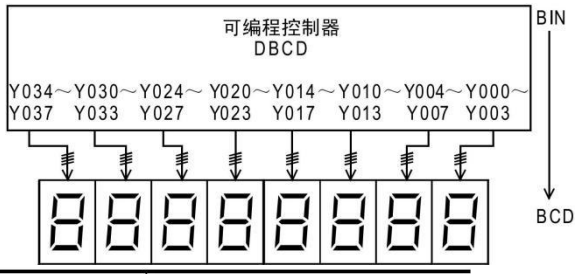
2. 32 位运算(DBCD、DBCDP)

将[S+1，S]的 BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到[D+1，D]中。

- [S+1，S]的数据，可以转换成 K0～K99999999 的BCD(10 进制数)。
- [S+1，S]和[D+1，D]指定位数的时候，参考下表。



如下图，当指定目标软元件地址 D 为 K8Y000 的情况。



[D+1, D]	位数	数据范围
K1Y000	1位数	0～9
K2Y000	2位数	00～99
K3Y000	3位数	000～999
K4Y000	4位数	0000～9999
K5Y000	5位数	00000～99999
K6Y000	6位数	000000～999999
K7Y000	7位数	0000000～9999999
K8Y000	8位数	00000000～99999999

11.7.3 注意要点

1. BCD 是用 4 位二进制数来表示 1 位十进制数中的 0～9 这10 个数的方法。

2. 为简便理解，可以视作将十进制数值拿出来直接变为十六进制数值(如K1234 转换后变为 H1234)。

3. 使用 16 位指令时，S 指定范围为 0～9999。小于 0 时，转换结果为 0；大于 9999 时，转换结果为 H9999。

4. 使用 32 位指令时，S 指定范围为 0～99999999。小于 0 时，转换结果为 0；大于 99999999 时，转换结果为H99999999。

5. 四则运算(+-×÷)和加一，减一指令等可编程控制器内的运算都以 BIN(2 进制数)执行。

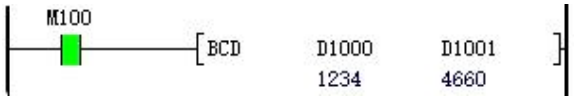
• 在将 BCD(10 进制数)的数字式开关信息读入可编程控制器时，使用BIN（BCD→BIN）转换传送指令。

• 在向 BCD(10 进制数)的7 段码显示器进行输出时，使用BCD（BIN→BCD）的转换传送指令。

11.7.4 程序应用

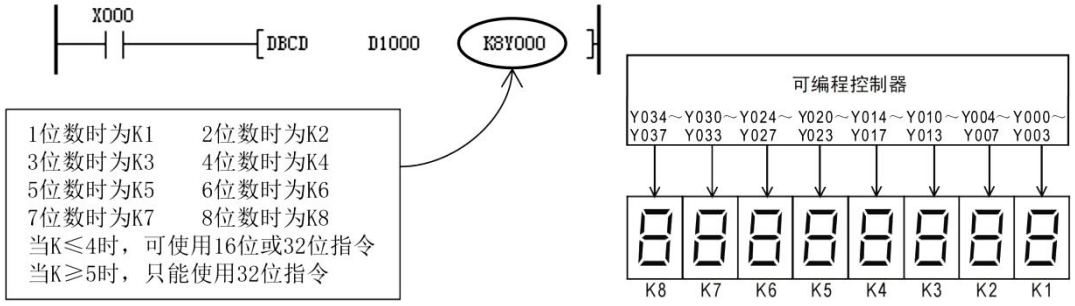
1. 数制转换

运行如下梯形图。M100 触点导通，执行二进制整数值转 BCD 码的指令，此时 D1000=1234，因为H1234 其十进制表达为 4660，所以转换后结果D1001=4660。



2. 7 段数码管显示情况

运行如下梯形图，用 8 位7 段数码管显示[D1001、D1000]的数值。若 D1000=K12345678，当X0 导通时，右图中 8 位7 段数码管显示为：12345678。

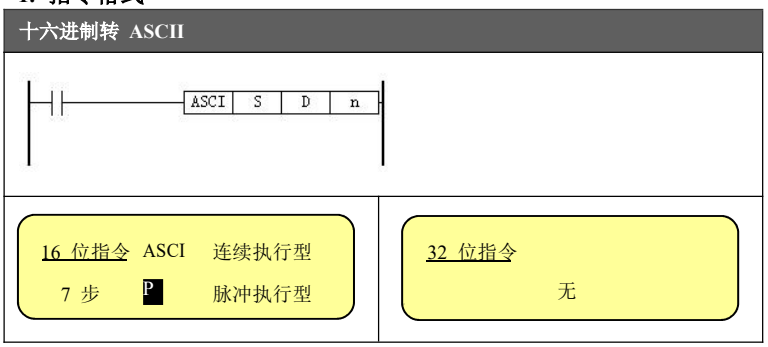


11.8 [ASCI][HEX→ASCII]十六进制 HEX 转 ASCII

11.8.1 概要

将指定软元件中的内容转换为 ASCII 码。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定源数据的软元件地址编号	2 位 HEX
D	指定目标软元件的首地址编号	ASCII
n	指定转换的 ASCII 码字符个数	16 位 BIN

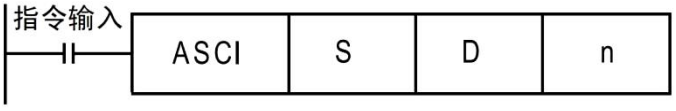
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S											•	•	•	•			•			
D											•	•	•	•			•			
n																		•	•	

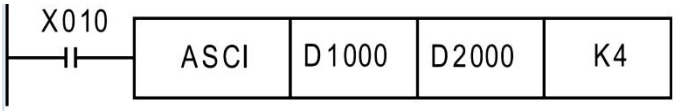
11.8.2 功能和动作说明

将S 开始的软元件中保存的各位 HEX 数据转换成 ASCII 码，然后传送到D 开始的各软元件的低 8 位(字节) 中的指令。用 n 指定转换的位数(字符数)。

D 的高八位为 0。



当为如下程序时，转换结果如图



S 开始的软元件D1000 = H0ABC D1001 = H1234 D1002 = H5678

指定位数(字符数)及转换结果

n / D	K1	K2	k3	k4	k5	k6	k7	k8	k9
D2000	[C]	[B]	[A]	[0]	[0]	[0]	[0]	[0]	[0]
D2001		[C]	[B]	[A]	[A]	[A]	[A]	[A]	[A]
D2002			[C]	[B]	[B]	[B]	[B]	[B]	[B]
D2003				[C]	[C]	[C]	[C]	[C]	[C]
D2004					[4]	[3]	[2]	[1]	[1]
D2005		不变化				[4]	[3]	[2]	[2]
D2006							[4]	[3]	[3]
D2007								[4]	[4]
D2008									[8]

ASCII码转换

[0]	=	H30	[1]	=	H31	[5]	=	H35
[A]	=	H41	[2]	=	H32	[6]	=	H36
[B]	=	H42	[3]	=	H33	[7]	=	H37
[C]	=	H43	[4]	=	H34	[8]	=	H38

n=K2时的位结构

D1000＝H0ABC

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D2000 ＝ B的ASCII码 ＝H42

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
								4				2			

D2001＝C的ASCII码 ＝H43

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
								4				3			

ASCII码转换表								
数值	H0	H1	H2	H3	H4	H5	H6	H7
ASCII码	H30	H31	H32	H33	H34	H35	H36	H37
数值	H8	H9	HA	HB	HC	HD	HE	HF
ASCII码	H38	H39	H41	H42	H43	H44	H45	H46

11.8.3 注意要点

1. D 的低 8 位接收 ASCII 数据，每 4 个D 为一组。

2. 指令执行的n（传送点数）最少为 1 个点，最多为 256 个点，n 设置过大会影响定时器的精确度。

11.8.4 程序应用



S 中数据：

D1000	D1001	D1002	D1003	D1004
H1234	H4567	H789A	HABCD	HDEF0

结果如图所示：

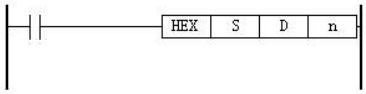
D2000	D2001	D2002	D2003	D2004	D2005	D2006	D2007	D2008	D2009
[1]	[2]	[3]	[4]	[4]	[5]	[6]	[7]	[7]	[8]
D2010	D2011	D2012	D2013	D2014	D2015	D2016	D2017	D2018	D2019
[9]	[A]	[A]	[B]	[C]	[D]	[D]	[E]	[F]	[0]

11.9 [HEX][ASCII→HEX]ASCII 转十六进制 HEX

11.9.1 概要

将指定软元件中的内容(必须为ASCII 码格式)转换为十六进制数。

1. 指令格式

ASCII 转十六进制	
	
16 位指令 HEX 连续执行型 7 步 P 脉冲执行型	32 位指令 无

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件的首地址编号	ASCII
D	指定目标软元件的首地址编号	2 位 HEX
n	指定转换的 ASCII 码字符个数	16 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S											•	•	•	•			•			
D											•	•	•	•			•			
n																		•	•	

11.9.2 功能与动作说明

将源数据 S 的低 8 位(字节)中的 ASCII 字符码转换成 HEX 数据，然后按照每 4 位数的方式传送到目标软元件D 中。用个数 n 指定要转换的字符数。



若将源数据的 S 软元件的首地址设为 D2000,目标软元件的首地址设为 D1000。以 D2000 位首地址的转换源数据设定值见下表：

	ASCII码	HEX转换
D2000	H30	H0
D2001	H41	HA
D2002	H42	HB
D2003	H43	HC
D2004	H31	H1
D2005	H32	H2
D2006	H33	H3
D2007	H34	H4
D2008	H35	H5
D2009	H36	H6

对于不同的 ASCII 码字符个数 n，对应的以D1000 位首地址的目标软元件转换结果见下表：

n	D1002	D1001	D1000
1	不变化		H0000
2			H000A
3			H00AB
4			H0ABC
5		H0001	H0ABC
6		H0012	H0ABC
7		H0123	H0ABC
8		H1234	H0ABC
9	H0005	H1234	H0ABC
10	H0056	H1234	H0ABC

11.9.3 注意要点

1. S 的低 8 位作为 ASCII 码转成 HEX 数据，每 4 位向 D 进行传送。转换的字符数用 n 指定。
2. S 内容必须设置高 8 位为 0，低 8 位为 ASCII 码转换表的 ASCII 码值 (参考 11.8 章节)，即 S 的内容在 H0030～H0045 之间时，否则转换结果不正确。
3. 设置 1≤n≤256。

11.9.4 程序应用

1. 将4 个数 ASCII 码转为 HEX 数据，存入到一个 D 寄存器内。梯形图如下：



当 X000 触点闭合，执行ASCII 转十六进制指令。

若 D1100=H0031，D1101=H0032，D1102=H0033，D1103=H0034。

执行结果：D1000=H1234。

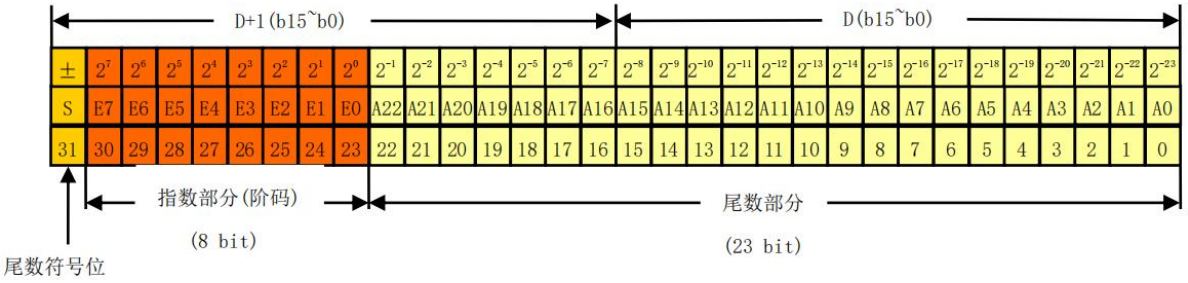
12. 浮点数运算指令

在本章中，主要介绍浮点数的转换、比较、四则运算、开方运算、三角函数等应用指令。

助记符、名称	功能	章节
DECMP	浮点数比较	12.1
DEZCP	浮点数区间比较	12.2
DEADD	浮点数加法	12.3
DESUB	浮点数减法	12.4
DEMUL	浮点数乘法	12.5
DEDIV	浮点数除法	12.6
DEBIN	十进制浮点数转二进制浮点数	12.7
DEBCD	二进制浮点数转十进制浮点数	12.8
DESQR	浮点数开方	12.9
SIN	浮点数 SIN 运算	12.10
COS	浮点数 COS 运算	12.11
TAN	浮点数 TAN 运算	12.12
ASIN	浮点数 ASIN 运算	12.13
ACOS	浮点数 ACOS 运算	12.14
ATAN	浮点数 ATAN 运算	12.15
RAD	浮点数 RAD 运算	12.16
DEG	浮点数 DEG 运算	12.17

单精度浮点数格式

- 用来表示带有小数部分的实数，一般用于科学计算。
- 占用4 个字节（32 位）存储空间，包括符号位 1 位，阶码8 位，尾数23 位。
- 其数值范围为-3.4E038～3.4E038，单精度浮点数最多有 7 位十进制有效数字， 指数用“E”表示。



$$二进制浮点数(实数)=\pm\left(2^0+A22\times2^{-1}+A21\times2^{-2}+...+A0\times2^{-23}\right)\times2^{(E7\times2^7+E6\times2^6+...+E0\times2^0)}/2^{127}$$

(例如) A22=1,A21=0,A20=1,A19～A0=0,E7=1,E6～E1=0,E0=1，

$$二进制浮点数(实数)=\pm\left(2^0+1\times2^{-1}+0\times2^{-2}+...+0\times2^{-23}\right)\times2^{(1\times2^7+0\times2^6+...+1\times2^0)}/2^{127}$$

$$=\pm1.625\times2^{129}/2^{127}=\pm1.625\times2^2=\pm6.5$$

正负由 b31 的符号位决定，0 为正，1 为负。

注意：32 位整数和 32 位单精度浮点数的存储格式不一样，整数运算和浮点数不能混用，须用（D）FLT 和（D）INT 指令转换之后，才能切换整数运算和浮点数运算指令。为避免结果出现较大偏差，建议一般在结果处才进行相互转换。

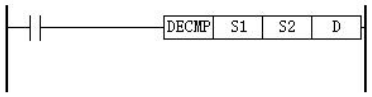
12.1 [DECMP]浮点比较

12.1.1 概要

比较 2 个数据(2 进制浮点数)，将结果(大于、等于或小于)输出到位软元件(3 点)中的指令。

1. 指令格式

浮点比较



16 位指令	32 位指令 DECMP 连续执行型
无	10 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定被比较的数据或软元件地址编号	32 位二进制浮点
S2	指定比较源的数据或软元件地址编号	32 位二进制浮点
D	指定比较输出结果的软元件地址编号	位

3. 对象软元件

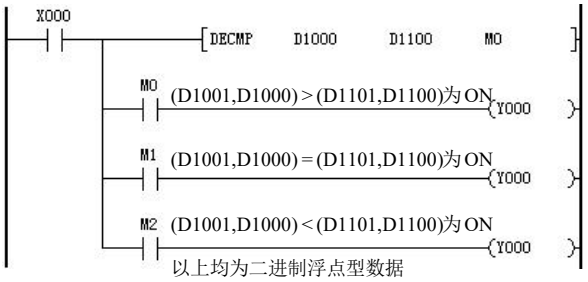
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
D		•	•			•											•			

12.1.2 功能和动作说明

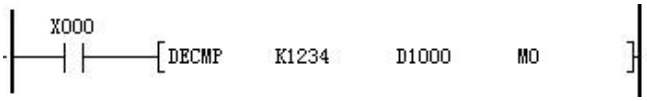
1. 32 位运算(DECMP、DECMPP)

将比较值[S1+1，S1]和比较源[S2+1，S2]作为浮点数数据进行比较，然后根据比较的结果(大于、等于、小于)将 D、D +1、D +2 中的对应位置 ON，另外两个位置 OFF。

● [S1+1，S1]、[S2 +1，S2]中指定了常数(K、H)时，会自动将数值从 BIN 转换成2 进制浮点数后再处理。



- 比较(D1001,D1000)和(D1101,D1100)当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制浮点型处理。
- 作为目标地址假如指定M0，如左记一样M0 (D1001,D1000)＞(D1101,D1100)为 ON M1、M2 被自动占有。
- 在执行 DECMP 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。



➢ (K1234):(D1001,D1000)→(M0,M1,M2)
常数 1234 自动转换成二进制浮点数。

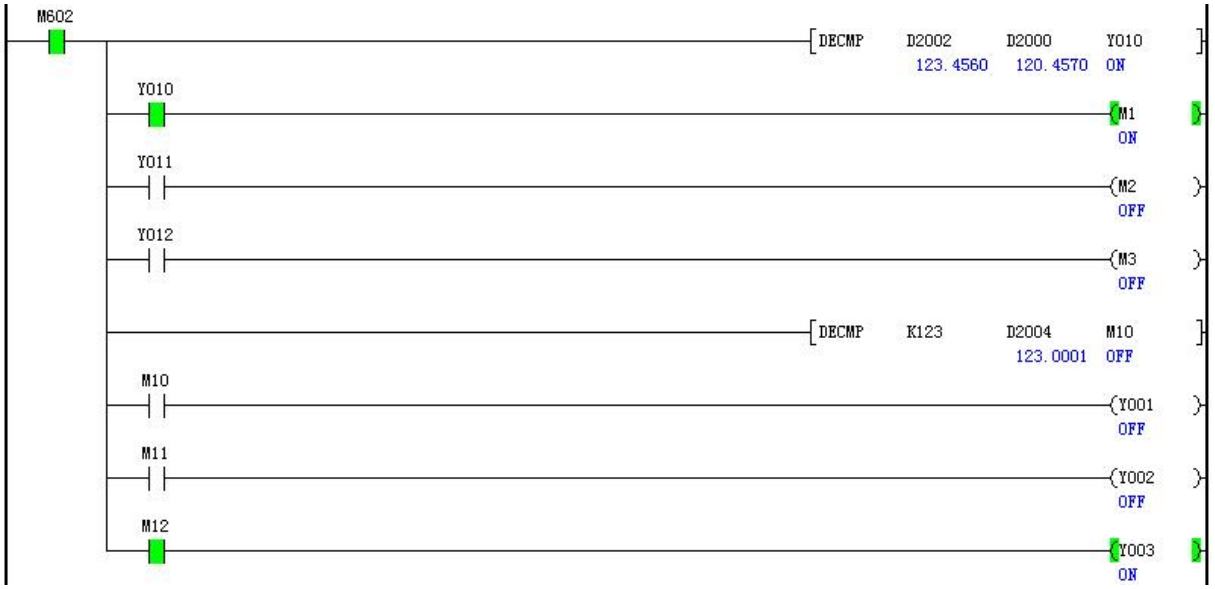
12.1.3 注意要点

1. 软元件的占用点数

D 占用 3 点。[D ， D+1，D +2]

请注意不要与用于其他用途的软元件重复。

12.1.4 程序应用



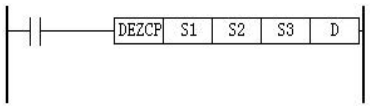
12.2 [DEZCP]浮点区域比较

12.2.1 概要

将上下 2 点的比较范围和数据(2 进制浮点数)进行比较，根据其结果输出到位软元件(3 点)中的指令。

1. 指令格式

浮点区域比较



16 位指令	32 位指令 DEZCP 连续执行型
无	13 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	32 位二进制浮点
S2	指定比较基准上限的数据或软元件地址编号	32 位二进制浮点
S3	指定当前数据或软元件地址编号	32 位二进制浮点
D	指定比较输出结果的软元件地址编号	位

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
S3													•	•			•	•	•	
D		•	•			•											•			

12.2.2 功能和动作说明

1. 32 位运算(DEZCP、DEZCPP)

将比较值[S1+1，S1](下区间比较值)、[S2 +1，S2](上区间比较值)和比较源[S3+1，S3]作为浮点数数据进行比较，然后根据结果(小于、等于或大于)将D 、D +1、D +2 中的对应一位置ON，另外两个位置 OFF。

● 在[S1+1，S1]、[S2+1，S2]、[S3 +1，S3]中指定了常数(K、H)时，会自动将数值转换成 2 进制浮点数后再处理。



- 比 较 (D1001,D1000) 和 (D1101,D1100) 与 (D1201,D1200)当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制浮点型处理。
- 作为目标地址假如指定 M0，如左记一样 M0、M1、M2 被自动占有。
- 在执行 DEZCP 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。



➢ (K-1000):(D1201,D1200):(K2000)→(M0,M1,M2)
常数-1000 和 2000 自动转换成二进制浮点数。

12.2.3 注意要点

1. 软元件的占用点数

D 的软元件占用 3 点。

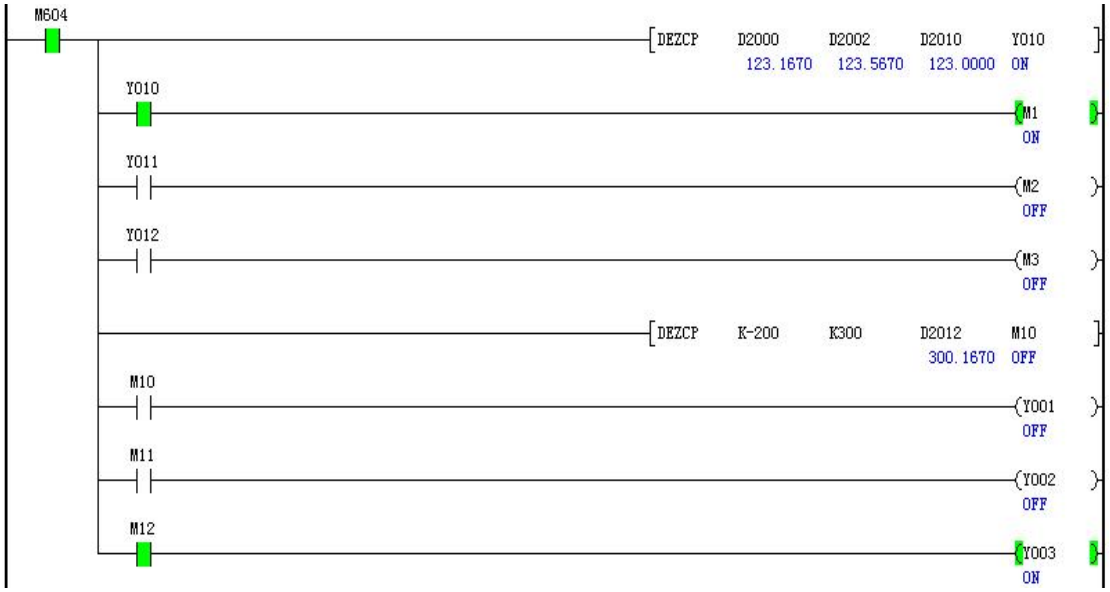
请注意不要与用于其他用途的软元件重复。

2. 关于 S1 和 S2 的比较数据

比较数据的大小关系请设置为[S1+1，S1]≦[S2 +1，S2]。

[S1+1，S1]＞[S2 +1，S2]的情况时，[S2 +1，S2]的数值视为与[S1 +1，S1]相同，从而进行比较。

12.2.4 程序应用



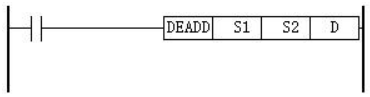
12.3 [DEADD]浮点加法

12.3.1 概要

2 个 2 进制浮点数加法运算的指令。

1. 指令格式

浮点加法



16 位指令	32 位指令 DEADD 连续执行型
无	10 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定进行加法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行加法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存加法运算结果的软元件地址编号	32 位二进制浮点

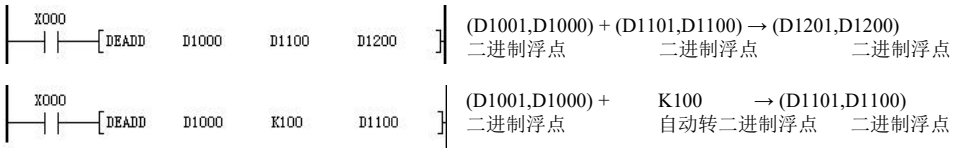
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
D													•	•			•			

12.3.2 功能和动作说明

1. 32 位运算(DEADD、DEADDP)

将[S1+1， S1]和[S2+1，S2]的 2 进制浮点数数据进行加法运算，并将其运算结果以 2 进制浮点数形式传送到[D +1，D]中。



12.3.3 注意要点

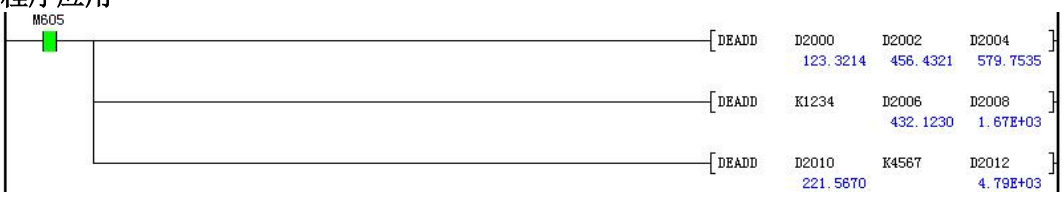
1 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

2 指定了同一软元件时

[S1+1，S1]以及[S2+1，S2]和[D+1，D]中也可以指定同一个软元件编号。

此时，如使用连续执行型的指令(DEADD)，则每一个运算周期其加法运算的结果都会改变，请务必注意。

12.3.4 程序应用



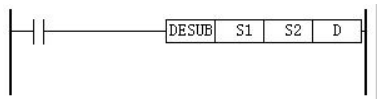
12.4 [DESUB]浮点减法

12.4.1 概要

2 个 2 进制浮点数减法运算的指令。

1. 指令格式

浮点减法



16 位指令	无

32 位指令	DESUB	连续执行型
10 步	P	脉冲执行型

2. 设定数据

操作数	作用	类型
S1	指定进行减法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行减法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存减法运算结果的软元件地址编号	32 位二进制浮点

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
D													•	•			•			

12.4.2 功能和动作说明

1. 32 位运算(DESUB、DESUBP)

从[S1 +1，S1]中减去[S2 +1，S2]的2 进制浮点数数据，并将其运算结果以2 进制浮点数形式传送到[D +1，D]中。



(D1001,D1000) - (D1101,D1100) → (D1201,D1200)
二进制浮点 二进制浮点 二进制浮点



(D1001,D1000) - K100 → (D1101,D1100)
二进制浮点 自动转二进制浮点 二进制浮点

12.4.3 注意要点

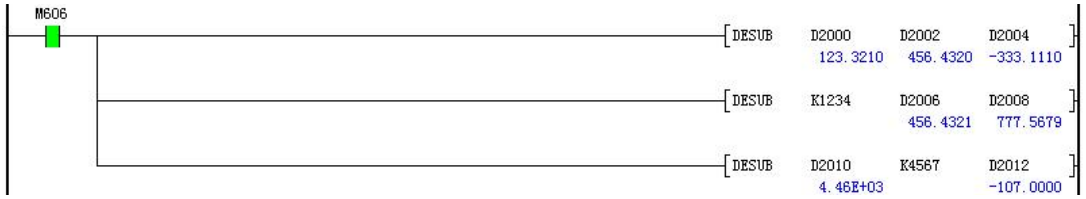
1 常数K/H 被指定为源数据时，自动转换成二进制浮点值处理。

2 指定了同一软元件时

[S1+1，S1]以及[S2+1，S2]和[D+1，D]中也可以指定同一个软元件编号。

此时，如使用连续执行型的指令(DESUB)，则每一个运算周期其减法运算的结果都会改变，请务必注意。

12.4.4 程序应用



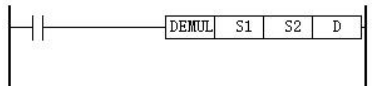
12.5[DEMUL]浮点乘法

12.5.1 概要

2 个 2 进制浮点数乘法运算的指令。

1.指令格式

浮点乘法



16 位指令　　无

32 位指令　DEMUL　连续执行型

10 步　　P　　脉冲执行型

2.设定数据

操作数	作用	类型
S1	指定进行乘法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行乘法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存乘法运算结果的软元件地址编号	32 位二进制浮点

3.对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
D													•	•			•			

12.5.2 功能和动作说明

1. 32 位运算(DEMUL、DEMULP)

将[S1 +1，S1]和[S2 +1，S2]的2 进制浮点数数据相乘，并将其运算结果以 2 进制浮点数形式传送到[D+1，D]中。



(D1001,D1000)×(D1101,D1100) → (D1201,D1200)
二进制浮点　　二进制浮点　　二进制浮点



(D1001,D1000) × K100 → (D1101,D1100)
二进制浮点　　自动转二进制浮点　　二进制浮点

12.5.3 注意要点

1 常数K/H 被指定为源数据时，自动转换成二进制浮点值处理。

2 指定了同一软元件时

[S1+1，S1]以及[S2+1，S2]和[D+1，D]中也可以指定同一个软元件编号。

此时，如使用连续执行型的指令(DEMUL)，则每一个运算周期其减法运算的结果都会改变，请务必注意。

3 负数乘以 0 时，结果中符号位不会被清除，其它位置清零。

12.5.4 程序应用

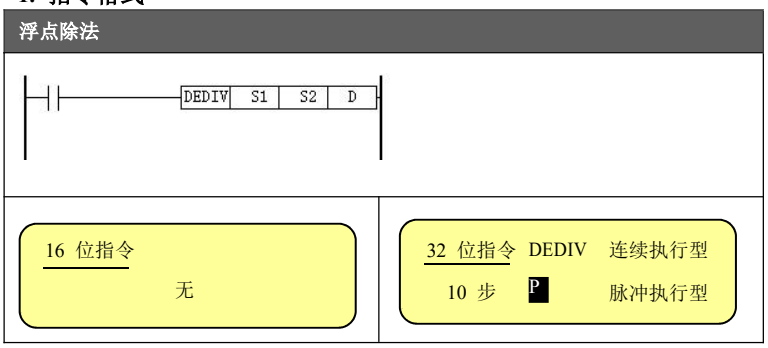

M607
DEMUL D2000 D2002 D2004
123.4560 2.0000 246.9120
DEMUL K1234 D2006 D2008
6.1230 7.56E+03
DEMUL D2010 K4567 D2012
-6.4560 -2.95E+04
DEMUL K-1234 D2014 D2016
-8.4560 1.04E+04
DEMUL D2018 K-4567 D2020
7.4560 -3.41E+04

12.6[DEDIV]浮点除法

12.6.1 概要

2 个 2 进制浮点数除法运算的指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S1	指定进行除法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行除法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存除法运算结果的软元件地址编号	32 位二进制浮点

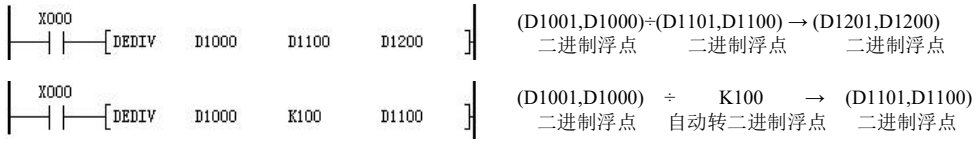
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•	•	•	
S2													•	•			•	•	•	
D													•	•			•			

12.6.2 功能和动作说明

1. 32 位运算(DEDIV、DEDIVP)

将[S1+1，S1]和[S2 +1，S2]的2 进制浮点数数据相除，并将其运算结果以 2 进制浮点数形式传送到[D+1，D]中。



12.6.3 注意要点

1 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

2 指定了同一软元件时

[S1+1，S1]以及[S2+1，S2]和[D+1，D]中也可以指定同一个软元件编号。

此时，如使用连续执行型的指令(DEDIV)，则每一个运算周期其减法运算的结果都会改变，请务必注意。

3 除数 S2=0 时：

被除数>0，则结果为单精度二进制浮点最大限值(3.4E038)。 被除数≤0，则结果为单精度二进制浮点最小限值(-3.4E038)。

12.6.4 程序应用



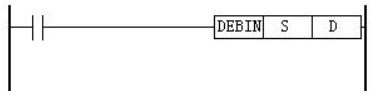
12.7[DEBIN]浮点十进制转浮点二进制

12.7.1 概要

将软元件中的 10 进制浮点数转换成 2 进制浮点数的指令。

1. 指令格式

浮点十进制转浮点二进制



16 位指令	32 位指令 DEBIN 连续执行型
无	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定源数据的软元件地址编号	32 位十进制浮点
D	指定目标软元件地址编号	32 位二进制浮点

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•			
D													•	•			•			

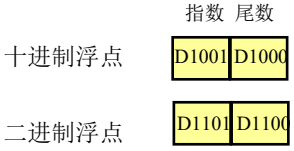
12.7.2 功能和动作说明

1. 32 位运算(DEBIN、DEBINP)

将[S+1，S]的 10 进制浮点数转换成 2 进制浮点数后，传送到[D+1，D]中。



(D1001,D1000) → (D1101,D1100)
十进制浮点 二进制浮点



浮点运算在可编程控制器内部全部以二进制浮点值为基础执行。但是，由于二进制浮点值为不易判别的数值，所以可通过能将其变换为十进制浮点值的外部设备很容易地监控。现在的主流的触摸屏和上位机，还有我们的编程软件，都支持二进制浮点值的显示。

。

使用 DEBIN 指令，可以使含有小数点的数值直接转换为二进制浮点数。

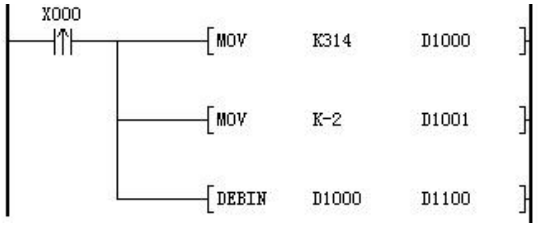
例：3.14 的二进制浮点转换。

3.14=314×10-2(十进制浮点)

将314 传送到(D1000)

将-2 传送到(D1001)。

利用 DEBIN 指令进行转换。(D1101,D1100)→3.14(2 进制浮点)



12.7.3 注意要点

1. 指数的输入范围

指数的输入范围通常为±38，结合尾数部分必须保证其转换值在±3.4E038 范围之内。

12.7.4 程序应用



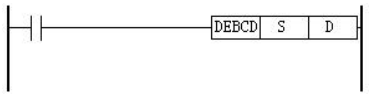
12.8[DEBCD]浮点二进制转浮点十进制

12.8.1 概要

将软元件中的 2 进制浮点数转换成 10 进制浮点数的指令。

1.指令格式

浮点十进制转浮点二进制



16 位指令	32 位指令 DEBCD 连续执行型
无	7 步 P 脉冲执行型

2、设定数据

操作数	作用	类型
S	指定源数据的软元件地址编号	32 位二进制浮点
D	指定目标软元件地址编号	32 位十进制浮点

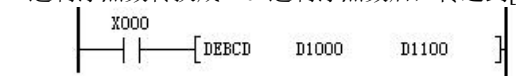
3、对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•			
D													•	•			•			

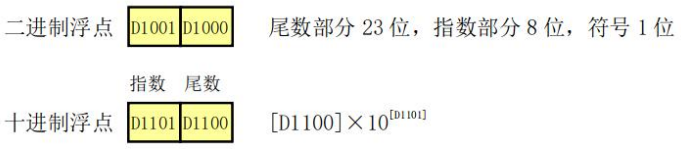
12.8.2 功能和动作说明

1. 32 位运算(DEBCD、DEBCDP)

将[S+1、S]的 2 进制浮点数转换成 10 进制浮点数后，传送到[D +1，D]中。



(D1001,D1000) → (D1101,D1100)二进制浮点　　十进制浮点



12.8.3 注意要点

1. 指数的输入范围

指数的输入范围通常为±38，结合尾数部分必须保证其转换值在±3.4E038 范围之内。

2. 尾数的精度

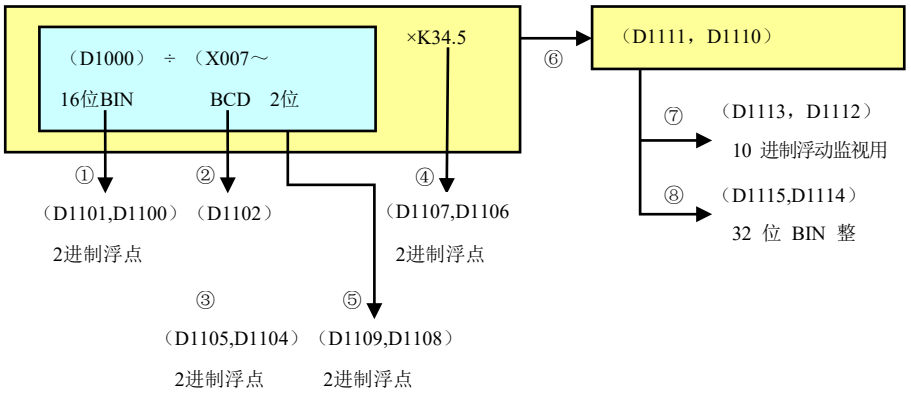
尾数保留 4 位有效数字。

12.8.4 程序应用

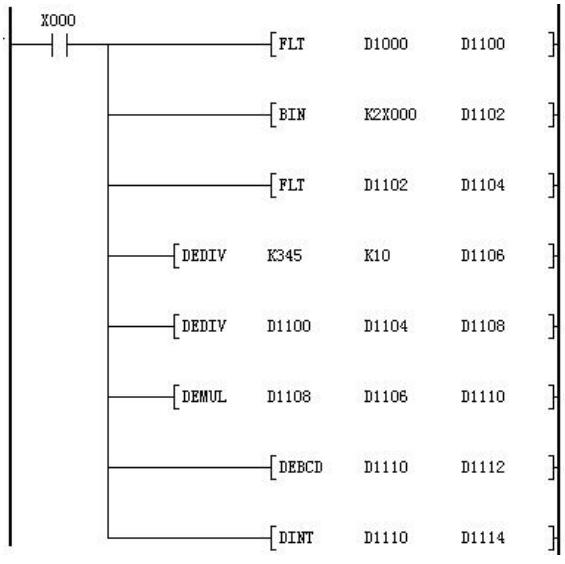


浮点综合运算示例

浮点运算综合应用的示例如下所示。



用梯形图实现上述结构图的运算过程：



- （D1000）→（D1101,D1100）
  BIN 2 进制浮点
- （X000～X007）→（D1102）
  BCD BIN
- （D1102）→（D1105,D1104）
  BIN2 进制浮点
- （K345 ÷ K10）→（D1107,D1106）
  常数自动转浮点2进制浮点
- （D1101,D1100）÷（D1105,D1104）→（D1109,D1108）
  2 进制浮点 2 进制浮点 2 进制浮点
- （D1109,D1108）×（D1107,D1106）→（D1111,D1110）
  2 进制浮点 2 进制浮点 2 进制浮点
- （D1110）→（D1113，D1112）
  2 进制浮点 10 进制浮动监视用
- （D1110） →（D1115，D1114）
  2 进制浮点 32 位 BIN 整数

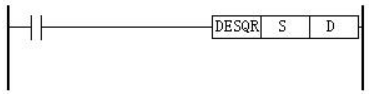
12.9 [DESQR]浮点开方

12.9.1 概要

2 进制浮点数开方(开根号)运算的指令。

1. 指令格式

浮点开方



16 位指令	32 位指令 DESQR 连续执行型
无	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行开方运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

3. 对象软元件

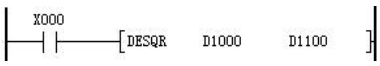
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.9.2 功能和动作说明

1. 32 位运算(DESQR、DESQRP)

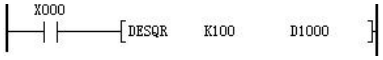
将[S +1，S]进行开方(开根号)运算(2 进制浮点数运算)后，将其结果传送到[D +1，D]中。

● 与整数型开方指令SQR 相比，其运算结果会保留小数位。



$\sqrt{(D1001,D1000)}$ → (D1101,D1100)

二进制浮点 二进制浮点



$\sqrt{K100}$ → (D1001,D1000)

自动转二进制浮点 二进制浮点

12.9.3 注意要点

1 常数K/H 被指定为源数据时，自动转换成二进制浮点值处理。

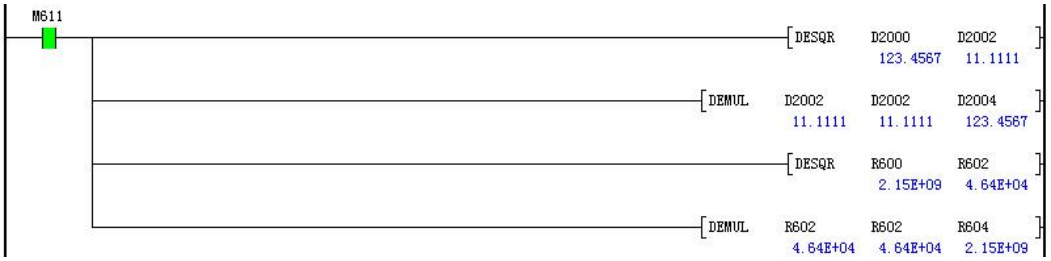
2 指定了同一软元件时[S+1，S]和[D+1，D]中也可以指定同一个软元件编号。

此时，如使用连续执行型的指令(DESQR)，则每一个运算周期其开方运算的结果都会改变，请务必注意。

3 非法使用的情况：

- ◆ S 不在单精度浮点数取值范围内(-3.4E038～3.4E038)，计算结果为 0。
- ◆ S＜0，运行将出错。

12.9.4 程序应用

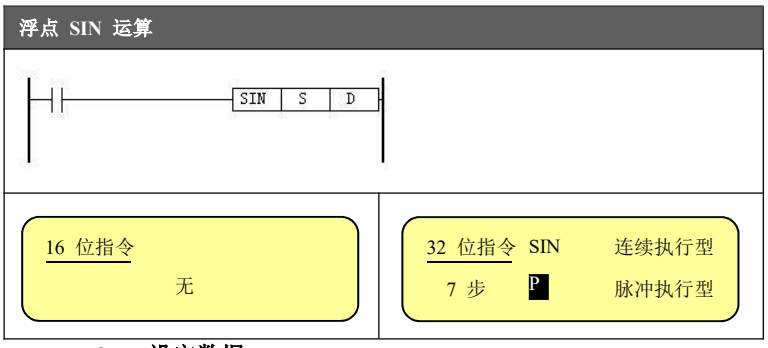


12.10 [SIN]浮点 SIN 运算

12.10.1 概要

求弧度(RAD)的 SIN 值的指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定进行 SIN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

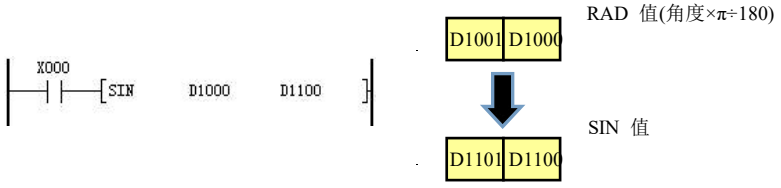
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.10.2 功能和动作说明

1. 32 位运算(SIN、SINP)

将[S+1，S]中指定的弧度值(2 进制浮点数)转换成 SIN 值(2 进制浮点数)后，传送到[D +1，D]中。



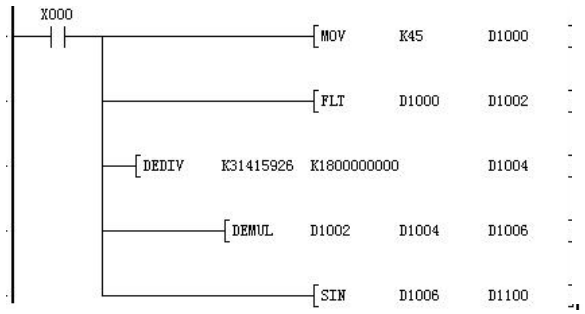
12.10.3 注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

12.10.4 程序应用

SIN 编程示例

求角度 45°的 SIN 值梯形图示例。



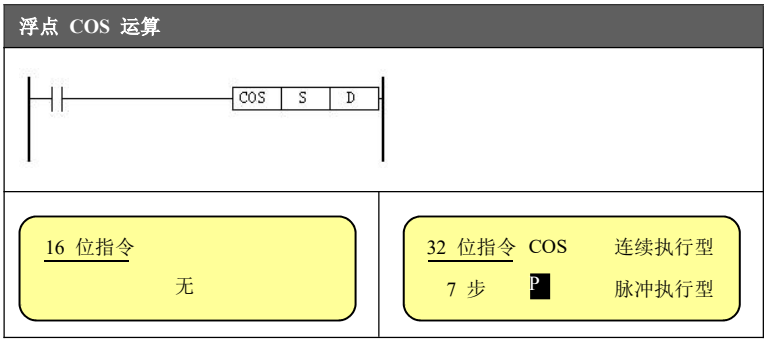
- 存入将要运算的角度
- 将运算角度转为二进制浮点
- 计算π÷180 二进制浮点数值
- 计算 RAD 值(角度×π÷180)
- RAD 值经过 SIN 变换得结果

12.11 [COS]浮点 COS 运算

12.11.1 概要

求弧度(RAD)的 COS 值的指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定进行 COS 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

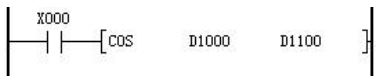
3. 对象软元件

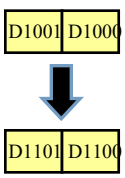
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.11.2 功能和动作说明

1. 32 位运算(COS、COSP)

将[S+1，S]中指定的弧度值(2 进制浮点数)转换成COS 值后，传送到[D +1，D]中。





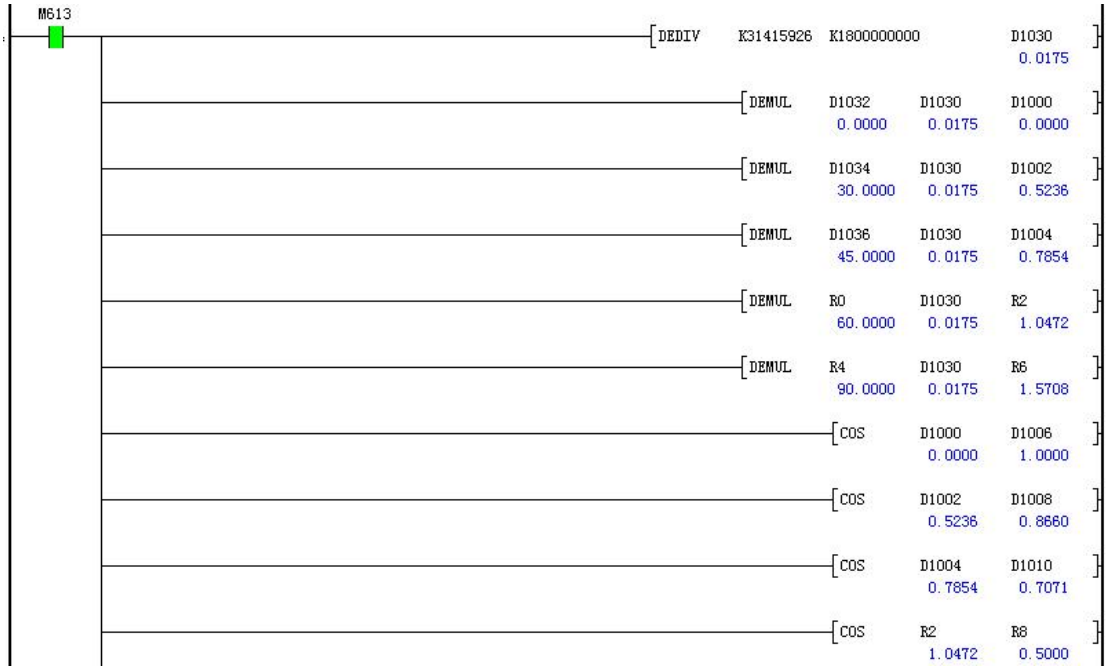
RAD 值(角度×π÷180)
指定二进制浮点值

COS 值
二进制浮点

12.11.3 注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

12.11.4 程序应用


M613
DEDIV K31415926 K1800000000 D1030 0.0175
DEMUL D1032 D1030 D1000 0.0000 0.0175 0.0000
DEMUL D1034 D1030 D1002 30.0000 0.0175 0.5236
DEMUL D1036 D1030 D1004 45.0000 0.0175 0.7854
DEMUL R0 D1030 R2 60.0000 0.0175 1.0472
DEMUL R4 D1030 R6 90.0000 0.0175 1.5708
COS D1000 D1006 0.0000 1.0000
COS D1002 D1008 0.5236 0.8660
COS D1004 D1010 0.7854 0.7071
COS R2 R8 1.0472 0.5000

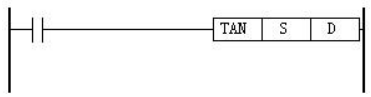
12.12 [TAN]浮点 TAN 运算

12.12.1 概要

求弧度(RAD)的 TAN 值的指令。

1. 指令格式

浮点TAN 运算



16 位指令	32 位指令 TAN 连续执行型
无	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行 TAN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

3. 对象软元件

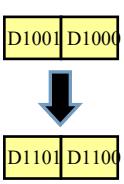
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.12.2 功能和动作说明

1. 32 位运算(TAN、TANP)

将[S+1，S]中指定的弧度值(2 进制浮点数)转换成TAN 值后，传送到[D+1，D]中。





RAD 值(角度×π÷180)

指定二进制浮点值

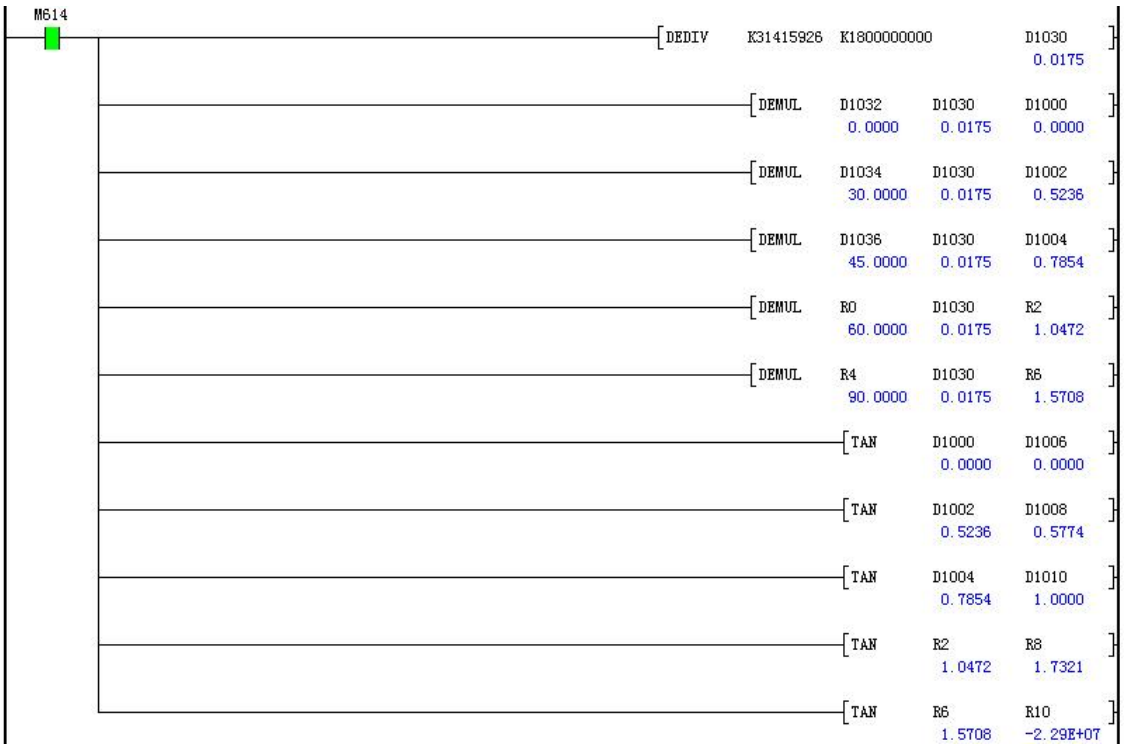
TAN 值

二进制浮点

12.12.3 注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

12.12.4 程序应用

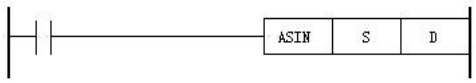

M614
DEDIV K31415926 K1800000000 D1030 0.0175
DEMUL D1032 D1030 D1000 0.0000 0.0175 0.0000
DEMUL D1034 D1030 D1002 30.0000 0.0175 0.5236
DEMUL D1036 D1030 D1004 45.0000 0.0175 0.7854
DEMUL R0 D1030 R2 60.0000 0.0175 1.0472
DEMUL R4 D1030 R6 90.0000 0.0175 1.5708
TAN D1000 D1006 0.0000 0.0000
TAN D1002 D1008 0.5236 0.5774
TAN D1004 D1010 0.7854 1.0000
TAN R2 R8 1.0472 1.7321
TAN R6 R10 1.5708 -2.29E+07

12.13 [ASIN]浮点 ASIN 运算

12.13.1 概要

执行 SIN^{-1}（反正弦）运算的指令。

1. 指令格式

浮点 ASIN 运算	
	
16 位指令 无	32 位指令 ASIN 连续执行型 7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行 ASIN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

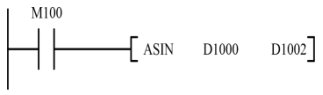
3. 对象软元件

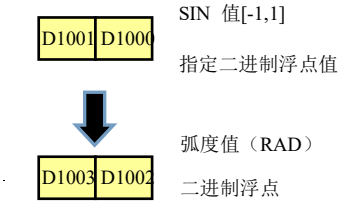
操作数种类	位软元件						字型软元件													其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.13.2 功能和动作说明

1. 32 位运算(ASIN、ASINP)

[S +1，S]的 SIN 值求出弧度，将运算结果保存到[D +1，D]中。此外，可以在 S 中直接指定实数。





12.13.3 注意要点

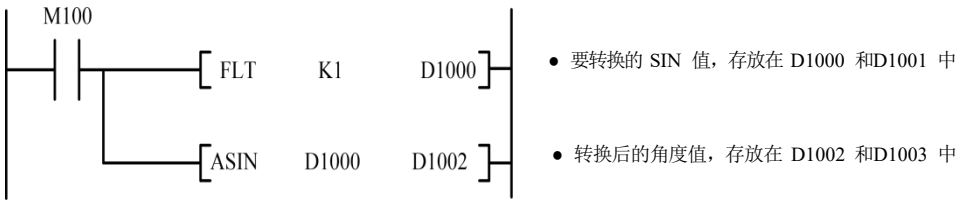
1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

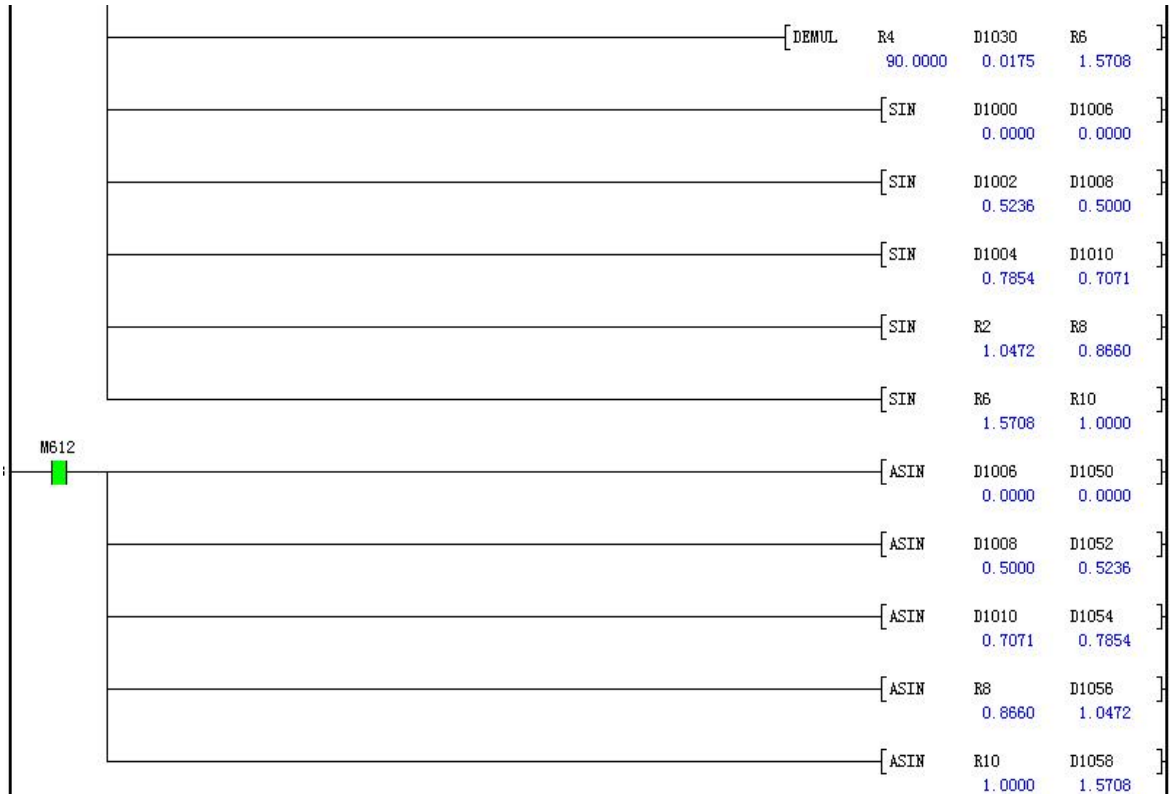
2.S 的取值范围为-1~1，若超出范围会导致控制器报错。

12.13.4程序应用

ASIN 编程示例

求SIN 值为 1 的 ASIN 弧度值梯形图示例。运算最后结果为D1002=1.5708.



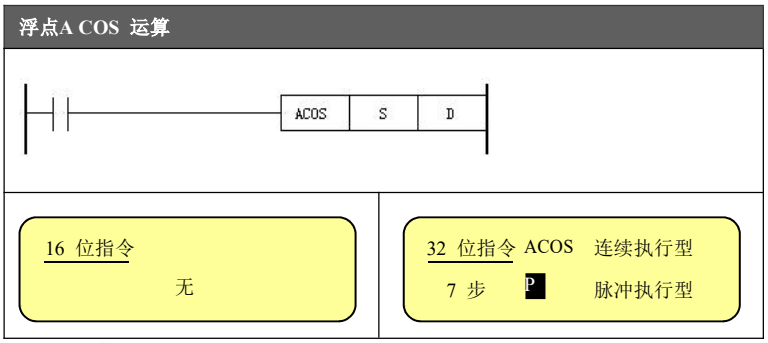


12.14 [ACOS]浮点 ACOS 运算

12.14.1 概要

执行 COS $^{-1}$（反余弦）运算的指令。

1. 指令格式



2. 设定数据

操作数	作用	类型
S	指定进行 ACOS 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

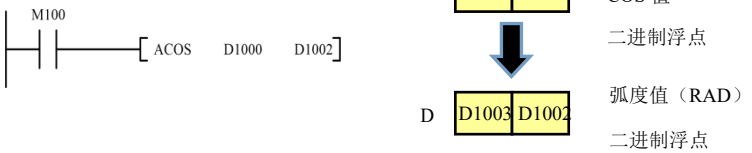
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.14.2 功能和动作说明

1. 32 位运算(ACOS、ACOSP)

[S +1，S]的 COS 值求出弧度，将运算结果保存到[D +1，D]中。此外，可以在 S 中直接指定实数。



12.14.3 注意要点

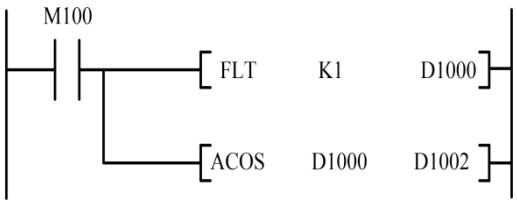
1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

2.S 的取值范围为-1~1，若超出范围会导致控制器报错。

12.14.4 程序应用

ACOS 编程示例

求COS 值为1 的ACOS 角度值梯形图示例。运算最后结果为 D1002=0.0000.



- 要转换的 COS 值，存放在 D1000 和D1002 中
- 转换后的角度值，存放在 D1002 和D1003 中



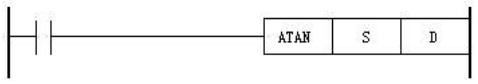
12.15 [ATAN]浮点ATAN 运算

12.15.1概要

执行 TAN^{-1}（反正切）运算的指令。

1. 指令格式

浮点A TAN 运算



16 位指令

无

32 位指令 ATAN 连续执行型

7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行 ATAN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

3. 对象软元件

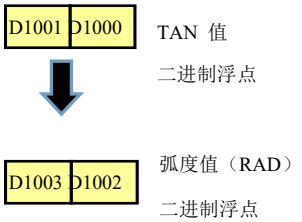
操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.15.2功能和动作说明

1. 32 位运算(ATAN、ATANP)

[S +1，S]的 TAN 值求出弧度，将运算结果保存到[D +1，D]中。此外，可以在 S 中直接指定实数。





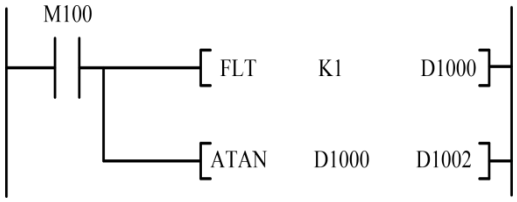
12.15.3注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

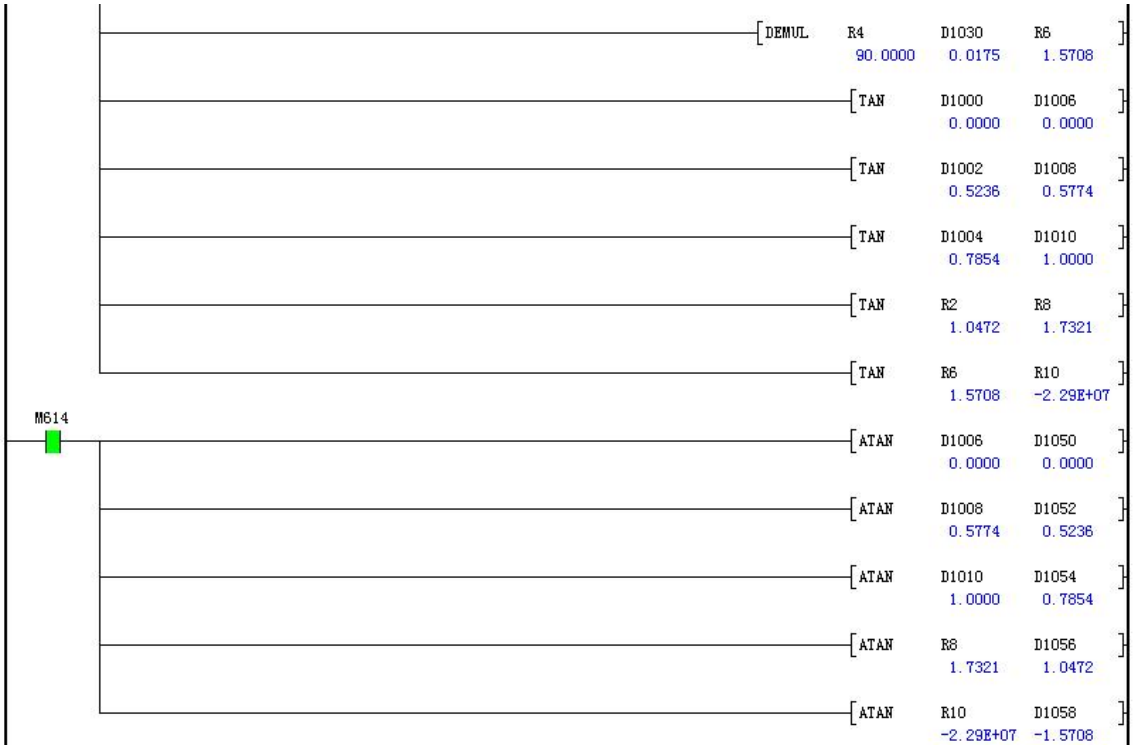
12.15.4程序应用

ATAN 编程示例

求TAN 值为 1 的 ATAN 角度值梯形图示例。运算最后结果为D1002=0.7854.



- 要转换的 TAN 值，存放在 D1000 和D1002 中
- 转换后的角度值，存放在 D1002 和D1003 中



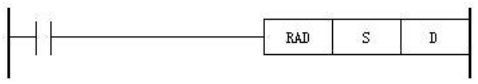
12.16 [RAD]浮点数RAD运算

12.16.1 概要

将角度单位的值转换成弧度(RAD)单位的指令。

1. 指令格式

浮点RAD 运算



16 位指令

无

32 位指令 RAD 连续执行型

7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行 RAD 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.16.2 功能和动作说明

1. 32 位运算(RAD、RADP)

[S +1，S]的单位从角度单位转换成弧度单位，并保存到[D +1，D]中。此外，可以在 S 中直接指定实数。



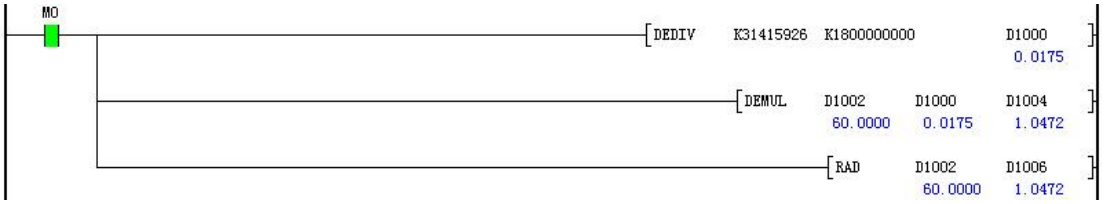
S D1001 D1000 角度值(RAD 弧度×180÷π) 二进制浮点值

D D1101 D1100 RAD 值(角度×π÷180) 二进制浮点值

12.16.3 注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

12.16.4 程序应用



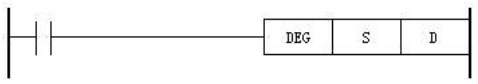
12.17 [DEG]浮点DEG运算

12.17.1 概要

将弧度单位的值转换成角度(DEG)单位的指令。

1. 指令格式

浮点 DEG 运算



16 位指令	32 位指令 DEG 连续执行型
无	7 步 P 脉冲执行型

2. 设定数据

操作数	作用	类型
S	指定进行 DEG 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

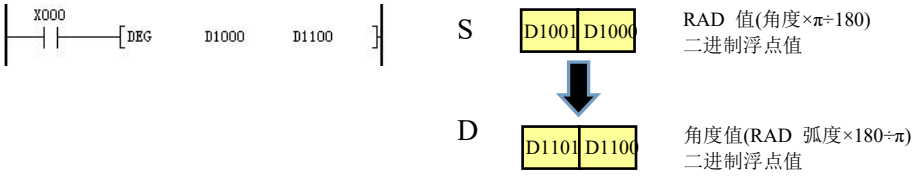
3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S													•	•			•	•	•	
D													•	•			•			

12.17.2 功能和动作说明

1. 32 位运算(DEG、DEGP)

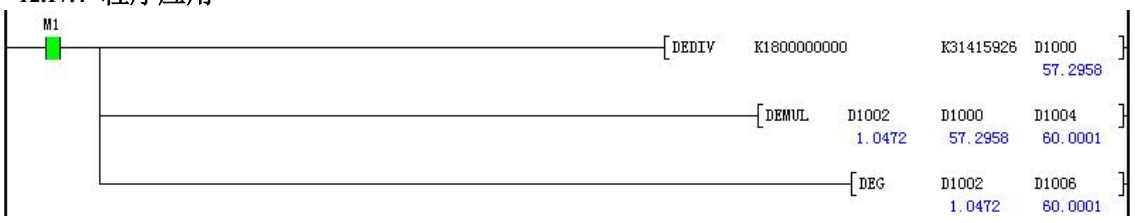
[S +1，S]的单位从弧度单位转换成角度单位，并保存到[D +1，D]中。此外，可以在 S 中直接指定实数。



12.17.3 注意要点

1. 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。

12.17.4 程序应用



13. 专用指令

在本章中，主要介绍 PID 运算指令。

助记符、名称	功能	章节
PID	PID 运算	13.1

13.1 [PID]PID 控制

13.1.1 概要

该指令用于执行根据输入的变化量而改变输出值的 PID 控制。

1. 指令格式

PID 控制



16 位指令	PID	连续执行型
9 步		脉冲执行型

32 位指令

无

2. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1													•	•			•			
S2													•	•			•			
S3													•	•			•			
D													•	•			•			

3.对象软元件

操作数	作用	类型
S1	设定目标值的软元件地址编号	16 位 BIN
S2	测定值的软元件地址编号	16 位 BIN
S3	设定控制参数的软元件首地址编号	16 位 BIN
D	输出值的软元件地址编号	16 位 BIN

13.1.2 功能和动作说明

1. 16 位运算(PID)

执行对目标值 S1、测量值S2 、参数S3 ～ S3+6 进行设定的程序后，每隔采样时间 S3 将运算结果(MV)保存到输出值 D 中。

功能与动作

- P-比例调节，I-积分调节，D-微分调节，应用于闭环控制系统。
- S3～S3+24 作为控制参数区被该指令占用，不可用作其他普通数据寄存器。
- 可通过 T 定时器或定时中断来设置 PID 指令的调节周期。
- PID 指令可在定时器中断、子程序、步进梯形图、跳转指令中使用，首次执行前必须清除偏差值(S3+7)。

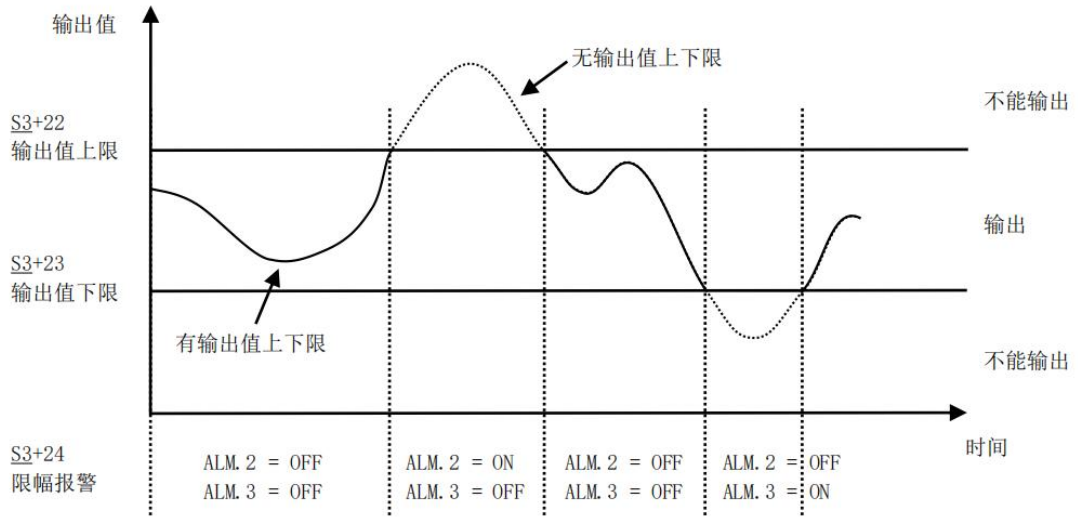
S3 参数设定定义

PID 指令运算参数设定表。

S3 + 偏移	名称	控制参数设置值	备注	读写
S3 + 0	采样时间	PID 控制采样时间，此参数决定 PID 运算的间隔时间，单位 ms。	梯形图需配合设定的时间执行指令。例如设置为 100ms 的采样时间，则梯形图可设计 100ms 定时器，定时器触点导通执行一次 PID 运算。	设置值
S3 + 1	运算方向	= 0，逆向计算；= 1，正向计算	推荐设置 1	设置值
S3 + 2	滤波系数	0～99 %	推荐设置 50，设置为 0 则无滤波	设置值
S3 + 3	比例增益	0～32767 %	推荐设置 2000	设置值
S3 + 4	积分时间	0～32767，单位 ms。	推荐设置 500	设置值
S3 + 5	微分增益	0～32767 %	推荐设置 0，无微分计算。	设置值
S3 + 6	微分时间	0～32767，单位 ms。	推荐设置 0，无微分计算。	设置值
S3 + 7(L)	偏差	内部计算使用	首次执行指令前清 0 二进制浮点格式(32 位)	显示值
S3 + 9(L)	偏差一阶倒数	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 11(L)	偏差二阶倒数	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 13(L)	滤波后测定值	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 15(L)	微分调整项	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 17(L)	PID 增量输出	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 19	PID 输出	内部计算使用	-	显示值
S3 + 20	上限报警	不能超过 32767	推荐设置值 20000	设置值
S3 + 21	下限报警	不能小于 0	推荐设置值 0	设置值
S3 + 22	上限幅	不能超过 32767	推荐设置值 10000	设置值
S3 + 23	下限幅	不能小于 0	推荐设置值 20	设置值
S3 + 24	报警输出	根据输出结果检测报警状态	助记符 ALM，按位显示上下限和上下限幅的报警状态。 ALM.Bit0 = ON，上限报警 ALM.Bit1 = ON，下限报警 ALM.Bit2 = ON，上限幅状态 ALM.Bit3 = ON，下限限幅状态	显示值

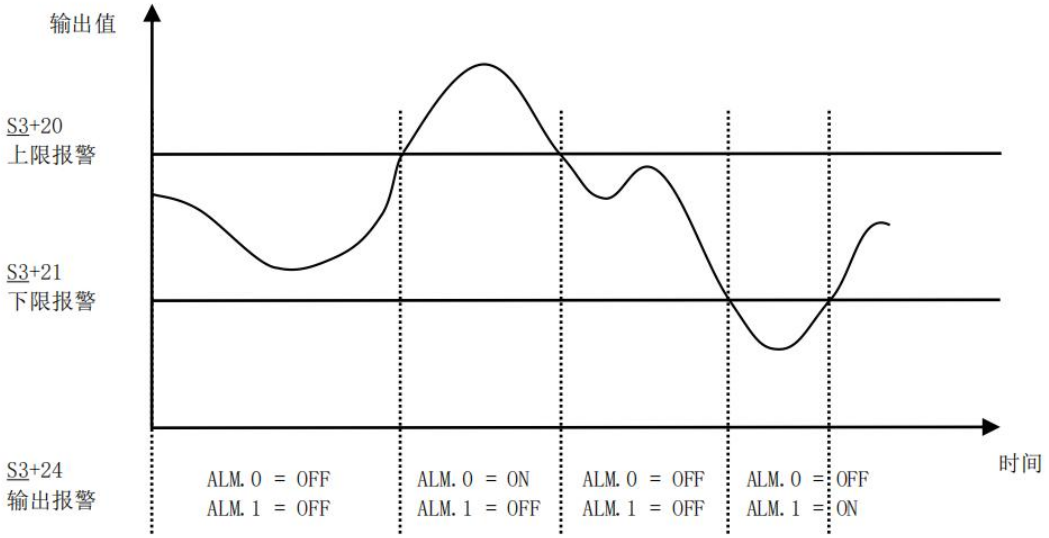
输出值
上下限幅

以下描述 PID 控制对于输出上下限幅的作用示意图。



报 警
上下限

以下描述PID 控制对于输出报警上下限的作用示意图。



13.1.3 程序应用

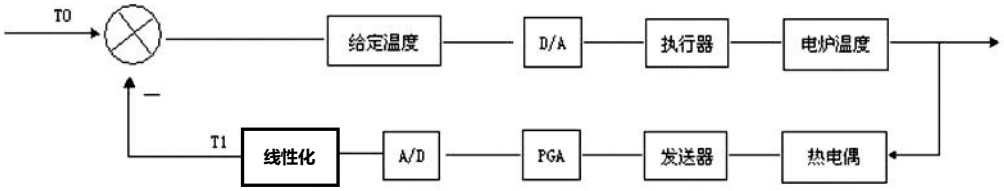
1. PID 指令的应用范围

PID 指令应用十分广泛，例如纺织行业的松紧架控制，供水行业、供气行业、汽车行业、窑炉行业、养殖行业中的恒压供水供气和恒温控制等。PLC 内置PID 控制可以适应各种复杂的控制、多变量的控制。

2. PID 指令控制电加热炉的工作原理

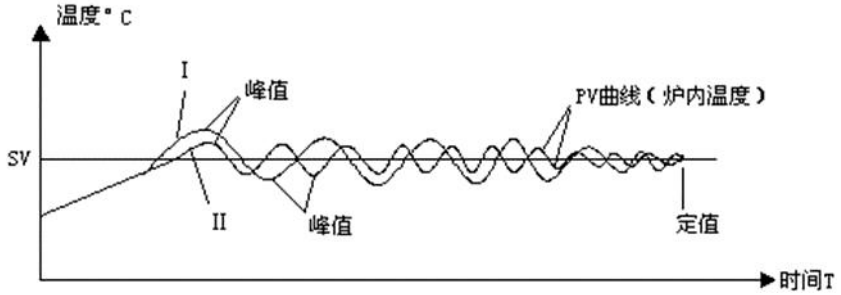
- 电加热炉的控温是闭环控制系统，控温原理框图如下图所示。控温的本质是利用给定温度值与炉内当前值（测得值）的偏差进行 PID 运算，最终使温度控制在某一特定的温度范围内，达到控制的目的。

为了提高控制精度，必须设定诸多控制参数及调节环节。



PID 控温原理框图

- 设计者总是力图炉内温度逼近给定温度值，如下图所示，PV 曲线围绕SV 直线，上下作小幅波动，波动峰值越小，控制精度越高。



炉内温度 PV 逼近给定温度 SV 直线变化曲线I：开关

量控制；曲线II：模拟量控制。

- 控制 PV 曲线变化的方式有两种：开关量控制和模拟量控制。

开关量控制：控制精度要求不高的场合采用开关量控制，其执行器多为接触器或固态继电器—控制电源的通、断。

控制原理是：首先将热电偶检测到的电压和热电偶环温补偿的温度所对应的电压（由热电偶环温补偿非线性表可查）相加，然后，再利用这一电压（电压和）及对应分度的热电偶的非线性表经底层运算得到相应的温度，此温度就是 T1；有了这些数据后方可进行 PID 运算。由运算的结果设置占空比—控制接触器的通与断的时间，到达控制炉温的目的。

开关量控制炉温过程的框图下图。



开关量控制炉温过程框图

开关量控制应用实例：主机接一个EX1-4TC 扩展模块，取第一通道设置为 K 分度热电偶输入，测量温度。通过 PID 运算结果控制 Y0 开关输出来控制加热执行机构动作，从而控制温度到设定目标温度。

例程中需设置参数：

1、PID 运算周期（D1000）

2、PID 运算设定目标值SV（D1300）

3、PWM 输出周期（D1008）

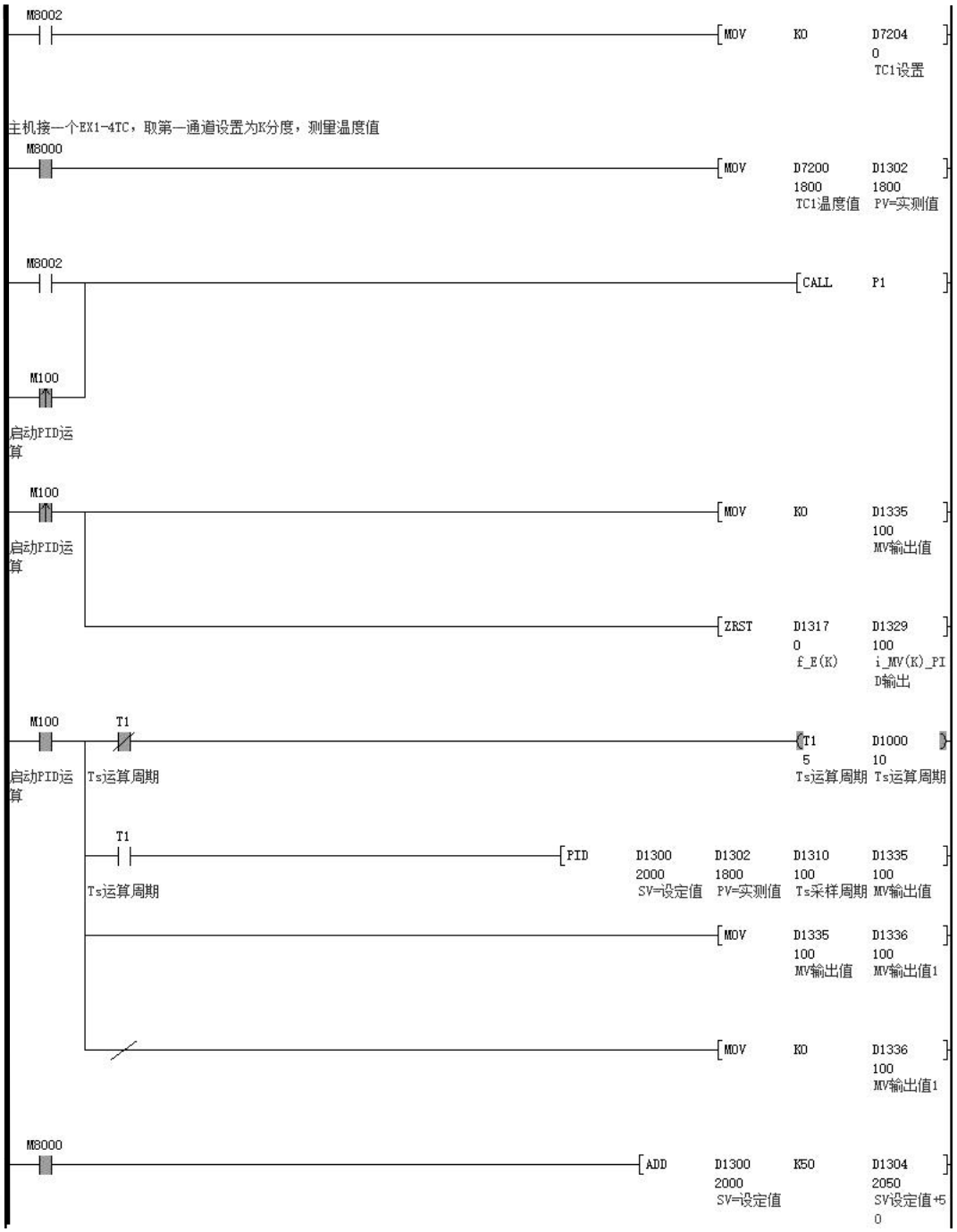
4、滤波系数（D1002）

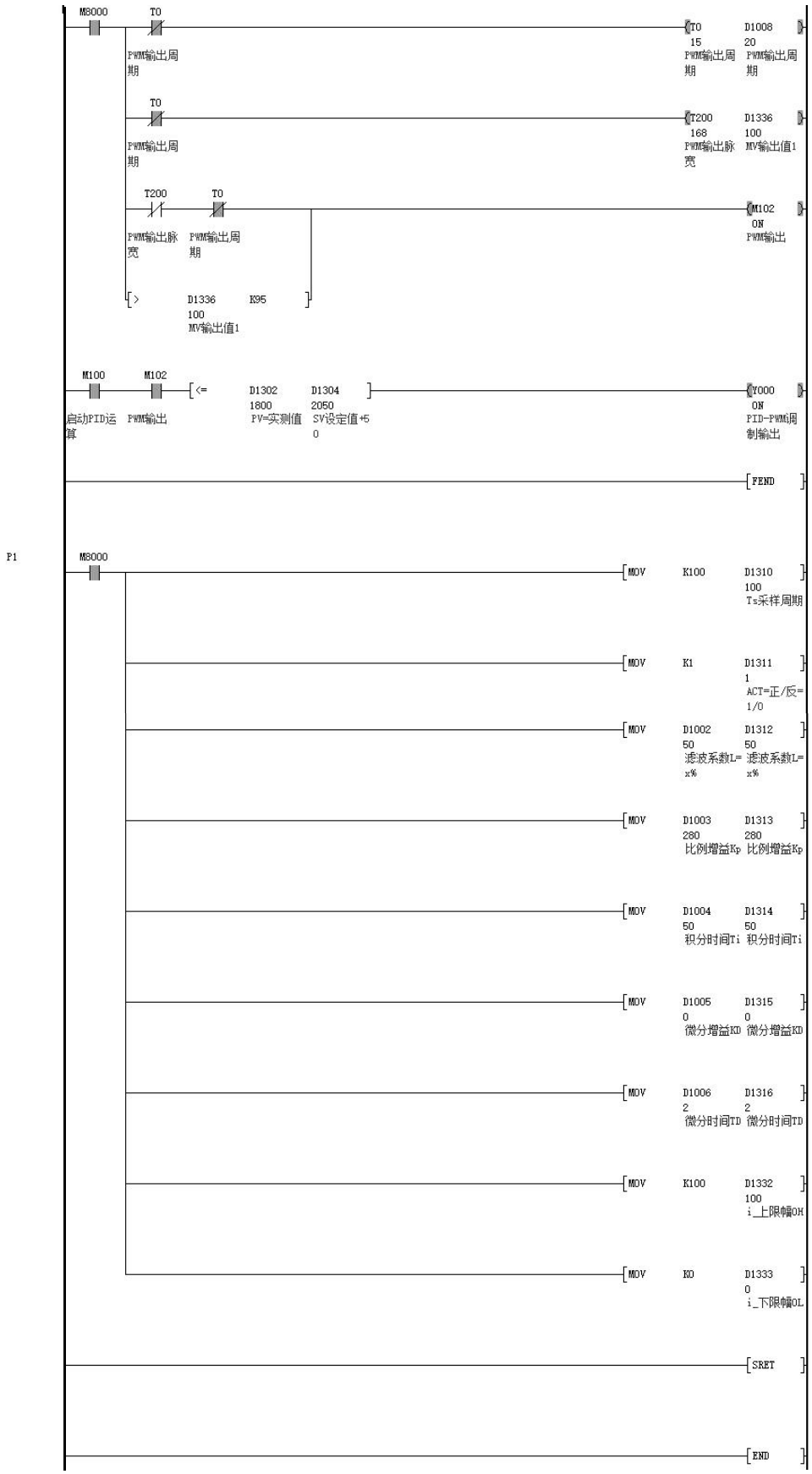
5、比例增益 P（D1003）

6、积分时间 I（D1004）

7、微分增益 KD（D1005）

8、微分时间 I（D1006）





模拟量控制：模拟量控制与开关量控制有许多相同之处，不同的是：模拟量控制的

执行器（变频器、可控硅等）可以接受模拟量信号，这样控制精度就提高了，电源电压（或炉内电功率）不是通和断，而是平滑调节，温度升、降梯度小。

模拟量控制应用实例：主机接一个EX1-4TC 扩展模块，取第一通道设置为 K 分度热电偶输入，测量温度。再接一个 EX1-4DA,取第一通道设置为 0-10V 输出。通过 PID 运算结果控制 0-10V 输出来控制加热执行机构动作，从而控制温度到设定目标温度。

例程中需设置参数：

1、PID 运算周期（D1000）

2、PID 运算设定目标值SV（D1300）

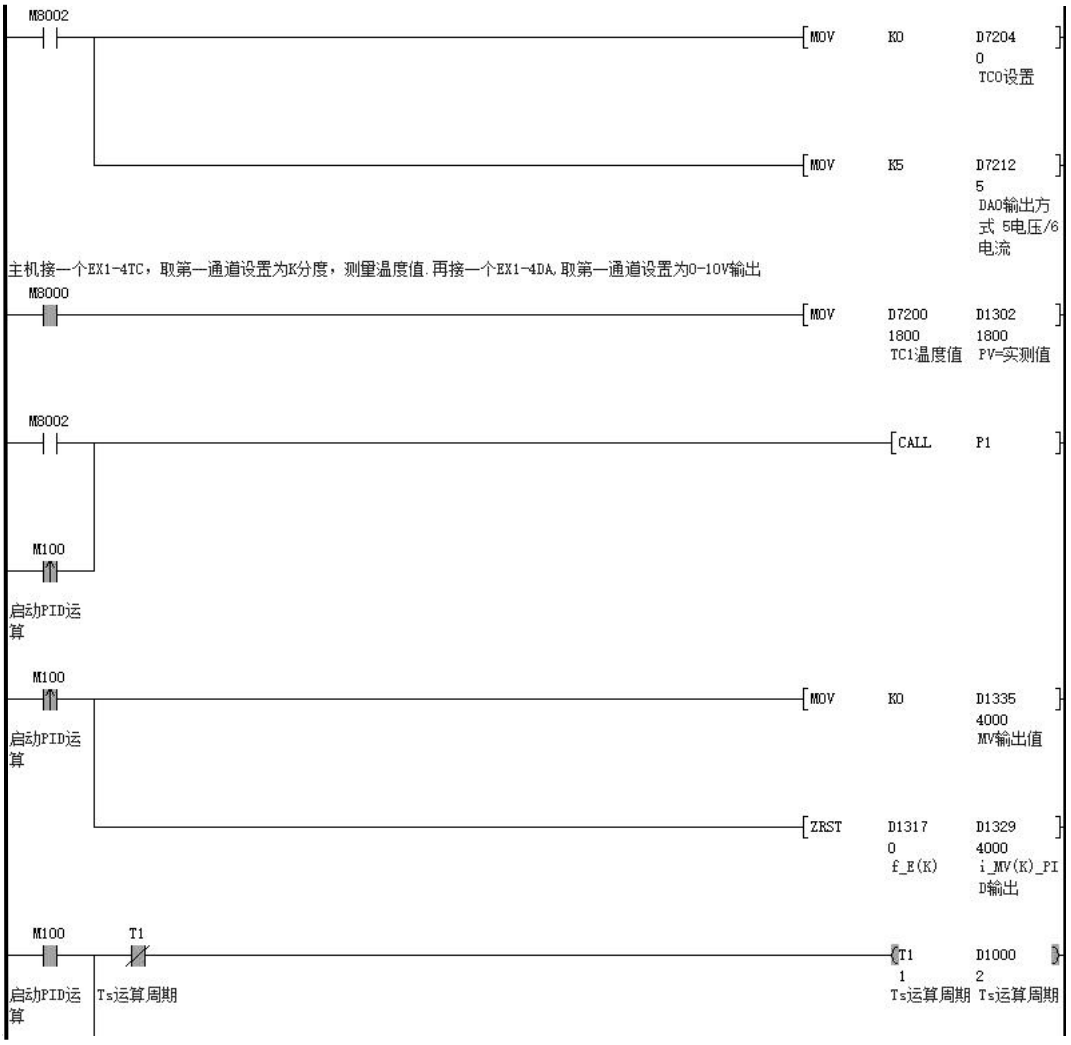
3、滤波系数（D1002）

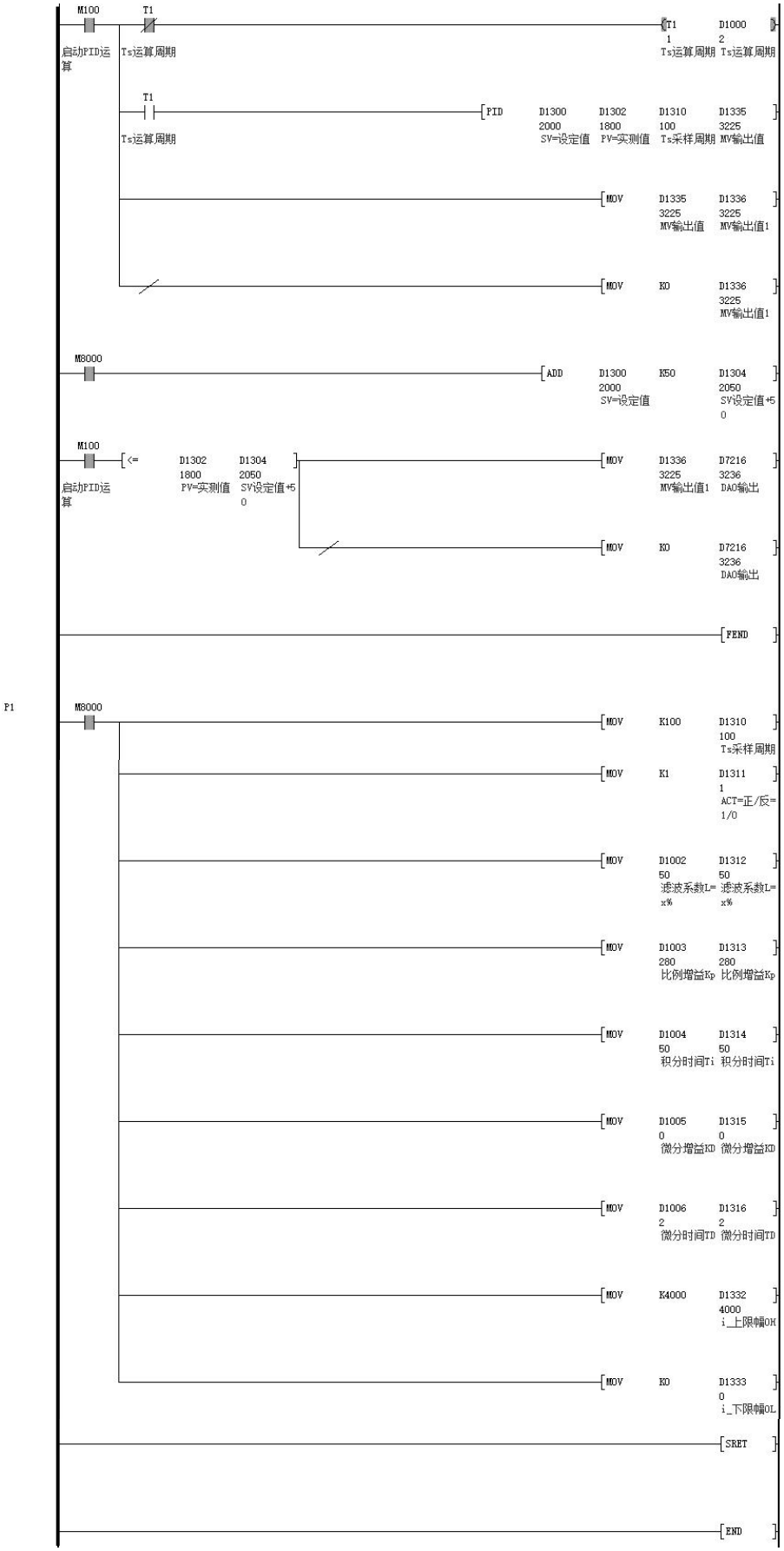
4、比例增益 P（D1003）

5、积分时间 I（D1004）

6、微分增益 KD（D1005）

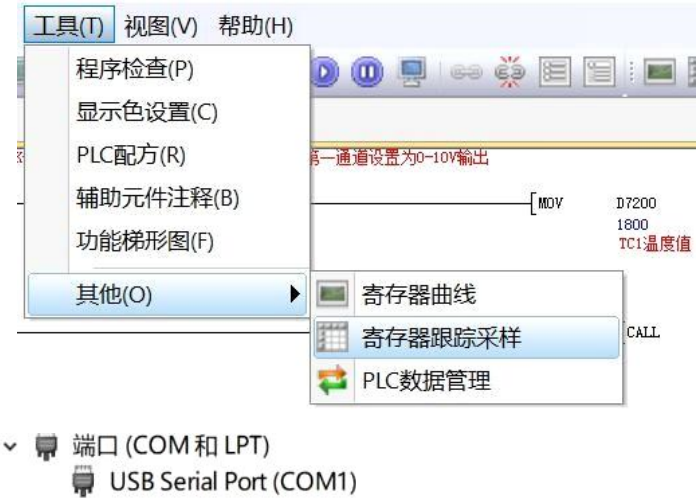
7、微分时间 I（D1006）




M100
T1
启动PID运算
Ts运算周期
T1
K
1
Ts运算周期
D1000
2
Ts运算周期
T1
Ts运算周期
PID
D1300
2000
SV=设定值
D1302
1800
PV=实测值
D1310
100
Ts采样周期
D1335
3225
MV输出值
MOV
D1335
3225
MV输出值
D1336
3225
MV输出值1
MOV
K0
D1336
3225
MV输出值1
M8000
ADD
D1300
2000
SV=设定值
K50
D1304
2050
SV设定值+50
M100
启动PID运算
<=
D1302
1800
PV=实测值
D1304
2050
SV设定值+50
MOV
D1336
3225
MV输出值1
D7216
3236
DAO输出
MOV
K0
D7216
3236
DAO输出
FEND
P1
M8000
MOV
K100
D1310
100
Ts采样周期
MOV
K1
D1311
1
ACT=正/反=1/0
MOV
D1002
50
滤波系数L=x%
D1312
50
滤波系数L=x%
MOV
D1003
280
比例增益Kp
D1313
280
比例增益Kp
MOV
D1004
50
积分时间Ti
D1314
50
积分时间Ti
MOV
D1005
0
微分增益KD
D1315
0
微分增益KD
MOV
D1006
2
微分时间TD
D1316
2
微分时间TD
MOV
K4000
D1332
4000
i_上限幅OH
MOV
K0
D1333
0
i_下限幅OL
SRET
END

13.1.4 数据分析方法及步骤

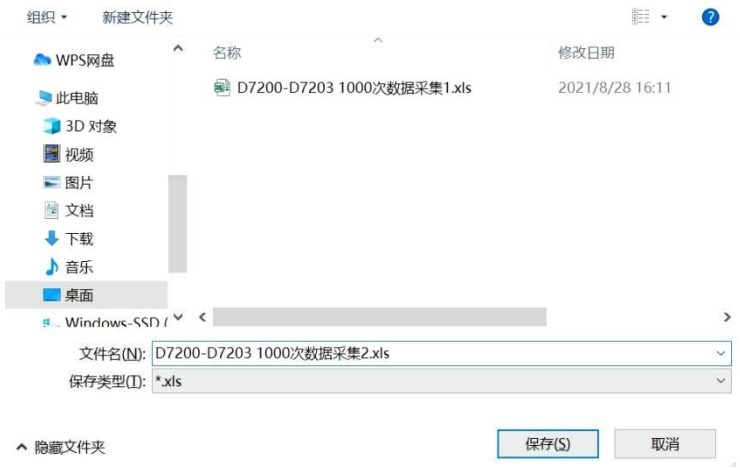
1、打开 KWPRO 软件，再打开工具-其它-寄存器跟踪采样（需关闭监控功能，确保串口不被占用， 硬件串口调为或选用 COM1。



2、采样设置



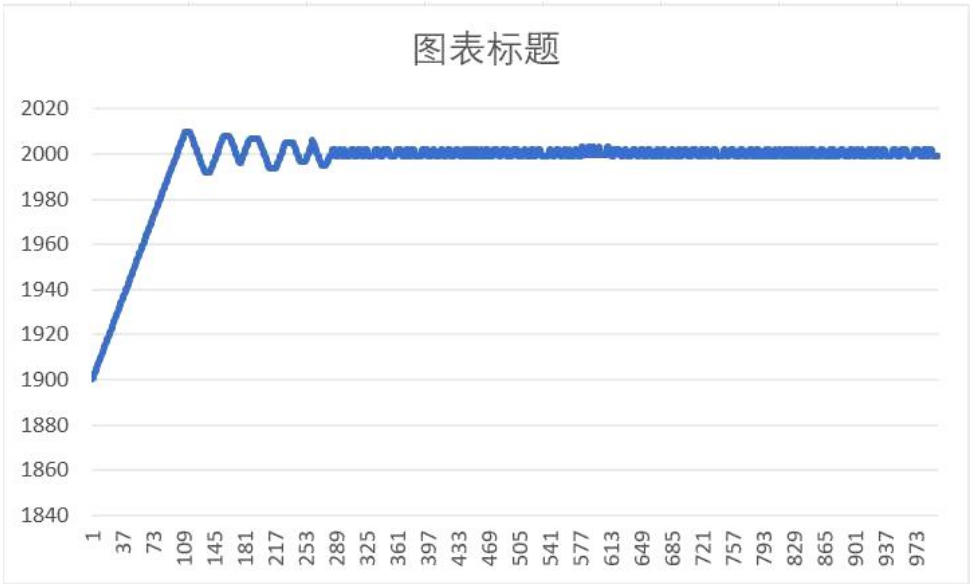
3、点击开始采样，选择文件夹目录和命名文件名，点击保存，开始采样。



4、采样结束后（可手动点击结束采样，提前结束采样），打开保存采集到数据的 EXCEL 表格，选中（B2:E1001） 数据区，找到下图中黄色按钮，将（以文本形式存储的数字）转换为数字。



5、选中（B2:B1001），插入折线图



结合生成的曲线图和前面介绍的 PID 参数整定规律，反复调节 PID 参数，最终得到理想的控温效果。

注意要点

1. 使用多个指令时

可以同时多次执行(环路数没有限制)，但是请注意运算中使用的 S3 和 D 的软元件编号不能重复。

2. 参数 S3 的占用点数

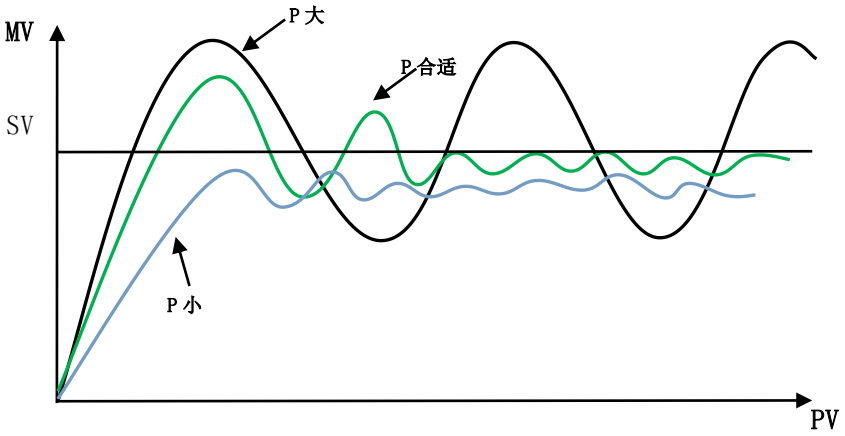
占用从 S3 中指定的起始软元件开始的 25 点软元件。

3. PID 指令可在定时器中断、子程序、步进梯形图、跳转指令中使用，首次执行前必须清除偏差值(S3+7)。

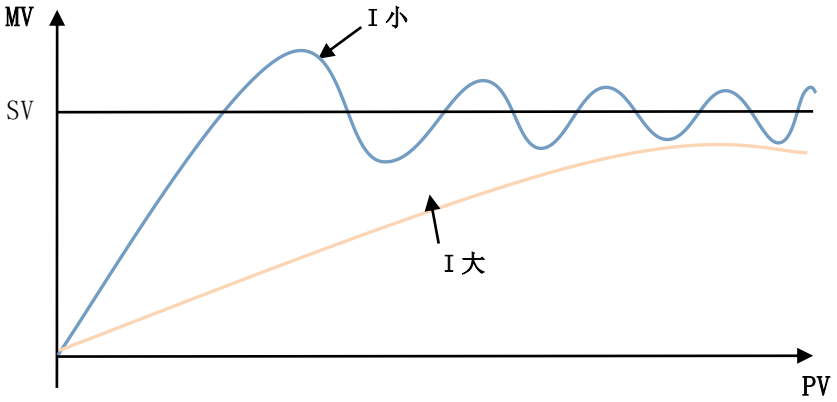
4. 对于MC系列PLC，由于没有热电偶接入接口和模拟量输出接口，故不能直接应用于控温系统。但可选配带热电偶 A/D 采样控制器或模块，利用通信方式将 A/D 值传输进行 PID 运算，将运算结果通信到控制器或模块进行输出调节。

13.1.5 PID参数的整定

1. **采样周期Ts**——PID控制的运算时间间隔。
   Ts大：实时性差，过大时影响控制质量。
   Ts小：实时性好，但不能小于扫描时间，运算频繁会加重运算负担，且执行机构调节频繁。
2. **比例增益P**——表示控制作用的强弱。
   P大：作用强，稳定性差，易振荡。
   P小：作用弱，稳定性好，偏差大。

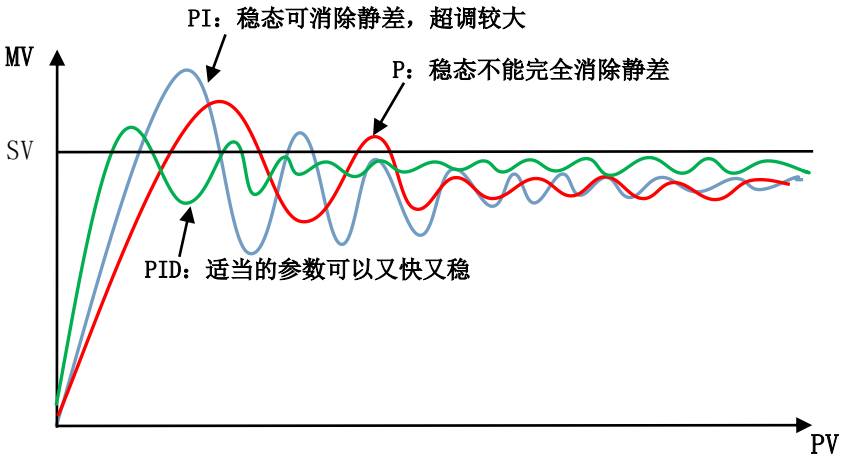


3. **积分时间I**——控制作用的强弱和到达无静差的时间。
   I大：作用弱，稳定性好，到达时间长。
   I小：作用强，易振荡，时间短。



4. **微分时间D**——保持系统稳定，抑制振荡，抑制变化过程。
   一般D=0，根据实际情况适当加入。

5. P.I.D参数控制作用图示



PID控制质量的好坏，与其参数Ts，P，I，D的设置有很大的关系。仅当PID的参数设置与被控对象相配合时，才能得到最佳控制效果。

6. PID参数整定口诀：

参数整定找最佳，从小到大顺序查
先是比例后积分，最后再把微分加
曲线振荡很频繁，比例度盘要调大
曲线漂浮绕大湾，比例度盘往小调
曲线偏离回复慢，积分时间往下降
曲线波动周期长，积分时间再加长
曲线振荡频率快，先把微分降下来
动差大来波动慢。微分时间应加长
理想曲线两个波，前高后低4比1
一看二调多分析，调节质量不会低。

7. 现场整定方法（仅供参考）

先选P，再加I，再加D。

（1）比例P的整定

- 反复调节P的数值，使调节过程达到最佳。
- 方法：优选法，缩短整定次数。
- 确定P值。（P=0.8P选）

（2）积分常数I的整定

- 先加入较大的I，记录波动过程。
- 按I减小，P减小和I加大，P加大的原则反复选取P.I对，直到调节过程满意为止。
- 确定P.I值。

（3）微分常数D的整定

- 一般D=0。
- 出现超调过大，反复振荡可加D。
- 逐步增大D，同时减小P. I；反复选取满意为止。

14. 触点比较指令

在本章中，主要介绍 LD□、OR□、AND□触点比较应用指令。

助记符、名称	功能	章节
[LD□]	开始比较	14.1
[OR□]	并联比较	14.2
[AN□]	串联比较	14.3

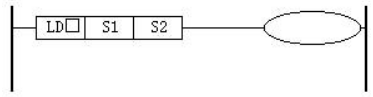
14.1 [LD□]开始比较

14.1.1 概要

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算指令。

1、指令格式

开始比较



16 位指令

5步 下述

32 位指令

7 步 下述

2、设定数据

操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

3、对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	

14.1.2 功能与动作说明

连接在母线上的触点比较指令。

对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

16 位指令	32 位指令	导通条件	非导通条件
LD＝	LDD＝	(S1)=(S2)	(S1)≠(S2)
LD>	LDD>	(S1)>(S2)	(S1)≤(S2)
LD<	LDD<	(S1)<(S2)	(S1)≥(S2)
LD<>	LDD<>	(S1)≠(S2)	(S1)=(S2)
LD>=	LDD>=	(S1)≥(S2)	(S1)<(S2)
LD<=	LDD<=	(S1)≤(S2)	(S1)>(S2)

14.1.3 注意要点

1. LD□是连接母线的触点比较指令，对源数据内容进行BIN 比较，对应其结果执行后段的运算。

2. 关于负数

- S1、S2 的数据最高位为 1 时，将其值当作负数进行比较。
- 16 位运算时: b15。
- 32 位运算时: b31。

3. 使用32 位计数器(包含高速计数器)时

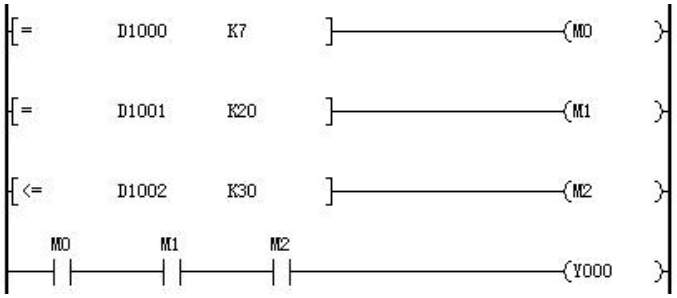
- 32 位计数器(C200～C255)的比较必须用 32 位(LDD=、LDD＞、LDD＜等)运算执行。
- 如指定了 16 位运算(LD=、LD＞、LD＜等)时，会发生程序错误或是运算错误。

14.1.4 程序应用

1. 定时闹钟

用PLC 控制一个电铃，要求每天早上 7：20 响铃 30s。即时间在 7：20～ 7：30时，响铃响起。假定电铃由 Y0 控制，当Y0=ON 时，电铃响；PLC 内部时、分、秒给定在 D1000、D1001、D1002 内。

程序如下：



当 D1000=7 时 ，M0=ON； 当 D1001=20 时，M1=ON； 当 D1002≤30 时，M2=ON；

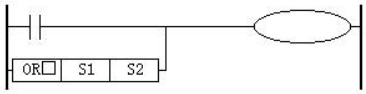
在 M0=ON，M1=ON，M2=ON 时，Y0=ON。

14.2 [OR□]并联比较

14.2.1 概要

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算指令。

1. 指令格式

并联比较	
	
~~16 位指令~~ 5步 下述	~~32 位指令~~ 7 步 下述

2. 设定数据

操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	

14.2.2 功能与动作说明

与其他触点并联的触点比较指令。

对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

16 位指令	32 位指令	导通条件	非导通条件
OR＝	ORD＝	(S1)=(S2)	(S1)≠(S2)
OR>	ORD>	(S1)>(S2)	(S1)≤(S2)
OR<	ORD<	(S1)<(S2)	(S1)≥(S2)
OR<>	ORD<>	(S1)≠(S2)	(S1)=(S2)
OR>=	ORD>=	(S1)≥(S2)	(S1)<(S2)
OR<=	ORD<=	(S1)≤(S2)	(S1)>(S2)

14.2.3 注意要点

1. OR□是连接母线的触点比较指令，对源数据内容进行 BIN 比较，对应其结果执行后段的运算。

2. 关于负数

- S1、S2 的数据最高位为 1 时，将其值当作负数进行比较。
- 16 位运算时: b15。
- 32 位运算时: b31。

3. 使用32 位计数器（包含高速计数器）时

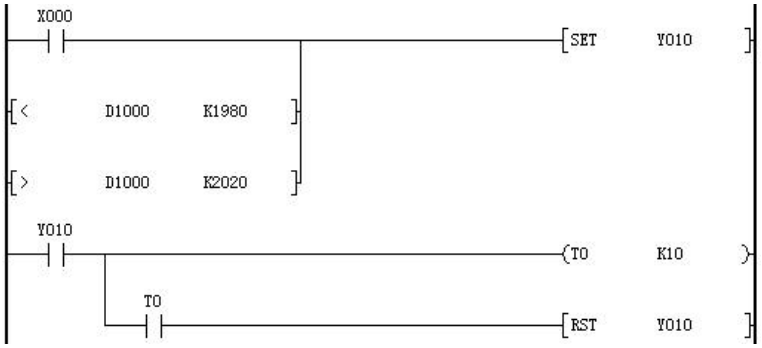
- 32 位计数器（C200～C255）的比较必须用 32 位（ORD=、ORD＞、ORD＜等）运算执行。
- 如指定了 16 位运算（OR=、OR＞、OR＜等）时，会发生程序错误或是运算错误。

14.2.4 程序应用

1. 筛选重量合格产品

某工厂生产螺丝，按每 200 个装袋，每个螺丝在 10g 左右，需保证每袋重量为 2000g（±20g）。假定每3S 完成一袋螺丝的筛选，重量称取后存入寄存器 D1000（g 位单位）。要求当肉眼观察到螺丝外观不符合要求时按下按钮控制 X0=ON，或者当重量不符合标准时，由 Y10 控制打开一个气缸，将本袋螺丝推出生产线作不合格产品处理，且 1S 后气缸回复初始状态。

程序如下：



当以下三个条件任意满足一个时，可使 Y10 置位。

①X0=ON 时；

②D1000＜1980 时；

③D1000＞2020 时。

当 Y10=ON 后 T0 计时，1s 后可复位 Y10 恢复初始状态。

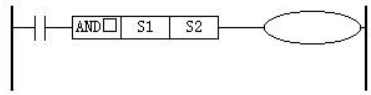
14.3 [AND□]串联比较

14.3.1 概要

AND□是连接母线的触点比较指令，对源数据内容进行 BIN 比较，对应其结果执行后段的运算。

1. 指令格式

串联比较



16 位指令	32 位指令
5步 下述	7 步 下述

2. 设定数据

操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

3. 对象软元件

操作数种类	位软元件						字型软元件											其它		
	系统·用户						位数指定				系统·用户				变址			常数		指针
	X	Y	M	T	C	S	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	P
S1							•	•	•	•	•	•	•	•	•	•	•	•	•	
S2							•	•	•	•	•	•	•	•	•	•	•	•	•	

14.3.2 功能和动作说明

与其他触点串联的触点比较指令。

对 S、D 的内容进行 BIN 比较，根据其结果控制触点的导通或是不导通。

16位指令	32位指令	导通条件	不导通条件
AND=	ANDD=	$S_1 = S_2$	$S_1 \neq S_2$
AND＞	ANDD＞	$S_1 > S_2$	$S_1 \leqq S_2$
AND＜	ANDD＜	$S_1 < S_2$	$S_1 \geqq S_2$
AND＜＞	ANDD＜＞	$S_1 \neq S_2$	$S_1 = S_2$
AND＜=	ANDD＜=	$S_1 \leqq S_2$	$S_1 > S_2$
AND＞=	ANDD＞=	$S_1 \geqq S_2$	$S_1 < S_2$

14.3.3 注意要点

1. 负数比较时

S1、S2 的最高位为 1 时，将其值当作负数进行比较。

- 16 位运算时，最高位：b15
- 32 位运算时，最高位：b31

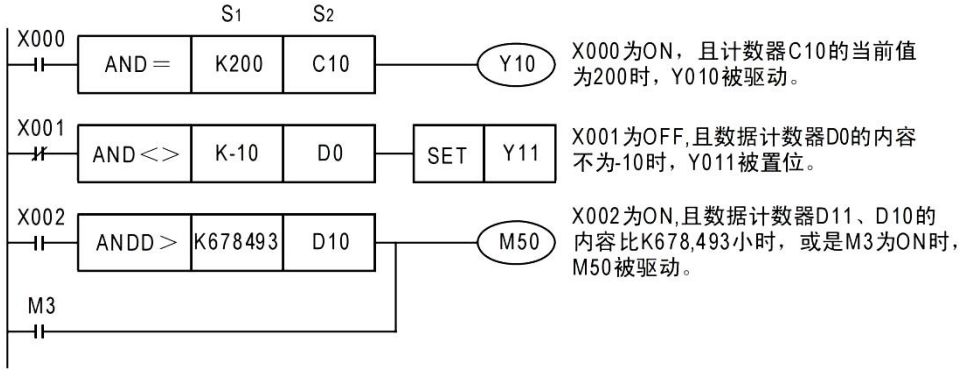
2. 使用 32 位计数器(包含高速计数器)时

如指定了 16 位运算(AND=、AND＞、AND＜等)时，将 32 位计时器的低位当做 16 位寄存器进行比较。

3. 触点比较指令的编程

编程时不能输入≦和≧符号。 ≦是分成＜和=，≧是分成＞和=进行输入的。

14.3.4 程序应用



15 虚拟示波器

MC系列PLC内含一个虚拟示波器，可对寄存器状态的实时变化进行记录，便于用户保存数据或是分析问题。

寄存器地址	参数名称	设定/显示范围	初始推荐设定	单位	生效方式
D8030	触发方式	0～2	0	--	条件生效
D8031	触发单元	0～999	0	--	条件生效
D8032	触发门限	-32768～32767	100	--	条件生效
D8033	采样周期	0～32767	2	ms	条件生效
D8034	采样点数	1～7000	2000	点数	条件生效
D8035	记录起始寄存器序号	0～7999	1000	--	条件生效
D8036	记录区间起始寄存器序号	0～7999	1000	--	条件生效
D8037	记录区间终止寄存器序号	0～7999	4999	--	条件生效
D8038	示波器状态设置/显示	0～4	3	--	立即生效
D8039	当前采样点	采样过程中系统显示当前的已采样点数	0	点数	系统给定
D8040	A通道工作模式	0,1,2,4	1	--	条件生效
D8041	A通道采样寄存器序号	0～7999, 8824~9823	1000	--	条件生效
D8042	B通道工作模式	0,1,2,4	1	--	条件生效
D8043	B通道采样寄存器序号	0～7999, 8824~9823	1100	--	条件生效

15.1 触发方式

D8030设定示波器的触发采样模式。

- D8030 = 0，设置示波器处于手动触发状态。

  当参数生效后，系统将检测触发单元 D8031当前数值：

  - D8031 ≠ 0，则立即开始进行数据采样。
  - D8031 = 0，则系统处于等待采样阶段，直到D8031 ≠ 0时，立即开始进行数据采样。

  所谓“手动”，即D8031的数值由用户进行填写，决定什么时候可以启动示波器采样。

D8030 =	D8031 =	示波器开始采样
0	0	×
	非0，比如1	○

- D8030 = 1或2，设置示波器处于门限触发状态。
  - 门限触发状态为触发单元值与触发门限设定值进行比较，根据其大小关系决定是否启动示波器采样。
    - ◇ D8030 = 1，触发单元值≥触发门限设定值，示波器开始采样。
    - ◇ D8030 = 2，触发单元值≤触发门限设定值，示波器开始采样。
  - 触发单元值通过参数触发单元D8031进行设定，与手动触发不同的是，门限触发模式下该设置值对应的是寄存器地址单元，而非1个常数值。

    例如D8031 = 12，实际与门限设定值进行比较的是D12的当前数值。
  - 触发门限设定值通过参数触发门限D8032进行设定。

D8030 =	D8031 = 12	D8032 =	示波器启动采样
1	D12 = 0	100	×
	D12 = 100		○
	D12 = 120		○
2	D12 = 0	100	○
	D12 = 100		○
	D12 = 120		×

● 若示波器已经开始采样，即使触发条件发生改变使启动条件不满足，也不会停止本次的采样过程。

注：若D8030填写数值不在0～2范围之内，将导致示波器初始化失败。

15.2 触发单元

D8031设定示波器的触发单元。

● 该参数设置的数值意义与参数(触发方式 D8030)的设置相关。

D8030 =	触发方式	D8031设置方法
0	手动触发	D8031 = 0时触发关闭，D8031≠0时开始触发采样
1或2	门限触发	D8031存放触发寄存器序号，为寄存器地址，而非常数

● 其他参考参数(触发方式 D8030)中描述。

● 若触发方式设置为门限触发，必须保证D8031填写数值在0～999范围之内。

15.3 触发门限

D8032设定示波器的触发门限。

● 该参数设置的数值意义与参数(触发方式 D8030)和参数(触发单元 D8031)的设置相关。

D8030 =	触发方式	D8031设置方法
0	手动触发	该参数设定数值与触发采样无关，可随意配置
1或2	门限触发	根据D8031存放寄存器序号的数值与D8032设定值的关系决定是否触发采样

15.4 采样周期

D8033设定示波器采样周期。

● 该参数设置的数值意义与示波器采样周期的换算关系如下所示：

D8033 =	采样周期
0	采样周期 = 0.1 ms，通常采样电流环运行时相关参数时使用。
1	采样周期 = 0.5 ms，通常采样速度环运行时相关参数时使用。
2	采样周期 = 1.0 ms，通常采样位置环运行时相关参数时使用。
N	采样周期 = (N + 1) ms，用户自定义采样周期时使用，N值必须 > 2。 例如N = 99，则采样周期 = (99 + 1) = 100.0 ms。

● D8033填写数值小于0，采样周期固定为0.1ms。

15.5 采样点数

D8034设定示波器的采样点数。

● 该参数设置采样过程中所需要的采样次数，每次采样数据按寄存器序号顺序存放在相应寄存器中。存放规则参考参数(A通道工作模式 D8040)中描述。

D8034与以下所示的参数之间存在着耦合关系，这些参数必须设置正确，否则有可能导致示波器初始化失败。

D8036	记录起始寄存器序号
D8037	记录区间终止寄存器序号
D8040	A通道工作模式
D8042	B通道工作模式

以上参数的配置意义请参考对应的参数描述。

由于用户可自由使用的寄存器序号为D0～D7999，对于以上参数的配置必须注意，存放的采样数据必须在该寄存器区间以内。

参数配置	情况分析	示波器初始化
D8034 = 2000 D8035 = 1000 D8037 = 6999 D8040 = 1 D8042 = 1	采样2000次，每次采样2个字的数据(A/B通道各占1个字)，故采样共需2000×2 = 4000个D寄存器。 从D1000开始存放采样数据，那么采样记录区所占用的寄存器为D1000～D4999。 由于6999 > 4999，故存放没有超限，示波器初始化成功。	○
D8034 = 1500 D8035 = 2000 D8037 = 6999 D8040 = 1 D8042 = 2	采样1500次，每次采样3个字的数据(A通道占1个字，B通道占2个字)，故采样共需1500×3 = 4500个D寄存器。 从D2000开始存放采样数据，那么采样记录区所占用的寄存器为D2000～D6499。 由于6999 > 6499，故存放没有超限，示波器初始化成功。	○
D8034 = 3000 D8035 = 1500 D8037 = 6999 D8040 = 2 D8042 = 0	采样3000次，每次采样2个字的数据(A通道占2个字，B通道不采样)，故采样共需3000×2 = 6000个D寄存器。 从D1500开始存放采样数据，那么采样记录区所占用的寄存器为D1500～D7499。 由于6999 <7499，故存放超限，示波器初始化失败。	×
D8034 = 2000 D8035 = 1000 D8037 = 10000 D8040 = 2 D8042 = 2	采样2000次，每次采样4个字的数据(A通道占2个字，B通道占2个字)，故采样共需2000×4 = 8000个D寄存器。 从D1000开始存放采样数据，那么采样记录区最后的寄存器序号应为8999。用户D寄存器最大序号只能到D7999，但由于D8037本身设置的不合理，导致没有超限，故示波器仍能初始化成功。 采样数据将覆盖大于D7999后的未知区域，可能产生不可预料的后果。	○(不推荐)

- 注意梯形图中其他工艺流程所用的寄存器不能与示波器采样数据的采样区间重叠，否则可能导致示波器采样数据覆盖用户数据。
- 若D8034与其他耦合参数的实际寄存器空间分配超限(如上例描述中的错误情况)，将导致示波器初始化失败。

15.6 记录起始寄存器序号

D8035设定示波器采样数据存放的起始寄存器序号。

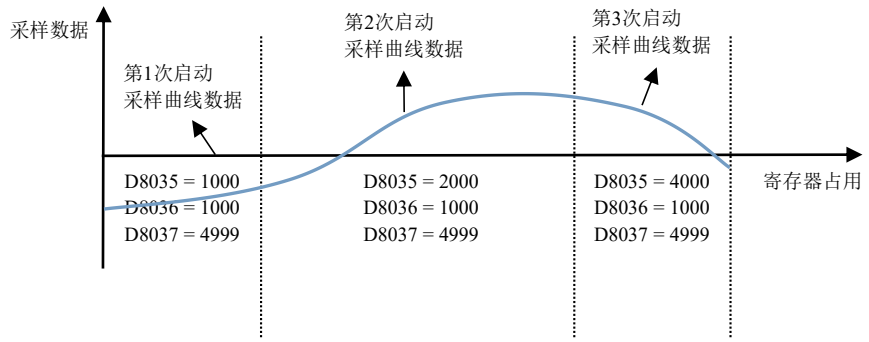
- 该参数设置的数值意义为示波器采样数据存放位置，其指向寄存器地址，而非一个常数项。

  例如D8035 = 1000，则采样数据存放寄存器地址从D1000开始。

  与参数(记录区间起始寄存器序号 D8036)和参数(记录区间终止寄存器序号 D8037)的关系为：

  - ◆ D8035为本次采样实际存放数据的起始地址，D8036和D8037为示波器采样数据用户规划存放的区间地址。必须保证D8035指向的寄存器序号在D8036和D8037之间。
  - ◆ 一个完整的曲线数据采样可能分为多次启用示波器采样完成，D8036规划为整个曲线数据的区间起始点，D8037

    规划为整个曲线数据的区间结束点。



- ◆ 若数据曲线采样仅单次执行，可以将D8035的设定数值与D8036相等，此也为通常情况下的推荐做法。
- 若D8035填写的寄存器序号在0～7999范围之外，可能产生不可预料的后果。
- 必须保证满足D8037 ≥ D8035 ≥ D8036，否则将导致示波器初始化失败。

15.7 记录区间起始寄存器序号

D8036设定示波器记录区间起始寄存器序号。

- 该参数设置的数值意义是规划示波器采样数据存放的起始位置，其指向寄存器地址，而非一个常数项。
  例如D8036 = 1000，则规划采样数据存放在D1000以上的寄存器区域。
- 该参数与(起始寄存器序号 D8035)和参数(记录区间终止寄存器序号 D8037)之间的关系可在参数D8035中查看。
- 若D8036填写的寄存器序号在0～7999范围之外，可能产生不可预料的后果。
- 必须保证满足D8037 ≥ D8035 ≥ D8036，否则将导致示波器初始化失败。

15.8 记录区间终止寄存器序号

D8037设定示波器记录区间终止寄存器序号。

- 该参数设置的数值意义是规划示波器采样数据存放的结束位置，其指向寄存器地址，而非一个常数项。
  例如D8037 = 4999，则本次采样数据存放在D4999以下的寄存器区域。
- 本次示波器采样的记录终止寄存器序号可通过以下参数算出：

D8034	采样点数
D8035	记录起始寄存器序号
D8040	A通道工作模式
D8042	B通道工作模式

记录终止寄存器序号 =

记录起始寄存器序号 + 采样点数×(A通道信号数据类型所占字数 +B通道信号数据类型所占字数) - 1

例如D8034 = 2000，D8035 = 1000，D8040 = 1，D8042 = 1，

则记录终止寄存器序号 = 1000 + 2000×(1 + 1) - 1 = 4999，即本次采样终止寄存器为D4999。

D8037的设定数值≥计算出的记录终止寄存器序号即可，若只执行单次采样，推荐设置两者相等。

- 若D8037填写的寄存器序号在0～7999范围之外，可能产生不可预料的后果。
- 必须保证满足D8037 ≥ D8035 ≥ D8036，否则将导致示波器初始化失败。

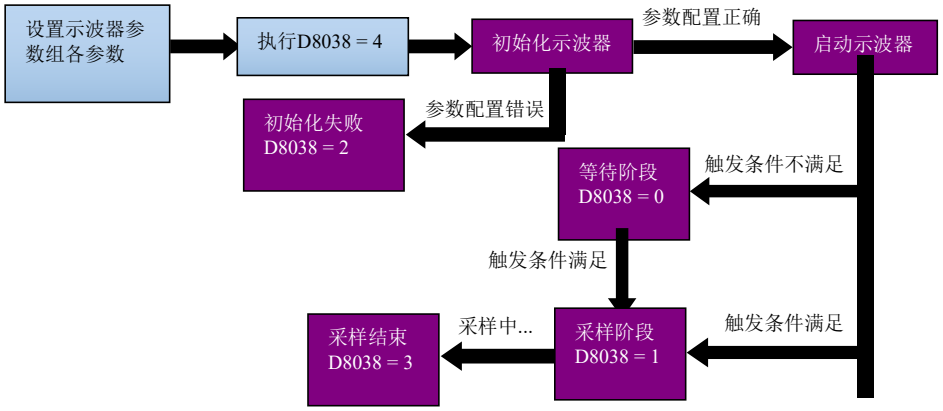
15.9 示波器状态设置/显示

D8038是示波器与用户的交互接口。用户可通过设置该参数申请启动示波器，也可通过其查看当前示波器的状态。

- D8038为用户设置参数时，仅作为申请启动示波器时使用。

D8038 =	设置/显示值	示波器状态
0	系统给定	启动示波器后，若触发条件不满足，D8038会被系统置0，此时示波器处于等待采样阶段。 当触发条件满足后，D8038会被系统置1。
1	系统给定	启动示波器后，若触发条件满足，D8038会被系统置1，此时示波器处于采样阶段。
2	系统给定	示波器初始化失败，即示波器相关参数设置出错所导致。
3	系统给定	当示波器采样结束，D8038会被系统置3。通过观测此数值判断示波器是否已经结束采样。
4	用户设置	示波器申请初始化。 ● 若示波器相关参数设置正确，系统启动示波器。根据参数设置的触发条件决定是否开始采样。 ● 若示波器相关参数设置不正确，系统给定D8038 = 2。

● 一个采样流程中，D8038数值的改变顺序如下所示：



● 示波器参数可能出错或不推荐的设置情况参考第15.6章节各参数的下划横线部分的描述。

15.10 当前采样点

D8039是示波器实时显示当前已采样点数的数值。

● 用户申请示波器初始化(D8038 = 4)后，D8039由系统赋初值0。

若触发条件满足开始采样，D8039的数值随着采样周期变化而递增，直至采样结束。

例如参数(采样点数 D8034) = 2000，参数(采样周期 D8033) = 2。

采样开始后D8039数值清零，并1ms递增1次，直至采样结束。其变化规律为0,1,2,3,...,1998,1999,2000。

15.11 通道工作模式

示波器可同时支持2个通道的数据采集，其中D8040为设定示波器A通道的工作模式，D8042为设定示波器B通道的工作模式。

对于A通道，其对应的采样寄存器序号通过参数(A通道采样寄存器序号 D8041)进行设定。

对于B通道，其对应的采样寄存器序号通过参数(B通道采样寄存器序号 D8043)进行设定。

D8040/D8042=	A/B通道工作模式	A/B工作模式描述	示例寄存器地址
0	A/B通道关闭	A/B通道不采样	--
1	A/B通道字信号	采样D8041/D8043所指向的寄存器的字数据	D11，D12，D224，D234，D260...
2	A/B通道双字信号	采样D8041/D8043所指向的寄存器的双字数据	D226，D228，D236，D238...
4	A/B通道四字信号	采样D8041/D8043所指向的寄存器的四字数据	D14，D220，D230... * 以上也可作为双字信号进行采样。

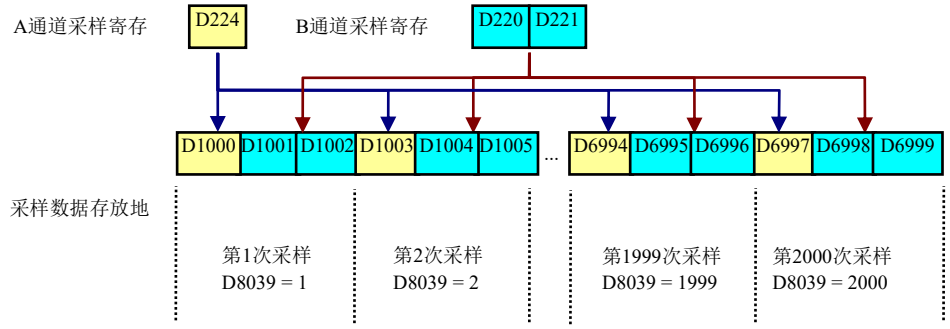
● 示波器采样数据为交替存放规则，以如下示例进行说明。假定示波器的采样参数设置如下：

◆ (采样点数 D8034) = 2000

◆ (记录起始寄存器序号 D8035) = 1000

◆ (A通道工作模式 D8040) = 1，(A通道寄存器序号 D8041) = 224

◆ (B通道工作模式 D8042) = 2，(B通道寄存器序号 D8043) = 220



● 示波器通道的工作模式可填3或大于4的设定值，只要满足采样结束地址不会超过(记录区间终止寄存器序号 D8037)的设定值，示波器仍然可以采样。

例如D8040 = 10，每次采样将A通道采样寄存器序号开始的连续10个D单元数值进行保存。

但此时所采样的数据为非规则的变量类型，难以利用软件进行分析，故不推荐这样设置。

15.12 通道采样寄存器序号

D8041为示波器A通道采样寄存器序号的设定值，D8043为示波器B通道采样寄存器序号的设定值。

● 该参数设置的数值意义是A/B通道每次采样的位置，其指向寄存器地址，而非一个常数项。

例如D8041 = 234，则A通道每次采样从D234寄存器开始，采样字数通过参数(A通道工作模式 D8040)进行设定。

D8043 = 220，则A通道每次采样从D220寄存器开始，采样字数通过参数(B通道工作模式 D8042)进行设定。

D8041/D8043=0~7999时，指向D0~D7999。D8041/D8043=8824~9823时，指向R0~R999。

● 当D8040 设置为0时，即代表A通道不参与本次采样，此时D8041不参与系统运算，用户可随意配置。

当D8042 设置为0时，即代表B通道不参与本次采样，此时D8043不参与系统运算，用户可随意配置。

16 特殊辅助继电器[M],特殊数据寄存器[D]

本章节就PLC内置的特殊辅助继电器(M8000～M8255)，特殊数据寄存器(D8000～D8255)的功能及作用进行列表说明。对于未使用或是未作记载和定义的软元件编号，请勿在程序内进行驱动和写入操作。否则可能出现意料之外的错误。

16.1 特殊辅助继电器[M]功能一览表

软元件编号	软元件功能
M8000	PLC运行ON,其他状态OFF
M8001	PLC运行OFF,其他状态ON
M8002	PLC运行首次梯形图扫描周期ON,其余扫描周期OFF
M8003	PLC运行首次梯形图扫描周期OFF,其余扫描周期ON
M8004	设置PLC运行状态(ON-PLC运行，OFF-PLC停止)，操控该位必须满足以下前提条件： ● D8004 = HFAC0 ● M8005 = OFF
M8005	远程控制设置位(ON-物理控制，OFF-远程控制)
M8006	梯形图运行错误ON
M8007	外部RAM检测(ON-外部RAM运行正常，OFF-未检测到外部RAM)
M8008	掉电检测状态(ON-上电后发生掉电，OFF-上电后未发生掉电)
M8009	暂不开放使用
M8010	暂不开放使用
M8011	暂不开放使用
M8012	100ms固定周期脉冲
M8013	1s固定周期脉冲
M8014	1min固定周期脉冲
M8015	暂不开放使用
M8016	暂不开放使用
M8017	暂不开放使用
M8018	暂不开放使用
M8019	暂不开放使用
M8020	零点标志，加减运算结果为0时ON(保留)
M8021	借位标志，减法运算结果小于负最大值时ON(保留)
M8022	进位标志,进位循环，移位指令运算使用
M8023	暂不开放使用
M8024	暂不开放使用
M8025	暂不开放使用
M8026	暂不开放使用
M8027	暂不开放使用
M8028	暂不开放使用
M8029	暂不开放使用
M8030	暂不开放使用
M8031	暂不开放使用
M8032	输入中断I000申请
M8033	输入中断I001申请
M8034	输入中断I100申请
M8035	输入中断I101申请
M8036	输入中断I200申请
M8037	输入中断I201申请
M8038	输入中断I300申请
M8039	输入中断I301申请
M8040	定时中断I6□□申请
M8041	定时中断I7□□申请
M8042	定时中断I8□□申请
M8043	定时中断I010申请
M8044	软中断I020申请
M8045	软中断I030申请
M8046	软中断I040申请
M8047	软中断I050申请
M8048	输入中断I000禁止
M8049	输入中断I001禁止

软元件编号	软元件功能
M8050	输入中断I100禁止
M8051	输入中断I101禁止
M8052	输入中断I200禁止
M8053	输入中断I201禁止
M8054	输入中断I300禁止
M8055	输入中断I301禁止
M8056	定时中断I6□□禁止
M8057	定时中断I7□□禁止
M8058	定时中断I8□□禁止
M8059	软中断I010禁止
M8060	软中断I020禁止
M8061	软中断I030禁止
M8062	软中断I040禁止
M8063	软中断I050禁止
M8064	禁止高级中断I900(ON:DISABLE,OFF:ENABLE)
M8065	禁止高级中断I901(ON:DISABLE,OFF:ENABLE)
M8066	禁止高级中断I902(ON:DISABLE,OFF:ENABLE)
M8067	禁止高级中断I903(ON:DISABLE,OFF:ENABLE)
M8068	禁止高级中断I904(ON:DISABLE,OFF:ENABLE)
M8069	禁止高级中断I905(ON:DISABLE,OFF:ENABLE)
M8070	禁止高级中断I906(ON:DISABLE,OFF:ENABLE)
M8071	禁止高级中断I907(ON:DISABLE,OFF:ENABLE)
M8072	高级中断I900激活（ON:ACTIVE,OFF:END）
M8073	高级中断I901激活（ON:ACTIVE,OFF:END）
M8074	高级中断I902激活（ON:ACTIVE,OFF:END）
M8075	高级中断I903激活（ON:ACTIVE,OFF:END）
M8076	高级中断I904激活（ON:ACTIVE,OFF:END）
M8077	高级中断I905激活（ON:ACTIVE,OFF:END）
M8078	高级中断I906激活（ON:ACTIVE,OFF:END）
M8079	高级中断I907激活（ON:ACTIVE,OFF:END）
M8080	高级中断I900申请（ON:ASK.OFF:NO）
M8081	高级中断I901申请（ON:ASK.OFF:NO）
M8082	高级中断I902申请（ON:ASK.OFF:NO）
M8083	高级中断I903申请（ON:ASK.OFF:NO）
M8084	高级中断I904申请（ON:ASK.OFF:NO）
M8085	高级中断I905申请（ON:ASK.OFF:NO）
M8086	高级中断I906申请（ON:ASK.OFF:NO）
M8087	高级中断I907申请（ON:ASK.OFF:NO）
M8088	高级中断0进程申请字（1-ASK 0-NO）
M8089	高级中断1进程申请字（1-ASK 0-NO）
M8090	高级中断2进程申请字（1-ASK 0-NO）
M8091	高级中断3进程申请字（1-ASK 0-NO）
M8092	高级中断4进程申请字（1-ASK 0-NO）
M8093	高级中断5进程申请字（1-ASK 0-NO）
M8094	高级中断6进程申请字（1-ASK 0-NO）
M8095	高级中断7进程申请字（1-ASK 0-NO）
M8096	运动指令开关量输入（替代X000输入）
M8097	运动指令开关量输入（替代X001输入）
M8098	运动指令开关量输入（替代X002输入）
M8099	运动指令开关量输入（替代X003输入）
M8100	运动指令开关量输入（替代X004输入）
M8101	运动指令开关量输入（替代X005输入）
M8102	运动指令开关量输入（替代X006输入）
M8103	运动指令开关量输入（替代X007输入）
M8104	运动指令开关量输入（替代X010输入）
M8105	运动指令开关量输入（替代X011输入）
M8106	运动指令开关量输入（替代X012输入）
M8107	运动指令开关量输入（替代X013输入）
M8108	运动指令开关量输入（替代X014输入）
M8109	运动指令开关量输入（替代X015输入）
M8110	运动指令开关量输入（替代X016输入）
M8111	运动指令开关量输入（替代X017输入）
M8112	暂不开放使用

软元件编号	软元件功能
M8113	暂不开放使用
M8114	暂不开放使用
M8115	暂不开放使用
M8116	暂不开放使用
M8117	暂不开放使用
M8118	暂不开放使用
M8119	暂不开放使用
M8120	暂不开放使用
M8121	暂不开放使用
M8122	暂不开放使用
M8123	暂不开放使用
M8124	暂不开放使用
M8125	暂不开放使用
M8126	暂不开放使用
M8127	暂不开放使用
M8128	暂不开放使用
M8129	暂不开放使用
M8130	暂不开放使用
M8131	暂不开放使用
M8132	暂不开放使用
M8133	暂不开放使用
M8134	暂不开放使用
M8135	暂不开放使用
M8136	暂不开放使用
M8137	暂不开放使用
M8138	暂不开放使用
M8139	暂不开放使用
M8140	暂不开放使用
M8141	暂不开放使用
M8142	暂不开放使用
M8143	暂不开放使用
M8144	暂不开放使用
M8145	暂不开放使用
M8146	暂不开放使用
M8147	暂不开放使用
M8148	暂不开放使用
M8149	暂不开放使用
M8150	暂不开放使用
M8151	暂不开放使用
M8152	暂不开放使用
M8153	暂不开放使用
M8154	暂不开放使用
M8155	暂不开放使用
M8156	暂不开放使用
M8157	暂不开放使用
M8158	暂不开放使用
M8159	暂不开放使用
M8160	暂不开放使用
M8161	暂不开放使用
M8162	暂不开放使用
M8163	暂不开放使用
M8164	暂不开放使用
M8165	暂不开放使用
M8166	暂不开放使用
M8167	暂不开放使用
M8168	暂不开放使用
M8169	暂不开放使用
M8170	暂不开放使用
M8171	暂不开放使用
M8172	暂不开放使用
M8173	暂不开放使用
M8174	暂不开放使用
M8175	暂不开放使用
M8176	暂不开放使用
M8177	暂不开放使用

软元件编号	软元件功能
M8178	暂不开放使用
M8179	暂不开放使用
M8180	暂不开放使用
M8181	暂不开放使用
M8182	暂不开放使用
M8183	暂不开放使用
M8184	暂不开放使用
M8185	暂不开放使用
M8186	暂不开放使用
M8187	暂不开放使用
M8188	暂不开放使用
M8189	暂不开放使用
M8190	暂不开放使用
M8191	暂不开放使用
M8192	暂不开放使用
M8193	暂不开放使用
M8194	暂不开放使用
M8195	暂不开放使用
M8196	暂不开放使用
M8197	暂不开放使用
M8198	暂不开放使用
M8199	暂不开放使用
M8200	内部计数器C200方向控制位
M8201	内部计数器C201方向控制位
M8202	内部计数器C202方向控制位
M8203	内部计数器C203方向控制位
M8204	内部计数器C204方向控制位
M8205	内部计数器C205方向控制位
M8206	内部计数器C206方向控制位
M8207	内部计数器C207方向控制位
M8208	内部计数器C208方向控制位
M8209	内部计数器C209方向控制位
M8210	内部计数器C210方向控制位
M8211	内部计数器C211方向控制位
M8212	内部计数器C212方向控制位
M8213	内部计数器C213方向控制位
M8214	内部计数器C214方向控制位
M8215	内部计数器C215方向控制位
M8216	内部计数器C216方向控制位
M8217	内部计数器C217方向控制位
M8218	内部计数器C218方向控制位
M8219	内部计数器C219方向控制位
M8220	内部计数器C220方向控制位
M8221	内部计数器C221方向控制位
M8222	内部计数器C222方向控制位
M8223	内部计数器C223方向控制位
M8224	内部计数器C224方向控制位
M8225	内部计数器C225方向控制位
M8226	内部计数器C226方向控制位
M8227	内部计数器C227方向控制位
M8228	内部计数器C228方向控制位
M8229	内部计数器C229方向控制位
M8230	内部计数器C230方向控制位
M8231	内部计数器C231方向控制位
M8232	内部计数器C232方向控制位
M8233	内部计数器C233方向控制位
M8234	内部计数器C234方向控制位
M8235	内部计数器C235方向控制位
M8236	暂不开放使用
M8237	内部计数器C237方向控制位
M8238	暂不开放使用
M8239	内部计数器C239方向控制位
M8240	内部计数器C240方向控制位
M8241	内部计数器C241方向控制位

软元件编号	软元件功能
M8242	内部计数器C242方向控制位
M8243	暂不开放使用
M8244	暂不开放使用
M8245	暂不开放使用
M8246	暂不开放使用
M8247	暂不开放使用
M8248	暂不开放使用
M8249	暂不开放使用
M8250	暂不开放使用
M8251	暂不开放使用
M8252	暂不开放使用
M8253	暂不开放使用
M8254	暂不开放使用
M8255	暂不开放使用

16.2 特殊辅助寄存器[D]功能一览表

软元件编号	软元件功能
D8000	扫描周期定时监视器限值，单位为ms。当梯形图扫描周期超过该值时梯形图报错。
D8001	指令系列号
D8002	程序容量
D8003	驱动系列号
D8004	设置PLC远程控制(设置为HFAC0，则PLC处于远程控制模式)
D8005	文件写入操作口令(内部使用)
D8006	梯形图出错时的错误步号
D8007	100微秒定时器
D8008	毫秒滴答定时器低字
D8009	毫秒滴答定时器高字
D8010	PLC运行扫描周期实时值
D8011	PLC运行扫描周期最小值
D8012	PLC运行扫描周期最大值
D8013	时钟秒数据(保留)
D8014	时钟分数据(保留)
D8015	时钟时数据(保留)
D8016	时钟日数据(保留)
D8017	时钟月数据(保留)
D8018	时钟年数据(保留)
D8019	时钟星期数据(保留)
D8020	数据掉电保护方式，参考软元件的掉电保护章节 ● =1111 不带停电保护 ● =2222 带停电保护
D8021	掉电电压AD(实测值)
D8022	暂不开放使用
D8023	暂不开放使用
D8024	暂不开放使用
D8025	暂不开放使用
D8026	主控芯片端口A组状态
D8027	暂不开放使用
D8028	主控芯片端口B组状态
D8029	暂不开放使用
D8030	虚拟示波器触发方式
D8031	虚拟示波器触发单元
D8032	虚拟示波器触发门限
D8033	虚拟示波器采样周期
D8034	虚拟示波器采样点数
D8035	虚拟示波器记录起始寄存器序号
D8036	虚拟示波器记录区间起始寄存器序号
D8037	虚拟示波器记录区间终止寄存器序号
D8038	虚拟示波器状态设置/显示
D8039	虚拟示波器当前采样点
D8040	虚拟示波器A通道工作模式
D8041	虚拟示波器A通道采样寄存器序号
D8042	虚拟示波器B通道工作模式
D8043	虚拟示波器B通道采样寄存器序号
D8044	50us时间片内任务时间位置，数值越大则任务执行时间越短(满值3999)
D8045	定时中断最大占用时间
D8046	暂不开放使用
D8047	暂不开放使用
D8048	暂不开放使用
D8049	暂不开放使用
D8050	暂不开放使用
D8051	暂不开放使用
D8052	暂不开放使用
D8053	暂不开放使用
D8054	暂不开放使用
D8055	暂不开放使用
D8056	暂不开放使用
D8057	暂不开放使用
D8058	暂不开放使用

软元件编号	软元件功能
D8059	暂不开放使用
D8060	暂不开放使用
D8061	暂不开放使用
D8062	暂不开放使用
D8063	暂不开放使用
D8064	暂不开放使用
D8065	暂不开放使用
D8066	暂不开放使用
D8067	暂不开放使用
D8068	暂不开放使用
D8069	暂不开放使用
D8070	暂不开放使用
D8071	暂不开放使用
D8072	暂不开放使用
D8073	暂不开放使用
D8074	暂不开放使用
D8075	暂不开放使用
D8076	暂不开放使用
D8077	暂不开放使用
D8078	暂不开放使用
D8079	暂不开放使用
D8080	暂不开放使用
D8081	暂不开放使用
D8082	暂不开放使用
D8083	暂不开放使用
D8084	暂不开放使用
D8085	暂不开放使用
D8086	暂不开放使用
D8087	暂不开放使用
D8088	暂不开放使用
D8089	暂不开放使用
D8090	暂不开放使用
D8091	暂不开放使用
D8092	暂不开放使用
D8093	暂不开放使用
D8094	暂不开放使用
D8095	暂不开放使用
D8096	暂不开放使用
D8097	暂不开放使用
D8098	暂不开放使用
D8099	暂不开放使用
D8100	V0指针寄存器
D8101	V1指针寄存器
D8102	V2指针寄存器
D8103	V3指针寄存器
D8104	V4指针寄存器
D8105	V5指针寄存器
D8106	V6指针寄存器
D8107	V7指针寄存器
D8108	Z0指针寄存器
D8109	Z1指针寄存器
D8110	Z2指针寄存器
D8111	Z3指针寄存器
D8112	Z4指针寄存器
D8113	Z5指针寄存器
D8114	Z6指针寄存器
D8115	Z7指针寄存器
D8116	暂不开放使用
D8117	暂不开放使用
D8118	暂不开放使用
D8119	暂不开放使用
D8120	UART0协议控制字
D8121	UART0字节控制字
D8122	UART0占用D寄存器起始单元（默认7000）

软元件编号	软元件功能
D8123	UART1协议控制字
D8124	UART1字节控制字
D8125	UART1占用D寄存器起始单元（默认7020）
D8126	暂不开放使用
D8127	暂不开放使用
D8128	暂不开放使用
D8129	扩展占用D起始单元(默认7100)
D8130	暂不开放使用
D8131	暂不开放使用
D8132	暂不开放使用
D8133	暂不开放使用
D8134	暂不开放使用
D8135	暂不开放使用
D8136	暂不开放使用
D8137	暂不开放使用
D8138	暂不开放使用
D8139	暂不开放使用
D8140	暂不开放使用
D8141	暂不开放使用
D8142	暂不开放使用
D8143	暂不开放使用
D8144	暂不开放使用
D8145	暂不开放使用
D8146	暂不开放使用
D8147	暂不开放使用
D8148	暂不开放使用
D8149	暂不开放使用
D8150	暂不开放使用
D8151	暂不开放使用
D8152	暂不开放使用
D8153	暂不开放使用
D8154	暂不开放使用
D8155	暂不开放使用
D8156	暂不开放使用
D8157	暂不开放使用
D8158	暂不开放使用
D8159	暂不开放使用
D8160	X00上升沿计数（2KHz）
D8161	X01上升沿计数（2KHz）
D8162	X02上升沿计数（2KHz）
D8163	X03上升沿计数（2KHz）
D8164	X04上升沿计数（2KHz）
D8165	X05上升沿计数（2KHz）
D8166	X06上升沿计数（2KHz）
D8167	X07上升沿计数（2KHz）
D8168	暂不开放使用
D8169	暂不开放使用
D8170	暂不开放使用
D8171	暂不开放使用
D8172	暂不开放使用
D8173	暂不开放使用
D8174	暂不开放使用
D8175	暂不开放使用
D8176	暂不开放使用
D8177	暂不开放使用
D8178	暂不开放使用
D8179	暂不开放使用
D8180	Y0输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8181	Y1输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8182	Y2输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8183	Y3输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8184	Y4输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8185	Y5输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8186	Y6输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位

软元件编号	软元件功能
D8187	Y7输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8188	暂不开放使用
D8189	暂不开放使用
D8190	暂不开放使用
D8191	暂不开放使用
D8192	暂不开放使用
D8193	暂不开放使用
D8194	暂不开放使用
D8195	暂不开放使用
D8196	暂不开放使用
D8197	暂不开放使用
D8198	暂不开放使用
D8199	暂不开放使用
D8200	暂不开放使用
D8201	暂不开放使用
D8202	暂不开放使用
D8203	暂不开放使用
D8204	暂不开放使用
D8205	暂不开放使用
D8206	暂不开放使用
D8207	暂不开放使用
D8208	暂不开放使用
D8209	暂不开放使用
D8210	暂不开放使用
D8211	暂不开放使用
D8212	暂不开放使用
D8213	暂不开放使用
D8214	暂不开放使用
D8215	暂不开放使用
D8216	暂不开放使用
D8217	暂不开放使用
D8218	暂不开放使用
D8219	暂不开放使用
D8220	暂不开放使用
D8221	暂不开放使用
D8222	暂不开放使用
D8223	暂不开放使用
D8224	暂不开放使用
D8225	暂不开放使用
D8226	暂不开放使用
D8227	暂不开放使用
D8228	暂不开放使用
D8229	暂不开放使用
D8230	暂不开放使用
D8231	暂不开放使用
D8232	暂不开放使用
D8233	暂不开放使用
D8234	暂不开放使用
D8235	暂不开放使用
D8236	暂不开放使用
D8237	暂不开放使用
D8238	暂不开放使用
D8239	暂不开放使用
D8240	暂不开放使用
D8241	暂不开放使用
D8242	暂不开放使用
D8243	暂不开放使用
D8244	暂不开放使用
D8245	暂不开放使用
D8246	暂不开放使用
D8247	暂不开放使用
D8248	暂不开放使用
D8249	暂不开放使用
D8250	暂不开放使用
D8251	暂不开放使用

软元件编号	软元件功能
D8252	暂不开放使用
D8253	暂不开放使用
D8254	加密控制。PLC停止状态时写D8254为以下数值，并下载一个与MC系列PLC内相异的梯形图后，系统即进行加密操作。 ● D8254 = K12345　禁读 ● D8254 = K-11215　禁读禁写 * 加密为不可逆操作，需谨慎操作，后果自负。
D8255	内核版本号

16.3 实时时钟

MC系列PLC提供年、月、日、时、分、秒和星期共7 种时间信息，PLC在供电或断电情况下能够正确计时，采用24小时制计时，年4位显示（可计到2099），带闰年自调整，可实现时间信息的读取和写入（设定）以及修正。

(1) 实时时钟占用特殊软元件资源及功能说明

序号	特殊软元件	名称	功能说明
1	M8015	时钟修改确认继存器	上升沿有效，将D中时间数据写入，用于设置时间。
2	M8016	时钟修改允许继存器	高电平时允许修改D8013~D8019中的数据。 修改完毕后，经M8015上升沿确认生效。
3	D8013	秒	正常工作时，存放读取的时钟数据。 设定或修改时，存放要设定的时间数据。 时间格式： 年为4 位显示（如2016），采用24小时制。 星期日~星期六分别对应0~6。
4	D8014	分	
5	D8015	时	
6	D8016	日	
7	D8017	月	
8	D8018	年	
9	D8019	星期	

(2) 正常读取时钟

当M8015=OFF，M8016=OFF，D8013~D8019上存放的值是当前时钟值，其具体定义见上表。

(3) 时钟修改

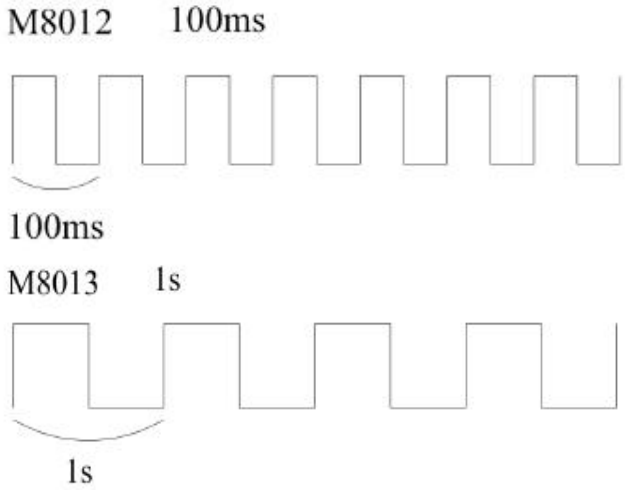
当M8016=ON时，按要求人为设定D8013~D8019的值。
当M8015由OFF → ON（上升沿）时，将D8013~D8019的值作为当前时钟值。
请按下列步骤修改时钟。
① 把M8016 置“高”，禁止读取时钟数值，允许修改D8013~D8019寄存器中的数值；
② 修改D8013~D8019寄存器中的数值；
③ 把M8015置“高”，即在M8015上升沿将D8013~D8019中时间数据写入时钟；
④ 把M8015复位，M8016复位。恢复时钟正常读取。

16.4 固定周期脉冲

提供100ms、1s、1min 的固定周期脉冲动作M8012、M8013、M8014。

固定周期	特殊软元件	说明
100ms	M8012	独立于梯形图之外，不受扫描周期影响。
1s	M8013	
1min	M8014	

不管是否使用，M8012、M8013 都会按照100ms、1s 的固定周期进行翻转。



三、程序流程控制指令使用手册

1. 程序流程控制指令之间的相互关系

1.1. 程序流程控制指令之间的相互关系

流程之间的相互/调用关系									
说明 1：左列调用上列：√表示可以，×表示不可以，◎表示谨慎使用									
说明 2：其中 CJ 的和其它指令的重叠包括CJ 指针 P 直接指向其它指令的开始指令，比如 CJ 的指针 P 指向 CJ 或FOR。									
上列 / 左列	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET	P-SRET	I-IRET /PRET	FEND-END	MODBUS 自由协议
MC-MCR	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
CJ-P	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 ◎	包含 ◎	包 含 ×	包 含 √
	重叠 ◎	重叠◎	重叠 √	重叠 ◎	重叠◎	重叠 ◎	重叠 ◎	重 叠 √	重 叠 ×
						调用 √			
EI-DI	包 含 √	包含√	包含 √	包 含 √	包含 √	包含 √	包含 √	包 含 √4	包 含 √
	重 叠 √	重叠√	重叠 √	重 叠 √	重叠 √	重叠 √	重叠 √	重 叠 √	重 叠 ×
FOR-NEXT	包 含 ×	包含√	包含 √	包 含 √	包含 ×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 √	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
STL-RET	包 含 ×	包含◎	包含 √	包 含 √	包含 ×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
P-SRET	包 含 ×	包含√	包含 √	包 含 √	包含 2×	包含 ×	包含 ×	包 含 ×	包 含 √
	重 叠 ×	重叠◎	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
I-IRET/PRET	包 含 ×	包含√	包含 √	包 含 √	包含 2×	包含 ×	包含 ◎	包 含 ×	包 含 √
	重 叠 ×	重叠 3×	重叠 √	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重 叠 ×	重 叠 ×
						调用 √			
FEND-END	包含 1√	包含√	包含 √	包 含 √	包含 1◎	包含 √	包含 √	包含 √	
	重 叠 ×	重叠×	重叠√	重 叠 ×	重叠 ×	重叠 ×	重叠 ×	重叠 √	

1：需配合 CJ 指令使用，否则主控执行不了。

2：可以使用，但不建议使用。

3：I 中跳转到 I 外部时必须要有跳回到 I 中的 CJ-P 语句。

4：程序中只有 EI，无DI 情况下，可类同包含关系。

定义 1：两个发生关系的流程指令的语句为 A1 和 B1、A2 和 B2。

定义 2：包含关系为，A1→A2→B2→B1，例，CJ（A1）→FOR（A2）→NEXT（B2）→P（B1）。

定义 3：重叠关系为，A1→A2→B1→B2，例，CJ（A1）→FOR（A2）→P（B1）→NEXT（B2）。

定义 4：调用的意思是在左列的指令中调用上列的指令，一般使用 CALL 指令调用上列指令。

定义 5：MODBUS 和自由协议表示的是两种通信协议的通信指令，只存在被其它流程指令包含的情况。

2. 程序流程控制指令

2.1. 主控指令[MC-MCR]

2.1.1. 概要

执行 MC 指令后，母线(LD、LDI 点)移动到 MC 触点之后。使用 MCR 指令，可以将其返回原来的母线位置。通过更改软元件编号 Y、M，可以多次使用 MC 指令。但使用同一软元件编号时，和OUT 指令相同，会出现双线圈输出。

- 指令格式及对象软元件

助记符	功能	回路表示	程序步
MC	公共串联触点的连接	MC N S1	5
MCR	公共串联触点的清除	MCR N	3

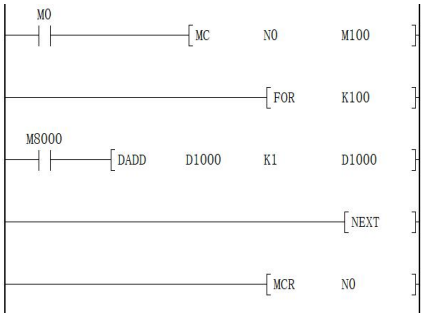
操作数	位软元件	变址	字软元件	其他
N	N0-N7			

S1	X	Y	M	S	V	Z	修饰	KnX	KnY	KnM	KnS	T	C	D	R	K	H
		√	√														

注：特殊辅助继电器 M 除外

2.1.2. 功能与动作说明

- MC 与MCR 之间的母线上连接的驱动指令（执行结果），只在MC 指令执行时才执行，不执行MC 指令时，无论驱动指令之前的触点状态（执行条件）如何，都按 OFF 执行(等同触点OFF 时的动作，与主程序存在双线圈输出，编程时注意软资源分配)。
- 在使用了主控的回路中，触点指令中如果有不需要触发条件的指令（如FOR-NEXT，EI，DI 指令等），则执行该指令，而与 MC 指令是否执行无关，如图 1



M0不导通，MC主控回路不执行，但FOR-NEXT会一直执行100次，
FOR-NEXT之间的指令不执行。
增大FOR循环次数，梯形图扫描时间增加，但D1000值不变

图 1.主控内直连母线驱动指令运行情况说明图

- 主控（MC）和主控复位 MCR 指令：当接通时正常运行其之间的程序，当未接通时或从接通变为未接通时，MC 与MCR 之间的指令动作如下表所示：

一般定时器	计时值归零，线圈失电，触点不动作
累计型定时器	线圈失电，计时值及触点保持目前状态
计数器	线圈失电，计数值及触点保持目前状态
OUT 指令驱动的线圈	全部不受电
SET， RST 指令驱动的元件	保持目前状态
应用指令	全部不动作，但 FOR-NEXT 循环回路仍会来回执行 N 次，但 FOR-NEXT 间的任何指令与 MC-MCR 之间其它指令动作相同（如图 1 所示）

目前状态：梯形图扫描到主控指令时刻，软元件的状态。

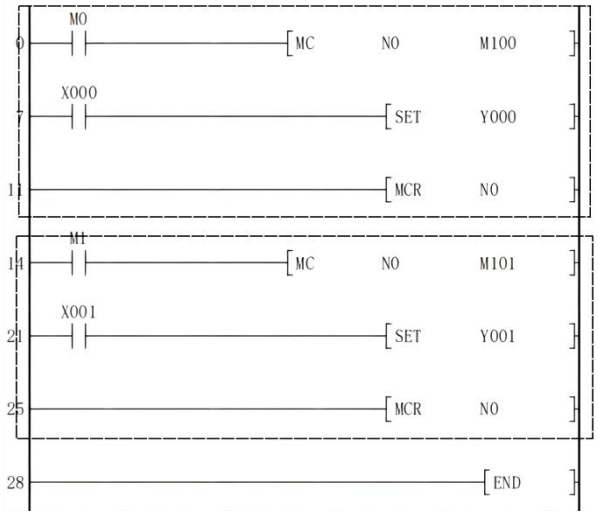
2.1.3. 注意要点

- 主控 MC 和主控复位 MCR 指令成对出现，不能缺少任何一个，否则 PLC 会报错，主控 MC 指令之前必须经过 LD（LDI）指令连接母线，否则梯形图编译不了，主控复位 MCR 指令直接连接母线，否则梯形图也编译不了。
- 主控 MC-MCR 指令之间的程序不管主控执行或不执行，都会扫描，执行时按正常逻辑扫描执行；不执行时，按新母线为 0 扫描执行。所以在主控 MC-MCR 之间的驱动指令与主程序驱动指令使用的软元件，同样存在双线圈输出问题，使用时要合理分配软资源。
- MC-MCR 指令不能在FOR-NEXT，STL-RET，P-SRET，I-IRET，FEND-END 中使用。
- MC-MCR 相对的N 值必须相同。

2.1.4. 使用说明

1）单层使用

没有嵌套结构的时候，可以再次使用 N0。N0 的使用次数没有限制。 通过更改软元件号 Y，M，可多次使用主控指令(MC)。 但是如果使用同一软元件号，将同OUT 指令一样，会出现双线圈输出，如图 2 所示：



M0有效

X000为ON，Y000为ON

母线返回

没有嵌套时可以再次使用N0
N0的使用次数没有限制
M1有效

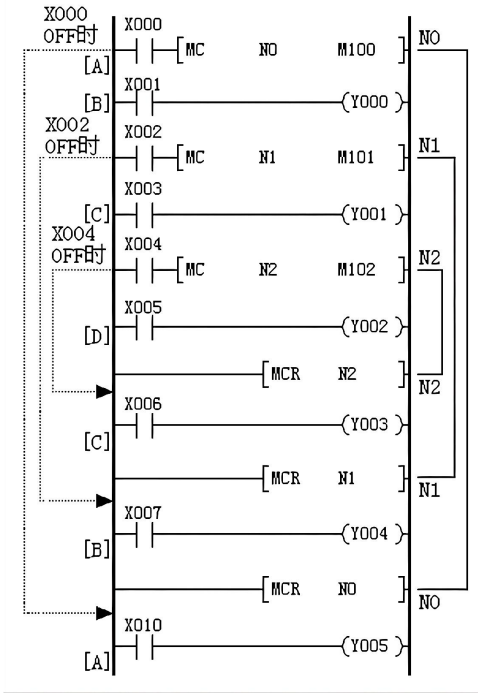
X001为ON，Y001为ON

母线返回

图 2.主控单层使用图例

2）多层嵌套使用

- 主控MC-MCR 允许多层嵌套使用，最多 8 层，即N 为N0-N7；
- 嵌套时要按顺序嵌套，使用MC指令时，嵌套等级 N 的编号依次增大(N0→N1→N2→N3→N4→N5→ N6→N7)。返回时，采用 MCR 指令，从大的嵌套等级开始解除(N7→N6→N5→N4→N3→N2→N1→N0)。图3 为3 层嵌套使用。例如，不对 MCR N6、MCR N7 编程时，如对 MCR N5 编程，则嵌套等级会一下子返回到5。



《嵌套级N0》
母线B在X000为ON时，呈激活状态。

《嵌套级N1》
母线C在X000，X002为ON时，呈激活状态。

《嵌套级N2》
母线D在X000，X002，X004为ON时，呈激活状态。

《嵌套级N1》
通过MCR N2，母线返回到C的状态。

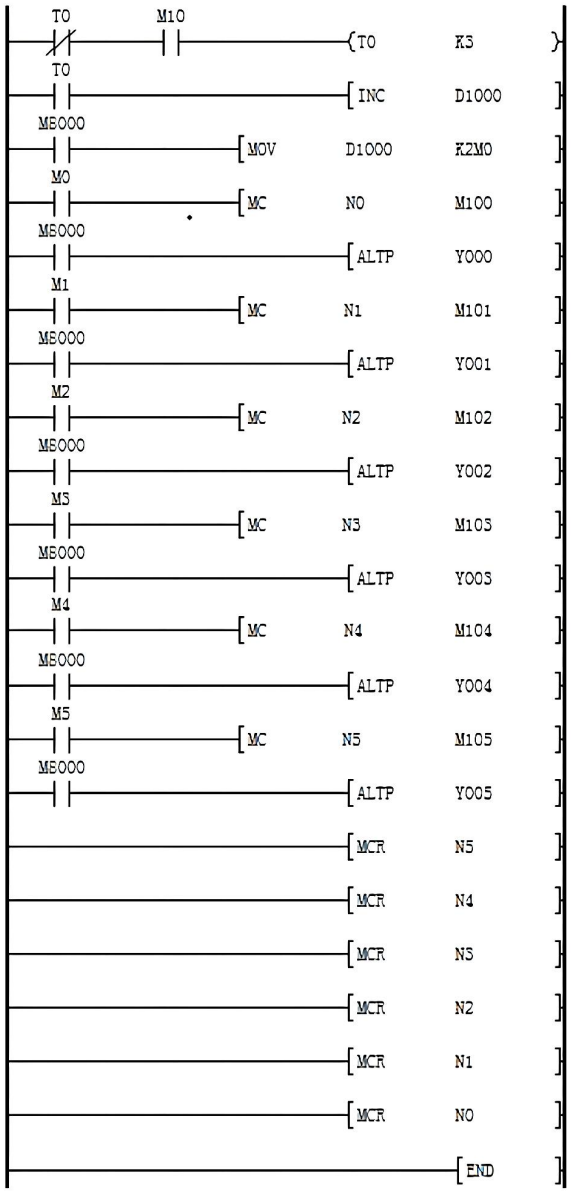
《嵌套级N0》
通过MCR N1，母线返回到B的状态。

《初始状态》
通过MCR N0，母线返回到最初A的状态。因此，Y005的接通/断开只取决于X010的接通断开状态，而与X000，X002，X004的状态无关。

图 3.主控 3 层嵌套调用

2.1.5. 案例分析

下面例程使用主控实现跑马灯效果，如图 4 所示：



M10启动，T0定时0.3S，

D1000加1，

再赋值给K2M0，

先让M0-M7实现跑马灯效果，然后根据M0-M5的效果在主控嵌套中实现Y0-Y5的跑马灯效果

这里用ALTP指令使得Y的状态翻转

Y0的状态维持0.6秒闪烁一次

D1000	1	2	3	4	5	6	7
Y0	1	1	0	0	1	1	0
Y1	0	0	1	1	1	1	0
Y2	0	0	0	0	0	0	1
Y3	0	0	0	0	0	0	0
Y4	0	0	0	0	0	0	0
Y5	0	0	0	0	0	0	0

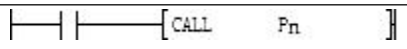
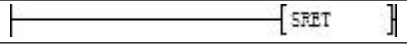
图 4.用主控实现跑马灯

2.2. 子程序调用指令[CALL/CALLP]

2.2.1. 概要

在顺控程序中，对共同处理的程序进行调用的指令，可以减少程序的步数，更加高效地设计程序。此外，编写子程序时，还需要使用 FEND 指令和 SRET 指令。

- 指令格式及对象软元件

助记符		功能		回路表示	程序步
CALL	P	调用子程序			3
SRET		子程序结束			1
操作数		位软元件	变址	字软元件	其他
指针 Pn		P0-P127			
	X Y M S V Z	修饰	KnX KnY KnM KnS T C D R	K H	
		√			

2.2.2. 功能与动作说明

- 当指令输入为 ON 时，执行 CALL/CALLP 指令，向指针 P 标记的步跳转。然后执行标记的子程序。直到执行 SERT 后，返回到 CALL/CALLP 指令的下一步继续运行。
- 当指令从 ON 变为 OFF 时，子程序不再扫描，除定时器 T 以外软元件保持原状，定时器 T 会继续计时（最大到 32767），当达到设定值后，输出触点会在下一次执行线圈指令时（下一次调用子程序时）动作，编程时要注意。

2.2.3. 注意要点

- 指针 P 所指定的子程序请在 FEND 指令后编写。
- 不可与 CJ 指令指定相同的指针编号P。
- 子程序调用指令必须与对应的指针编号一一对应（即必须有对应的子程序），指针只能 P0-P127，否则会出现报警或死机现象。
- 只使用 CALL/CALLP 指令时，可不限次数调用同一指针编号的子程序。
- 子程序中可使用 CALL/CALLP 指令调用其它子程序，包括本身最多 8 层（嵌套调用）。
- 子程序调用指令与跳转指令的区别在于：

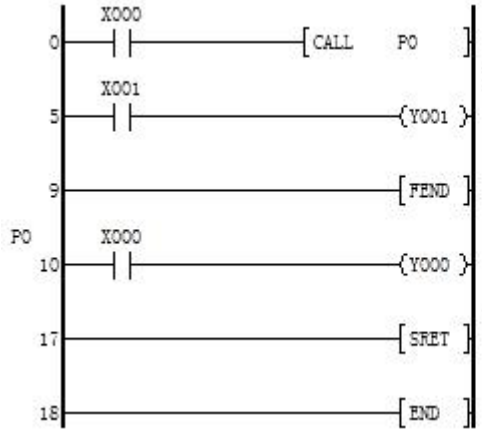
✯ CALL Pn 一旦执行，首先执行其指针编号 Pn 所指的下一步程序，直至遇到 SRET，再返回子程序调用指令的下一步继续执行后面程序。

✯ CJ Pn 一旦执行，直接执行其指针编号 Pn 所指的下一步程序，并不返回原调用处。

- 子程序使用软元件与主程序存在双线圈问题，要合理分配软元件。

2.2.4. 使用说明

1）1 层使用



主程序
X0由OFF变为ON，则执行调用程序，
跳转到标记为P0的子程序，执行子程序。

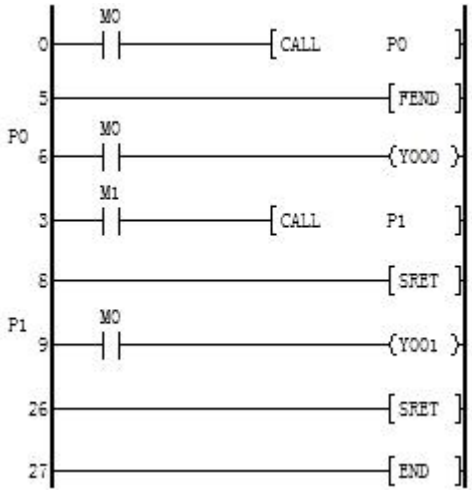
通过执行SRET指令后，返回到原来的主程序步，接着继续执行后续的主程序。

子程序
在FEND指令后面对子程序编程

图 5.子程序单层调用

2）嵌套使用

子程序内的 CALL 指令嵌套使用最多允许 7 次，即包括本身为 8 层嵌套。



主程序
M0=OFF→ON后标记P0跳转

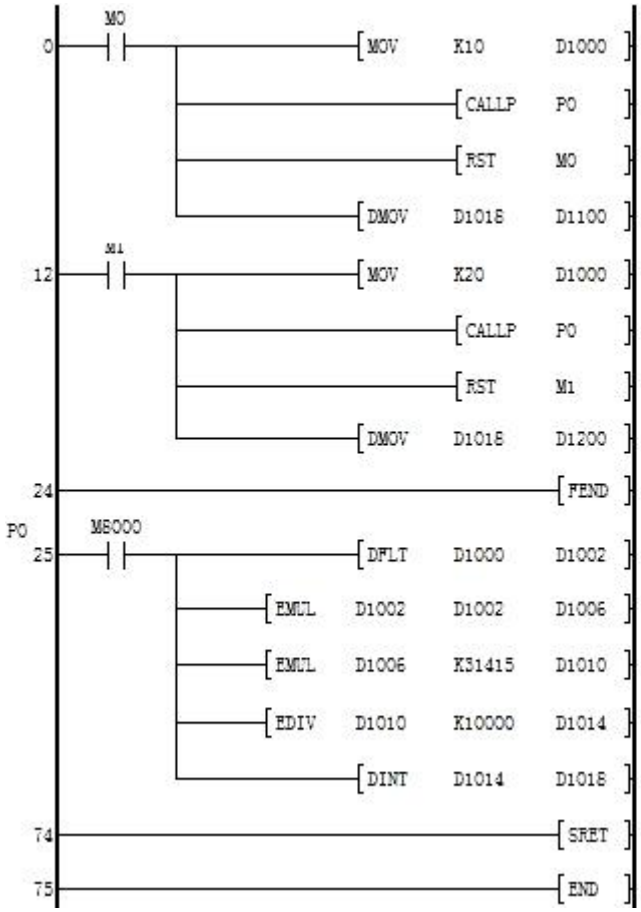
子程序1
执行SRET指令后返回到主程序。执行子程序1时，若M1为ON，则向标记P1的步跳转。

子程序2
执行P1的子程序后，使用SRET指令返回到P0的子程序。

图 6.子程序嵌套调用

2.2.5. 案例分析

不同的参数，可以调用同一子程序，得到最后的运算结果。



求不同半径圆的面积
M0置位，调用子程序P0，
计算半径为10的圆面积，
存入D1100中。

M1置位，调用子程序P0
计算半径为20的圆面积，存入
D1200中。

子程序
把半径转32位浮点数
运用面积公式$s=\pi r^2$
运算结果放入D1018中

图 7.子程序调用例程

2.3. 循环指令[FOR-NEXT]

2.3.1. 概要

从 FOR 指令开始到 NEXT 指令之间的程序按指定次数重复运行。

- 指令格式及对象软元件

助记符	功能	回路表示	程序步
FOR	循环范围开始	FOR S	3
NEXT	循环范围结束	NEXT	1

操作数	位软元件				变址			字软元件								其他	
S	X	Y	M	S	V	Z	修饰	KnX	KnY	KnM	KnS	T	C	D	R	K	H
					√	√	√	√	√	√	√	√	√	√	√	√	√

2.3.2. 功能与动作说明

- FOR～NEXT 指令的处理重复 N 次（源数据中指定的次数），重复 N 次数后，执行 NEXT 指令后面的程序处理，执行示意图如图 8 所示。
- 使用范围循环指令，搭配软元件的变址使用，可有效处理数据区间的运算问题。简化编程的同时减少梯形图步数，但一般情况下不能缩短梯形图演算时间。

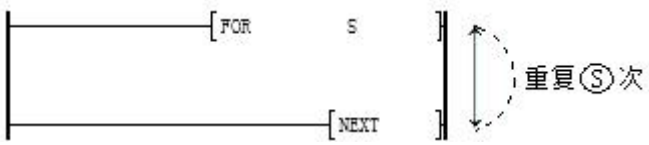


图 8.FOR-NEXT 执行示意图

2.3.3. 注意要点

- 由FOR 指令指定 FOR～NEXT 循环执行N 次后跳出 FOR～NEXT 循环往下继续执行。
- 指定次数范围 N＝K1～K32767，当指定次数范围N≦K1 时，都当作 K1 来处理。
- 当不需要执行 FOR～NEXT 循环时，可使用 CJ 指令来跳出循环，见 CJ-P 指令说明。
- 下列使用情况会导致 PLC 报错：

1. NEXT 指令在 FOR 指令之前。
2. 有FOR 指令没有NEXT 指令。
3. FEND 或 END 指令之后有 NEXT 指令时。
4. FOR 和 NEXT 指令个数不同时。

- FOR～NEXT 回路最多可嵌套使用 5 层，但要注意循环次数过多时，会使 PLC 扫描周期增加有可能造成逾时监视定时器动作，而导致错误产生，可使用 WDT 指令来改善。

2.3.4. 使用说明

1）1 层使用

利用循环指令，把 D1000-D1009 赋值 K1234，如图 9 所示。

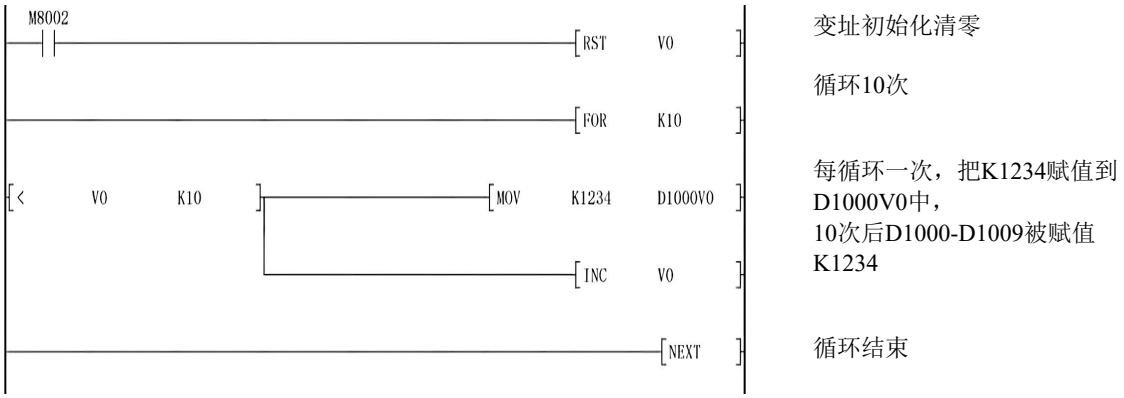


图 9.单层循环示例

2）多层嵌套

如图 10 所示，A 程序执行 10 次后再到对应的NEXT 指令以后的程序继续执行。而 A 程序每执行一次 B 程序会执行 5 次，所以B 程序合计共执行 10×5 ＝50 次，B 程序每执行一次C 程序执行 2 次，所以 C 程序合计共执行 10×5×2 ＝100 次。

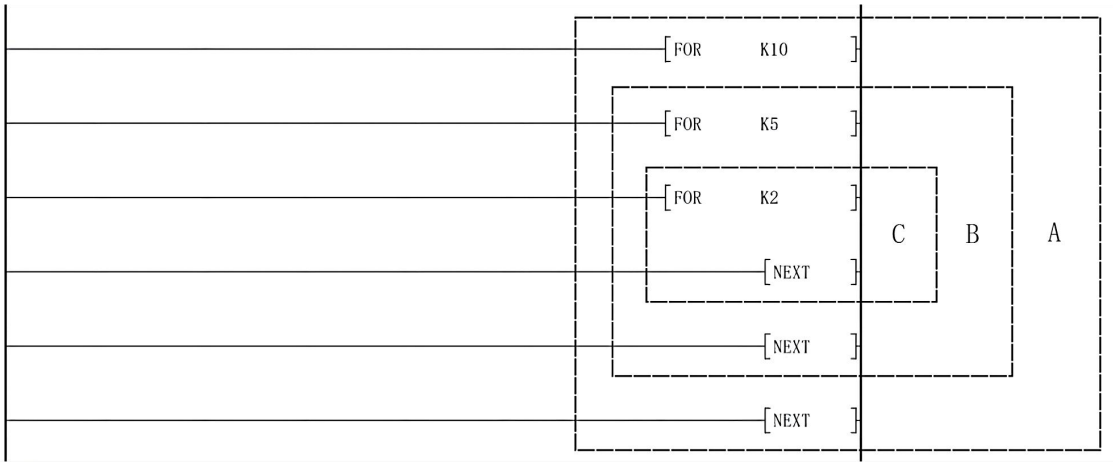


图10.FOR-NEXT 循环嵌套示意图

2.3.5. 案例分析

利用 FOR 循环的重复性，很好的简化运算，提高效率，图 11 是用双重 FOR-NEXT 循环很好的解决大量数据的排列顺序。

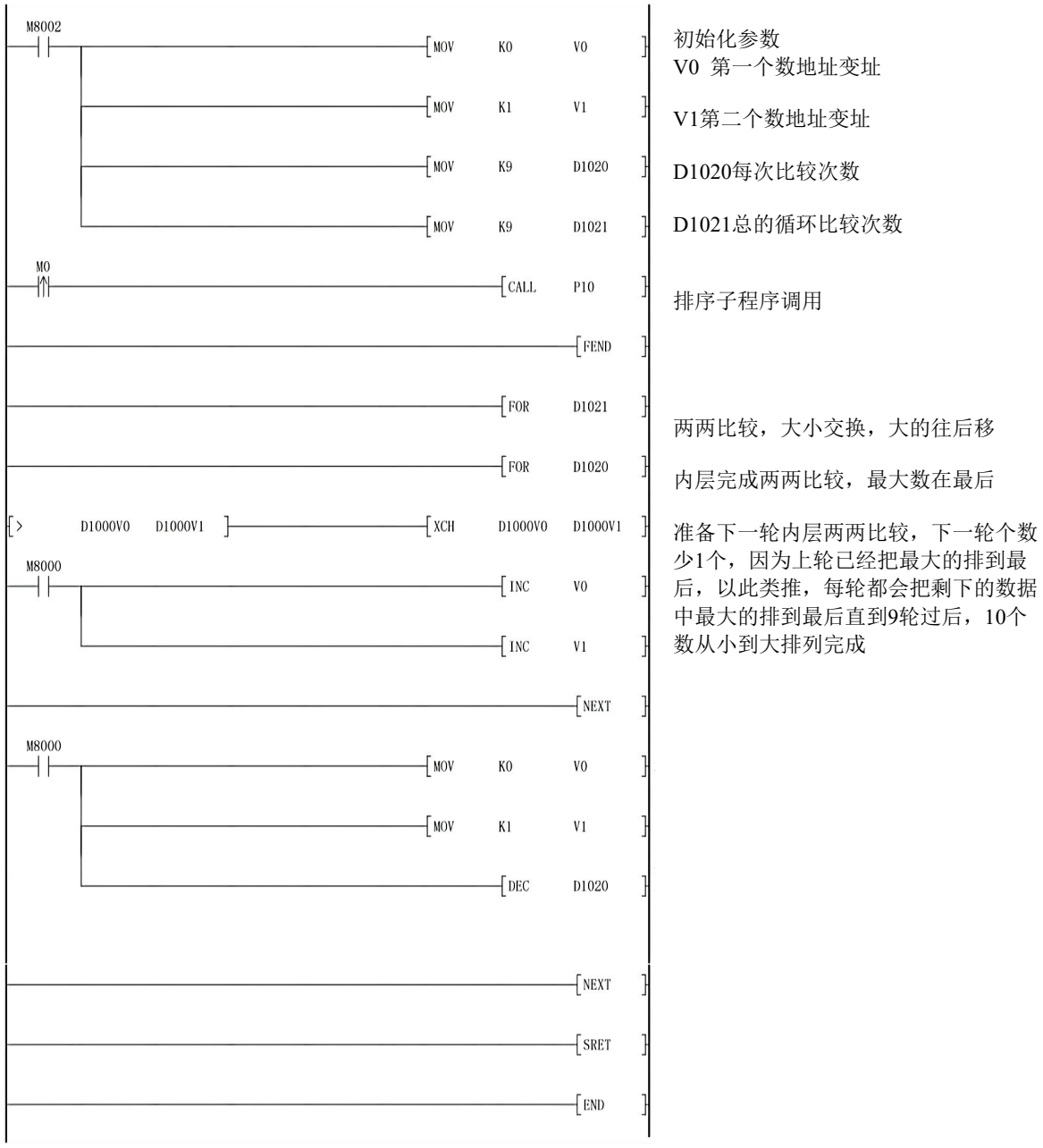


图 11.冒泡排序法排列数据示例

2.4. 跳转指令[CJ-P]

2.4.1. 概要

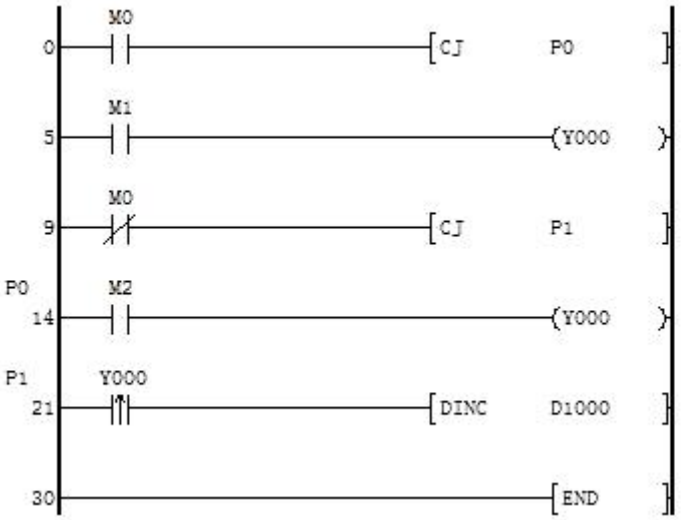
使 CJ、CJP 指令开始到指针(P)为止的程序不执行的指令。可以缩短循环时间(运算周期)和执行使用双线圈的程序。

- 指令格式及对象软元件

助记符		功能																程序步
CJ	P	程序跳转							回路表示 CJ Pn									3
操作数		位软元件				变址			字软元件									其他
指针 En		P0-P127																
		X	Y	M	S	V	Z	修饰	Kn	KY	Kim	Ens	T	C	D	R	K	H
								√										

2.4.2. 功能与动作要点

- 使用 CJ 或 CJP 指令使 PLC 程序中的某一部份不执行，以缩短扫描周期，以及使用双重输出问题， 如图 12 所示。



M0=ON，
则Y000受M2控制
M0=OFF，
则Y000受M1控制
通过CJ指令很好的解决双线圈输出问题。

图 12.CJ 解决双线圈输出问题

- 跳转执行中的程序（CJ 与其对应指针 Pn 之间的程序）各软元件动作情形说明：

1. Y、M、S 保持跳转发生前的状态。
2. 跳转前执行的定时器会继续计时。
3. 跳转前计数中的高速计数器会继续计数，且输出接点正常动作，执行的运动指令继续正常执行。
4. 一般计数器会停止计数，不会清零。
5. 一般应用指令不会被执行。
6. 计数器及定时器触点信号保留，待下次执行其指令时有效。

2.4.3．注意要点

- Pn：条件转移目的指针，使用 CJ 跳转时，必须有对应的指针编号，否则报错或出现异常。
- 指针 P 所指的程序若在 CJ 指令之前，需注意会导致 WDT 逾时错误，PLC 停止运转，请注意使用。
- 多个CJ 指令可重复指定同一指针P，但CJ 不可指向CALL 指定的指针 P，否则会产生错误。
- 使用CJ 跳转要注意程序结构，不能随意跳转，以免引起程序复杂或者错误等问题，例如尽量不要在

MC-MCR、FOR-NEXT、STL-RET、P-SRET、I-IRET 等两两之间跳转。

- 指定的指针可以重复，但标记号不能重复，如图 13 所示。

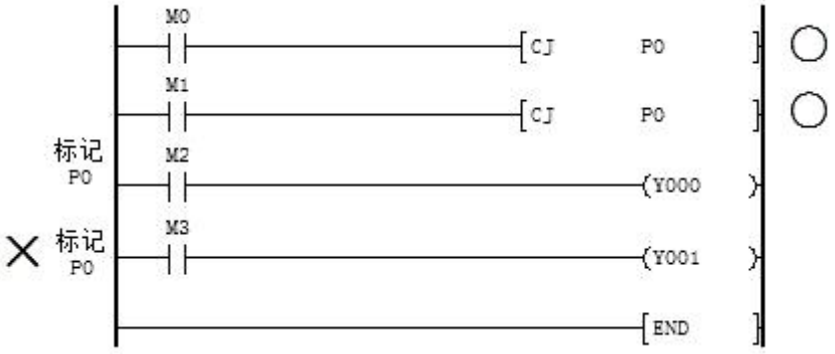


图 13.CJ 指针可重复而标记不可重复

2.4.4．使用说明

1）在主程序中的使用，

如图 14 所示，CJ 到标记之间的程序不再执行；END 语句行前不允许输入指针编号，否则程序编译不了。

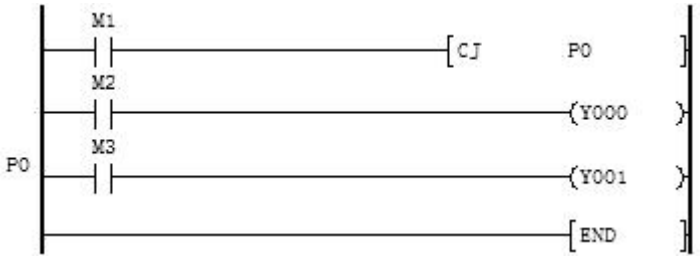


图 14.CJ 在主程序中使用示意图

2）在 MC-MCR、FOR-NEXT、STL-RET、P-SRET 中使用

- 主控指令 MC-MCR 和跳转指令的关系及动作内容如图 15 所示。存在 5 种情

况，但是②、④、⑤的运行会使运算变得复杂，所以请避免使用。

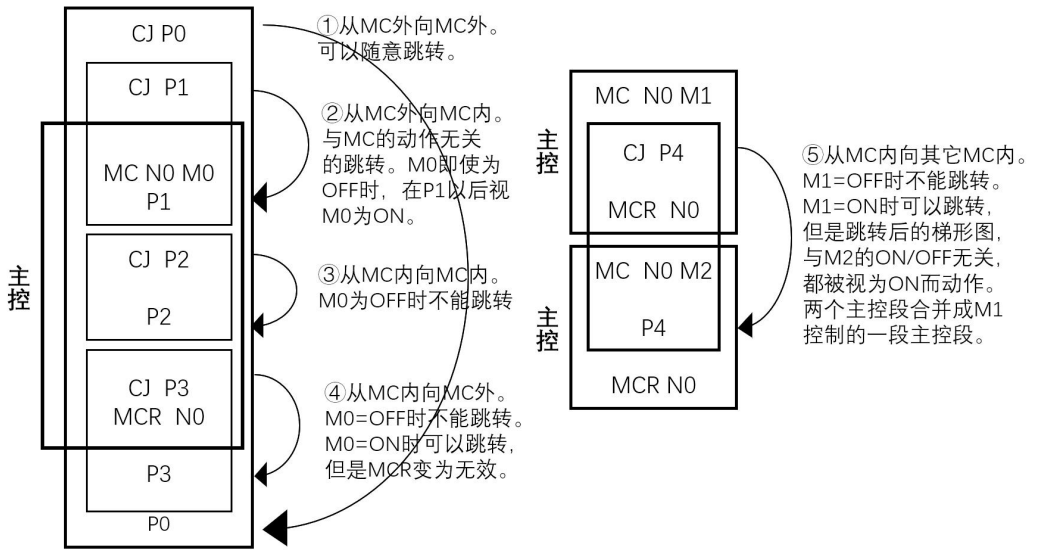
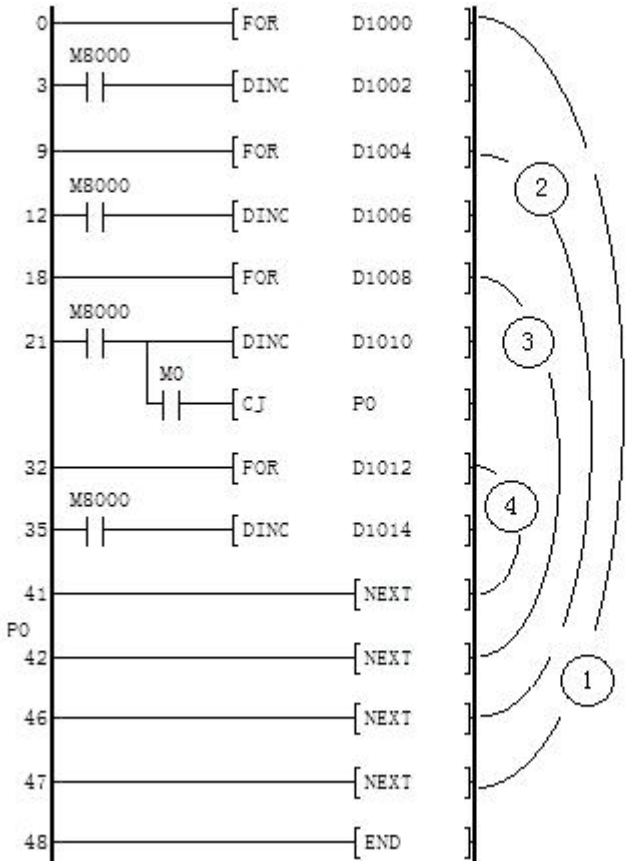


图 15.主控指令与跳转指令关系

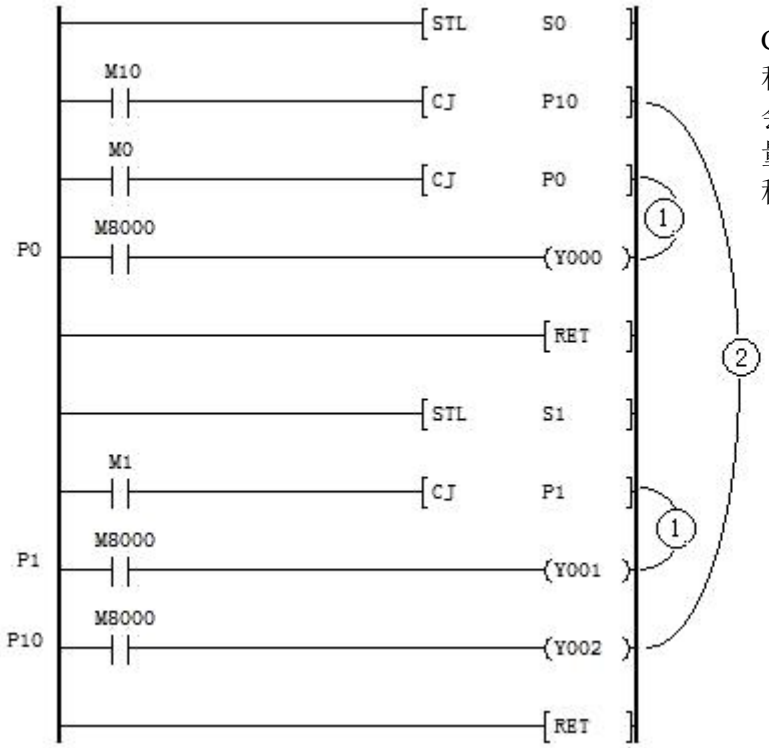
- FOR-NEXT 循环和跳转指令的关系及动作内容如图 16 所示，因循环指令都是NEXT 结束，所以跳转指针编号随便在哪个 NEXT 都可以，但运行情况可能比较复杂，所以除了在同一个 FOR-NEXT 内跳转外， 其他情况应避免使用。



CJ跳转在FOR-NEXT循环中，应尽量是跳转到对应的NEXT处，也就是说尽量在一个FOR-NEXT中实现跳转，而不要从一个FOR-NEXT跳转到另一个FOR-NEXT中，虽然都是NEXT结尾而不会报错，但运行结果比较复杂，所以尽量避免使用。

图 16.FOR-NEXT 指令与跳转指令关系

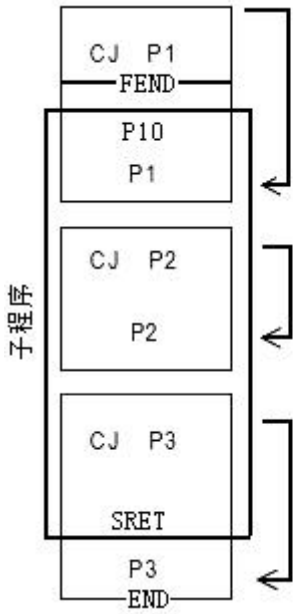
- CJ 不可以从一个STL 跳转到另一个 STL，以免程序运算复杂，如图 17 所示。



CJ与STL存在左边①②两种使用情况，第②种情况会使程序运行复杂，应尽量避免使用，第①种用法程序执行简单，明朗

图 17.步序指令与跳转指令关系

- CJ 不可以从一个 P-SRET 跳转到另一个 P-SRET，会导致子程序返回错误，程序运算复杂或错误，只能在子程序内部跳转，如图 18 所示。



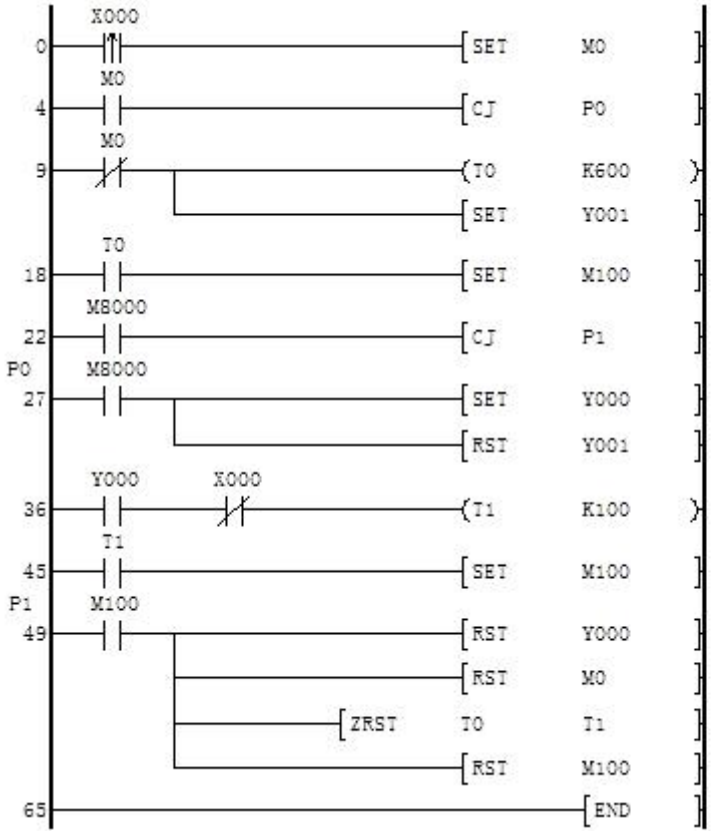
①从外部跳转到子程序内部，P1后面的程序会执行，但SRET返回会没有目标，执行后果无法预估，应避免这种跳转

②子程序内部跳转，子程序有入口，有返回，不会引起异常

③跳出子程序外部，会导致子程序P10无返回指令，无法预估执行结果，应避免这种跳转

2.4.5. 案例分析

- 例程中模拟路灯或楼道灯的控制过程，如图 19 所示。



模拟路灯高亮和低亮状态

X0有人经过，跳转到P0，

灯高亮(Y0)，10秒钟后若无人，则高亮关闭，低亮打开，维持1分钟，然后关闭低亮，若中间有人过来，则继续转高亮，如此往复

图 19.路灯或楼道灯模拟

2.5. 步进控制指令[STL-RET]

2.5.1. 概要

步进梯形图指令的程序，是以机械的动作为基础，对各工序分配状态 S，作为连接在状态触点(STL 触点)中的回路，对输入条件和输出控制的顺序进行编程。使用步进指令不仅可以简单、直观地表示顺序操作的流程图，而且可以非常容易地设计多流程顺序控制，并且能够减少程序指令数，程序易于理解。

- 指令格式及对象软元件

助记符	功能							回路表示									程序步
STL	步进梯形图开始							STL Sn									2
RET	步进梯形图结束							RET									1
操作数	位软元件				变址			字软元件									其他
Sn																	
	X	Y	M	S	V	Z	修饰	KnX	KnY	KnM	KnS	T	C	D	R	K	H
				√													

- 指令说明

★ 步进梯形图指令（STL）是利用内部软元件状态（S），在顺控程序上面进行工序步进控制的指令。返回（RET）是表示状态（S）流程的结束。置位指令 SET 是 STL 指令状态开始转移必备条件（SET 一旦为ON，不需自保，解除用 RST）。

★ 每一个流程书写都是独立的，写法上不能嵌套书写。在流程执行时，不一定要按 S0、S1、S2……的顺序执行，流程执行的顺序在程序中可以按需求任意指定。可以先执行 S10 再执行 S5，再执行 S0。

★ 执行 RET 指令，意味着步进梯形图回路的结束，在希望中断一系列的工序而在主程序编程时，同样需要 RET 指令，该指令可多次编程。

★ STL 指令和 RET 指令必须配对使用。在一系列 STL 指令之间，可以不书写 RET 指令但在 STL 指令的最后，若没有书写 RET 指令，则会导致运行出错。

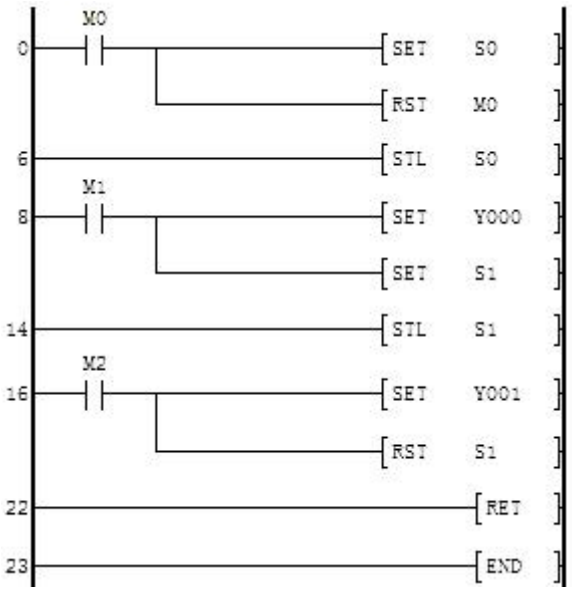


图 20.STL 步序使用示意图

如图 20 所示为 STL 步序功能的简单应用编程，M0 为 ON，进入步序 S0，在 S0 中，M1 为 ON 则退出 S0

（S0 自动复位），进入 S1，然后在 S1 中 M2 为 ON，退出步序 S1，整个步序流程结束。

2.5.2. 功能与动作要点

步进指令是专为顺序控制而设计的指令。在工业控制领域许多的控制过程都可用顺序控制的方式来实现， 使用步进指令实现顺序控制既方便实现又便于阅读修改。

- 步进梯形动作：

步进梯形是由很多个步进点组成，每一个步进点代表控制流程的一个动作，每一个步进点都将提供三个功能：

a) 驱动输出线圈。

b) 指定转移条件。

c) 指定步进点的控制权要转移给下一个步进点。

其中后两项功能是必不可少的（如图21，简称步序三要素）。

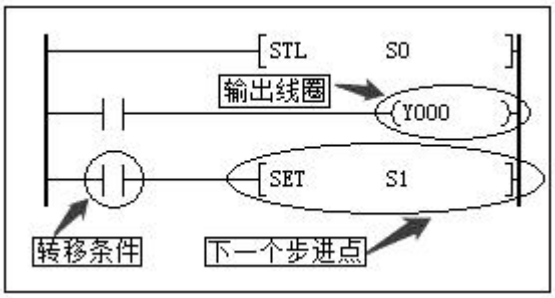


图 21.步序三要素

- STL-RET 步序指令：当为 ON 时正常运行步序内的程序，当为 OFF 时或从 ON 变为OFF 时，步序状态内的软元件动作如下表所示：

一般定时器	计时值归零，线圈失电，触点不动作
累计型定时器	线圈失电，计时值及触点保持目前状态
计数器	线圈失电，计数值及触点保持目前状态
OUT 指令驱动的线圈	全部不受电
SET， RST 指令驱动的元件	保持目前状态
应用指令	全部不动作，但 FOR-NEXT 循环回路仍会来回执行 N 次，但 FOR-NEXT 间的任何指令与步序内其它指令动作相同

注：上表状态在其步序为 OFF 时，不影响主程序及其他正在执行的步序，即不存在双线圈输出问题。

- CPU 在步序为 OFF 时，也会扫描步序程序，所以步序程序并不能节约扫描周期，扫描结果不影响其他程序的运行。
- 当前状态（S0）向下一个状态（S1）转移时，在当前扫描周期，S0=OFF，S1=ON，此时两个状态内的动作均会执行；下一个扫描周期执行时，状态（S0）内

的所有动作不被执行，所有 OUT 元件的输出均被断开，如图 22 所示。

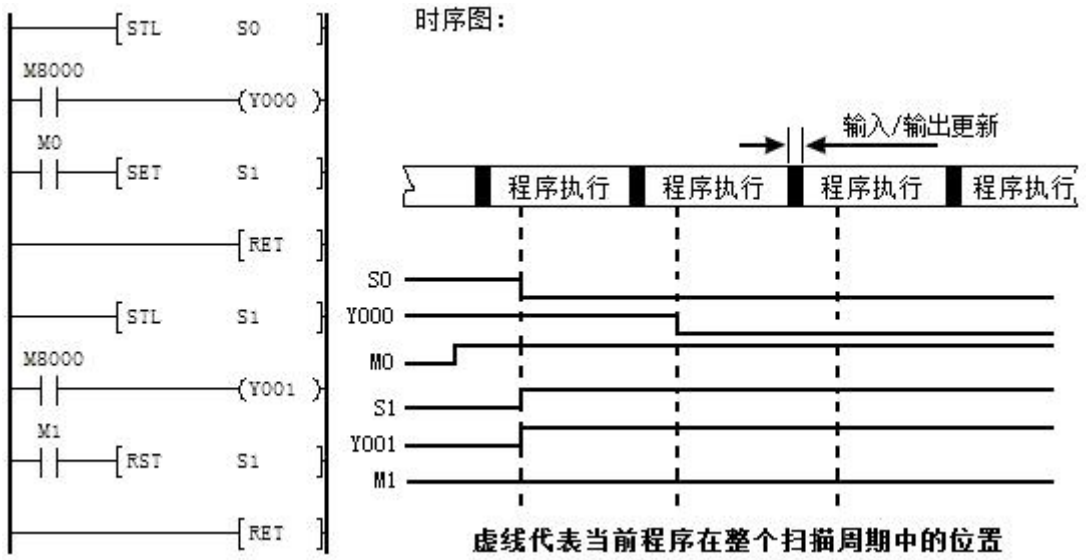


图 22.步序转移时序图

2.5.3. 注意要点

- STL 最前头的步进点称之为初始步进点，使用初始步进点做为流程的开始，以 RET 指令做结束构成一个完整的流程。
- 当STL 指令完全不被使用时，步进点 S 可当成一般辅助继电器来使用，有 STL 指令使用时，要慎重使用（STL 内除跳转转移外不使用其他任何 S，步序外不使用跟步序点相同的 S，S 作为条件可随便使用）。
- 当STL 指令使用时，步进点 S 的号码不可重复使用。
- 一个分支流程所使用的分支步进（并行及选择性分支）最多 8 个。
- 流程中的某一步进点可指定跳到另一个流程的任一个步进点。
- 步序编程顺序：先进行驱动，再进行转移，不能颠倒，如图 23 所示。

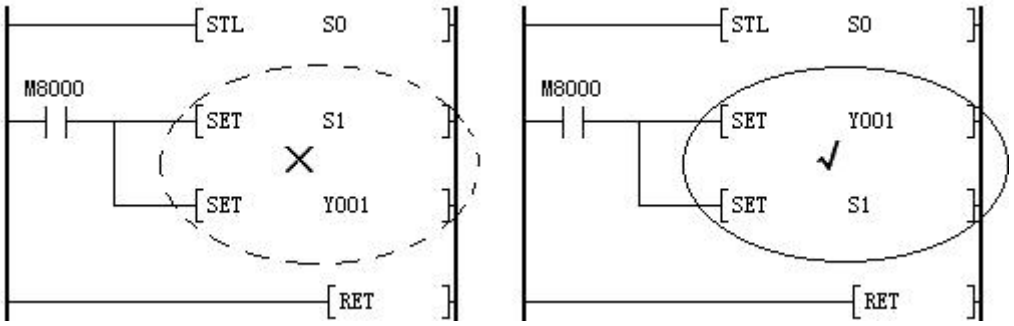


图 23.先驱动再转移

- 步序中不可使用 MC/MCR 指令。STL 指令不可使用于一般子程序内及中断服务子程序内。STL 指令中并不禁止使用 CJ 指令，但会使动作更加复杂，应尽量避免。

- 当同一负载需要连续多个步序状态驱动时，可使用多重输出，在步序程序中，不同时“激活”的双线圈是允许的，如图 24（a）所示。另外，相邻步序使用 T，C 软元件，不能使用同一编号，如图 24（b）所示。

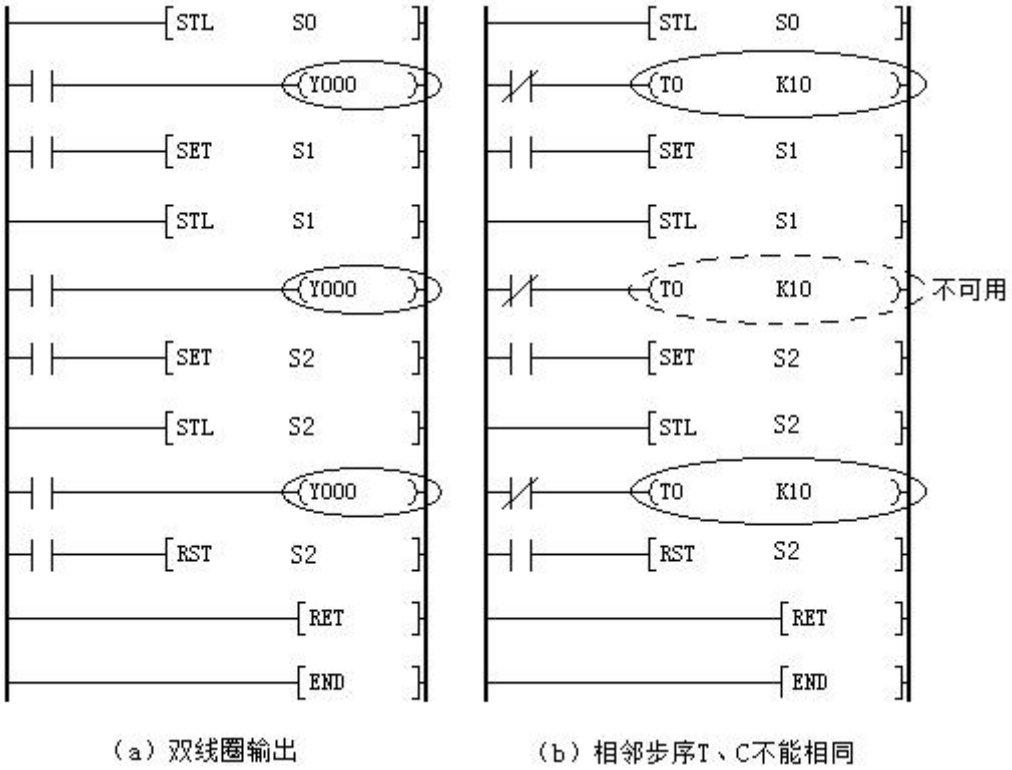


图 24.相邻步序线圈操作

- 输出的互锁：在状态转移过程中，只有一瞬间(1 个运算周期)两个状态会同时为 ON。因此，在不可以同时接通的一对输出之间（例如正转与反转），为了避免同时为 ON，请在程序中执行如图 25 所示的相互的互锁。

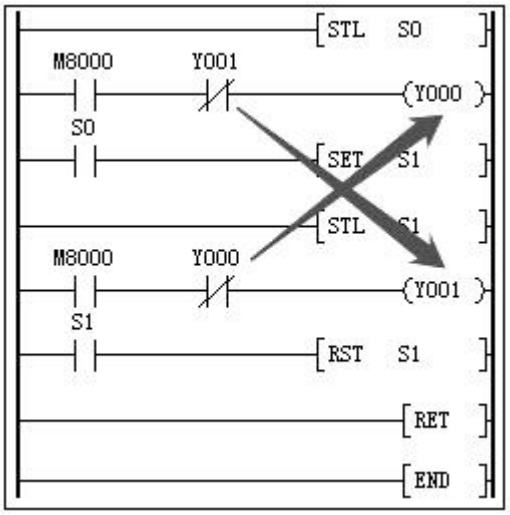


图 25.输出的互锁

2.5.4. 使用说明

1）流程分类

单一流程：一个程序中只有一个流程且不含选择分支、选择汇合、并行分支、并行汇合的简单流程。即在同一时刻，只有一个状态在工作，如 图 26 所示。

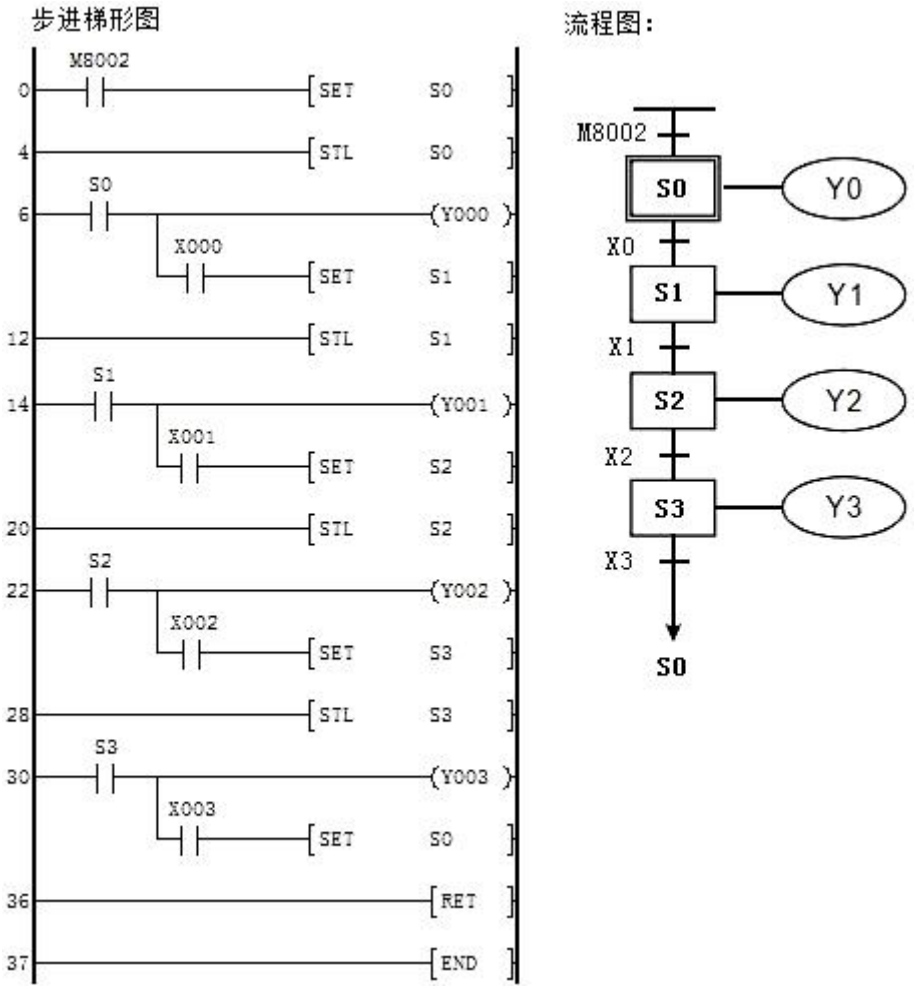


图 26.单一流程步序

复杂单一流程：一个程序中只有一个流程包含选择分支、选择汇合、并行分支、并行汇合等流程（详见2.5.4第三小节<分支结构>）。

复数流程：一个程序中有复数个单一流程，数量暂时没有限定，建议最多 10 个。

2）循环与跳转

应用步进顺控指令时，SET Si 是状态转移的首要条件。跳过某些状态直接向下一状态转移或向系统外转移，称为跳转。以符号↓表示转移的目标状态。向上面状态转移称为重复或循环。同样↓表示转移的目标状态，如图 27 所示。

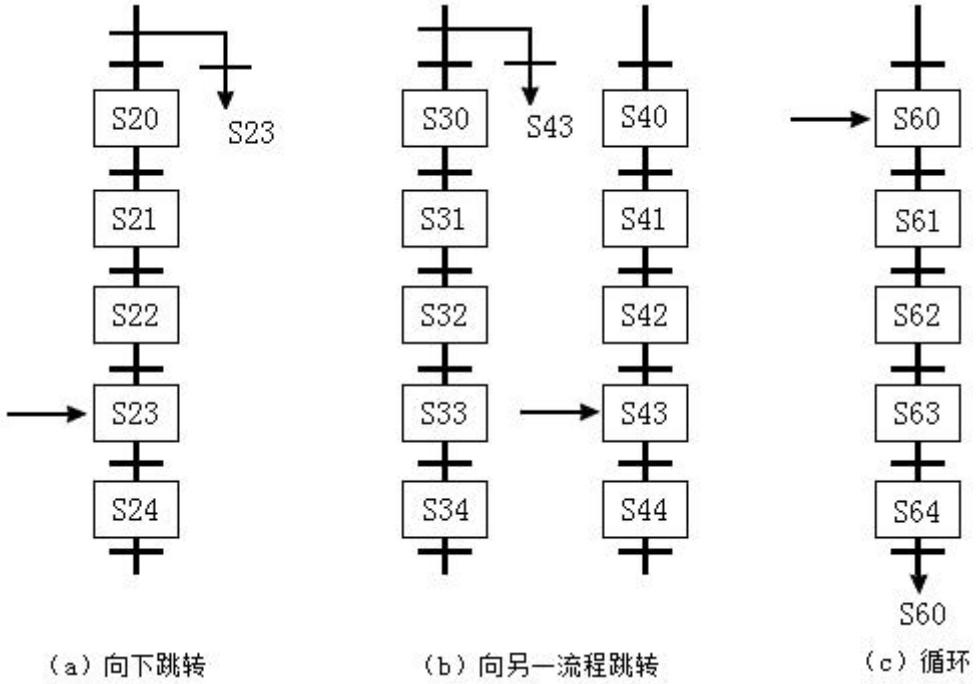


图.27 循环与跳转示意图

3） 分支结构

1>并行分支

现在的状态在条件成立时，同时转移到多个状态，属于并行分支结构，如图所示，状态从 S0 转移，当定时器 T0=0N 时， 同时转移到S1， S2， S3，S4 状态。

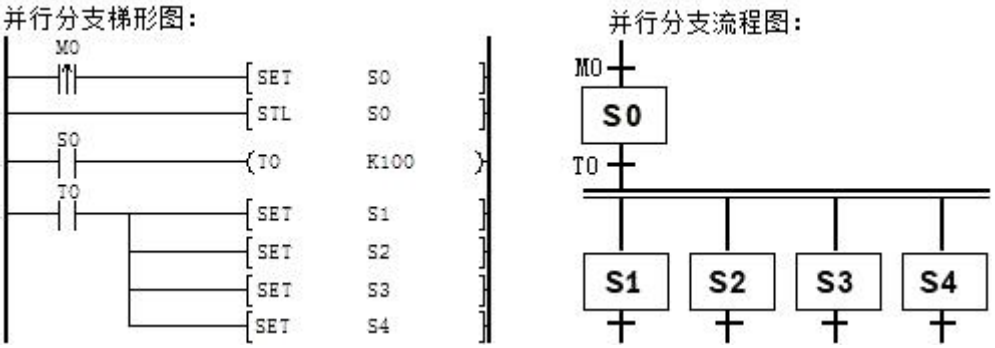


图 28.并行分支

2>选择性分支

现在的状态在个别条件成立时，转移至个别状态，属于选择性分支结构，如图所示，状态是从 S0 转移，当 M1=On 时，转移到S1， 当 M2=On 时，转移到 S2，当M3=On 时，转移到 S3，当M4=On 时，转移到 S4。

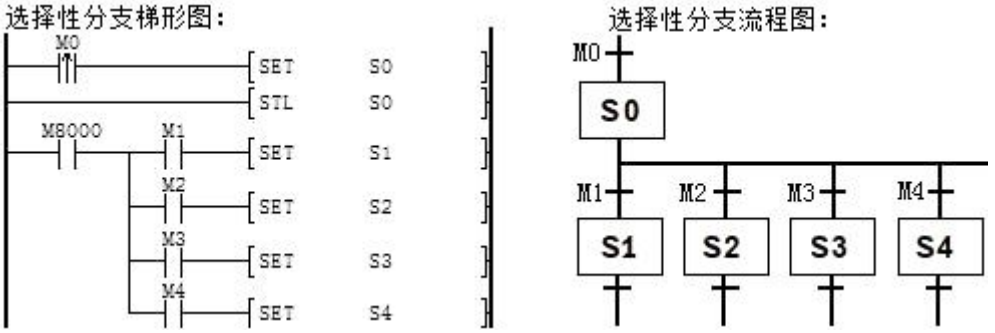
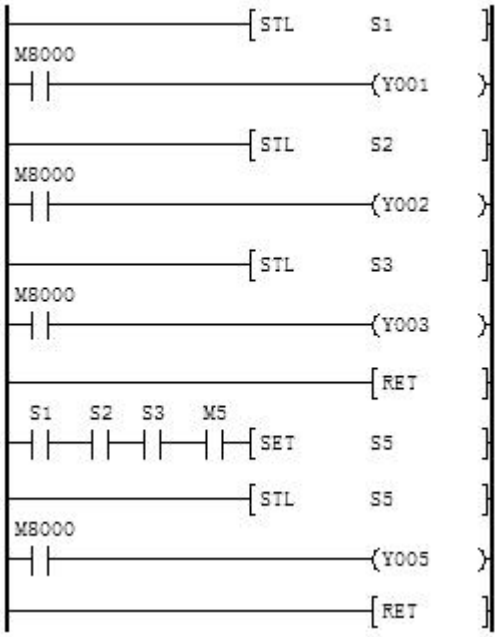


图 29.选择性分支

3>并行汇合

如图所示，连续的 STL 指令并行汇合结构，连续的状态输出后在条件成立时，转移到下一个状态。并行汇合的意思是指几个状态要同时成立时，才可以状态转移。 状态 S1，S2，S3 同时有效，再在 M5=ON 时， 才会状态转移到 S5。

并行汇合梯形图：



并行汇合流程图：

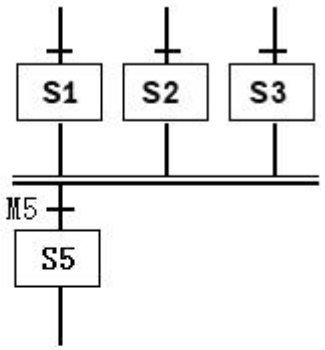
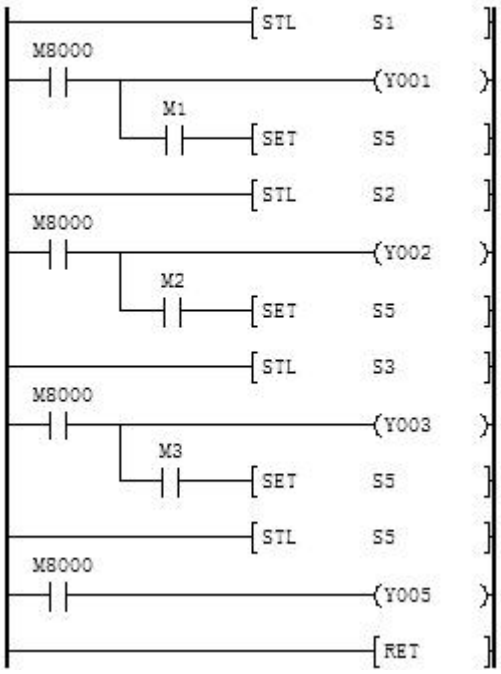


图 30.并行汇合

4>选择性汇合

如图所示是属于选择汇合，S1、S2、S3 三种状态，某一状态中的转移条件先成立就状态转移至 S5。

选择性汇合梯形图：



选择性汇合流程图：

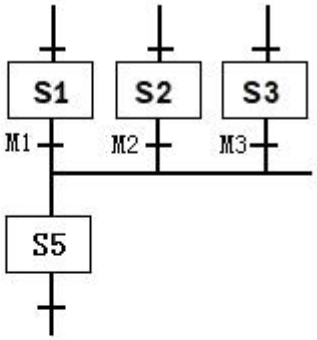


图 31.选择性汇合

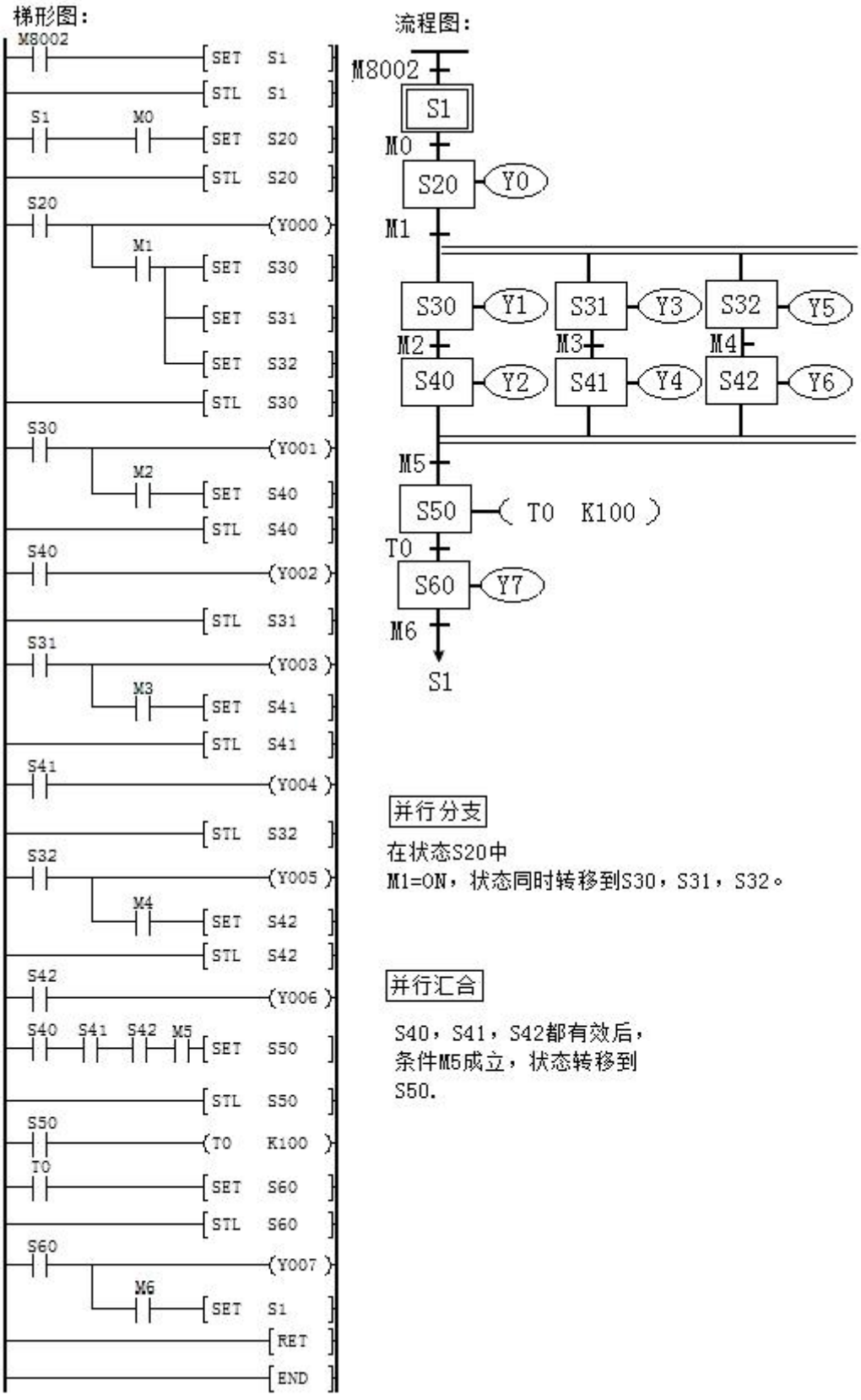


图 32.并行分支、并行汇合

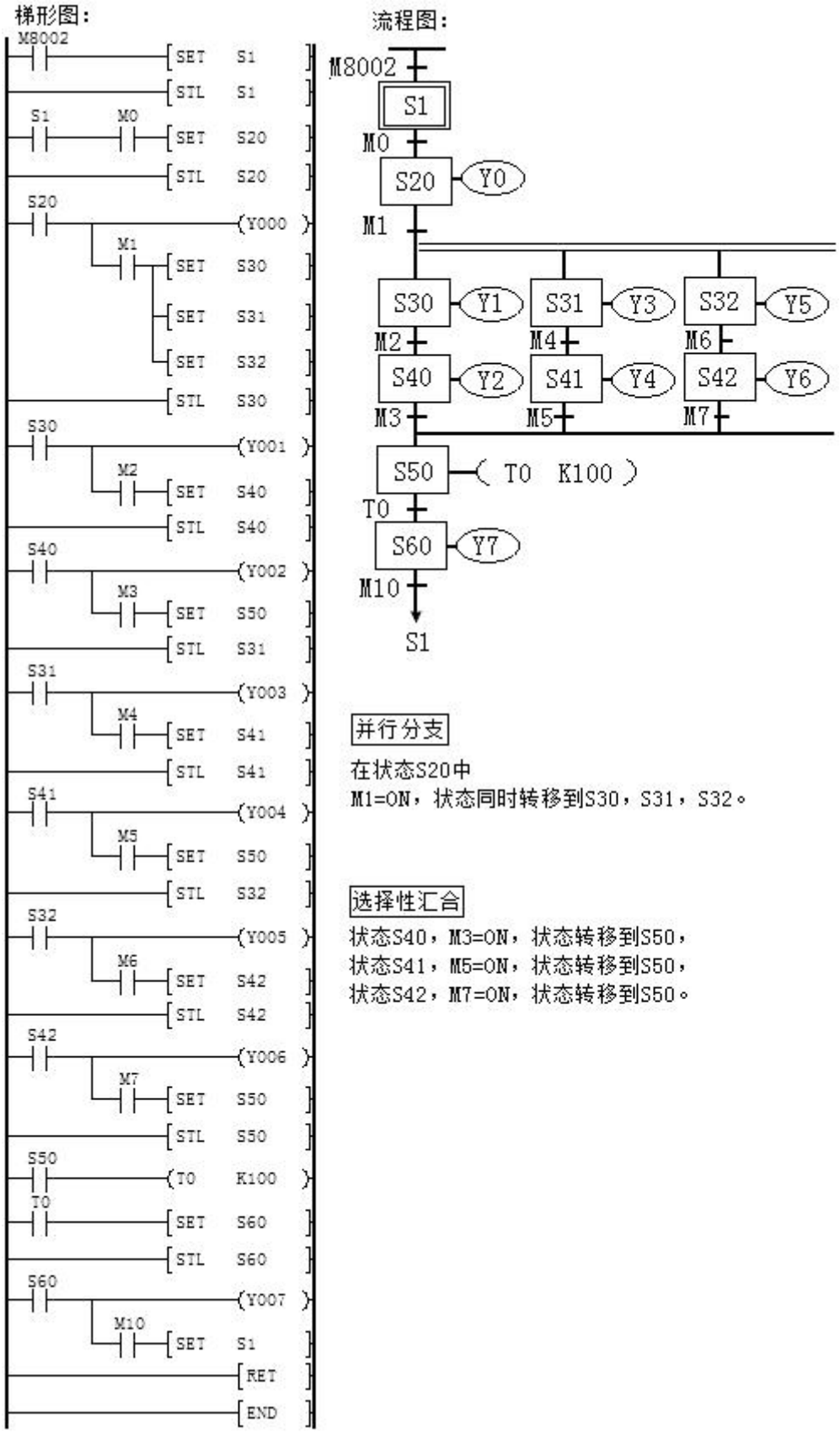


图 33.并行分支、选择性汇合

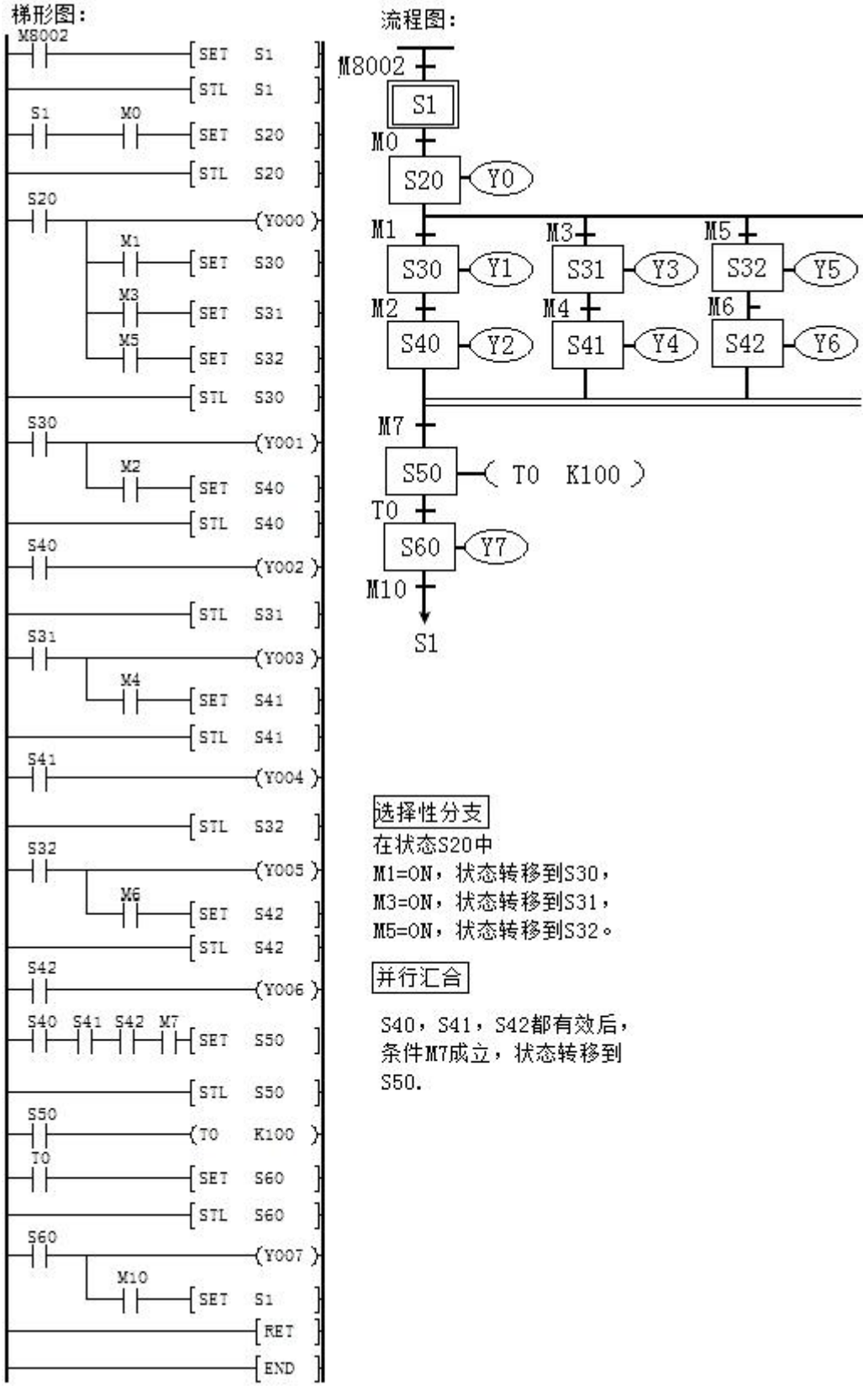
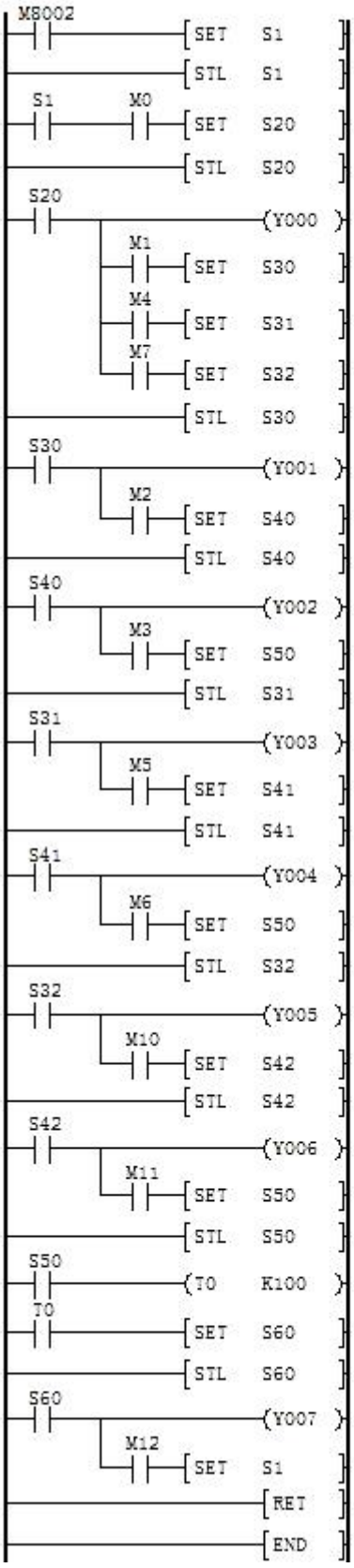
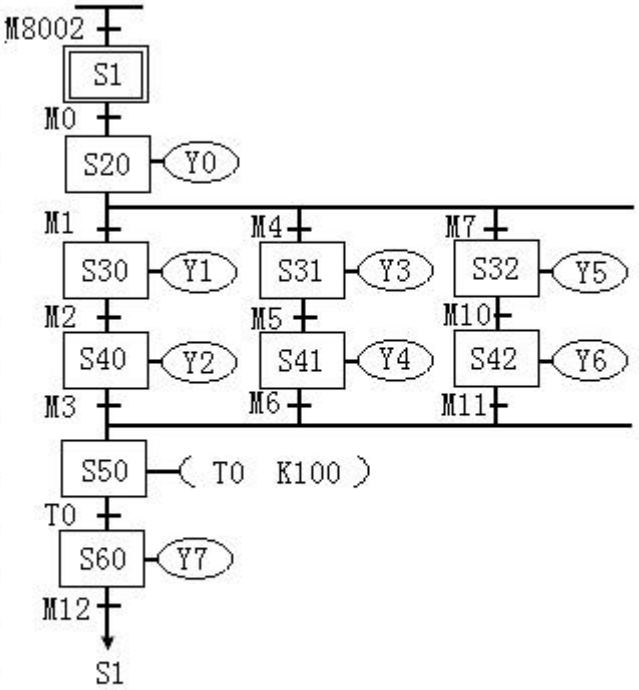


图 34.选择性分支、并行汇合

梯形图：



流程图：



选择性分支

在状态S20中

M1=ON，状态转移到S30，

M4=ON，状态转移到S31，

M7=ON，状态转移到S32。

选择性汇合

状态S40，M3=ON，状态转移到S50，

状态S41，M6=ON，状态转移到S50，

状态S42，M11=ON，状态转移到S50。

图 35.选择性分支、选择性汇合

某糖果包装需要 30 个糖果 A，20 个糖果B，10 个糖果 C，启动包装按钮 X0 后，开始包装，糖果 A、B、C 传送计数都是单独的，互不影响，但最后包装需要三种糖果都完成才能一起包装，再通过传送带把糖果一起送走，梯形图及流程图如图 36 所示。

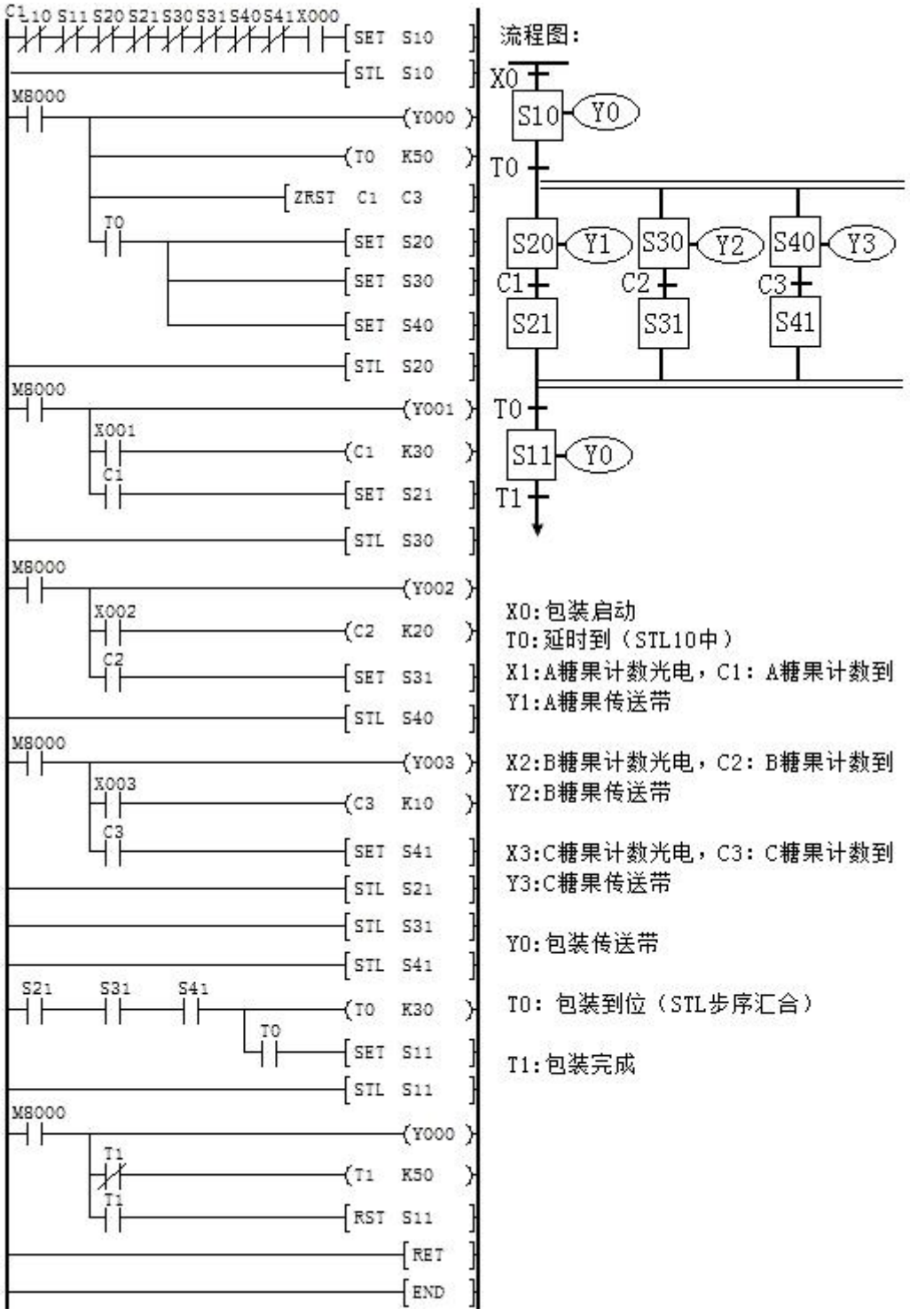


图 36.步序实际应用示意图

2.6．主程序结束指令[FEND]

2.6.1. 概要

此指令代表着主程序结束，当 PLC 执行至此指令时，与 END 指令相同 。

助记符	功能							回路表示								程序步	
FEND	主程序结束							FEND								1	
操作数	位软元件				变址			字软元件								其他	
无																	
	X	Y	M	S	V	Z	修饰	KnX	KnY	KnM	KnS	T	C	D	R	K	H

2.6.2. 功能与动作要点

执行 FEND 指令后， 会执行与 END 指令相同的输出处理、输入处理、监视定时器的刷新，然后返回到0步的程序。在编写子程序和中断程序时需要使用这个指令。

1.CJ 指令的情况

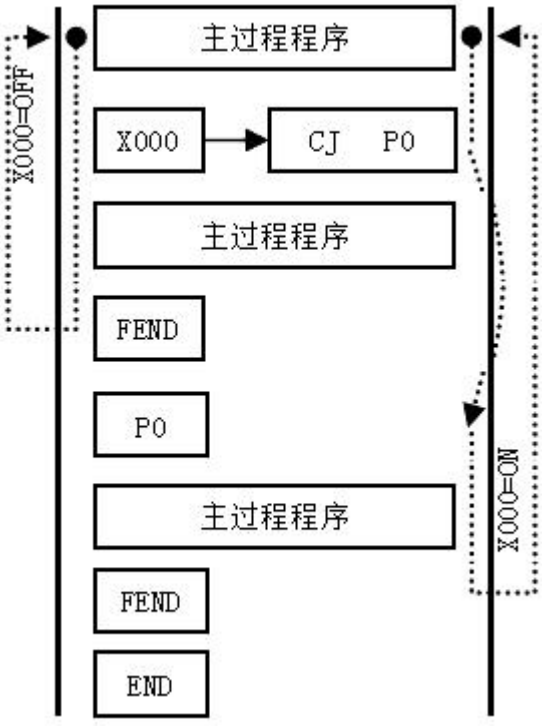


图 37.CJ 指令的情况

2.CALL 指令的情况

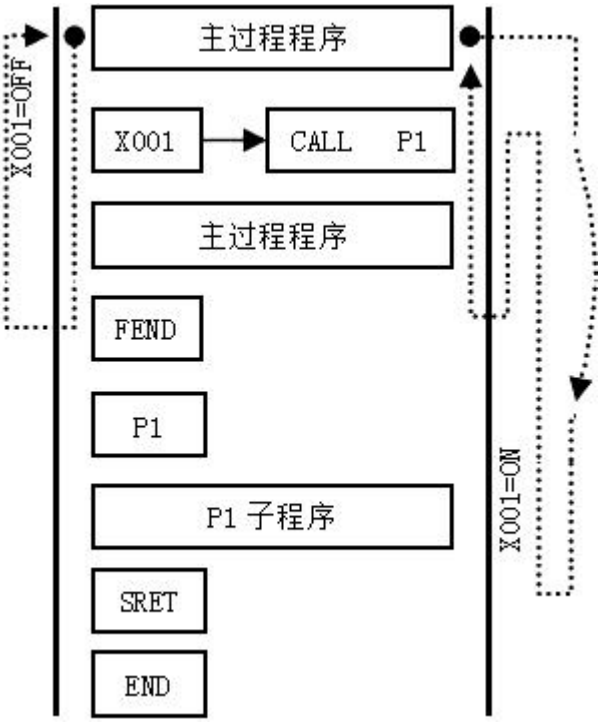


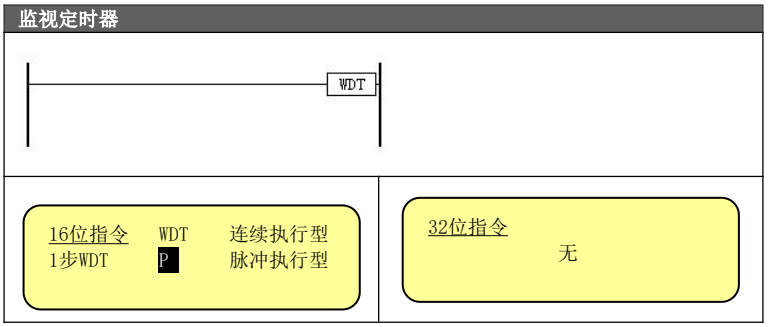
图 38.CALL 指令的情况

2.6.3. 注意要点

1. 多次应用FEND 指令时必须在最后的 FEND 指令和 END 指令之间编写子程序和中断子程序。
2. 子程序P-SRET 的编号必须在 FEND 指令后，且必须使用SRET 指令结束。
3. 子程序P-SRET 在编号与 SRET 间不允许编写 FEND。
4. 中断子程序 I-IRET 指针编号与 IRET 间不允许编写 FEND。
5. 循环指令FOR—NEXT 之间，不允许使用 FEND 指令。

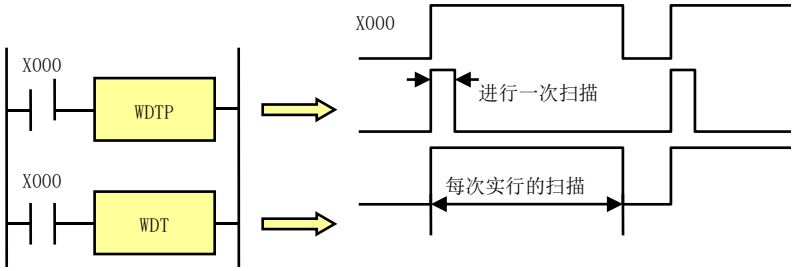
2.7. 监视定时器[WDT]

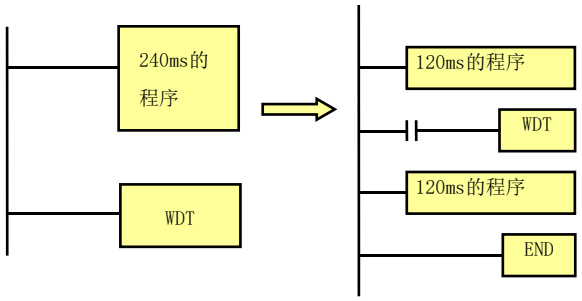
在顺控程序中防止梯形图演算时间较长导致系统脱机，可执行监视用定时器刷新指令。



功能与动作

- 当PLC的运算周期（0-END及FEND指令执行时间）超过D8000(监视定时器时间)的设定值时，PLC出错指示灯点亮同时停止工作，因此需要在编程过程中插入使用该指令。
- 在CJ指令标签的步号比CJ指令小时，可在标签后编写WDT指令。
- 若循环次数较多，可在FOR/NEXT指令之间使用该指令。

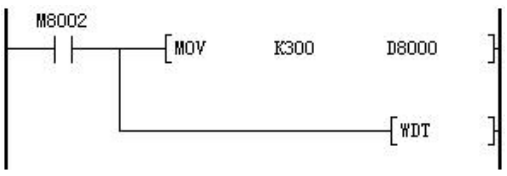




例如，将240ms的程序一分为二，在中间插入WDT指令，则前后两部分程序都在200ms以下。

监视定时器设置

通过改写D8000(监视定时器时间)的内容，可改变监视定时器的检出时间。输入下述程序，此段后的PLC程序将采用新的监视定时器时间执行监视。



- 初始演算周期，将监视定时器时间设置为300毫秒。
- 如果程序内未运行监视定时器时间刷新指令(WDT)，则在END处理时，D8000的值才有效。

系统构成中连接有很多台定位，凸轮开关、ID接口、链接、模拟量等特殊扩展设备时，PLC运行时进行的缓冲存储器初始化时间会延长，从而使扫描时间延长。
在这种情况下，监视定时器有可能出错，因此请在起始步附近输入上述程序，将监视定时器时间延长。

四、中断使用手册

MC系列PLC具有丰富的中断资源，用户可在程序中调用不同的中断实时处理多种不同类型的任务。综合运用不同的中断，可以实现MC系列PLC与多个外设同时运行，增强系统实时性，增强系统对故障的处理能力。

1、中断资源

1.1 中断种类

MC 系列PLC共有四个中断资源：软中断、定时器中断、外部输入中断和高级中断。

中断名称	个数	中断标号	说明
外部输入中断	4	I0□□、I1□□、I2□□、I3□□	外部输入信号 X010～X013 作为中断源，当外部输入信号到来，执行相应中断程序。
定时器中断	3	I6□□、I7□□、I8□□	定时器定时时间作为中断源，当定时时间到就执行相应中断程序。
系统软中断	5	I010、I020、I030、I040、I050	设置中断申请位执行相应中断程序。
高级中断	8	I900～I907	用于实现高速并发任务的处理以及插补算法的实现。

1.2 中断标志位

MC系列PLC共四种中断资源，合计 24 个中断指针号，其中普通中断总共有 16 个，每个中断对应有两个相关的特殊辅助继电器，一个是中断禁止标志，一个是中断申请字，高级中断共有 8 个，每个高级中断除了中断禁止标志位和中断申请字外，还有一个中断激活标志位和一个进程申请字。

- 中断禁止标志：准许和禁止中断申请进入，ON 禁止中断申请进入，此时即使中断申请字信号有效也不会进入中断，OFF 允许中断申请进入。
- 中断申请字：在中断禁止标志位复位的前提下，申请进入中断，ON 申请应答，进入中断后变成OFF。

中断标号	中断禁止标志	中断申请字
	ON（禁止中断）OFF（使能中断）	ON（申请应答）OFF（申请不应答）
I000	M8048	M8032
I001	M8049	M8033
I100	M8050	M8034
I101	M8051	M8035
I200	M8052	M8036
I201	M8053	M8034
I300	M8054	M8038
I301	M8055	M8039
I6□□	M8056	/
I7□□	M8057	/
I8□□	M8058	/
I010	M8059	M8043
I020	M8060	M8044
I030	M8061	M8045
I040	M8062	M8046
I050	M8063	M8047

- 进程申请字：高级中断特有的辅助继电器，当高级中断中配置了进程中断时，进程申请字起作用，ON 状态申请进入进程中断，进入进程中断后为 OFF。
- 中断激活标志：高级中断特有的辅助继电器，进入中断并且中断未退出时，中断激活标志位 ON。

中断标号	中断禁止标志	中断激活标志	中断申请字	进程中断申请字
	ON（禁止中断） OFF（使能中断）	ON（进入中断） OFF（未进入中断）	ON（申请应答） OFF（申请不应答）	ON（申请应答） OFF（申请不应答）
I900	M8064	M8072	M8080	M8088
I901	M8065	M8073	M8081	M8089
I902	M8066	M8074	M8082	M8090
I903	M8067	M8075	M8083	M8091
I904	M8068	M8076	M8084	M8092
I905	M8069	M8077	M8085	M8093
I906	M8070	M8078	M8086	M8094
I907	M8071	M8079	M8087	M8095

1.3 系统时间

MC系列PLC和中断相关的系统时间主要有三个：系统基时、系统检测时间、1ms 定时。

(1) 系统基时：PLC内部最小时间单位，50us 一个时间刻度，主要用途是作为其它时间的最小单位， 其它时间都是基于这个最小时间单位；

(2) 系统检测时间：两个系统基时（100us），主要用来检测系统的相关标志位以及高速计数等功能， 比如中断相关的申请位等、高速计数的分辨率等；

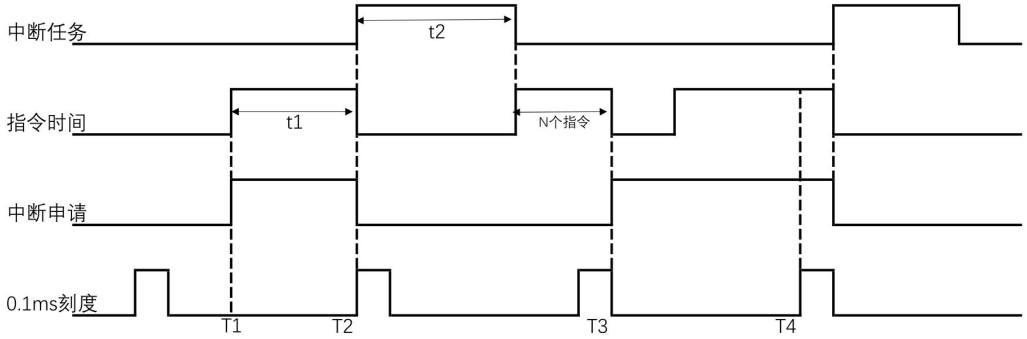
(3) 1ms 定时：和运动相关的时间，主要应用于系统两个 1ms 中断的中断申请位（I010 的中断申请位 M8043）和进程申请位（I900 的进程申请位 M8088）控制。

(4) 上述三个时间单位都是系统软件计时，不是绝对意义上的时间，根据系统内部执行的任务类别和个数不同，时间有细微的差别。

2、中断处理流程

2.1 中断执行流程

中断就是指某事件的发生引起 CPU 暂停当前程序的运行，转入对所发生事件的处理，处理结束又回到程序被打断处继续执行这样的一个过程。MC 所有的中断都基于系统内部的 50us 定时基时处理，比如中断申请位是每间隔 0.1ms（2 个定时基时）检测一次。梯形图中系统每执行完一条梯形图指令后，扫描一次中断申请，在梯形图指令执行过程中不会扫描中断申请，因此，中断最快的反应时间约为一个指令运行时间， 最慢响应时间为 0.1ms。



如上图所示为系统扫描中断申请和中断进入的时序图，从左向右看。

中断最快反应时间：T1 时刻在两个 0.1ms 时刻之间，正好扫描到中断申请，并且执行完一条梯形图指令后，刚好到达下一个 0.1ms（T2）时刻，中断进入。

中断最慢反应时间：T3 时刻刚好上一个 0.1ms 时刻过去，中断申请未扫描到，那么中断只有在下一个 0.1ms 时刻（T4）并且执行完一条梯形图指令后进入中断。

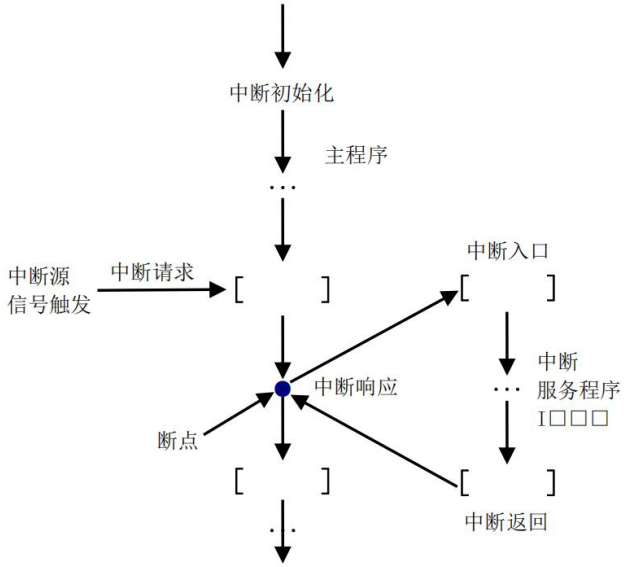
中断处理分为四个阶段：

(1) 保存被中断程序的现场，其目的是为了在中断处理完之后，可以返回到原来被中断的地方继续执行；

(2) 分析中断源，判断中断原因，当有多个中断同时请求的时候还要考虑中断的优先级；

(3) 转去执行相应的处理程序；

(4) 恢复被中断程序现场，继续执行被中断程序。



●梯形图是周期扫描型的处理机制，为了能够响应实时性比较强的任务，引入了梯形图中断的概念。

●可利用在梯形图母线左端添加I 标号的方式指定一个梯形图中断。

●系统每扫描完一条指令都会扫描一次所有的中断申请。若中断源满足触发条件且申请有效，则执行相应的中断服务程序。

●中断执行的流程如左图所示。

2.2 中断执行条件

- 中断进入的条件：下面 3 个条件缺一不可
  1) 程序中有 EI 指令，且会被梯形图扫描到；
  2) 中断禁止标志复位；
  3) 中断申请位置位，进入中断；
- 中断不进入的条件：下面 3 个条件任意 1 个即可
  1) 程序中无 EI，或者在 EI 和DI 之间；
  2) 中断禁止标志置位；
  3) 无中断申请位置位（OFF）。
- 高级中断进程的进入条件：
  1) 满足中断进入的 3 个条件；
  2) 中断申请字置位（中断激活位变成 ON）；
  3) 在中断激活位为 ON 时，进程申请字置位一次，进程进入一次。
- 高级中断进程的退出条件：
  1) PEND 前面触点导通；
  2) 复位中断激活位（退出中断）。

2.2.1 中断助记符

1、助记符类别

软中断、定时器中断和外部输入中断的指令主要有以下三条。

助记符、名称	功能	回路表示	程序步
EI 开中断	开始中断申请扫描	EI 操作元：无	1
IRET 中断返回	中断结束返回主程序	IRET 操作元：无	1
DI 中断禁止	禁止中断申请扫描（禁止中断）	DI 操作元：无	1

高级中断和中断进程的指令主要有以下三条。

助记符、名称	功能	回路表示	程序步
PROC 进程开始	进程中断入口	PROC 操作元：无	1
PEND 进程结束	结束当前的进程	PEND 操作元：无	1
PRET 高级中断返回	整个高级中断结束返回主程序	PRET 操作元：无	1

2.2.2 指令用法说明

- 以上指令除了进程结束指令其它都直接与母线连接，中间不能添加其他任何指令；
- EI 和DI 指令的作用相反，为开/关中断检测的指令，EI 和DI 只需要使用一次（当不需要禁止中断时， 不用DI 指令）。
- 中断子程序的返回指令：IRET 用于普通中断，PRET 用于高级中断；
- 中断进程的开始指令：PROC 进程中断的入口；
- 高级中断进程的退出和返回：PEND 处结束当前进程；
- 每一个I 标号都必须有一个返回指令进行匹配，否则申请进入中断时将出错。

标记指针 I 用于中断程序标号，其对应的中断类型有定时中断，外部输入中断，软中断及高级中断。中断 I 插入的动作须搭配 EI （中断插入允许）、DI （中断插入禁止）、IRET/PRET （中断插入返回）等指令组合而成。当 EI 指令开启中断插入之后，PLC 每执行完成一条指令时，将自动检查是否有中断发生，若是“已发生”，则 PLC 将暂停主程序的指令执行动作，并执行中断服务程序内指令；若是“未发生”，则继续执行主程序指令。

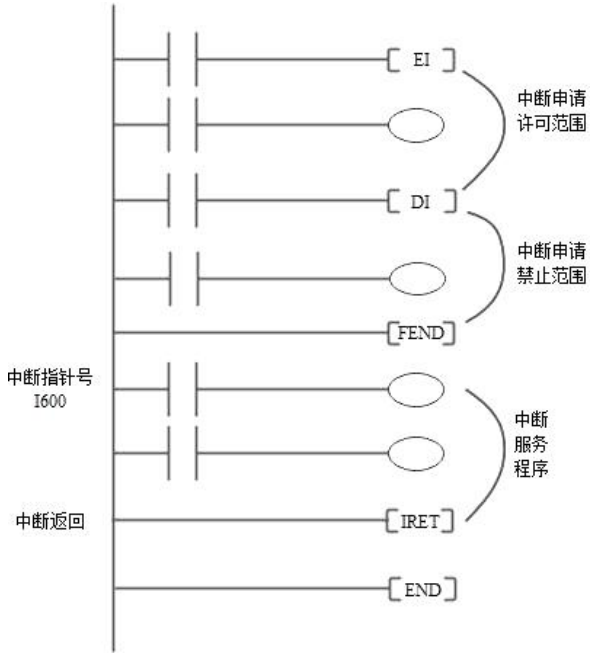
2.2.3 进程语句 PROC 和 PEND

- PROC 指令表示从此处开始准许进程申请位的扫描；
- PEND 表示当前进程结束（结束内容为PEND 语句之前所有的内容），但并不表示中断内容结束；
- PROC 和 PEND 建议成对使用，但不是必须的，可以只有 PROC，也可以只有 PEND；
- 高级中断中 I 和PRET 之间出现 PROC 和 PEND 有三种情况：只有 PROC、只有PEND、两者都有。

2.3 中断格式

2.3.1 中断在程序中的位置

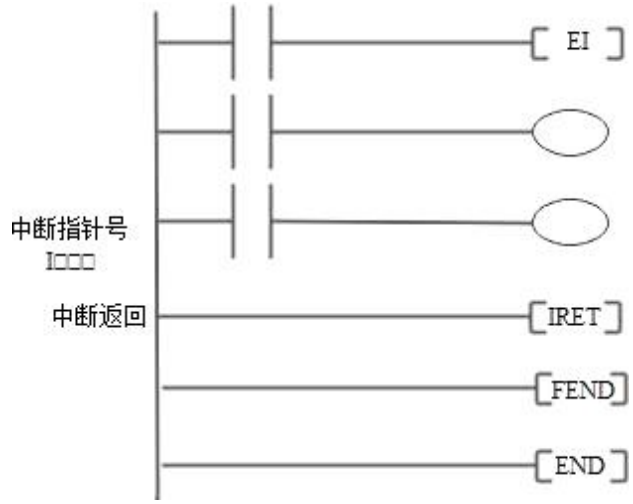
- 中断子程序在程序中的位置：在 FEND 之后和 END 之前。
- 中断子程序在程序中的格式：I 和IRET（PRET）之间写中断服务程序。



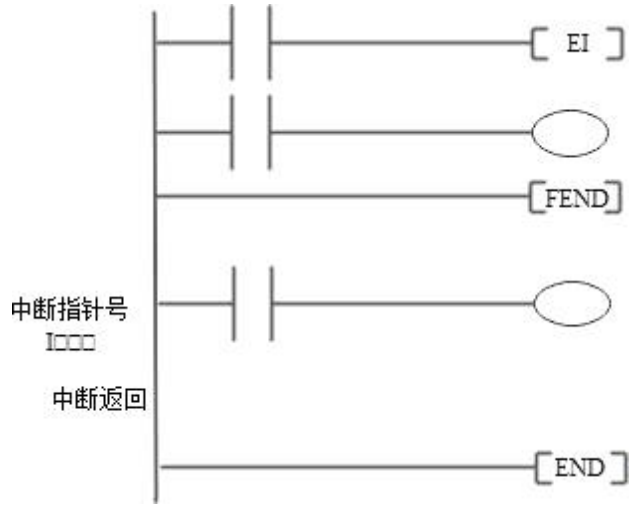
- 左侧梯形图中，EI 为开中断，即在此条指令后，系统每扫描完一条指令都会扫描一次中断申请。
- DI 为禁止中断，即在 DI 到 FEND 指令之间区间，禁止接收中断信号，如果中断信号在梯形图运行到此区间内到来，则会延迟到此区间运行结束后进入中断子程序。如果 DI 到 FEND 指令之间的程序运行时间较长，则中断响应有较明显的延迟。
- FEND 表示主程序结束，中断子程序只能在 FEND 指令之后。
- IRET 指令表示中断返回，如果是高级中断中断返回用 PRET。
- 程序结束。

2.3.2 中断的书写异常

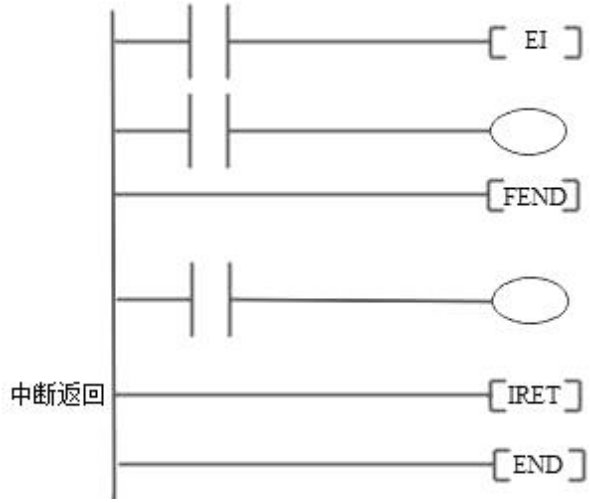
- 中断子程序写在 FEND 之前，梯形图报错。



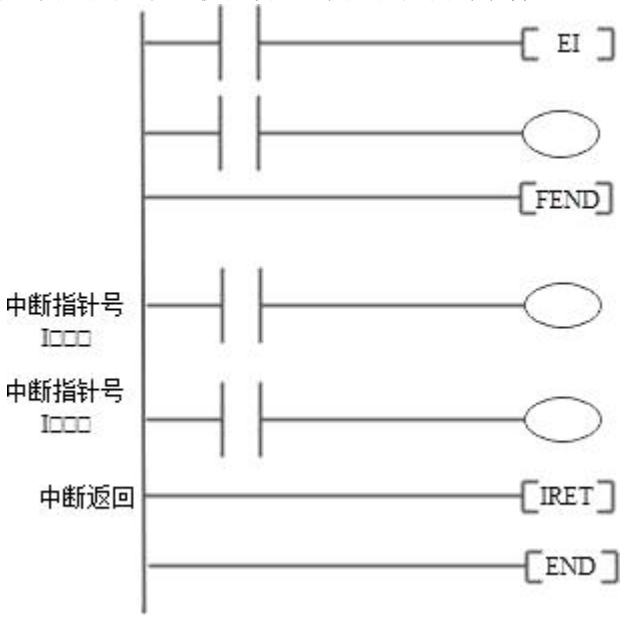
- 中断无返回指令（IRET 或者 PRET），中断可以进入一次，此后不会再进入，如果中断服务程序后面紧跟 END 则主程序程序的执行不受影响，否则主程序执行受到影响。



- 没有中断指针号，复位相关中断禁止位，允许扫描到（有EI）中断申请位置位时，控制器报错。



- 同一个指针号重复，执行最后一个指针号的中断任务。



2.4 中断优先级

- 不同类型的中断优先级：高级中断>（外部中断、定时中断、软中断），外部中断、定时中断和软中断三者不分优先级，三者按照中断申请位地址排序，中断申请位地址号越小，系统越早扫描到，可以相互抢占嵌套。
- 同一种类型的中断触发条件同时来时：进入的顺序为中断指针号小的先进入。

2.5 中断嵌套

2.5.1 嵌套概念

中断嵌套，当一个中断正在执行的时候，如果事先设置了中断优先级，那么当一个更高优先级的中断到来的时候会发生中断嵌套，如果没有设置则不会发生任何嵌套。

比如在执行一个外部设备的中断服务程序期间允许中断，同时已经建立了一种优先链结构，那么优先级更高的设备就能中断现行的服务程序。这种嵌套的结构将使高速设备能进入中断，而暂时挂起较低速设备的中断服务。在完成了较高速设备的服务后，就自动返回去完成较低速设备的服务程序。

2.5.2 嵌套的条件

MC 系列PLC可以实现相互抢占和嵌套的中断：

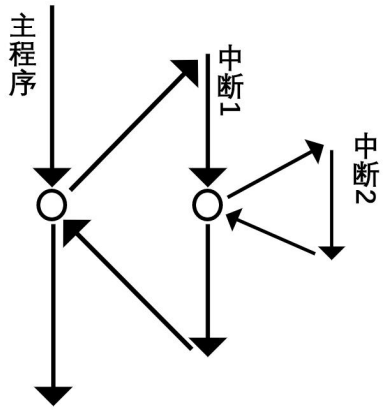
- 外部中断
- 定时中断
- 软中断

MC 系列PLC可以抢占其它中断，但是不能够被抢占和嵌套的中断：

- 高级中断 I900-I907

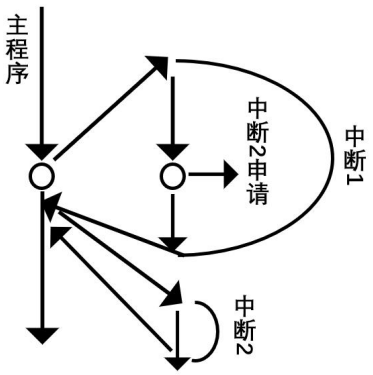
2.5.3 嵌套执行流程

MC 系列PLC最多可以支持 8 层中断嵌套，中断嵌套方式如下图。中断 2 的优先级不低于中断 1，即中断 2 可以抢占中断 1。



- 左图即为中断嵌套的流程。
- 主程序中每扫描完一条指令都会扫描一次所有的中断申请。若中断源满足触发条件且申请有效，则执行相应的中断服务程序。
- 当在主程序中某一时刻扫描到中断 1 的申请后，暂停此时的主程序， 从而进入中断 1，执行中断 1 的服务程序。
- 在中断 1 服务程序执行过程中，系统扫描到中断 2 的申请，且有效， 如果中断 2 的优先级不低于中断 1，暂停中断 1。
- 中断 2 执行完后，返回到中断 1 中执行完中断 1 剩余的服务程序，然后返回到主程序暂停的地方，继续执行主程序程序。

中断 2 的优先级低于中断 1，即中断 2 不能够抢占中断 1。



- 左图即为中断不能够嵌套的流程。
- 主程序中每扫描完一条指令都会扫描一次所有的中断申请。若中断源满足触发条件且申请有效，则执行相应的中断服务程序。
- 当在主程序中某一时刻扫描到中断 1 的申请后，暂停此时的主程序， 从而进入中断 1，执行中断 1 的服务程序。
- 在中断 1 服务程序执行过程中，系统扫描到中断 2 的申请，且有效， 但是中断 2 的优先级低于中断 1，中断 1 继续执行完，然后返回到主程序断点处，然后再进入中断 2。
- 中断 2 执行完后，返回到主程序暂停的地方，继续执行主程序程序。

2.5.4 嵌套注意事项

- 当主中断任务小于 0.1ms 时，主中断可以被抢占，但是在抢占中断执行之前主中断的任务可能已经执行完。
- 中断不可以执行 9 层嵌套，因为第 9 层的中断在第 8 层执行完后才执行，即第 9 层抢占不了第 8 层。
- 高级中断不影响低级中断的嵌套层数。
- 高级中断之间不可以嵌套。
- 高级中断可以抢占低级中断。
- 被抢占的中断的被抢占时间等于抢占中断任务的时间之和。

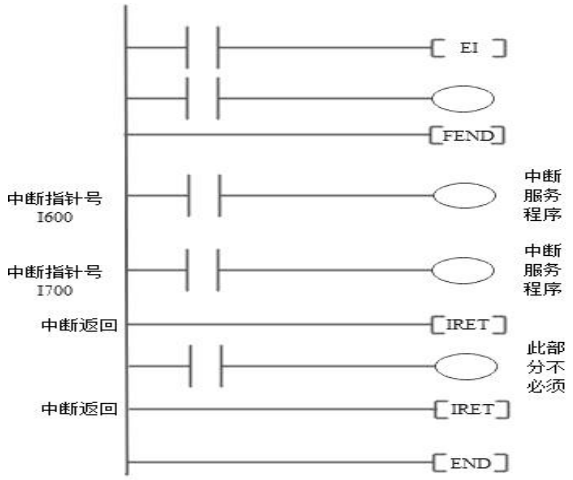
2.6 多重中断

2.6.1 多重中断概念

多重中断是在一个中断的入口和返回之间（I-IRET 或I-PRET）包含另一个或多个其它的中断的入口和返回（I-IRET 或I-PRET）。多重中断语法上不会出错，但中断执行的时候需要注意，中断指针号靠前的中断服务程序会包含中断指针号靠后的中断服务程序。

2.6.2 多重中断实现方式

多重中断实现方式如下示意图。



- 左图即为多重中断流程。
- 在程序中有两个中断服务程序，书写格式如左图所示。
- 此种语法格式不会导致程序错误，并且程序和中断服务程序也能够正常执行。
- 中断指针号书写靠前的中断服务程序会包含中断指针号书写靠后的中断服务程序。
- 中断返回从第一个 IRET 返回到主程序。
- 两个 IRET 之间的语句不会执行。

2.6.3 多重中断注意事项

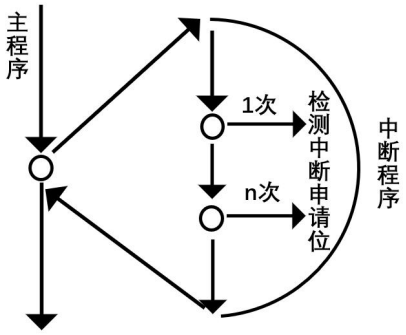
- 不建议使用多重中断。
- 多重中断语法上和程序执行上不会出错。
- 中断执行的时候，中断指针号靠前的中断服务程序会包含中断指针号靠后的中断服务程序。

2.7 中断溢出

2.7.1 中断溢出的概念

中断溢出：中断任务时间大于中断申请的间隔时间，即在中断服务程序执行过程中，中断申请位再次被检测到1 次或者多次。

2.7.2 中断溢出的表达的方式



- 左图即为中断溢出的表现方式。
- 在中断服务程序执行过程中检测到的n 次中断申请位都不会导致中断任务停止执行。
- 中断执行完后，返回到主程序暂停的地方，继续执行主程序。
- 如果中断执行完返回到主程序中，又检测到中断申请位置位，则再次进入中断。

2.7.3 中断溢出注意事项

- 中断处理中断服务过程中中断申请位再次检测到会记录下来，但不会导致中断任务停止执行， 等中断任务处理完，返回到主程序，然后主程序中再次检测到中断申请位后，再进入一次。
- 当中断任务或者进程任务时间过长时，会严重影响到主程序的扫描周期，尤其是在中断处理过程中再次检测到（中断重复进入）中断申请位时，会使中断接近连续进入状态，极端情况有可能一直在执行中断服务程序，因此中断服务程序不能够过长，并且要避免中断的重复进入。

3、中断详解

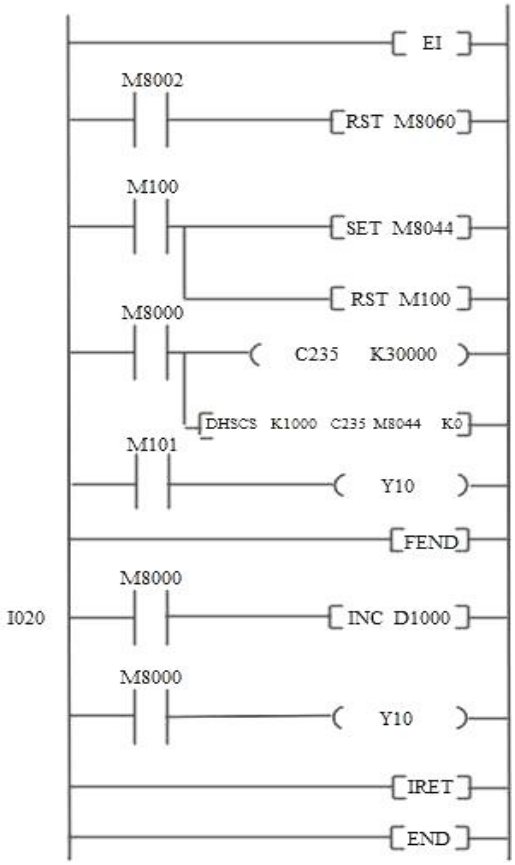
3.1 软中断

3.1.1 软中断概述

系统软中断是以中断申请标志位作为中断源的中断，主要适用于配合高速计数置位指令 DHSCS 使用， 同时也可以通过其它条件控制中断申请位进入中断。MC系列PLC提供 5 个系统软中断信号。

中断标号	中断申请标志位	中断禁止标志位
I010	M8043（系统1ms 控制）	M8059
I020	M8044	M8060
I030	M8045	M8061
I040	M8046	M8062
I050	M8047（DHSCT输出可触发）	M8063

3.1.2 软中断示例



- 左图即为软中断 I020 的进入和执行程序。
- EI 允许检测中断申请。
- 复位 I020 中断禁止位，允许进入中断。
- 进入 I020 方式一：置位 M100 后，置位 I020 中断申请位申请进入 I020 中断服务程序，然后复位 M100，不要一直置位中断申请位，仅在需要的时候进入即可。
- 进入 I020 方式二：当启用 X0 高数计数值 C235 从小于比较置位脉冲数值变成大于等于比较置位脉冲数值时或者从大于等于比较置位脉冲数值变为小于比较置位脉冲数值，置位 M8044 一次，进入 I020 中断一次，进入中断后 M8044 自动复位。
- 主程序中 M101 控制 Y10 的输出，M101 置位 Y10 输出为 ON。
- 主程序结束。
- 中断服务程序，每进入中断一次，D1000 增加 1
- 中断执行完，退出中断后 Y10 状态保持出中断时的状态，然后再同返回到主程序中断进入时刻点后 Y10 的状态（M101 控制）进行“与运算”输出。如果主程序 M101 为 OFF 则最终 Y10 不会进行输出。
- 中断程序返回。
- 主程序结束。

3.1.3 注意事项

- I010 为系统 1ms 定时软中断，无需用户设置中断申请位（M8043 由系统 1ms 自动申请）。
- I050 为DHSCT高速表格比较输出关联软中断，当梯形图程序中存在DHSCT指令，每次满足输出条件输出时就会自动触发一次I050（须I050中断允许状态），同时捕获高速计数器当前计数值，虚拟输入轴0的当前高速计数实时值传送到R114（L），虚拟输入轴1的当前高速计数实时值传送到R164（L）虚拟输入轴3的当前高速计数实时值传送到R214（L），虚拟输入轴3的当前高速计数实时值传送到R264（L）。
- I010 到I050 中断禁止位上电系统默认为 ON（中断处于禁止状态）。
- 通过梯形图相关元件（M 或者 X）操作中断申请位进入软中断时，中断进入的时刻可能会受到程序扫描周期的影响。
- 通过 DHSCS 指令置位中断申请位，可以保证中断的进入不受主程序扫描周期的影响，但是需要注意的是，高速计数值当前值与设定的比较值进行比较时，由**当前值<比较值**变为**≥比较值**，或由**当前值≥比较值**变为**<比较值**时，都会置位指定的位元件，这两种状态变化都会进入中断。

3.2 定时中断

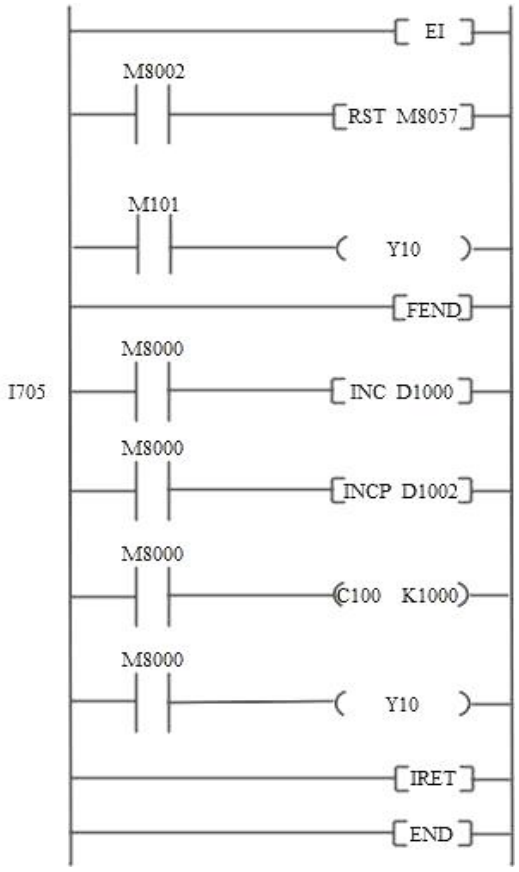
3.2.1 定时中断概述

定时中断即是以定时器的定时时间作为中断源的中断，适用于需要以固定的时间（有别于程序扫描周期） 执行某些任务的场合。MC系列PLC提供三个定时中断，中断标号以及对应的控制字如下表：

输入编号	中断周期(ms)	中断申请标志位	中断禁止标志位
I6□□	在□□中输入 00-99 的整数，表示定时周期。如	M8040（系统控制）	M8056
I7□□	I650，则表示每隔 50ms 执行一次定时器中断，	M8041（系统控制）	M8057
I8□□	I600，则表示每隔 100ms 执行一次定时器中断。	M8042（系统控制）	M8058

3.2.2 定时中断示例

以 I705 中断为例详解定时中断的使用方式。



- 左图即为定时中断 705 的进入和执行程序。
- EI 允许检测中断申请。
- 复位 I705 中断禁止位，按照指针标号对应的时间间隔进入中断。
- 主程序中 M101 控制 Y10 的输出，M101 置位 Y10 输出为 ON。
- 主程序结束。
- 中断服务程序，每进入中断一次，D1000 增加 1
- 中断第一次进入时 D1002 增加1。
- 中断第一次进入时 C100 计数一次。
- 中断执行完，退出中断后 Y10 状态保持出中断时的状态，然后再同返回到主程序中断进入时刻点后 Y10 的状态（M101 控制）进行“与运算” 后输出。如果主程序 M101 为 OFF 则最终 Y10 不会进行输出。
- 中断程序返回。
- 主程序结束。

3.2.3 注意事项

- 三个定时中断指针号（I6/I7/I8）同一个指针号的指针不可重复使用。如果重复使用，系统只识别FEND 和 END 之间最后一个中断编号。
- 系统默认为禁止中断状态，即 M8056～M8058 默认处于置位状态，如果需要使用某个定时中断，需要在程序中将对应的中断禁止控制位进行复位。
- 定时中断最大定时间为 100ms，中断标号后两位为0（如I700），则中断定时时间为 100ms。
- 定时中断的中断申请位不能通过梯形图控制，系统按照用户设定的时间自动置位申请，比如用户使用了 I602 中断指针号，那么系统每间隔 2ms 置位 I6 指针号对应的中断申请位 M8040 一次。

- 定时中断的中断任务时间应小于定时时间。
- 定时中断的中断任务时间等于定时时间时，系统不能执行其他任务。
- 如果大于定时时间（中断溢出），程序不会出错，但是中断不能够按照设定的定时时间进入中断，此时定时中断进入的时间间隔为：（（中断任务时间）÷（理论中断进入时间间隔））（有小数点时取整后加 1），然后再乘以（理论中断进入时间），比如定时中断为 3ms 进入一次，中断任务时间为 7ms，那么中断前后两次进入的时间间隔为 9ms（7÷3=2.333，结果 2.333 取整为2，然后再加 1 等于 3，最后 3 乘以定时时间 3ms 等于 9ms）。

3.3 外部输入中断

3.3.1 外部输入中断概述

外部输入中断是以MC系列PLC的特定的外部输入信号作为中断源的中断，中断源不受扫描周期的影响， 适用于当外部信号到来时需要执行某些任务的场合，通过外部中断可以处理比扫描周期更短的信号，因此可以在短时脉冲处理控制中使用，同时MC系列PLC可以直接获取外部中断来时脉冲输入轴的当前脉冲数，当脉冲输入轴接入外部脉冲信号时，通过外部中断可以获得外部中断信号来时的外部位置信息， 这种方式可以应用在测长（宽）度，尤其是高速测很小长（宽）度，比如色标信号的宽度等，同时也可以应用在色标捕获功能上。

MC系列PLC提供 4 个外部输入事件输入源(X010～X013)，其对应端口动作特性和中断标号的关联如下所示：

外部输入信号	中断标号	中断特性	中断申请标志位	中断禁止标志位
X010	I000	下降沿	M8032	M8048
	I001	上升沿	M8033	M8049
X011	I100	下降沿	M8034	M8050
	I101	上升沿	M8035	M8051
X012	I200	下降沿	M8036	M8052
	I201	上升沿	M8037	M8053
X013	I300	下降沿	M8038	M8054
	I301	上升沿	M8039	M8055

3.3.2 外部输入中断进入方式

外部输入中断进入的方式可以分为两种：

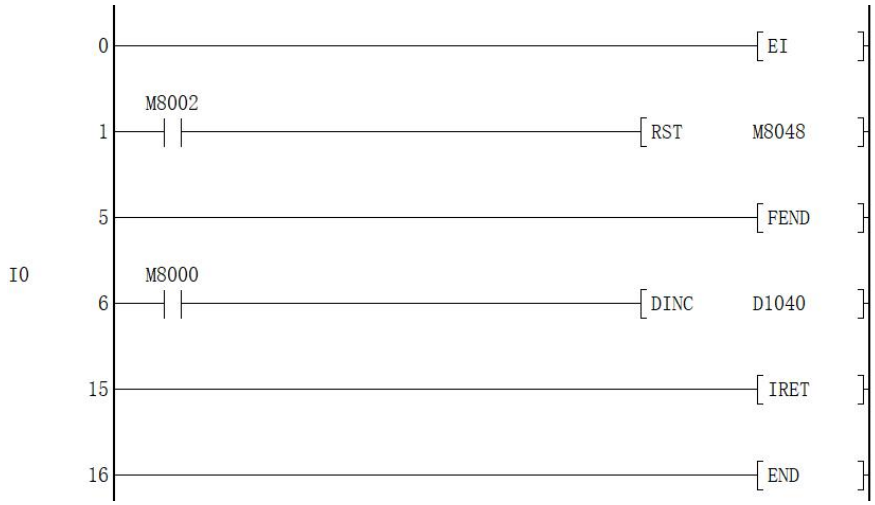
(1) 通过中断申请位进入。

(2) 通过外部硬件端子触发。

当中断禁止标志位为OFF时，直接置位中断申请标志位，可进入对应中断子程序并复位中断申请标志位。当外部输入中断使用PLC硬件端子触发进入时，会置位对应的中断申请标志位。当中断禁止标志位为ON时，不会进入中断子程序。当中断禁止标志位为OFF时，会进入对应中断子程序并复位中断申请标志位。

例如：M8048为OFF时，直接置位 M8032，会进入I000 (X10下降沿中断子程序)并自动复位M8032。

或X10下降沿信号到来时会置位M8032，进入I000 (X10下降沿中断子程序)后自动复位M8032。



MC系列PLC允许I000/I001/I100/I101/I200/I201/I300/I301同时使用，外部中断最高连续响应速度为2000Hz。

3.3.3 外部输入中断捕获的轴脉冲

当外部事件输入需要捕获外部脉冲计数值时，其相关的硬件资源和软件资源如下：

- X0、X1端子对应的高速计数器（5个）：C235、C239、C243、C247、C251；
- X0、X1端子对应虚拟输入轴0；
- 虚拟输入轴0参数地址为R100起始的50个R寄存器，其中R111为“脉冲捕获方式”寄存器，R106和R108为“外部事件捕获绝对计数”。

脉冲捕获方式寄存器	脉冲捕获方式	外部事件特性	捕捉高速计数值
R111	HX1	X010下降沿	R108
	HX0	X010上升沿	R108
	H1X	X011下降沿	R106
	H0X	X011上升沿	R106

R106（L）记录该虚拟输入轴在X11产生外部事件时的脉冲计数值。

R108（L）记录该虚拟输入轴在X10产生外部事件时的脉冲计数值。

- X2、X3端子对应的高速计数器（5个）：C236、C240、C244、C248、C252；
- X2、X3端子对应虚拟输入轴1；
- 虚拟输入轴1参数地址为R150起始的50个R寄存器，其中R161为“脉冲捕获方式”寄存器，R156和R158为“外部事件捕获绝对计数”。

脉冲捕获方式寄存器	脉冲捕获方式	外部事件特性	捕捉高速计数值
1261	HX1	X012下降沿	R158
	HX0	X012上升沿	R158
	H1X	X013下降沿	R156
	H0X	X013上升沿	R156

R156（L）记录该虚拟输入轴在X13产生外部事件时的脉冲计数值。

R158（L）记录该虚拟输入轴在X12产生外部事件时的脉冲计数值。

- X4、X5端子对应的高速计数器（5个）：C237、C241、C245、C249、C253；
- X4、X5端子对应虚拟输入轴3；
- 虚拟输入轴3参数地址为R200起始的50个R寄存器，其中R211为“脉冲捕获方式”寄存器，R206和R208为“外部事件捕获绝对计数”。

脉冲捕获方式寄存器	脉冲捕获方式	外部事件特性	捕捉高速计数值
R211	HX1	X014下降沿	R208
	HX0	X014上升沿	R208
	H1X	X015下降沿	R206
	H0X	X015上升沿	R206

R206（L）记录该虚拟输入轴在X15产生外部事件时的脉冲计数值。

R208（L）记录该虚拟输入轴在X14产生外部事件时的脉冲计数值。

- X6、X7端子对应的高速计数器（5个）：C238、C242、C246、C250、C254；
- X6、X7端子对应虚拟输入轴3；

- 虚拟输入轴3参数地址为R250起始的50个R寄存器，其中R261为“脉冲捕获方式”寄存器，R256和R258为“外部事件捕获绝对计数”。

脉冲捕获方式寄存器	脉冲捕获方式	外部事件特性	捕捉高速计数值
R261	HX1	X016下降沿	R258
	HX0	X016上升沿	R258
	H1X	X017下降沿	R256
	H0X	X017上升沿	R256

R156（L）记录该虚拟输入轴在X17产生外部事件时的脉冲计数值。

R158（L）记录该虚拟输入轴在X16产生外部事件时的脉冲计数值。

3.3.4 外部输入中断捕获特性

当外部输入中断使用PLC硬件端子触发进入时，外部输入中断的中断捕获特性可以通过运动输入轴参数的“脉冲捕获方式”寄存器进行配置，其中R111控制X10/X11的外部事件捕获方式，R111的第0位控制X10，第4位控制X11；其中R161控制X6/X7的外部事件捕获方式，R1261的第0位控制X12，第4位控制X13；其中R211控制X14/X15的外部事件捕获方式，R211的第0位控制X14，第4位控制X15，其中R261控制X14/X17的外部事件捕获方式，R261的第0位控制X16，第4位控制X17。当触发外部事件捕捉功能时，对应的标志位会置位。详细配置如下表：

脉冲捕获方式寄存器	脉冲捕获方式	外部事件特性	外部捕获标志位
R111	HX1	X010下降沿	M8168
	HX0	X010上升沿	M8169
	H1X	X011下降沿	M8170
	H0X	X011上升沿	M8171
R161	HX1	X012下降沿	M8172
	HX0	X012上升沿	M8173
	H1X	X013下降沿	M8174
	H0X	X013上升沿	M8175
R211	HX1	X014下降沿	M8176
	HX0	X014上升沿	M8177
	H1X	X015下降沿	M8178
	H0X	X015上升沿	M8178
R261	HX1	X016下降沿	M880
	HX0	X016上升沿	M8181
	H1X	X017下降沿	M8182
	H0X	X017上升沿	M8183

更改外部事件捕捉特性时需RST对应的高速计数器，具体参所如下：

修改R111的值后，需使用RST复位高速计数器（C235、C239、C243、C247、C251）当中任意一个或者多个。

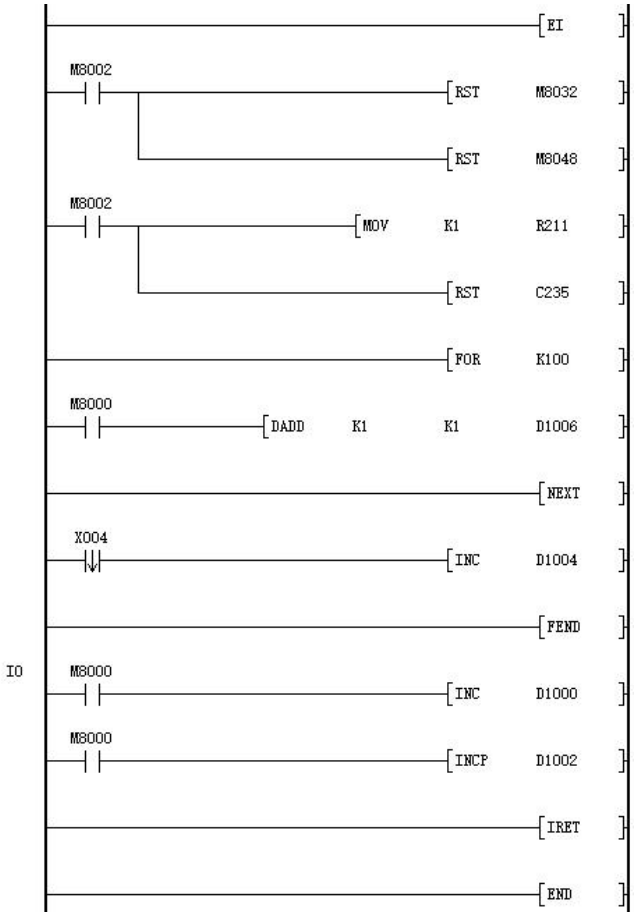
修改R161的值后，需使用RST复位高速计数器（C236、C240、C244、C248、C252）当中任意一个或者多个。

修改R211的值后，需使用RST复位高速计数器（C237、C241、C245、C249、C253）当中任意一个或者多个。

修改R261的值后，需使用RST复位高速计数器（C238、C242、C246、C250、C254）当中任意一个或者多个。

3.3.5 外部输入中断示例

外部输入中断的使用方式和示例。



- EI 允许检测中断申请。
- 复位 I0 中断申请位，防止外部信号变化造成的中断申请位置位。
- 复位 I0 中断禁止位，允许 X10 下降沿来时进入 I0 中断。
- 给R111赋值，设置为X10下降沿中断。
- 复位本路高速计数器C235。
- 主程序中 FOR 循环控制增加整个程序的扫描周期。
- 当程序扫描周期大于 X10 边沿信号变化时间时，在主程序中检测 X10 的边沿信号可能会丢失部分信号，因为主程序 X10 的状态检测会受到程序扫描周期的影响。如果使用 X10 的中断，则可以避免这个问题。
- 主程序中 X10 下降沿计数。
- 主程序结束。
- 中断服务程序，X10 下降沿触发每进入中断一次，D1000 增加 1
- 中断第一次进入时 D1002 增加1。
- 中断程序返回。

3.3.6 注意事项

- 通过中断申请位可以控制 X10/X11/X12/X13的中断进入，此时中断进入方式（上升沿或者下降沿） 不受对应的捕获特性寄存器R111/R161/R211/R261数值的影响。
- 当使用中断申请位调用外部输入中断服务程序时，如果不希望连续进入中断服务程序，不要使用常开或常闭触点去置位中断申请标志位。可以使用 LDP，LDF，PLS 等边沿指令实现单次置位中断申请标志位。
- 在复位中断禁止标志位之前，先清除相应的中断申请标志位，以免进入一个非预期的外部中断服务程序。比如 X010有下降沿动作，M8032会被置位。那么一旦复位M8048，就会立即进入I000 中断。
- 如果复位了对应中断的禁止位，但是程序中没有中断指针号，当中断申请位申请时，PLC报错。

3.4 高级中断

3.4.1 高级中断概述

在运动系统的控制中，一般需要更优化的中断方案。对此，MC系列PLC采用高级中断模式满足高速并发任务的处理。高级中断处理具备普通中断的特性外还有自身的特点，比如支持进程调度语句。

在高级中断中完成一个运动过程需要进行多次插补运算，而插补脉冲输出需要一定的时间，这就可以让系统不断以进程中断的方式进出高级中断，来完成某一运动过程。如果需要完成的运动控制是由多个运动过程组成的，那么可以在高级中断中定义多个进程中断。使用进程调度语句调配这多个进程中断，进程中断的触发信号可以由位置插补周期定时触发（I900 特有），也可以通过其它方式触发（I901-I907）。

3.4.2 高级中断控制位

MC 系列PLC有 8 个高级中断，每个高级中断对应四个控制位：中断禁止位、中断申请位、中断激活位、进程申请位。

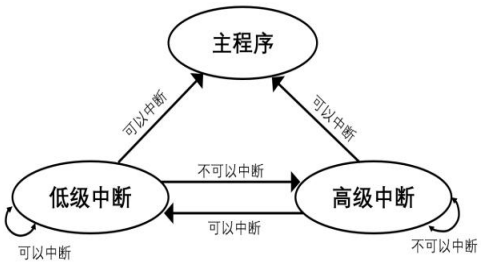
中断标号	中断禁止标志	中断激活标志	中断申请字	进程中断申请字
	ON（禁止中断） OFF（使能中断）	ON（进入中断） OFF（未进入中断）	ON（申请应答） OFF（申请不应答）	ON（申请应答） OFF（申请不应答）
I900	M8064	M8072	M8080	M8088
I901	M8065	M8073	M8081	M8089
I902	M8066	M8074	M8082	M8090
I903	M8067	M8075	M8083	M8091
I904	M8068	M8076	M8084	M8092
I905	M8069	M8077	M8085	M8093
I906	M8070	M8078	M8086	M8094
I907	M8071	M8079	M8087	M8095

高级中断的执行过程分两种情况：

- 没有进程调度语句时，其执行过程和低级中断的执行过程一样，每触发一次执行一次。
- 有进程调度语句，碰到 PROC 进程调度语句时，进程中断每触发一次，执行一次。多次入出本次高级中断(进程中断执行时，需多次重复执行，如插补运算)直至 PEND 语句得电，执行下一条梯行图语句。最终碰到 PRET 高级中断返回。

3.4.3 高级中断同其它中断的关系

主程序、低级中断和高级中断之间的关系如下图所示，主程序可以被低级中断和高级中断中断，高级中断可以中断低级中断，而低级中断不能中断高级中断。

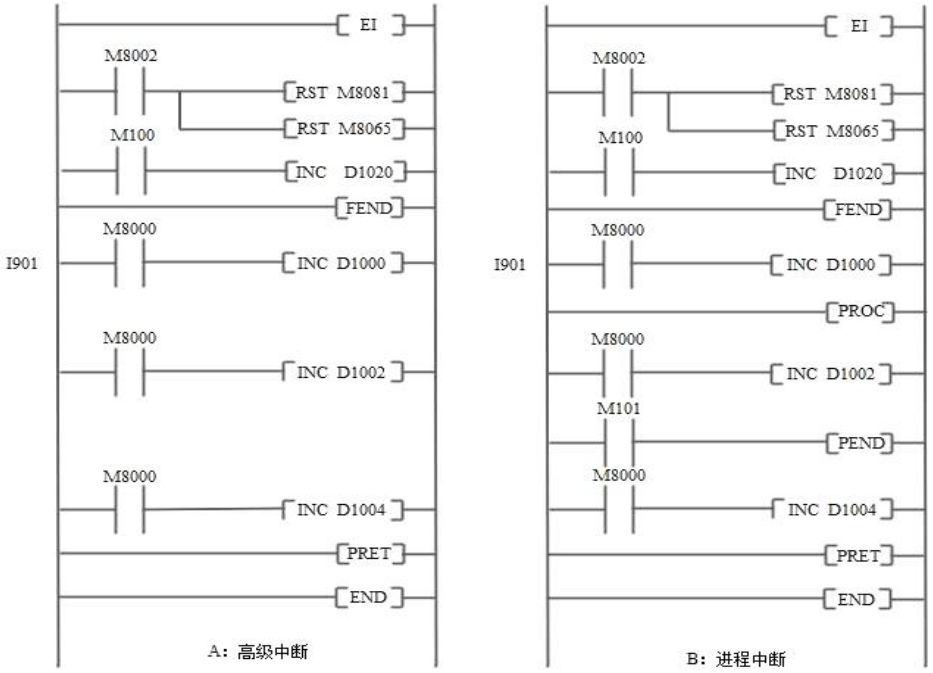


- 低级中断和高级中断都可以中断主程序，转而执行中断程序，待中断程序执行完后，再返回继续执行主程序。
- 当低级中断正在执行时，高级中断可以中断低级中断，转而执行高级中断，待高级中断执行完后，再返回继续执行低级中断，最后再返回执行主程序。
- 低级中断可以相互中断。
- 高级中断执行时，低级中断不能中断高级中断。
- 高级中断之间不能够相互中断。
- 高级中断中断低级中断不占用低级中断的嵌套层数。

3.4.4 高级中断和进程的书写格式

以I901 为例说明高级中断任务和进程中断区别。

- 如下图 A 中：表示的是高级中断书写格式，中断进入后执行完中断任务从 PRET 返回到主程序，并且退出中断（中断激活位复位）；
- 如下图 B 中：表示的是进程中断书写格式，进程中断进入后执行完进程任务从 PEND 返回到主程序， 当所有的进程执行完后从 PRET 返回主程序，并且退出中断（中断激活位复位）；



图文示意：

- 上面 A 高级中断 I901 的执行过程为：

(1) 复位中断禁止位 M8065；

(2) 置位中断申请 M8081，程序从主程序中跳到 I901 指针入口处开始执行 D1000 加1，接着执行D1002 加1 和 D1004 加1，然后执行到 PRET 语句返回到主程序中；

(3) 在执行中断任务过程中，中断激活位 M8073=ON，退出中断返回到主程序时中断激活位为 OFF；

(4) A 中无进程语句 PROC 和 PEND，因此中断申请位 M8089 无意义；

(5) （2）到（3）过程中 D1000、D1002 和 D1004 各增加 1；

- 上面 B 进程中断的执行过程为：

(1) 复位中断禁止位 M8065；

(2) 置位中断申请 M8081，程序从主程序中跳到 I901 指针入口处开始执行 D1000 加1，接着执行D1002 加1；

(3) 接（2）（M101=OFF）执行到 PEND 语句返回到主程序中，此时中断激活位M8073=ON 表示中断没有退出，仅仅是从进程出口 PEND 返回到主程序中；

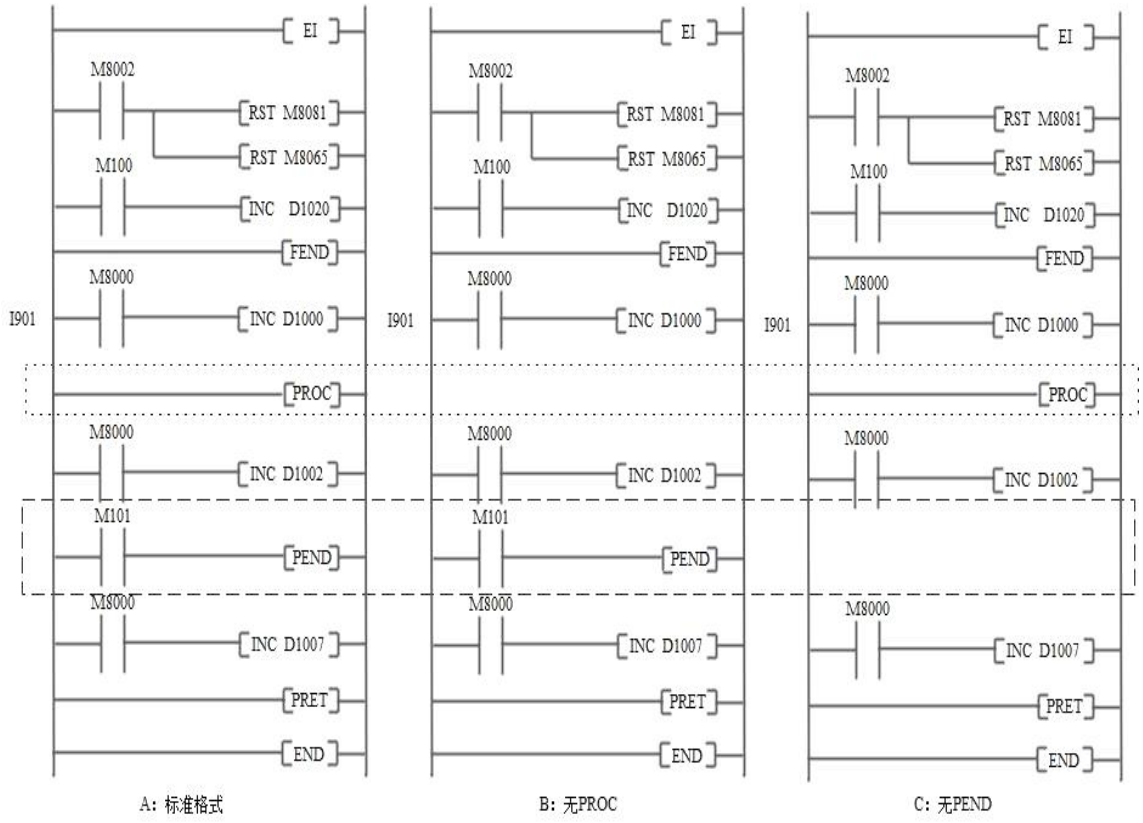
(4) 接（3）（导通 M101=ON）置位进程申请M8089，程序从主程序中跳到 I901 中的 PROC 处开始执行 D1002 加1，I901 到PROC 之间的语句 D1000 加1 并不会执行，此时依旧从进程出口 PEND返回到主程序中，中断激活位依旧 M8073=ON；

(5) 接（4）（导通 M101=ON）置位进程申请M8089，程序从主程序中跳到 I901 中的 PROC 处开始执 行 D1002 加1 和 PEND 后面的 D1004 加1，I901 到 PROC 之间的语句 D1000 加1 并不会执行，最后从中断返回指令 PRET 返回到主程序中，中断激活位变为 M8073=OFF，退出中断；

(6) 到（5）的过程中 D1000 增加了 1，D1002 增加了 2，D1004 增加了 1；

以I901 为例说明进程中断的三种书写方式。

- 如下图 A 中：适用于进程任务区分明确，同一种进程任务需要多次进入执行的情况，是进程的标准写法；
- 如下图 B 中：无PROC 语句，PEND 和I 之间或者 PEND 和上一个 PEND 之间的语句都为当前的进程任务，如果 I 和IRET 之间所有的语句都通过进程来控制，可以省略掉 PROC，仅仅使用 PEND 区分每个进程之间的关系；
- 如下图 C 中：无 PEND 语句，进入中断后无论是否有进程申请位置位，中断中的所有任务执行完后， 从PRET 返回到主程序此种方式同高级中断任务执行方式一样，通过中断申请位来进入中断执行中断任务，此时进程申请位无意义，同时 PROC 语句也无意义；
- 为了程序的可读性，如果使用进程中断时，推荐使用 A 方式书写，不建议 B 和C 的方式。

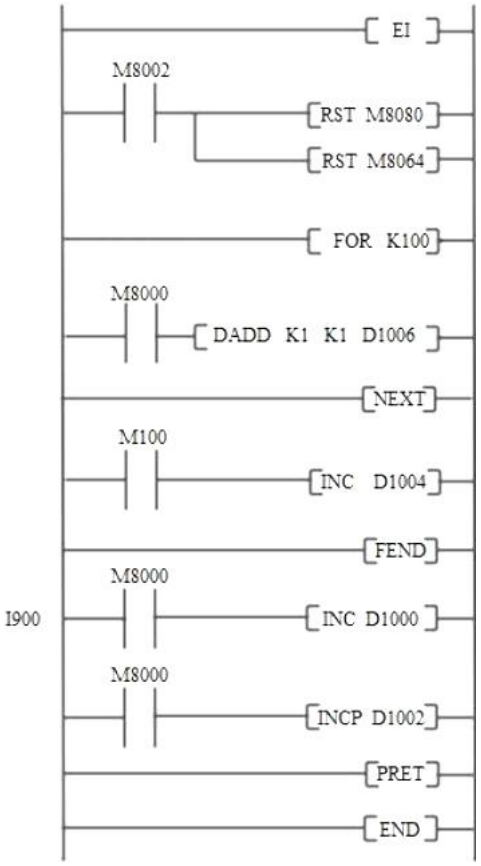


图文示意：

- 上面A 图为高级中断 I901 中一个进程中断的标准书写格式：PROC 和PEND 成对出现；
- 上面B 图为高级中断 I901 中一个进程中断的标准书写格式：无 PROC；
- 上面C 图为高级中断 I901 中一个进程中断的标准书写格式：无 PEND；
- A 和B 图的区别在于申请进入中断后：当 M101=OFF，进程申请位每检测到一次，A 中只有 D1002 会增加一次，而 B 中D1000 和D1002 都会增加一次，原因在于 B 中没有 PROC 引导进程的入口，默认I 和PEND 之间的语句都为进程程序；当 M101=ON，进程申请执行完 D1007 加一后，碰到 PRET 退出中断；
- A 和C 图的区别在于申请进入中断后：进程申请位每检测到一次，A 中只有 D1002 会增加一次，C 中无论有无进程申请位置位，C 中所有程序执行完后退出中断，原因在于 C 中没有PEND 控制进程的结束，此时和高级中断执行方式一样，从 I 指针号开始执行直到碰到 PRET 退出中断；

3.4.5 高级中断示例

高级中断示例，以I900 为例。



左图即为定时中断I900 的进入和执行程序。

EI 允许检测中断申请。

复位 I900 中断申请位，防止在复位中断禁止位前为 ON。

复位 I900 中断禁止位，允许中断申请位置位时进入 I900 中断。

主程序中 FOR 循环程序，仅仅作为主程序内容的一种形式，主程序内容可以多样化。

主程序中内容。

主程序结束。

中断服务程序，M8080每置位一次进入中断

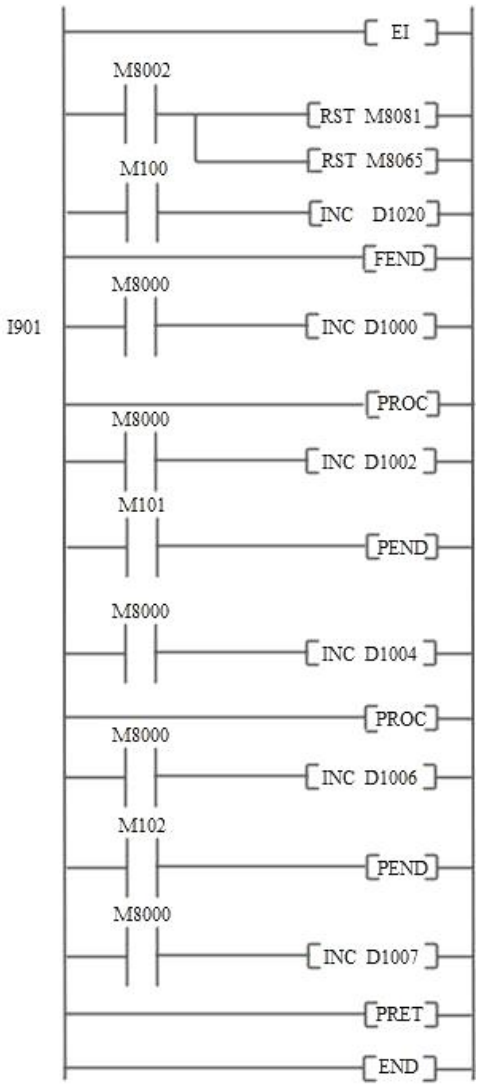
一次，D1000 增加 1。

中断第一次进入时 D1002 增加1。

中断程序返回。

主程序结束。

高级中断中进程中断示例，以 I901 为例。



EI 允许检测中断申请。

复位I901 中断申请位，防止在复位中断禁止位前为 ON。

复位I901 中断禁止位，允许中断申请位置位时进入 I901 中断。

主程序内容，仅仅作为主程序内容的一种形式，主程序内容可以多样化。

主程序结束。

中断任务 1，置位 M8081 进入中断（中断激活位 M8073=0N），D1000增加 1。

中断进程 1 入口。

进程1 任务，中断进入时以及中断进入后（中断激活位 M8073=0N）检测到进程申请位M8089 置位时进程 1 进入1 次，D1002 加1。

进程 1 退出：进程 1 任务执行完后，扫描到 M101 后面的 PEND 退出进程中断1，若M101 为OFF 则返回到主程序，若 M101 为ON 则在下一次进程申请时执行中断任务 2 和进程任务 2。

中断任务 2，进程 1 执行完后，在进入进程 2 时执行 D1004 增加 1。

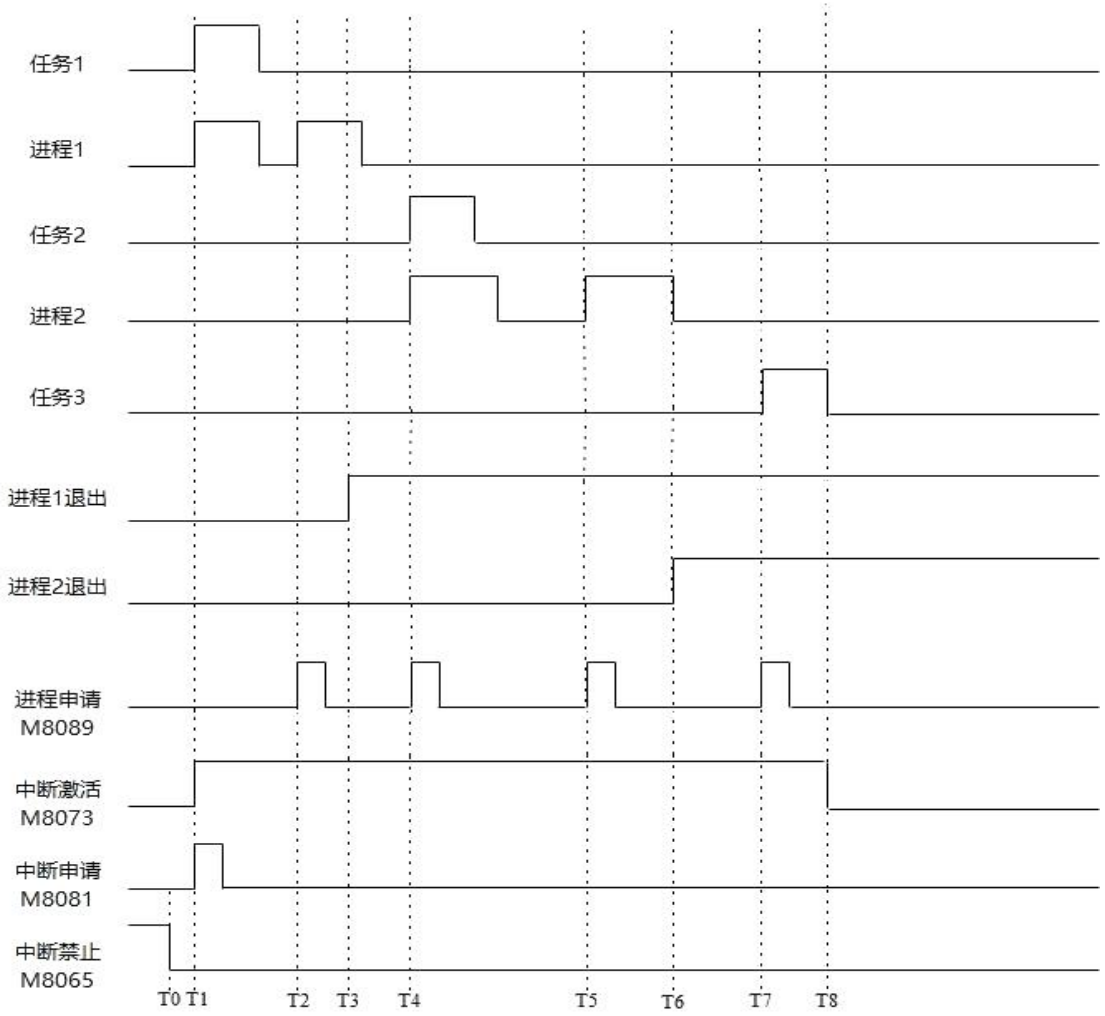
进程2 任务，进程 1 执行完时检测到进程 1 退出条件满足时，在接下来的进程中断申请时进入进程 2 执行D1006 增加 1。

进程 2 退出：进程 2 任务执行完时，若 M102 为OFF 则返回到主程序， 若M102 为ON 则在下一次进程申请时执行中断任务 3。

中断任务 3，进程 2 执行完时检测到进程 2 退出条件满足时，在接下来的进程中断申请时进入中断任务 3 执行 D1007 增加 1，然后检测到PRET 退出中断（中断激活位 M8073=OFF），返回到主程序。

3.4.6 高级中断任务和进程的时序

以 I901 为例讲解下述中断任务和进程任务时序图。



图文示意：

- 任务 1、进程 1、任务 2、进程 2、任务 3 在I901 中断服务程序中的顺序为从 I901 开始从上到下排列到PRET；
- T0 时刻复位中断禁止位M8065，表示可以准许中断进入；
- T1 时刻表示第一次置位中断申请 M8081，此时进入中断 I901，中断激活位 M8073=ON，中断任务 1执行、进程 1 执行一次（此时进程 1 退出条件不满足）后从进程 1 的返回语句 PEND 返回到主程序；
- T2 时刻（M8073=ON），置位进程申请位M8089，跳过任务 1 直接执行进程 1 任务，在 T3 时刻进程1 执行时或者执行完时进程 1 退出条件满足，此次并不会执行任务 2 和进程 2；
- T4 时刻（M8073=ON），置位进程申请位M8089，跳过任务 1 和进程 1，任务 2 进程 2 任务各执行一次（此时进程 1 退出条件满足、进程 2 退出条件不满足）后，从进程 2 的返回语句 PEND 返回到主程序；
- T5 时刻（M8073=ON，此时进程 1 退出条件满足、进程 2 退出条件不满足），置位进程申请位 M8089， 跳过任务 1、进程 1 和任务 2，直接执行进程 2， 在 T6 时刻进程 2 执行完时进程 2 退出条件满足，此次并不会执行任务 3；
- T7 时刻（M8073=ON，此时进程 1 退出条件满足、进程 2 退出条件满足），置位进程申请位 M8089， 跳过任务 1、进程 1、任务 2 和进程 2，直接执行任务 3，任务 3 执行完后T8 时刻退出中断（M8073=OFF）。

- 四个标志位都可以通过手动进行操作：置位和复位。
- 中断申请位 OFF 和中断激活 OFF 的前提下，中断激活位可以手动置位也可以通过中断申请字置位来进行置位，其中中断申请字处于自复位状态（变成 ON 后自动变成 OFF）。
- 在中断激活位为 ON 的前提下，置位中断申请位不会自复位。
- 在没有首次通过中断申请位进入中断时，置位中断激活和进程申请，PLC报错。

3.5 中断的应用和区别

3.5.1 1ms 中断

MC 系列PLC中有 3 种方案实现 1ms 中断：

(1) 3 个定时中断 I601/I701/I801；

(2) 1ms 的 I010 软中断；

(3) I900 的1ms 进程中断。

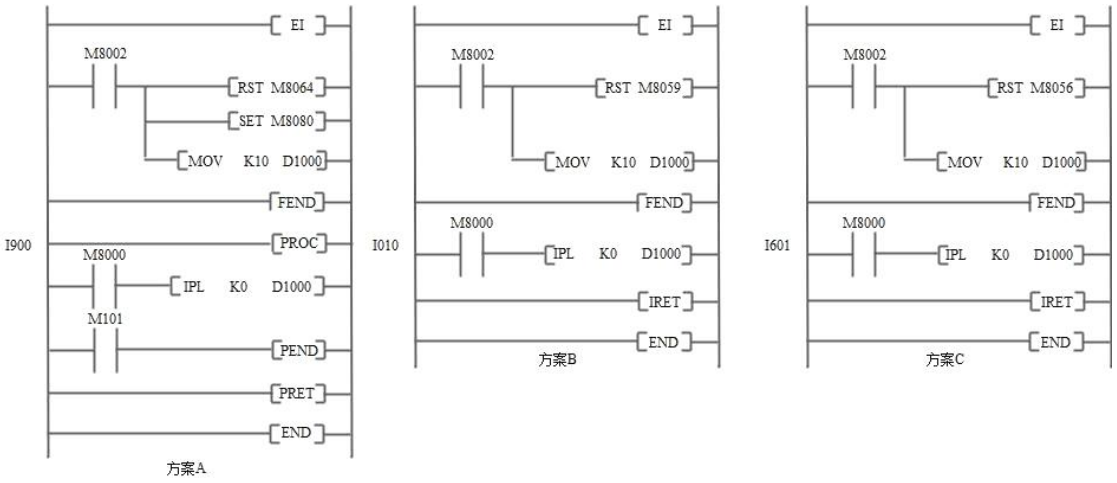
它们之所以可以实现 1ms 中断，在于PLC系统内部直接控制他们的中断进入方式，在梯形图中使用时，只需要复位它们对应的中断禁止位，系统就会自动申请进入它们的中断，其中 I900 的1ms 进程中断需要置位一次中断申请位（M8080）。

当要使用 1ms 中断时，三种 1ms 中断方案的优选顺序为：I900 进程中断、I010 软中断、定时中断。但是梯形图中不建议使用多个 1ms 中断。

3.5.2 1ms 中断用法

MC 系列PLC的 1ms 中断可以用来实现运动的插补运算或输出，关于运动相关的指令和作用参见“MC系列PLC编程手册.pdf”运动章节。

MC 系列PLC通过三种 1ms 中断方案实现插补的示例如下：

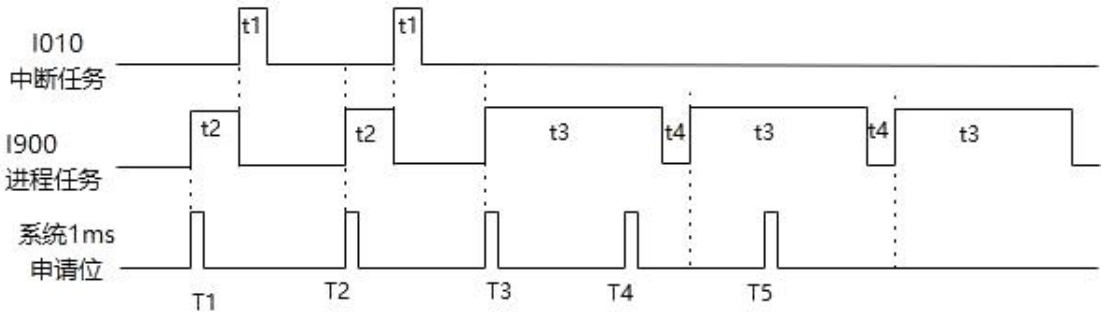


图文示意：

- 方案A、方案B 和方案C 是实现 1ms 插补的三种方案；
- 方案A 中只要M101 不导通，每 1ms 系统自动进入 I900 进程中断一次，可以实现 1ms 的插补输出；
- 方案B 和C 只要中断禁止位不置位，则每 1ms 自动进入中断一次，也可实现 1ms 的插补输出；
- 在一个程序中，如果只有一个 1ms 中断，再无其它中断，则上述A、B、C 三种方案效果一样，如果还有其它的中断，则 I900 的执行是最有保障的，因为 I900 为高级中断不可能被其它中断抢占，而 B 和C 存在被其它中断抢占的风险，因而，在程序中，如果需要使用 1ms 中断时，建议使用 I900 进程中断。

3.5.3 I900 进程同 I010 软中断的执行关系

MC 系列PLC中有两个申请位是和系统 1ms 中断有关的：I010 软中断的中断申请位 M8043，I900 的进程中断申请位 M8088。当I010 允许中断进入后，系统每间隔 1ms 自动置位一次中断申请位 M8043，I010 中断服务程序执行一次，当 I900 中断中开一个进程中断时，且进程中断不满足退出条件时，系统每间隔 1ms 自动置位一次进程申请位 M8088，I900 的进程任务执行一次。I900 进程中断同 I010 软中断执行时序如下图：



图文示意：

- T1-T5 表示 5 个1ms 对应的时刻，T1-T5 对应的 5 个宽度表示第几个 1ms 时刻，不代表时间长度；
- 系统 1ms 申请位：PLC系统内部自动每 1ms 置位 I010 的中断申请位 M8043 和I900 的进程中断申请位 M8088；
- I900 进程任务：高电平维持的时间表示单次进程任务的时间，用t2（任务时间小于 1ms）、t3（任务时间超过 1ms）表示；
- I010 中断任务：高电平维持的时间表示单次进程任务的时间，用 t1 表示；
- t4：时间不固定，最大值为 0.1ms（系统检测中断申请位的最大时间间隔），最小值为一个指令的周期（大约 50us）；
- T1 和T2 这两个时刻触发中断申请时，I900 进程任务没有超过 1ms，系统检测到中断申请时，先执行完I900 进程任务，而后再执行I010 中断任务；
- T3、T4 和T5 这三个时刻触发中断申请时，I900 进程任务超过 1ms，系统检测到中断申请时，先执行完I900 进程任务，而后再返回到主程序，在 t4 时间范围内检测到中断申请后会再次进入中断，如果I900 进程任务一直准许进入，那么 I010 永远不会进入，只有当 I900 中断退出后，且 I010 准许进入（对应的中断禁止位为 M8059=OFF），I010 中断才会进入执行中断任务；
- 特别说明：当中断任务或者进程任务时间过长时，会严重影响到主程序的扫描周期，尤其是在中断处理过程中再次检测到（中断重复进入）中断申请位时，会使中断接近连续进入，极端情况梯形图主程序每执行一条语句进入中断服务程序一次。

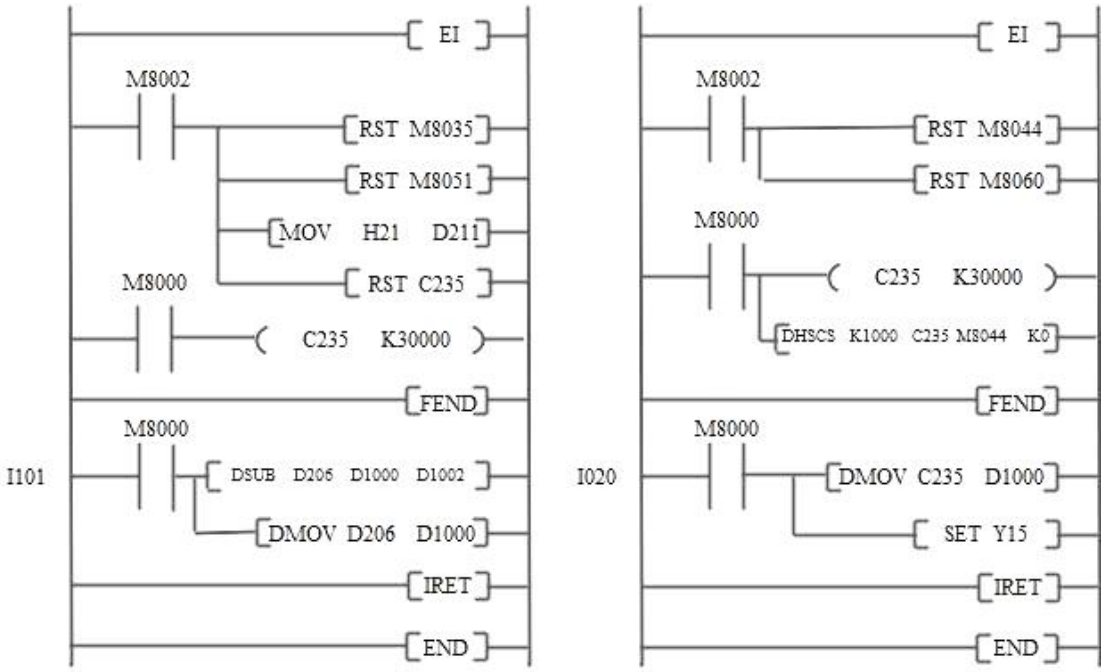
3.5.4 中断捕获高速计数值

MC 系列PLC中有四路高速计数口X0 、X2、X4和X6，可以外接编码器反馈脉冲等脉冲信号，当有外部信号来时， 获取此时编码器的反馈实时脉冲数值，比如在包装行业需要用到的色标捕获功能，需要色标信号来时获得此时外部编码器的位置信息，此时可以用外部中断自带的捕获功能（下述称之为方式 A）获取，也可以使用高速比较置位 DHSCS 指令开启中断（下述称之为方式 B）进行获取：

方式 A：外部中断 X10-X13 四个端子对应的外部中断，获取的输入轴中断脉冲寄存器 R106、R108（X0 高速计数口对应的输入轴中断脉冲数）、R156、R158（X2 高速计数口对应的输入轴中断脉冲数）、R206、R208（X4 高速计数口对应的输入轴中断脉冲数）、R256、R258（X6 高速计数口对应的输入轴中断脉冲数），此种方式对中断的触发时刻无要求。

方式 B：DHSCS 高速计数比较置位相关中断的申请位，进入中断获取当前的高速计数器 C对应的数值，一般选择中断 I020 到 I050 这 4 个软中断，此种方式只能应用在中断触发时刻已知的情况。

方式 A 和B 的实现方法如下：



图文示意：

- 方式 A 采用的是 X11 的上升沿中断捕获 X0 端口输入的脉冲数，当X11 上升沿中断触发时，X0 端口此时对应的脉冲数存放在输入轴 AXIN0 中的 R106 寄存器中，I101 为 X11 上升沿中断指针号，中断服务程序中将 R106 的值传给 D1000，此时获得的外部脉冲数值可以在程序其它地方参与运动控制的运算等，比如需要测得两个色标之间的距离，可以分别获取前后两个 X11 上升沿中断时R106 的数值，然后对他们做减法，即可求得两个色标之间的相差的脉冲数 D1002；
- 方式 B 实现的是当 X0 端口高速计数值到达设定的比较值（C235=K1000）时，触发中断 I020 中断服务程序执行相关的动作获取此时的 C235 数值和置位 Y15。

附录：中断异常自查表

表1：中断异常自查表

序号	异常现象	可能发生原因	建议对策
1	中断无法进入	没有写 EI 指令	程序中增加 EI 指令
		中断禁止位没有复位	复位中断禁止位
		中断申请位没有置位	需要的时候置位中断申请位
		指针号重复	检查中断指针号
2	中断程序执行异常	语法错误	检查中断程序书写格式，参考本章 2.3.2 节
		中断溢出	增大定时时间
			减少任务时间
		中断没有返回	检查是否有中断返回语句
		多重中断	检查多个中断的指针和返回语句是否一一对应
		中断申请位一直在申请	检查程序中处理中断申请位的方式
		程序中出现单条语句执行时间过长，大于 0.25ms	检查程序指令的使用方式，更改和简化耗时时间长指令。
3	中断嵌套不成功	语法错误	检查中断程序书写格式是否正确
		中断等级错误	检查嵌套中断的中断等级
		超过嵌套层数	查看嵌套层数是否超过 8 层
		发生嵌套时被抢占的中断任务剩余时间小于 0.1ms	正常现象

五、通信应用手册

1. 支持的通信协议介绍

1.1 下载监控协议

该协议兼容的是三菱 FX2N 系列 PLC 的串行监控协议，可以用于 PC 下载用户程序，监控或者修改PLC的内部运行参数，一般 HMI 与 MC 系列PLC 通信也可以采用该协议。

需要注意的是MC系列PLC下载程序时，默认波特率是 57600，且不能修改；在处于监控状态或者与 HMI通信时，波特率可以修改成 19200、38400和57600 等参数。

1.2 计算机链接协议

MC系列PLC内置的计算机链接协议兼容的是三菱计算机链接协议格式1，可以分别作为主站或者从站，当MC系列PLC之间或者与本公司其他产品进行数据传送时，该协议通信效率最高，推荐用户选择该协议。

1.3 标准 MODBUS协议（RTU 模式）

MC系列PLC内置标准 modbus 协议，仅支持 RTU模式。作为主站时，利用 modbus 通信指令与从站进行通信，作为从站时，只需要设置好通信参数，读取MC系列PLC规定的 modbus 地址即可。

1.4 自由协议

MC系列PLC内置报文发送指令 USEND 和报文接收指令 URCV，编程人员必须结合现场实际工况，来编写自己需要的梯形图，比如说设定串口的收发模式、通信参数，通信超时判断，指令本身给出的通信标志位等，才能达到预期想要的通信效果。

1.5 KLink总线协议

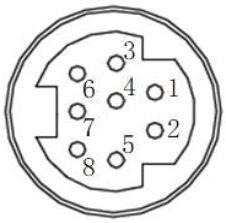
MC系列PLC内置的KLink总线协议是KEWEI高速通信协议，可以分别作为主站或从站，实现MC系列PLC之间或者与本公司其他产品间的高速通信。

2. 串口初始化

2.1 串口介绍

MC系列PLC配备有三个串口，串口 0、串口 1和串口2，三个串口上电默认均为 RS232 协议，但是也可以通过串口初始化指令配置成 RS485协议，不同的是串口 0 接口电平为标准 232 电平，配置成 485 后，需要利用 RS232 转 485，完成电平转换；串口 1 接口电平为标准 RS485 电平，配置成 RS232 通信协议后，需要利用 RS485 转232 完成电平转换。

串口 0 为 MINI8 芯插座孔，端口定义如下



Mini Din 8 芯插座（孔）

管脚号	信号	描述
1	NC	空脚
2	TXD	发送
3	RXD	接收
4	NC	空脚
5	GND	地线
6	NC	空脚
7	RTS	请求发送
8	CTS	允许发送

注：7、8 引脚内部已短接

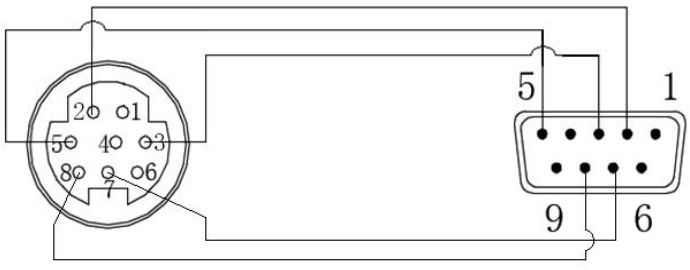
串口 2 为螺丝端子，端口定义如下



RSA2 RSB2

- 串口 2 用作 RS485 与第三方通信时，RSA2 和 RSB2 需和对方一一对应连接。
- 串口 2 如果用作 RS485 与其他设备连接后，必须利用串口初始化指令将串口配置成相对应协议的 RS485 口，如果未配置，可能对通信链路造成干扰，这是因为串口 2 默认 RS232 编程口协议，RS232 协议在遇到 02H 或者 05H 报文时，会进行报文回复，从而影响到通信报文内容。

科威 PLC 编程电缆内部接线如下图：



Mini Din 8 芯插头（针） DB9 插头（孔）

科威 PLC 编程电缆引脚定义

ini Din 8 芯插头（针）			DB9 插头（孔）		
管脚号	信号	描述	管脚号	信号	描述
1	NC	空脚	1	NC	空脚

2	TXD	发送	2	TXD	发送
3	RXD	接收	3	RXD	接收
4	NC	空脚	4	NC	空脚
5	GND	地线	5	GND	地线
6	NC	空脚	6	NC	空脚
7	RTS	请求发送	7	RTS	请求发送
8	CTS	允许发送	8	CTS	允许发送
			9	NC	空脚

- **串口1为网口座孔：**



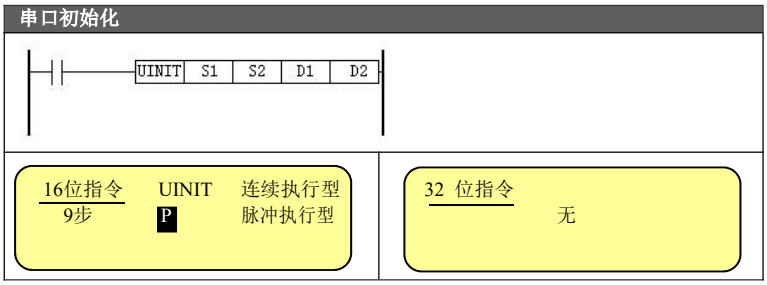
- 串口 1 用作标准 RS485 与第三方通信时，RSA 和 RSB 需和对方一一对应连接。
- 串口 1 如果用作 RS485 与其他设备连接后，必须利用串口初始化指令将串口配置成相对应协议的 RS485 口，如果未配置，可能对通信链路造成干扰，这是因为串口 1 默认 RS232 编程口协议，RS232 协议在遇到 02H 或者 05H 报文时，会进行报文回复，从而影响到通信报文内容。
- 串口1可作为高速485通信协议，可以分别作为主站或从站，实现MC系列PLC之间或者与本公司其他产品间的高速通信。

2.2 串口初始化

2.2.1 串口初始化指令

通过串口初始化指令指定通信端口、协议、串口参数等内容。

1) **指令格式**



2) **设定数据**

操作数	作用	类型
S1	指定协议控制字	16 位 BIN
S2	指定通信控制字	16 位 BIN
D1	指定资源占用起始 D 寄存器	16 位 BIN
D2	指定串口号	16 位 BIN

3) **对象软元件**

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S1	●	●								
S2	●	●								
D1	●	●								
D2	●	●								

功能与动作

当执行串口初始化指令时，填装串口功能控制字，重新初始化串口，使之工作于新的通信模式。一般而言，串口初始化指令只需要执行一次，执行完毕后系统将该串口指定为配置的协议。

协议控制字S1

协议控制字	数值	说明
高8位	0	HEX传输协议（内部协议，用户无需使用）
	1	编程口协议（初始默认）
	2	计算机链接主站协议
	3	计算机链接从站协议
	4	MODBUS协议
	5	自由协议
低8位	1~255	指定设备从站号，只有当选择计算机链接从站协议或MODBUS协议时生效

通信控制字S2

通信控制字	说明	数值	意义
高8位	波特率	0	9600 bps
		1	19200 bps
		2	38400 bps
		3	57600 bps(保留，用户不能使用)
		4	115200 bps（保留，用户不能使用）
		>4	9600 bps
第7~4位	保留	补0	
第3位	停止位	0	2位停止位
		1	1位停止位
第2~1位	校验位	00	无奇偶校验
		01	奇校验
		11	偶校验
第0位	数据位	0	8位数据
		1	7位数据



- 上电首运行触点 M8002 执行串口初始化指令。
- 参数的配备如下：
  - 计算机链接协议(主站)
  - 串口通信参数为 9600，n，8，1
  - 通信缓存区首地址指向 D7900
  - 指定串口 1 为通信对象

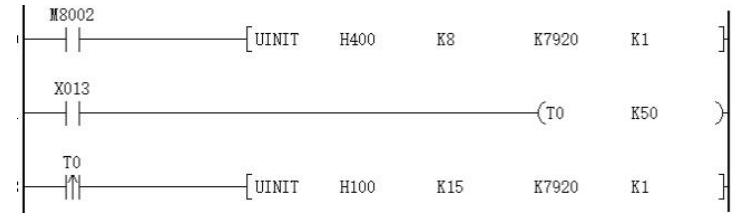
注意：串口 0 通信缓存区首地址默认是 D7900，而串口 1 通信缓存区首地址默认 D7920，上例中直接将串口 1 通信缓存区地址设置成 D7900，看似与串口 0 首地址冲突，实际上是由于编程口协议内部并不占用通信地址，故可以使用，但如果是串口 0 和串口 1 同时都设置成RS485 协议，则必须遵循串口 0 与串口 1 通信缓存区不冲突规则，如串口 0 通信缓存区首地址设置为 D7900，串口 1 通信缓存区首地址设置为 D7920 。

2.2.2 串口初始化指令应用举例

- 串口 1 的双重应用

在一些特殊应用场合，串口 0 可能会被占用，导致下载梯形图只能利用串口 1，同时串口 1 还需要与第三方设备进行 RS485 通信（如 modbus 主站），这种情况下灵活利用串口初始化指令可轻松解决此问题，举例如下：

程序说明



- 上电串口 1 配置成 MODBUS 主站梯形图上电后串口 1 配置成 modbus 主站协议。
- 当用户需要下载梯形图时，将 X13 端口（可指定任意端口）导通 5 秒后，串口 1 将切换成 RS232 编程口协议，此时用户利用 RS232 转 485 转换头连接上电脑，对LP3系列PLC进行下载、上传、监控等操作。

2.2.3 串口协议与可配置格式对应表

通信控制字		通信格式			协议控制字 （高8位 bit8-bit15）					
					0	1	2	3	4	5
bit0-bit3		停止位 Bit3	校验 Bit2-1	数据位 Bit0	HEX传输协议	编程口协议	计算机链接主站协议	计算机链接从站协议	modbus RTU协议	自由协议
0	0	2	N无	8	√				√	√
1	1	2	N无	7						
2	2	2	O奇	8					√	√
3	3	2	O奇	7		√	√	√		√
4	4	2		8						
5	5	2		7						
6	6	2	E偶	8					√	√
7	7	2	E偶	7		√	√	√		√
8	8	1	N无	8	√	√	√	√	√	√
9	9	1	N无	7						
A	10	1	O奇	8					√	√
B	11	1	O奇	7		√	√	√		√
C	12	1		8						
D	13	1		7						
E	14	1	E偶	8					√	√
F	15	1	E偶	7		√	√	√		√

√ 上表中，√标示部分，表示当前协议支持设置为此通信格式。

如协议控制字 （高8位 bit8-bit15）配置为1（协议字为H100），通信控制字的bit0-bit3可设置格式有H3(7数据位，2停止位，O奇校验)、H7(7数据位，2停止位，E偶校验)、H8(8数据位，1停止位，N无校验)、HB(7数据位，1停止位，O奇校验)、HF(7数据位，1停止位，E偶校验)。

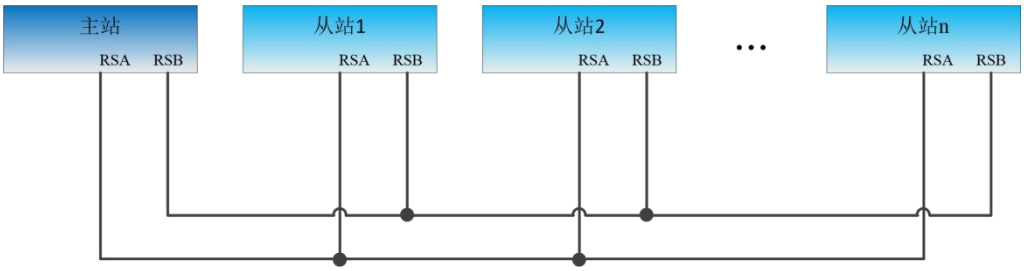
配置为不支持的格式（上表中空白未打勾标注的部分）时，会自动重置为9600bps，7数据位，1停止位，E偶校验格式的编程口协议。

3. 计算机链接协议

3.1 计算机链接协议介绍

科威 RS485 计算机链接协议加载的是三菱计算机链接协议1的子集，即只能对 D 寄存器进行读写操作，不能对位元件和线圈进行操作。用户在使用 MC系列PLC过程中如果需要和本公司其他产品（或者其他公司产品支持计算机链接协议，如组态软件昆仑通态、海泰克、威伦通等）进行互联交换数据时，推荐使用计算机链接协议，当然也可以用 modbus 和自由协议，选择计算机链接协议是因为它的通信效率比较高，用户只需要将待通信的产品的 RSA 和 RSB 一一对接，然后软件进行简单设置就能够正常通信了。

计算机链接协议的网络结构示意图如下：



3.2 计算机链接协议格式

3.2.1 通信格式

计算机链接协议支持的通信格式如下

通信格式				梯形图设置(作为主站)
数据长度	停止位	奇偶校验	波特率	
8	1	无	9600	UINIT H200 H8 K7900 K1
8	1	无	19200	UINIT H200 H108 K7900 K1
8	1	无	38400	UINIT H200 H208 K7900 K1
7	2	奇	9600	UINIT H200 H3 K7900 K1
7	2	奇	19200	UINIT H200 H103 K7900 K1
7	2	奇	38400	UINIT H200 H203 K7900 K1
7	2	偶	9600	UINIT H200 H7 K7900 K1
7	2	偶	19200	UINIT H200 H107 K7900 K1
7	2	偶	38400	UINIT H200 H207 K7900 K1
7	1	奇	9600	UINIT H200 HB K7900 K1
7	1	奇	19200	UINIT H200 H10B K7900 K1
7	1	奇	38400	UINIT H200 H20B K7900 K1
7	1	偶	9600	UINIT H200 HF K7900 K1
7	1	偶	19200	UINIT H200 H10F K7900 K1

7	1	偶	38400	UINIT H200 H20F K7900 K1

● 注意：● 从上面表格可以看出计算机链接协议通信格式只支持 8 个数据位、1 个停止位，无校验，波特率 支持 9600/19200/38400，推荐使用 9600，此时通信最稳定，且能够满足一般通信的实时要求。

●

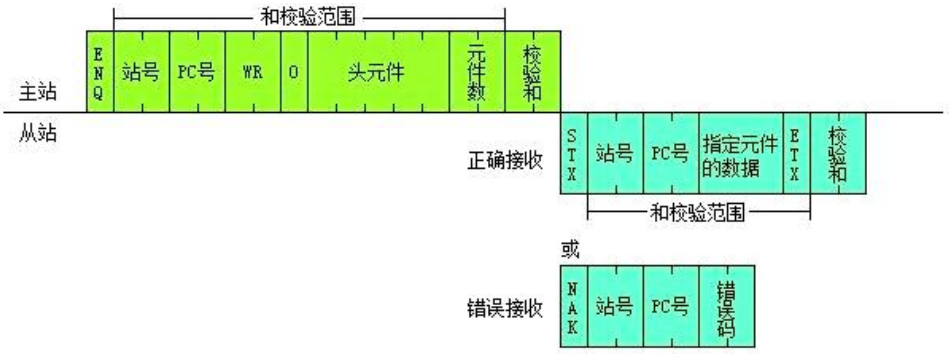
● ● 当作为计算机链接协议时，只需要修改串口初始化指令中第一个操作数，从上表格中可以看出， 作为主站时，第一个操作数固定为 H200；作为 1 号从站时则第一个操作数为 H301，作为 2 号从站时，第一个操作数为 H302，后面以此类推。

3.2.2 协议内容

读报文协议格式

主站读取从站数据协议格式如下所示。

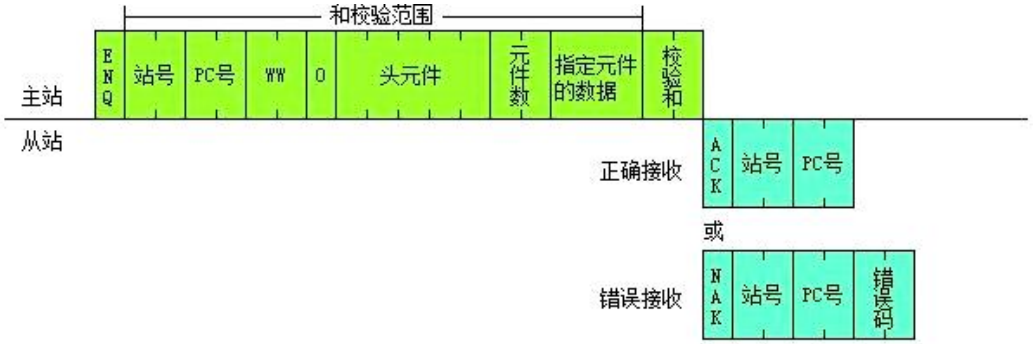
- 横线上方表示主站发送的数据,横线下方表示从站发送的数据。
- 数据发送顺序从左到右。



写报文协议格式

主站向从站写数据协议格式如下所示。

- 横线上方表示主站发送的数据,横线下方表示从站发送的数据。
- 数据发送顺序从左到右。



- 站号：2 字符十六进制数,范围 1-3F，对应十进制 1-63，表示可带 63 个从站。
- PC 号：规定为 FF。
- 头元件：通信的软元件首地址，只能是数据寄存器 D。用 5 个字符表示，例如 D1000。
- 元件数：用 2 位十六进制的字符表示,例如 0A,表示 10 个字元件。
- 指定元件数据：一个元件数据用 4 个十六进制的字符表示,如 AB96。
- 错误码：保留使用。
- 校验和 CRC：将和校验区 ASCII 码字符的十六进制数相加后，取后两位字符（00-FF）。

主站报文协议格式解释表：

名称	代码	定义	字符类型	发送方式
ENQ	05H	询问，不计入和校验区	控制型	字符
站号	01H～3FH	从站站号（地址）	数据型	ASCII 码
PC 号	FFH	固定值	数据型	ASCII 码
命令	WR/WW	读/写，以 1 个字为单位进行操作	数据型	ASCII 码
等待时间	0H	1 字符，规定为 0H 表示无等待	数据型	ASCII 码
头元件	D□□□□	5 字符表示	数据型	ASCII 码
元件数	00H～40H	需要通信的 D 软元件个数	数据型	ASCII 码
CRC	00H～FFH	检验和 CRC	数据型	ASCII 码

从站报文协议格式解释表：

名称	代码	定义	字符类型	发送方式
ACK	06H	确认	控制型	字符
STX	02H	文本起点，不计入和校验区	控制型	字符
NAK	15H	不确认	控制型	字符
站号	01H～3FH	从站站号	数据型	ASCII 码
PC 号	FFH	固定值	数据型	ASCII 码
指定元件数据	0H～FH	指定 D 软元件的数据，1 个 D 软元件需 4 位字符表示	数据型	ASCII 码
ETX	03H	文本结束	控制型	字符
CRC	00H～FFH	检验和 CRC	数据型	ASCII 码

3.2.3 协议示例

通信报文应用举例

主站与1号从站通信的示例报文描述。

读 1 号站数据：从 D7101 开始，共 1 个单元。则发送报文内容如下：

询问协议	和校验区（ASCII码—16 进制数）						累加取后两位字符
	站号	PC 号	命令	等待时间	头元件	元件数	
ENQ	0A	FF	WR	0	D7101	0 1	CRC
05H	30H 31H	46H 46H	57H 52H	30H	44H 37H 31H 30H 31H	30H 31H	33H 34H

CRC = 30H+31H + 46H+46H + 57H+52H + 30H + 44H+37H+31H+30H+31H + 30H+31H = 334H

取最后 2 个字符 34H，34H 用 ASCII 码表示是 33H 34H。ASCII 码与十六进制转换如下表

数值	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	HA	HB	HC	HD	HE	HF
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46

从站回报文内容如下：

询问	和校验区（ASCII码—16 进制数）				累加取后两位字符
	站号	PC 号	指定元件的数据（见注释）	文本结束	
STX	01	FF	0064	ETX	CRC
02H	30H 31H	46H 46H	30H 30H 36H 34H	03H	42H 41H

CRC = 30H+31H + 46H+46H + 30H+30H+36H+34H +03H =1BAH

取最后 2 个字符 BAH，用 ASCII 码表示是 42H 41H。

注：指定元件的数据，每一元件必须用 4 位字符来表示。本例 1 个 D 寄存器软元件，值为 0064（此为示例值，实际以通信数值为准）。

3.3 计算机链接协议主从互联

3.3.1 RS485 网络拓扑结构

（1）名词解释

- 主节点：主动向其它各节点发送命令并取得信息的站点，主从结构网络必须有主节点。又称主站。
- 从节点：在网络中接受主节点命令，返回要求信息的节点，称为从节点。又称从站。

（2）网络特征：

- 485 网络是主从结构网络。
- 通过梯形图进行网络设置，设置内容包括：网络设备地址、网络通讯数据内容、网络数据通讯速度等。
- 网络为半双工通信，即通信链路上数据可以双向发送接收，但是不能同时进行。

（3）注意事项：

- 主站节点必须是有平台节点，主站节点只有一个，地址固定为 0。
- 一个网络最多有 63 个从节点，从节点地址可设定，但不能有相同的地址。
- 主站与任一从站都能通讯，而从站与从站之间交换数据必须通过主站才能进行。

3.3.2 任务调度配置

1) MC系列PLC作为计算机链接主站协议时，需要对通信协议调度区进行配置。

通信协议调度区起始地址由串口初始化指令 UINIT 的操作元 D1 指定，指定调度区 D 寄存器的起始号，调度区将占用从起始号开始的 20 个D 寄存器。



当梯形图串口初始化如上图配置时，即 D7900-D7919 为该串口的任务调度配置区。

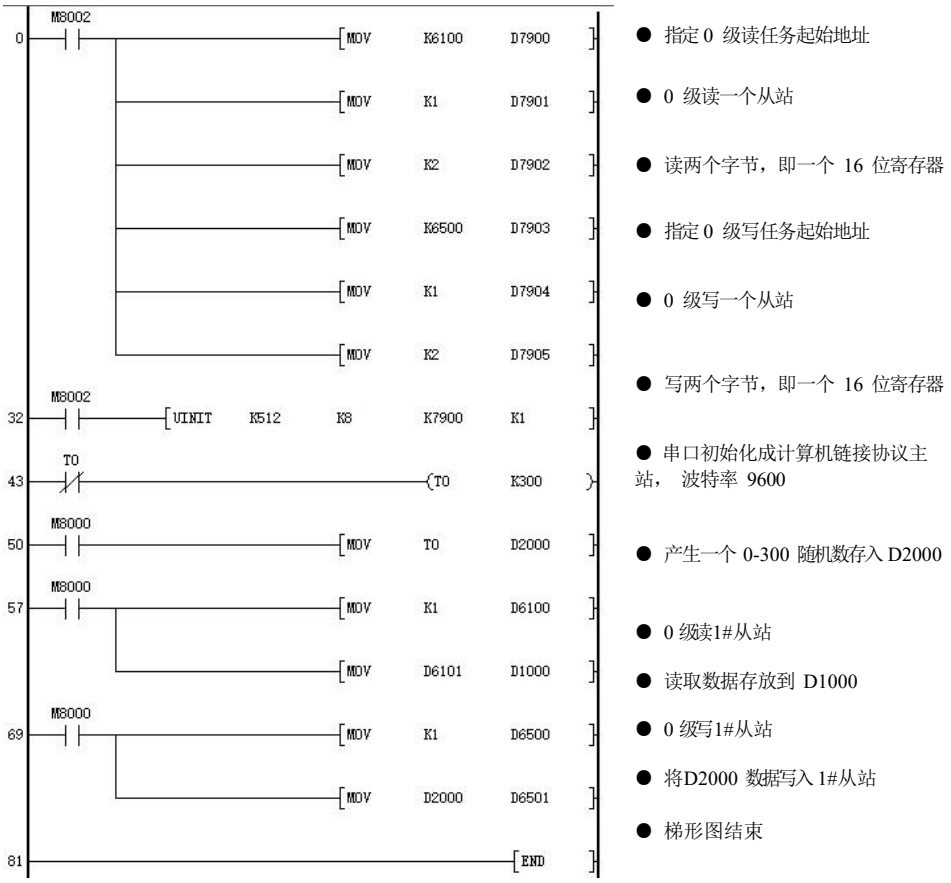
地址单元	说明	推荐初始化数值
D7900	0 级读任务起始单元	用户自定义，一般选取 D6000 以后的单元
D7901	0 级读任务数（需要读的从站个数）	用户自定义，最大不能超过 63
D7902	0 级读任务数据长度(字节数)	用户自定义，最大不能超过 96
D7903	0 级写任务起始单元	用户自定义，一般选取 D6000 以后的单元
D7904	0 级写任务数（需要写的从站个数）	用户自定义，最大不能超过 63
D7905	0 级写任务数据长度(字节数)	用户自定义，最大不能超过 96
D7906	1 级读任务起始单元	用户自定义，一般选取 D6000 以后的单元
D7907	1 级读任务数（需要读的从站个数）	用户自定义，最大不能超过 63
D7908	1 级读任务数据长度(字节数)	用户自定义，最大不能超过 96
D7909	1 级写任务起始单元	用户自定义，一般选取 D6000 以后的单元
D7910	1 级写任务数（需要写的从站个数）	用户自定义，最大不能超过 63
D7911	1 级写任务数据长度(字节数)	用户自定义，最大不能超过 96
D7912	00#～15# 设备状态：0 在线 1 离线	0xFFFF
D7913	16#～31# 设备状态：0 在线 1 离线	0xFFFF
D7914	32#～47# 设备状态：0 在线 1 离线	0xFFFF
D7915	48#～63# 设备状态：0 在线 1 离线	0xFFFF
D7916	从站正确返回数据的报文数	0
D7917	从站不能正确返回数据的报文数	0
D7918～D7919	保留使用	-

注意：任务调度区的初始化工作必须在 UINIT 指令执行前完成，通常采用上电第一个扫描周期进行配置。

MC系列PLC作为从站时，无须进行任务调度配置，只需要利用串口初始化指令配置从站号以及相关的通信参数。

2) 单个主从站通信为例说明任务调度配置的使用

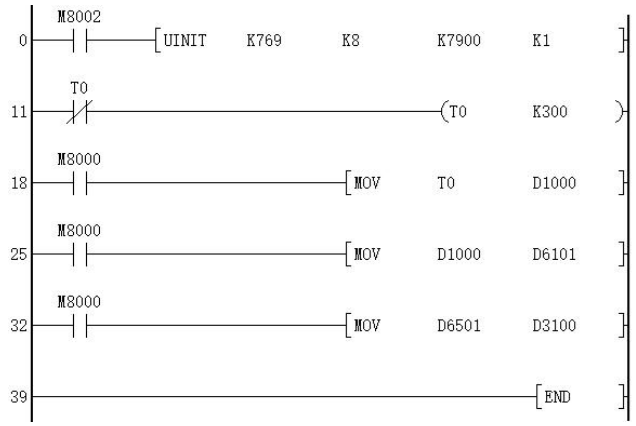
MC系列PLC A 主站梯形图



主站梯形图解释

- 上电第一个扫描周期配置 0 级任务 D7900-D7905， 1 级任务D7906-D7911 此例未作调用，可以不作配置。
- 任务调度配置完成后，利用 UINIT 指令进行串口初始化，UINIT 参数配置请参考本章 2.2 章节，
- 设置为计算机链接协议主站，波特率 9600，8 个数据位，1 个停止位，无奇偶校验。
- 读取从站 1 个D 寄存器内容，从站将内容放入缓存寄存器 D6101，主站直接读取 D6101 数值。
- 将D2000 数值写入从站，写入的值放入缓存寄存器 D6501，从站通过 D6501 来获取。
- 可通过监控 D7916 和D7917 数据寄存器查看当前通信是否正常，D7916 数据一直累加，且 D7917 数据为 0，或者有少量数值，存在少量数值是由于主从站之间上电时序有误差导致，则代表通信正常；如果D7917 数据一直累加，则代表通信异常。

MC系列PLC B 从站梯形图



- 指定成 1#从站，波特率 9600
- 产生 0-300 随机数
- 将随机数存放到 D1000
- 将 D1000 值上传到主站
- 将主站写入的值存放到 D3100

从站梯形图解释

- 上电利用串口初始化指令配置成计算机链接协议 1#从站，通信参数与主站保持一致。
- 将D1000 寄存器内容转移到缓存寄存器 D6101，主站通过 D6101 来获取其数据内容。
- 将主站写入的寄存器内容通过缓存寄存器 D6501，转移到 D3100。

3) 多从站任务调度说明

当主站与多个从站进行通信时，需遵循读/写任务数据长度取大值的原则。

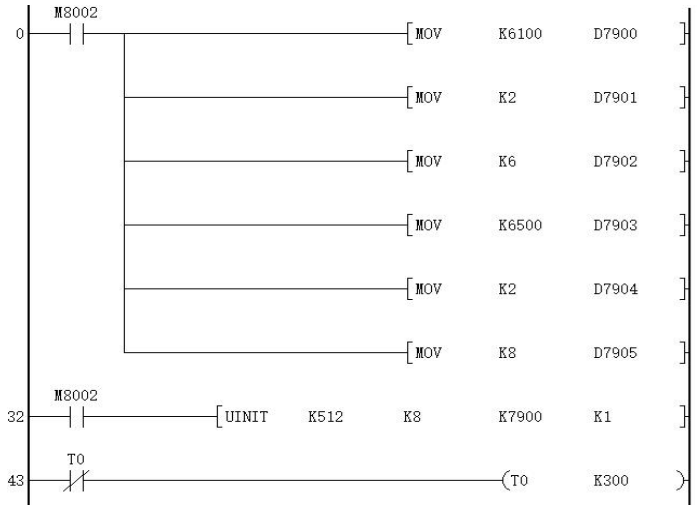
假定主站与从站A和从站B进行通信，根据需求分配读写任务数和任务数据长度。

- 与从站A的实际通信量为读取2个D寄存器数据，写入4个D寄存器数据。
- 与从站B的实际通信量为读取3个D寄存器数据，写入3个D寄存器数据。 可以分配0级读任务2个，0级写任务2个，用来和从站A以及从站B进行通信。

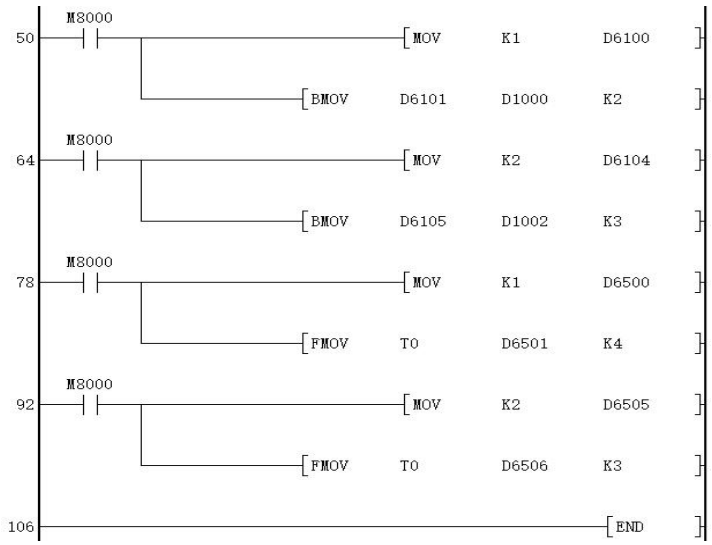
0级读任务数据长度必须填写6，即取从站中读任务数据长度最长的数值。对于从站A而言，前2个D寄存器作为有效数据应用即可。

0级写任务数据长度必须填写8，即取从站中写任务数据长度最长的数值。对于从站B而言，前3个D寄存器作为有效数据应用即可。

①MC系列PLC主站梯形图



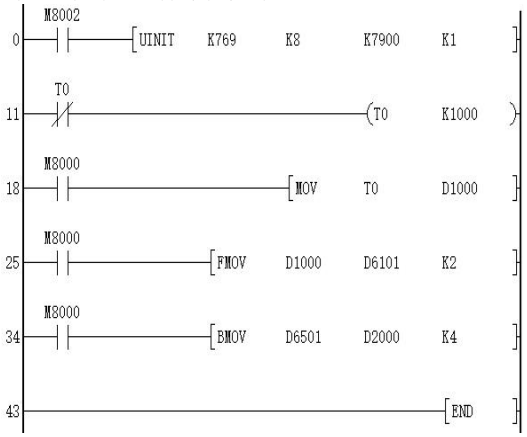
- 指定 0 级读任务起始地址
- 主站 0 级读2 个从站
- 读6 个字节，即 3 个寄存器
- 指定 0 级写任务起始地址
- 主站 0 级写2 个从站
- 写8 个字节，即 4 个寄存器
- 串口初始化 9600 8 1 N
- 0-300 随机数写入从站



- 0 级读1 号从站首地址
- 0 级读取1 号从站2 个寄存器内容
- 0 级读取2 号从站首地址
- 0 级读取2 号从站3 个寄存器内容
- 0 级写入1 号从站首地址
- 0 级将T0 值写入1 号从站4 个寄存器
- 0 级写入2 号从站首地址
- 0 级将T0 值写入2 号从站3 个寄存器

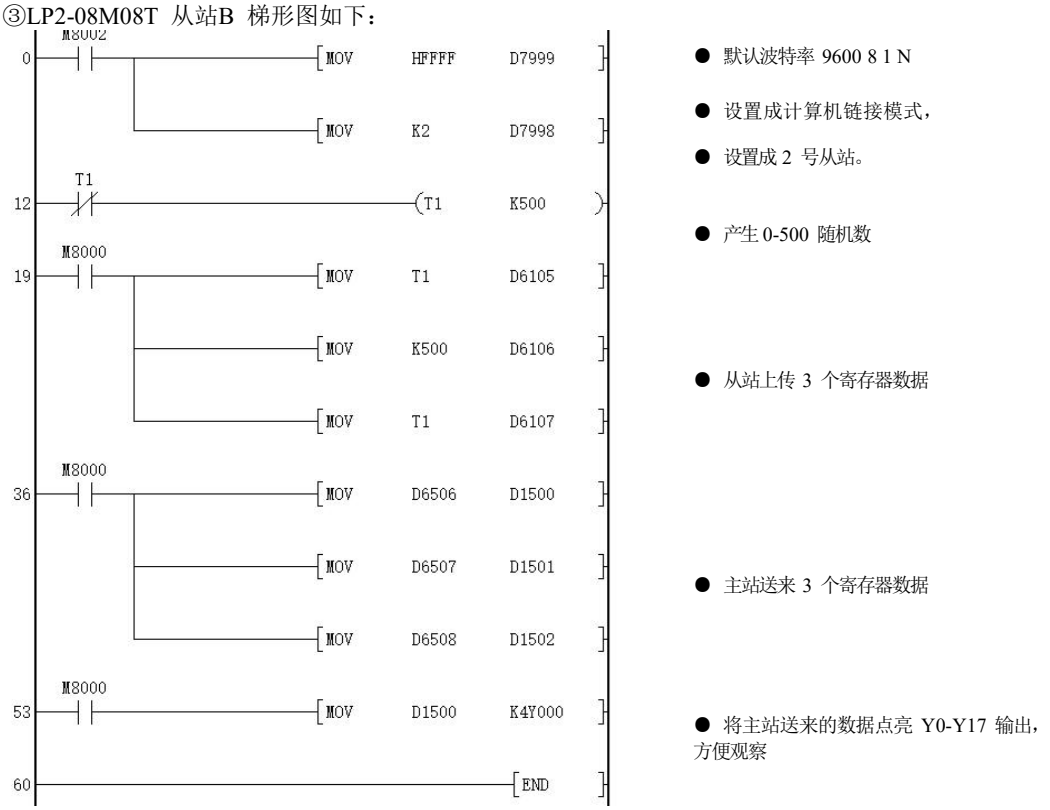
注意：对于 1 号从站 0 级读任务来说，虽然只需要读 2 个寄存器内容，但是由于读任务是 6 个字节， 会自动占用 3 个寄存器内容 D6101-D6103，所以 2 号从站读起始地址开始为 D6104。

②MC系列PLC从站 A 梯形图如下：



- 设置成 1 号从站，波特率 960081N
- 0-1000 随机数返回主站
- 将随机数赋值给 D1000
- 从站上传数据
- 主站送来数据

③LP2-08M08T 从站B 梯形图如下：

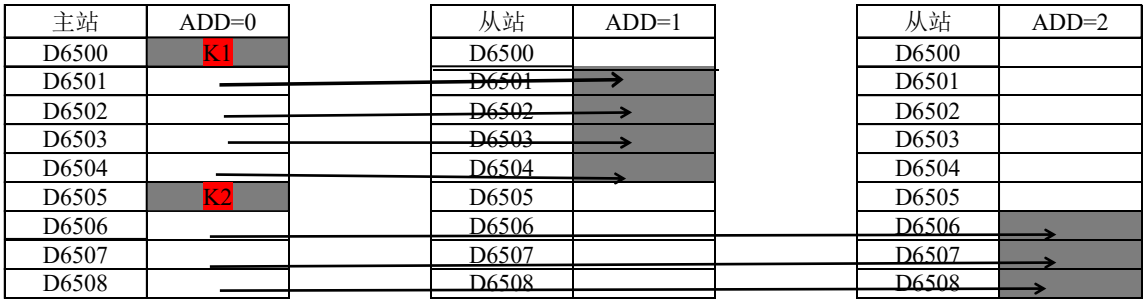


注意：由于计算机链接协议主从站只能互传D 寄存器，对于主从之间需要通信位单元，如 X 输入，Y 输出，M 中间继电器，可以利用字转位或者位转字指令进行中转，达到控制要求。如本例中最后将 D1500 寄存器的16 位 Bit 状态从低到高分别传送到 Y0-Y17。

④上例中主站读取从站的数据流向示意图如下：



⑤主站写入从站的数据流向图如下：



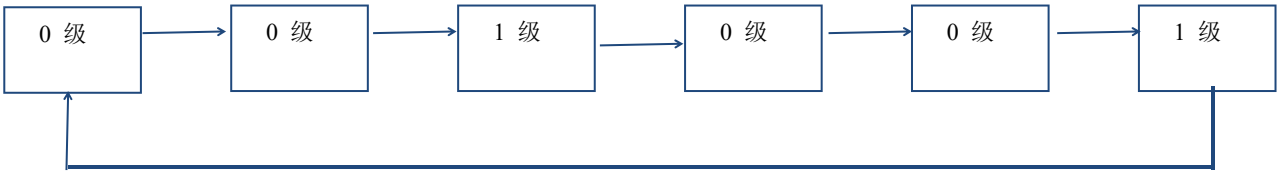
3.3.3 报文流量

1) RS485 计算机链接协议规划了两级任务：

- 0 级任务：发送报文流量占发送报文总流量的 2/3T。
- 1 级任务：发送报文流量占发送报文总流量的 1/3T。

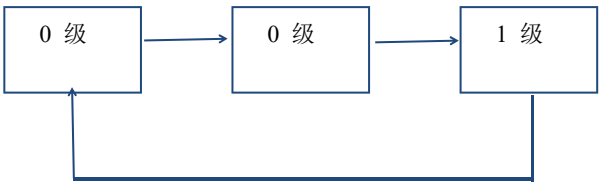
2) ① 在数据量相同情况下，0 级数据循环比 1 级数据循环快 1 倍。

例如主站A 对从站 B 通信调度有 0 级读任务 1 个，0 级写任务 1 个，1 级读任务 1 个，1 级写任务 1 个，则串口对 4 个任务的每个周期发送报文循环如下：



流程图解释：当 0 级任务和 1 级任务一样多时，一个周期发 6 次报文，其中 0 级任务占 4 次，满足 0 级任务占发送报文总流量 2/3，1 级任务占 2 次，满足 1 级任务占报文总流量 1/3。在一个周期内，0 级读任务发送 2 次，1 级读任务发送 1 次，0 级写任务发送 2 次，1 级写任务发送 1 次，故 0 级数据循环比 1 级数据循环快 1 倍。

② 如果 0 级数据量等于 1 级数据量两倍，则通讯循环次数一样。例如主站 A 对从站 B 通信调度任务有 0 级读任务 1 个，0 级写任务 1 个，1 级读任务 1 个，则串口对 3 个任务的每个周期发送报文循环如下：



流程图解释：当 0 级任务比 1 级任务多 1 倍时，此例中 0 级读和 0 级写，总共 2 个任务，一个周期内发送 3 次报文，其中 0 级任务占 2 次，满足 0 级任务占发送报文总流量 2/3；1 级任务占 1 次，满足 1 级任务占报文总流量 1/3。在一个周期内，0 级读任务发送 1 次，0 级写任务发送 1 次，1 级读任务发送一次，0 级任务和 1 级任务数据循环次数相同。

③ 0 级报文数据量超出 1 级数据量 2 倍以上时，则 0 级比 1 级循环次数少。

3) 实际应用中用户可以根据现场情况灵活使用 0 级和 1 级报文，来满足不同的应用要求。

例如，每秒报文总流量为 15 个/秒，则0 级每秒报文流量为 10 个/秒，1 级报文流量为 5 个/秒，如果规划 0 级报文个数为 2 个，1 级报文个数为 5 个，则每秒 0 级报文循环次数为 10/2=5 次，1 级报文循环次数 5/5=1 次，从而能够达到不同的实时要求。

4) 流量估算

以通信波特率 9600 为例，理论上每秒通信字节最大 V_{MAX}=9600bps/8=1200 字节/秒，但由于实际通信过程中每帧报文都有发送接收间隔时间，该时间为 5ms，并且通信数据量对其也有影响，通信数量增多，则接近其理论最大值，经过实际测算 V_{min} 在900 字节/秒左右。

每秒钟的报文流量估算如下：

单独执行 0 级或者 1 级读任务时,报文流量：

Q1=V/(25+N1*4)+5，N1 为读取数据字的个数，公式中分母为单次读任务所需要的字节数。

单独执行 0 级或者 1 级写任务时，报文流量：

Q2=V/(22+N2*4)+5，N2 为写出数据字的个数，公式中分母为单次写任务所需要的字节数。

当写报文和读报文任务同时进行时，报文流量：

Q3=M*V/(25+N1*4)+(25+N1*4+（5*M）)，其中 M 为总任务数，公式中分母为读写任务所需要的字节数之和。

注意：上述公式中计算出来的 Q 值均为估算值，与实际值存在误差，仅作参考。

3.4 应用案例

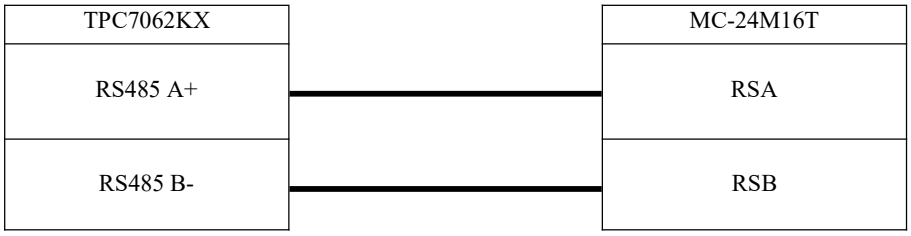
昆仑通态触摸屏与MC系列PLC利用计算机链接协议通信。

1) 硬件配置

触摸屏型号：TPC7062KX

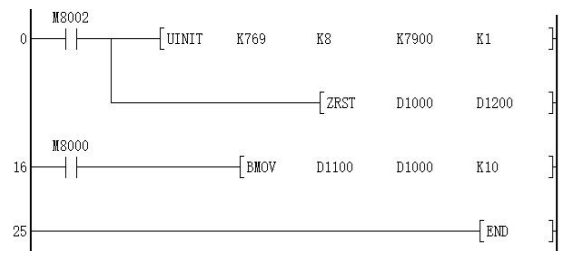
MC系列PLC型号：MC-24M16T

硬件连线



2) 软件设置

梯形图设置如下



- 设置成 1 号从站，波特率 9600 8 1 N
- 清空与主站通信区，方便查看
- 将主站写入数据存放主站读取区

3) 触摸屏设置：

设备窗口选择通用串口父设备，关联三菱 FX 系列串口。

通信设置如下：

串口父设备设置

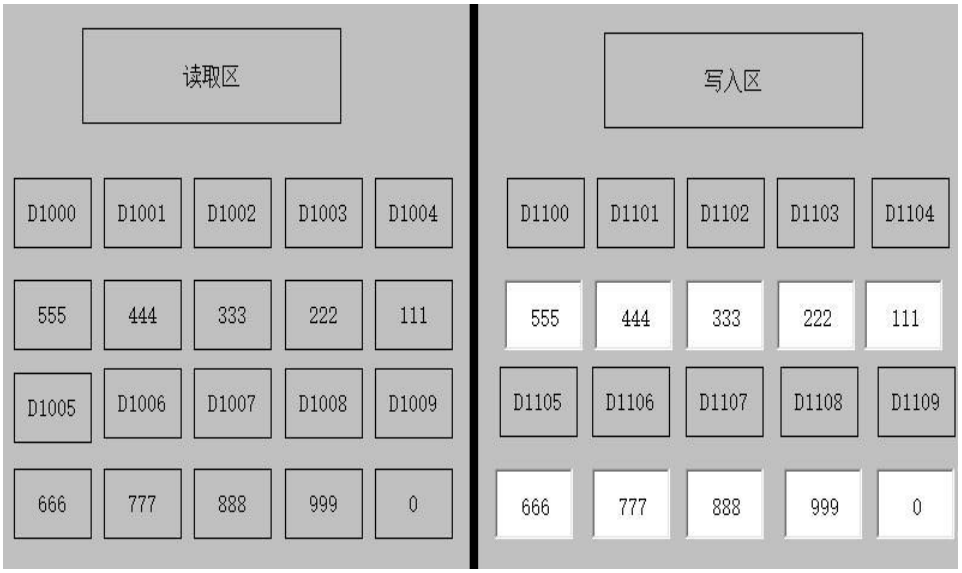
设备属性名	设备属性值
设备名称	通用串口父设备0
设备注释	通用串口父设备
初始工作状态	1 - 启动
最小采集周期(ms)	1000
串口端口号(1~255)	1 - COM2
通讯波特率	6 - 9600
数据位位数	1 - 8位
停止位位数	0 - 1位
数据校验方式	0 - 无校验

三菱 FX 系列串口

设备属性名	设备属性值
[内部属性]	设置设备内部属性
采集优化	1-优化
设备名称	设备0
设备注释	三菱_FX系列串口
初始工作状态	1 - 启动
最小采集周期(ms)	100
设备地址	1
通讯等待时间	200
快速采集次数	0
协议格式	0 - 协议1
是否校验	1 - 求校验
PLC类型	4 - FX2N

4) 功能说明

触摸屏将 D1100-D1109 数值写入MC系列PLC，并读取 D1000-D1009 数值，而MC系列PLC将触摸屏写入的数值存放到触摸屏读取区，故改变写入区数据寄存器值，读取区值会相应变化。见下图：



4. MODBUS 协议

4.1 协议说明

MC系列PLC具备完整的 MODBUS 协议通信功能，符合国际标准(RTU 模式，不支持 ASCII 模式)。用户可通过编写梯形图来实现MC系列PLC同其它设备的通信。

MODBUS 串行链路协议是一个主从协议。在同一时间，总线上只能有一个主站，和一个或多个（最多 247 个）从站。MODBUS 通信总是由主站发起。当从站没有收到来自主站的请求时，不会发送数据， 从站之间不能互相通信。主站同时只能启动一个 MODBUS 事务处理。

主站用两种模式向从站发出 MODBUS 请求：

- 单播模式，主站寻址单个从站。从站接收并处理完请求之后，向主站返回一个报文。

在这种模式下，一个 MODBUS 事务处理包含两个报文：一个是主站的请求，另一个是从站的应答。每个从站必须有唯一的地址（1～247），这样才能区别于其它站并独立地被寻址。

- 广播模式，主站可以向所有的从站发送请求。

对于主站发送的广播请求从站不会应答返回；广播请求必须是写命令。所有设备必须接受广播方式的写命令。地址 0 被保留用来识别广播通信。

4.2 MODBUS主站配置

4.2.1 支持 MODBUS 主站协议的端口

MC系列PLC的所有串口（COM0/COM1/COM2）都支持 MODBUS 主站协议。

作为 MODBUS 的主站，MC系列PLC的用户程序根据设备工艺要求，采用 MODBUS 指令对从站相关资源进行读入和写出。

4.2.2 端口配置方法

通过串口初始化指令(UINIT)将串口 1 配置为 MODBUS 主站，才能与从站通信，关于 UINIT 指令的详细用法请参见 4.2.4 章节。

正常情况下，一般将串口 1 配置为 MODBUS 通信端口，串口 0 作为编程口使用，如果梯形图没有对两个串口进行配置，则都默认配置为编程口，在梯形图运行第一个扫描周期内，需对串口进行配置。特殊情况下的端口配置方法请参见 4.5.3 章节。

推荐参数配置：传输速率 9600 bps，无校验，数据长度 8 位,停止位 1 位。传输速率也可选择 19200 bps 和 38400 bps，数据长度只能为 8 位，其它格式可配置，具体配置方式参照下表。

MODBUS主站支持配置如下：

通信格式				梯形图设置
数据长度	停止位	奇偶校验	波特率	
8	2	无	9600	UINIT H400 H0 K7900 K1
8	2	无	19200	UINIT H400 H100 K7900 K1
8	2	无	38400	UINIT H400 H200 K7900 K1
8	2	奇	9600	UINIT H400 H2 K7900 K1
8	2	奇	19200	UINIT H400 H102 K7900 K1
8	2	奇	38400	UINIT H400 H202 K7900 K1
8	2	偶	9600	UINIT H400 H6 K7900 K1
8	2	偶	19200	UINIT H400 H106 K7900 K1
8	2	偶	38400	UINIT H400 H206 K7900 K1
8	1	无	9600	UINIT H400 H8 K7900 K1
8	1	无	19200	UINIT H400 H108 K7900 K1
8	1	无	38400	UINIT H400 H208 K7900 K1
8	1	奇	9600	UINIT H400 HA K7900 K1
8	1	奇	19200	UINIT H400 H10A K7900 K1
8	1	奇	38400	UINIT H400 H20A K7900 K1
8	1	偶	9600	UINIT H400 HE K7900 K1
8	1	偶	19200	UINIT H400 H10E K7900 K1
8	1	偶	38400	UINIT H400 H20E K7900 K1



配置为主站
通信格式为9600，n，8，1

上述梯形图表示：将串口 1 配置为 MODBUS 主站，传输速率 9600 bps，无校验，数据长度 8 位,停止位 1 位。

4.2.3 从站通信协议的要求

- 从站通信端口支持 MODBUS协议RTU 模式，从站与主站的通信参数（如波特率、数据位、校验位、停止位）必须保持一致
- 从站相关软元件资源的 MODBUS 通信地址必须明确

4.2.4 任务调度配置[UINIT]

MC系列PLC使用 MODBUS 协议时，需要对通信协议调度区进行配置。

通信协议调度区起始地址由串口初始化指令 UINIT 的操作元 D1 指定，指定调度区 D 寄存器的起始号，调度区将占用从起始号开始的 7 个 D 寄存器，关于 UINIT 指令的详细使用说明请参见本章2.2节。

如 D1 = 7900，则 D7900 作为该串口的任务调度数据区被系统占用。

地址单元	说明	推荐初始化数值
D7900	modbus协议标志位状态字(定义下述)，用来观测。	由系统填写，可手动清零
D7901	从站正确返回数据的报文数	由系统填写，可手动清零
D7902	已缓存指令数（系统规划的缓存区最多缓存9条指令）	由系统填写，0
D7903	从站错误返回/超时的报文个数（无正确返回报文的报文数）	由系统填写，可手动清零
D7904	最后一个不能正常通讯的从站地址	由系统填写，可手动清零
D7905	主站发送延时（主站修改有效，支持在线修改）	用户自定义，0
D7906	从站回复延时（从站修改有效，不支持在线修改）	用户自定义，0

D7906的值修改后，须再次执行一次串口初始化指令 UINIT生效，故从站D7906赋值操作须在UINIT之前。

主站报文实际发送延时时间 = 3.5 byte时间（与波特率有关） + D7905×0.1ms

从站报文实际回复延时时间 = 3.5 byte时间（与波特率有关） + D7906×0.1ms

对于9600波特率来说，系统通信超时重试时间为48ms，重试次数为2，当有从站掉线时，每通信到掉线的从站时就会卡顿144ms。

对于19200波特率来说，系统通信超时重试时间为24ms，重试次数为2，当有从站掉线时，每通信到掉线的从站时就会卡顿72ms。

对于38400波特率来说，系统通信超时重试时间为12ms，重试次数为2，当有从站掉线时，每通信到掉线的从站时就会卡顿36ms。

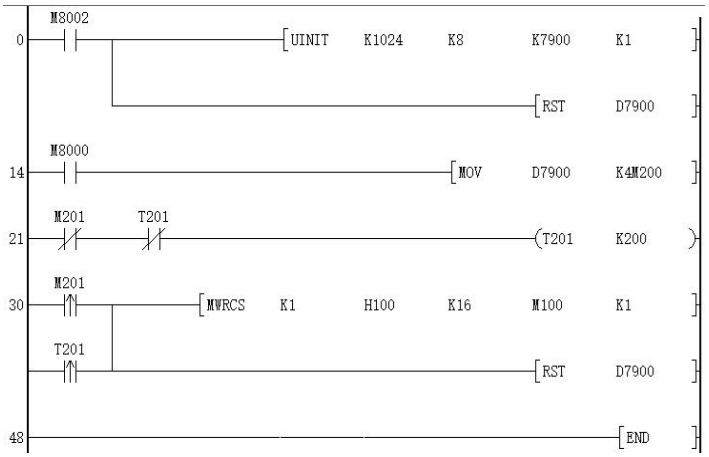
modbus协议标志位状态字

控制字(按位定义)	对应功能
Bit0	发送成功标志位 当一帧报文发送完成后，该位置 1，由用户程序复位。
Bit1	接收成功标志位 当一帧报文接收完成后，并且 CRC 检验正确，该位置 1，由用户程序复位。
Bit2	CRC 检验错误标志位 当一帧报文接收完成后，如果 CRC 检验错误，该位置 1，由用户程序复位。
Bit3	缓存区溢出标志 系统规划的缓存区最多缓存9条指令，缓存区存满后，再驱动了一条或多条modbus指令，缓存区溢出标志就会被置位。
Bit15～Bit4	保留使用

MC系列PLC底层规划了一个modbus指令缓存区，系统规划的缓存区最多缓存9条指令，每驱动一次modbus指令就会缓存一次驱动的指令，缓存区存满后，再驱动了一条或多条modbus指令，缓存区溢出标志就会被置位。缓存的指令会依次执行，每条指令执行完就退出缓存区。溢出的指令系统会自动舍弃不做处理。此功能可实现最多同时驱动9条modbus指令，使用方式参考参见本章4.6.2节modbus通信应用案例《4.6.2 三台MC系列控制器一主两从通信》。

一般情况下，如果用户的 MODBUS 主站程序采用定时方式与其它从站轮流通信，则这三个状态标志位只用于观测 MODBUS 通信链路的状态，不参与程序控制。如果用户需要快速与其它从站进行通信， 主站程序可以采用查询方式，当查询到接收成功标志位置 1 时，可启动下一次 MODBUS 通信，并且用户程序需要立即复位接收成功标志位；查询方式下，启动 MODBUS 指令后，用户程序同时启动一个定时器作为接收超时检测，在该时间点到达之后，如果接收成功标志位未置 1，用户程序直接启动下一次 MODBUS 通信。

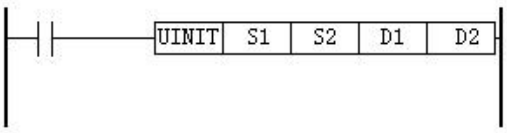
以下梯形图中，主站采用查询方式与从站进行 MODBUS 通信：



- 串口 1 初始化为 MODBUS 主站
- 通信状态寄存器清零
- 更新通信状态寄存器的标志
- 接收超时检测定时器计时
- 有两种情况启动 MODBUS 通信：接收成功标志位置 1；接收超时检测时间到，启动通信后， 立即将通信状态寄存器清零。

MC系列PLC MODBUS 从站程序中，这三个状态标志位只用于观测 MODBUS 通信链路的状态， 不参与程序控制。

为方便用户使用,当选择通信波特率为 9600 时，也可以按照以下表格的参数直接初始化串口。指令格式：



MODBUS 串口初始化配置表：

指令操作数	主站	从站	备注
S1	H400	H4XX	协议控制字，XX 表示从站地址，如配置 15 号从站时，S1 应为 H40F，十六进制
S2	K8	K8	通信控制字，十进制
D1	K7900	K7900	占用 D 寄存器资源的起始地址，一般选 D7900，也可以选其它地址
D2	K1	K1	串口号，一般选K1(串口 1)，也可以选 K0（串口 0）

4.3 MODBUS 从站配置

MODBUS从站只需要简单配置后，就可以对主站发送的符合 MODBUS RTU 格式的报文作出响应和处理。

1、支持Modbus从站协议的端口

MC系列PLC的所有串口（COM0/COM1）都支持 Modbus 从站协议。

2、作为Modbus的从站，用户程序只需将使用的通信端口配置为从站、配置本地站号就可以了，运行时MC系列PLC会接收解析主站或上位机的通信指令，并自动应答，无需用户程序干预。端口配置方法

通过串口初始化指令(UINIT)将串口 1 配置为 MODBUS 从站，才能与主站通信，关于 UINIT 指令的详细用法请参见 4.2.4 章节。

正常情况下，一般将串口 1 配置为 MODBUS 通信端口，串口 0 作为编程口使用，如果梯形图没有对两个串口进行配置，则都默认配置为编程口，在梯形图运行第一个扫描周期内，需对串口进行配置。特殊情况下的端口配置方法请参见 4.5.3 章节。

推荐参数配置：传输速率 9600 bps，无校验，数据长度 8 位,停止位 1 位。通信格式支持8个数据位、1/2个停止位、无校验/奇/偶检验，波特率支持9600/19200/38400，推荐使用9600。



上述梯形图表示：将串口 1 配置为 MODBUS 5 号从站，传输速率 9600bps，无校验，数据长度 8 位,停止位 1 位。

3、MC系列PLC作为 MODBUS 从站时软元件通信地址

主站或上位机利用 Modbus 协议访问MC系列PLC时，可以访问 D、R、T、C 等字变量，也可以访问 X/Y/M/T/C/S 等位变量。这里的“访问”是通信读、通信写等操作。上位机必需按照MC系列PLC的地址定义来发送读或写指令。

元件名称	MODBUS 地址（16进制）	元件名称	MODBUS 地址（16进制）
位软元件		字软元件	
M0～M3071	0200H～0DFFH	D0～D7999	0170H～20AFH
X000～X377	0000H～00FFH	D8000～D8255	20B0H～21AFH
Y000～Y377	0100H～01FFH	T0～T255	21B0H～22AFH
S0～S1023	0F00H～12FFH	C0～C199	22B0H～2377H
M8000～M8255	0E00H～0EFFH	C200～C255(32位)	2378H～23E7H
T0～T255(线圈)	1300H～13FFH	R0～R15999	23E8H～6267H
T0～T255(触点)	1400H～14FFH	采用寄存器读写命令对软元件 C200～C255 访问时，每个元件对应的寄存器数量为 2。	
C0～C255(线圈)	1500H～15FFH		
C0～C255(触点)	1600H～16FFH		

4.4 MODBUS 协议主站的指令应用方法

MC系列PLC Modbus 通讯传输模式为 MODBUS RTU 模式；它定义了报文域的位数据在线路上串行地传送；它确定了信息如何打包为报文和解码；Modbus 串行链路上所有设备的传输模式（包括串行口参数）必须相同。

4.4.1 RTU 模式

当设备使用 RTU（Remote Terminal Unit）模式在 Modbus 串行链路通信， 报文中每个 8 位字节含有两个 4 位十六进制字符。这种模式的主要优点是较高的数据密度，在相同的波特率下比 ASCII 模式有更高的吞吐率。每个报文必须以连续的字符流传送。

RTU 模式帧检验域：循环冗余校验（CRC）。

RTU 模式帧描述：

Modbus 站号	功能代码	数据	CRC	
1 字节	1 字节	0~252 字节	2 字节	
			CRC 低	CRC 高

格式：

START	保持无输入信号大于等于 10ms
Address	通信地址：8-bit 二进制地址
Function	功能码：8-bit 二进制
DATA（n - 1）	资料内容： N*8-bit 资料，N<=252，最大 252 个字节
……	
DATA 0	
CRC CHK Low	CRC 校验码低位
CRC CHK High	CRC 校验码高位
END	保持无输入信号大于等于 10ms

4.4.2 通信地址

00H：对所有科威系列PLC 广播（broadcast）——广播时候下位机不回复数据。

01H：对 01 地址 PLC 通信。

0FH：对 15 地址 PLC 通信。

10H：对 16 地址 PLC 通信。以此类推……，最大可到 247（F7H）。

4.4.3 功能码（Function ）

MODBUS 通信协议下，当主站向从站发送信息时，由功能代码向从站说明应执行的动作。如读一组离散式线圈或输入信号的 ON/OFF 状态，读一组寄存器的数据，写线圈（或寄存器）等。

功能码定义及MC系列PLC MODBUS 通信指令如下表所示：

功能码	通信指令	名称	功能	可操作软元件
01H	MRDC	读线圈	读一组逻辑线圈的当前状态	X Y M S T C
02H	MRDI	读输入线圈	读一组开关输入的当前状态	X Y M S T C
03H	MRDD	读寄存器	读一个或多个寄存器的当前值	D R T C
04H	MRDAI	读输入寄存器	读一个或多个输入寄存器的当前值	D R T C
05H	MWRC	写单个线圈	强置一个逻辑线圈的通断状态	Y M S T C
0FH	MWRCS	写多个线圈	强置一串连续逻辑线圈的通断	Y M S T C
06H	MWRD	写单个寄存器	强置一个特定的值到一个寄存器	D R T C

10H	MWRDS	写多个寄存器	强置一系列特定的值到一系列寄存器	D R T C

a）功能码 0x01（01）/0x02（02）：读线圈/读输入线圈

请求帧格式：从机地址+0x01/0x02+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x01/0x02（功能码）	1 个字节	读线圈/读输入线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈数量	2 个字节	高位在前，低位在后（N）
5	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x01/0x02+字节数+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x01/0x02（功能码）	1 个字节	读线圈/读输入线圈
3	字节数	1 个字节	值：[（N+7）/8]取整数，N 为线圈数量
4	线圈状态	[（N+7）/8] 个字节	每 8 个线圈合为一个字节，最后一个若不足 8 位，未定义部分填 0。前 8 个线圈在第一个字节，地址最小的线圈在最低位。依次类推
5	CRC 校验	2 个字节	低位在前，高位在后

注：N 最大不超过 256

错误响应：见错误响应说明

b） 功能码 0x03（03）/0x04（04）：读寄存器/读输入寄存器

请求帧格式：从机地址+0x03/0x04+寄存器起始地址+寄存器数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x03/0x04（功能码）	1 个字节	读寄存器/读输入寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后（N）
5	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x03/0x04+字节数+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x03/0x04（功能码）	1 个字节	读寄存器/读输入寄存器
3	字节数	1 个字节	值：N*2
4	寄存器值	N*2 个字节	两字节表示一个寄存器值，高位在前，低位在后。寄存器地址小的排在前面
5	CRC 校验	2 个字节	低位在前，高位在后

注：N 最大不超过 64

错误响应：见错误响应说明

c) 功能码 0x05（05） ：写单线圈

请求帧格式：从机地址+0x05+线圈地址+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x05（功能码）	1 个字节	写单线圈
3	线圈地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈状态	2 个字节	0xff00，表示 1；0x0000，表示 0。高位在前，低位在后
5	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x05+线圈地址+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x05（功能码）	1 个字节	写单线圈
3	线圈地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈状态	2 个字节	0xff00，表示 1；0x0000，表示 0。高位在前，低位在后
5	CRC 校验	2 个字节	低位在前，高位在后

错误响应：见错误响应说明

d) 功能码 0x06（06）：写单个寄存器

请求帧格式：从机地址+0x06+寄存器地址+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x06（功能码）	1 个字节	写单寄存器
3	寄存器地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器值	2 个字节	高位在前，低位在后
5	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x06+寄存器地址+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x06（功能码）	1 个字节	写单寄存器
3	寄存器地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器值	2 个字节	高位在前，低位在后
5	CRC 校验	2 个字节	低位在前，高位在后

错误响应：见错误响应说明

e) 功能码 0x0f（15）：写多个线圈

请求帧格式：从机地址+0x0f+线圈起始地址+线圈数量+字节数+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x0f（功能码）	1 个字节	写多个单线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址

4	线圈数量	2 个字节	高位在前，低位在后。N，最大为 256
5	字节数	1 个字节	值：[（N+7）/8]取整
6	线圈状态	[（N+7）/8]个字节	每 8 个线圈合为一个字节，最后一个若不足 8 位，未定义部分填 0。前 8 个线圈在第一个字节，地址最小的线圈在最低位。依次类推
7	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x0f+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x0f（功能码）	1 个字节	写多个单线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈数量	2 个字节	高位在前，低位在后。
5	CRC 校验	2 个字节	低位在前，高位在后

错误响应：见错误响应说明

f） 功能码 0x10（16）：写多个寄存器

请求帧格式：从机地址+0x10+寄存器起始地址+寄存器数量+字节数+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x10（功能码）	1 个字节	写多个寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后。N，最大为 64
5	字节数	1 个字节	值：N*2
6	寄存器值	N*2	
7	CRC 校验	2 个字节	低位在前，高位在后

响应帧格式：从机地址+0x10+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247
2	0x10（功能码）	1 个字节	写多个寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后。N，最大为 64
5	CRC 校验	2 个字节	低位在前，高位在后

错误响应：见错误响应说明

g）错误响应说明

PC 通过串口助手向MC系列PLC发送异常报文，下划线标识处为异常参数。

①功能码错误：01 11 02 00 FF 00 BD 81，MC系列PLC不识别“11”的功能码。

②非法变量地址：01 03 01 69 00 02 15 EB，读 D 寄存器的范围为 0x0170~0x20af

③错误数据：01 05 02 00 01 01 0C 22，数据应为 0xff00 或 0x0000

④变量个数越界：01 01 02 00 01 01 FD E2, 读位变量的最大个数不能超过 256

⑤CRC 校验错误：01 05 02 00 FF 00 82 8D ,检验码应为 8D 82

对于各种异常报文，MC系列PLC作为从站时的响应情况如下表所示，表格中还包括 MODBUS 通信状态寄存器中三个标志位的状态（具体说明请参见 4.2.4 章节），分别是 D0 位（报文发送成功标志）、D1 位（报文接收成功标志）、D2 位（CRC 检验错误标志）。

错误报文类型	PC 发送报文内容	MC系列PLC返回报文	MODBUS 状态寄存器
功能码错误	01 11 02 00 FF 00 BD 81	无	d2d1d0=011b
非法变量地址	01 03 01 69 00 02 15 EB	01 03 04 00 00 00 00 FA 33	d2d1d0=011b
错误数据	01 05 02 00 01 01 0C 22	01 05 02 00 01 01 0C 22 实测 M0 被复位	d2d1d0=011b
错误数据	01 05 02 00 FF 01 4C 42	01 05 02 00 FF 01 4C 42 实测 M0 被置位	d2d1d0=011b
变量个数越界	01 01 02 00 01 01 FD E2	可读取数据，但越界的数据不正确	d2d1d0=011b
CRC 校验错误	01 05 02 00 FF 00 82 8D	无	d2d1d0=100b

从

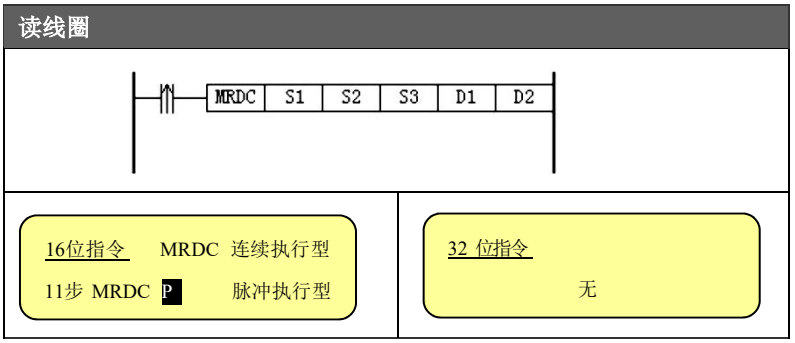
上表发现，MC系列PLC作为从站时，对于非法变量地址、错误数据、变量个数越界等类型的错误报文，并没有进行纠错处理，对于功能码错误和 CRC 检验错误的报文有纠错处理。

4.4.4 MODBUS 指令

本小节介绍MC系列PLC支持的 MODBUS 指令的功能及作用。

4.4.4.1. 读线圈[MRDC]，读输入线圈[MRDI]

读取远程设备一个或多个线圈/输入开关量的当前状态。



	作用	类型
S1	从站站号	16 位BIN
S2	从站软元件首地址	16 位BIN
S3	通信元件个数	16 位BIN
D1	主站软元件首地址编号	位
D2	通信串口号	16 位BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

读取远程设备一个或多个线圈/输入开关量的状态，建立远程设备与本地设备之间的读取数据通信功能。

《读线圈》



将站号 1 远程设备地址 0 开始的100 个位读入本地设备的M0～M99，串口号为 1。

《读输入线圈》



将站号 1 远程设备地址 0 开始的 10 个位读入本地设备的Y000～Y007、Y010～Y011，串口号为 1。

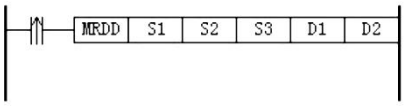
使用注意事项

- 读线圈与读输入线圈两条指令的处理机制相同。
- MC系列PLC做从站时，对于软元件通信个数 S3 的范围限定：
  - ◆ 软元件每次通信个数不超过 256 个。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件的 MODBUS 地址。

4.4.4.2. 读寄存器[MRDD]，读输入寄存器[MRDAI]

读取远程设备中一个或多个寄存器/输入寄存器的当前值。

读寄存器



16位指令 MRDD 连续执行型

11步 MRDD P 脉冲执行型

32 位指令

无

操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件首地址	16 位 BIN
S3	通信元件个数	16 位 BIN
D1	主站软元件首地址编号	16 位 BIN
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

读取远程设备一个或多个寄存器/输入寄存器当前值，建立远程设备与本地设备之间的读取数据通信功能。

《读寄存器》



将站号 10 远程设备地址 1000 开始的10 个字读入本地设备的D1000～D1009，串口号为 1。

《读输入寄存器》



将站号 10 远程设备地址 2000 开始的10 个字读入本地设备的D2000～D2009，串口号为 1。

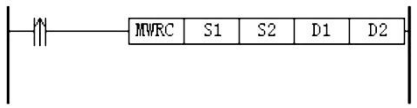
使用注意事项

- 读寄存器与读输入寄存器两条指令的处理机制相同。
- MC系列PLC做从站时，对于软元件通信个数 S3 的范围限定不超过 64 个。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件的 MODBUS 地址。

4.4.4.3. 写线圈[MWRC]

强置远程设备一个逻辑线圈的通断状态。

写线圈



16位指令	MWRC 连续执行型	32 位指令
9步 MWRC	P 脉冲执行型	无

操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件地址	16 位 BIN
D1	主站软元件编号	位
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

强置远程设备一个逻辑线圈的通断状态。



将 M0 软元件状态送入站号 5 远程设备地址 300 内，串口号为 1。

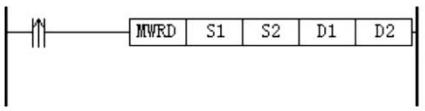
使用注意事项

- 确认从站的地址 S2 是一个可写入的位地址。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 主站软元件编号 D1 不能使用变址修饰。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件 MODBUS 地址。

4.4.4.4. 写寄存器[MWRD]

改变远程设备一个字地址的当前数值。

写寄存器



16位指令	MWRD 连续执行型	32 位指令	
9步 MWRD	P 脉冲执行型		无

操作数	作用	类型
S1	从站站号	16 位BIN
S2	从站软元件地址	16 位BIN
D1	主站软元件编号	16 位BIN
D2	通信串口号	16 位BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

改变远程设备一个字地址的当前数值。



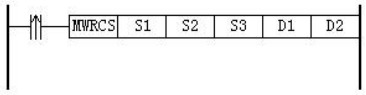
将 D2000 软元件的内容送入站号 5 远程设备地址 2000 内，串口号为 1。

使用注意事项

- 确认从站的地址 S2 是一个可写入的字地址。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 主站软元件编号 D1 不能使用变址修饰。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件的 MODBUS 地址。

4.4.4.5. 写多个线圈[MWRCS]

强置远程设备一个或多个逻辑线圈的通断状态。

写多个线圈	
	
16位指令 MWRCS 连续执行型 11步 MWRCS P 脉冲执行型	32 位指令 无

操作数	作用	类型
S1	从站站号	16 位BIN
S2	从站软元件地址	16 位BIN
S3	通信元件个数	16 位BIN
D1	主站软元件编号	位
D2	通信串口号	16 位BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

强置远程设备一串连续逻辑线圈的通断，建立远程设备与本地设备之间的读取数据通信功能。



将 M0～M4 软元件状态送入站号 1 远程设备地址 300～304 内，串口号为 1。

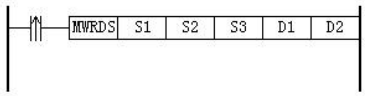
使用注意事项

- 确认从站的地址 S2 是一个可写入的位地址。
- MC系列PLC做从站时，对于软元件通信个数 S3 的范围限定：
  - ◆ 软元件每次通信个数不超过 256 个。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触 发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件的 MODBUS 地址。

4.4.4.6. 写多个寄存器[MWRDS]

改变远程设备一个或多个寄存器的当前数值。

写多个寄存器



16 位指令	MWRDS	连续执行型
11 步	P	脉冲执行型

32 位指令：无

操作数	作用	类型
S1	从站站号	16 位BIN
S2	从站软元件地址	16 位BIN
S3	通信元件个数	16 位BIN
D1	主站软元件编号	16 位BIN
D2	通信串口号	16 位BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

赋一系列特定的值到一系列寄存器，建立远程设备与本地设备之间的读取数据通信功能。



将 D1000～D1004 软元件的内容送入站号 1 远程设备地址 2000～2004 内，串口号为 1。

使用注意事项

- 确认从站的地址 S2 是一个可写入的字地址。
- MC系列PLC做从站时，对于软元件通信个数 S3 的范围限定不超过 64 个。
- MC系列PLC做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表MC系列PLC的软元件的 MODBUS 地址。

4.5 MODBUS 应用注意事项

4.5.1 指令使用规范

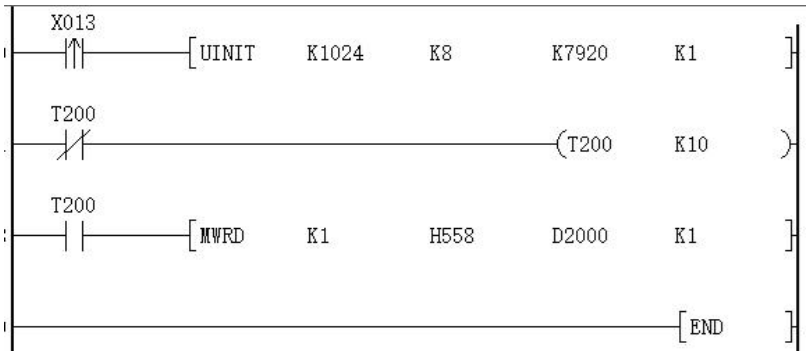
- MODBUS 指令的输入触点必须是边沿触发类触点或单次导通类触点，启动下一条 MODBUS 指令或再次启动当前的 MODBUS 指令之前，需查询 MODBUS 接收成功标志位的状态，该状态位为 1 时，才能启动下一次 MODBUS 指令，用户程序需要将该标志位立即复位。
- 不能超出 MODBUS 指令各操作数可寻址的变量范围
- 使用 MODBUS 指令读写多个线圈或多个寄存器时，必须保证指令操作数中主站和从站的软元件寻址的最大地址不能超过相对应的地址范围。如：主站把 10 个 D 寄存器的数据写到从站 D7995 开始的 10 个单元中，如果从站 D 寄存器的最大范围为D0~D7999，此时主站写从站第 5 个D 寄存器时对应的地址是D7999， 以后的写地址全部是无效地址。

4.5.2 与其他流程指令的相互关系

在 MC-MCR、CJ-P、FOR-NEXT、STL-RET、P-SRET(子程序)、I-IRET（普通中断）、I-PRET（高级中断）等流程指令中，都可以使用 MODBUS 指令。

4.5.3 特殊情况下的端口配置方法

1) 一般情况下，用户程序中没有使用 UINIT 指令对串口 0 和串口 1 进行初始化时，两个串口都默认为编程口协议，串口 1 必须通过 RS485 转RS232 转换器转换后才能与电脑或其它设备的RS232 接口连接， 实现编程口协议监控功能。串口 0 可通过科威 PLC 编程电缆直接与电脑或其它设备的 RS232 接口连接，实现编程口协议监控功能。
2) 一般情况下，使用 MODBUS 主站或从站协议时，串口 0 不进行初始化，保留为编程口使用，串口 1 在梯形图上电运行第一个周期内初始化为 MODBUS 协议。
3) 特殊情况下，如果用户不希望用串口 0 下载程序，串口 1 也可以复用为编程口和MODBUS 通信口， 此时用户程序中只能在梯形图上电运行第一个周期之后将串口 1 初始化为 MODBUS 主站或从站，建议使用某一个输入X 的上升沿作为MODBUS 初始化指令的输入条件，MC系列PLC上电运行后如果该输入点处于接通状态，则串口 1 作为 MODBUS 通信口使用，如果需要把串口 1 作为编程口使用，则必须将串口 1 与其它设备的通信线断开并接上 RS485/RS232 转换器，再将该输入点断开并重新上电。



• 上电后，串口 1 默认为编程口，X13 的上升沿将串口 1 初始化为 MODBUS 主站，如果串口 0 接触摸屏，也可以在屏的画面指定一个M 点的上升沿作为串口 1 初始化的输入条件。

4) 复杂的应用系统中，串口 0 和串口 1 都可以作为 MODBUS 通信口使用，分别与其它通信设备一起组成两个独立的 MODBUS 通信链路，串口 1 在梯形图上电运行第一个周期内初始化为 MODBUS 协议，必须保证串口 0 可以复用为编程口和 MODBUS 通信口，方法与上述第（3）项对串口 1 的描述相同。串口 0 作为 MODBUS 通信口使用时必须接 RS232/RS485 转换器。串口 0 作为编程口使用时必须将与之连接的 RS232/RS485 转换器断开。用户程序对串口 0 和串口 1 进行初始化时，通信协议调度区寄存器要指向不同的 D 寄存器。

4.5.4 MC系列PLC作为 MODBUS 从站的响应时延

MC系列PLC作为 MODBUS 从站时，从主站每一帧报文发送完成的时刻，到MC系列PLC经过内部报文处理后向主站返回一帧报文的起始时刻，这段时间称为从站的响应时延，其时间间隔不超过 10ms。

4.5.5 MC系列PLC作为 MODBUS 主站时，对其他从站的响应时间要求

MC系列PLC作为 MODBUS 主站时，对其他从站的响应时间根据不同波特率最大为 50ms,如果超过这个时间从站没有返回报文，LP3系列PLC将重发本次报文，重发次数最多两次。

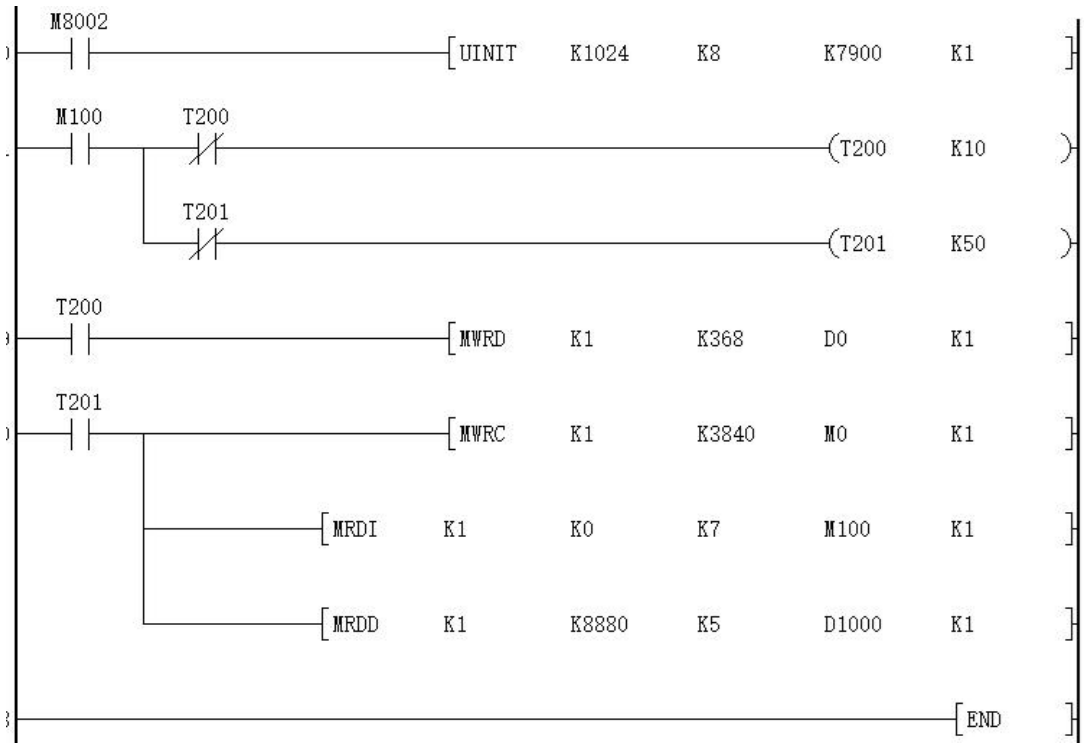
4.6 MODBUS 通信应用案例

4.6.1 两台MC PLC一主一从通信分级别通信

可将其中任意一台或多台设备换成MC系列PLC，无需修改程序。

按照各数据所需的刷新频率，将Modbus指令用定时方式分成多个级别通信（下例中分成两个等级）。

主站程序如下：



上电使用串口初始化指令配置通信格式：Modbus 主站；9600bps、无奇偶校验、8 数据位、1 停止位；通信缓存区首地址为D7900；COM1口。

T200计时器每100ms置位一次，即程序每100ms执行以下Modbus指令：

①将主站PLC D0的数据写入从站PLC的D0中

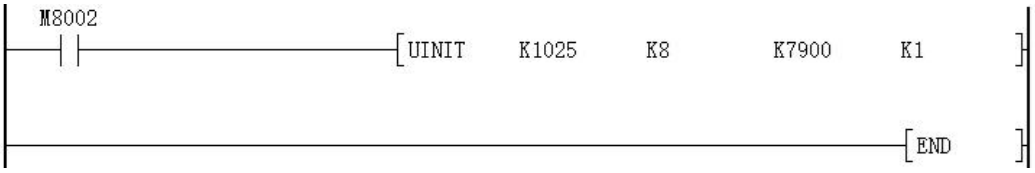
T201计时器每500ms置位一次，即程序每500ms执行以下3条Modbus指令：

②将主站PLC M0的状态写入从站PLC的S0中；

③将从站PLC X0-X7的状态读入主站PLC的M100-M107中；

④将从站PLC C0-C4的数据读入主站PLC的D1000-D1004中；

从站程序如下：



上电使用串口初始化指令配置通信格式：Modbus 1#从站；9600bps、无奇偶校验、8 数据位、1 停止位；通信缓存区首地址为D7900；COM1口。

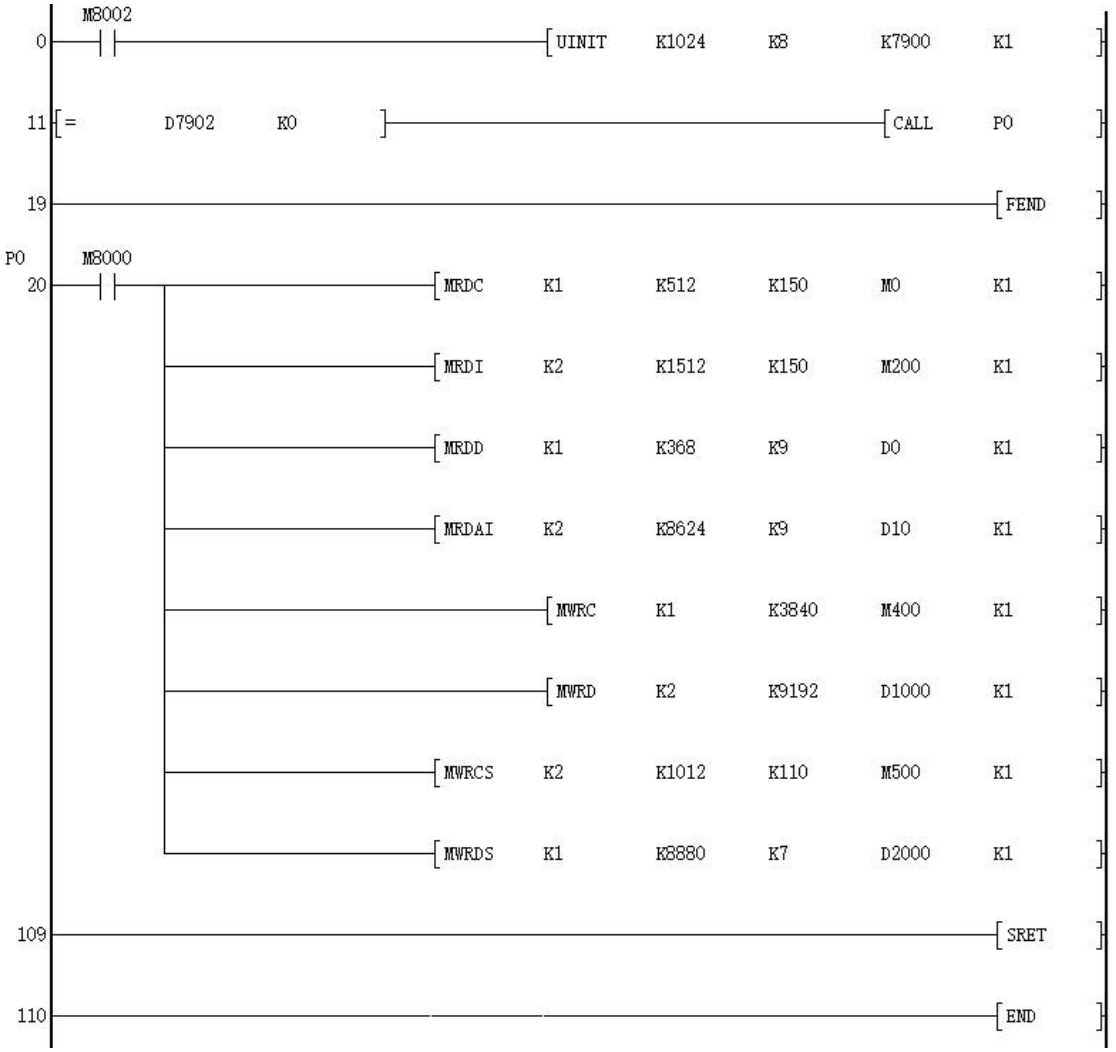
上例中，置位M100后，每经过500ms，Modbus指令①执行5次，Modbus指令②③④分别执行1次。在梯形图扫描周期影响不大的前提下，长时间运行，四条执行执行次数比例为5:1:1:1，从而达到分别通信的目的。

4.6.2 三台MC PLC一主两从通信

可将其中任意一台或多台设备换成MC系列PLC，无需修改程序。

实现多条指令（最多9条）长时间均匀排布通信。

主站程序如下：



上电使用串口初始化指令配置通信格式：Modbus 主站；9600bps、无奇偶校验、8 数据位、1 停止位；通信缓存区首地址为D7900；COM1口。

当缓存区Modbus指令数为0时，跳转至子程序P0执行8条Modbus指令后返回主程序，其中8条指令注释如下：

将1#从站PLC M0-M149的状态读入主站PLC的M0-M149中；

将2#从站PLC M1000-M1149的状态读入主站PLC的M200-M349中；

将1#从站PLC D0-D9的数据读入主站PLC的D0-D9中；

将2#从站PLC T0-T9的数据读入主站PLC的D10-D19中；

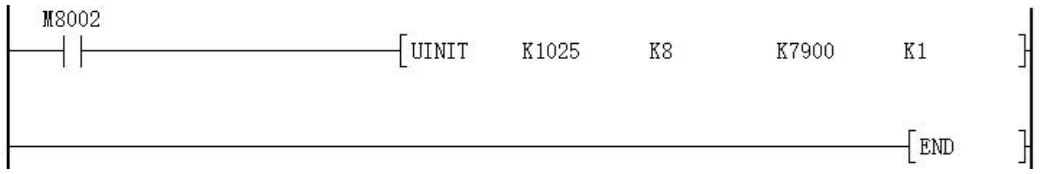
将主站PLC M400的状态写入1#从站PLC的S0中；

将主站PLC D1000的数据写入2#从站PLC的R0中；

将主站PLC M500-M609的状态写入2#从站PLC的M500-M609中；

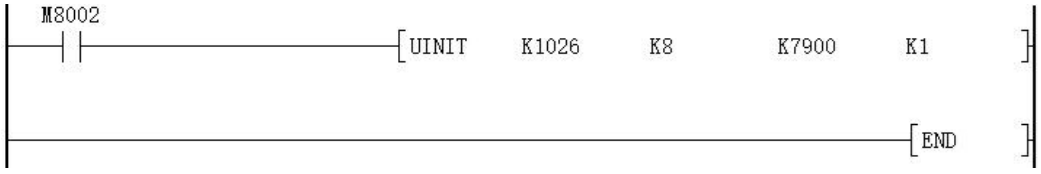
将主站PLC D2000-D2006的数据写入1#从站PLC的C0-C6中；

1#从站程序如下：



上电使用串口初始化指令配置通信格式：Modbus 1#从站；9600bps、无奇偶校验、8 数据位、1 停止位；通信缓存区首地址为D7900；COM1口。

2#从站程序如下：



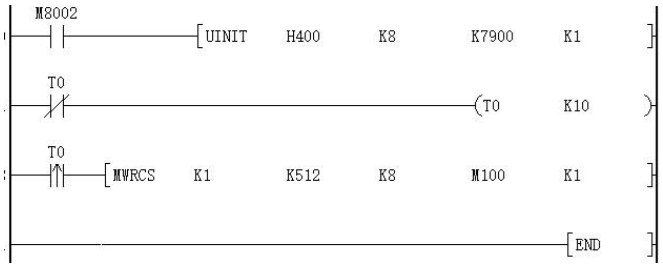
上电使用串口初始化指令配置通信格式：Modbus 2#从站；9600bps、无奇偶校验、8 数据位、1 停止位；通信缓存区首地址为D7900；COM1口。

4.6.3 两台MC系列PLC进行一主一从通信（定时通信方式）

定时通信方式：主站每隔一个固定时间与从站通信一次

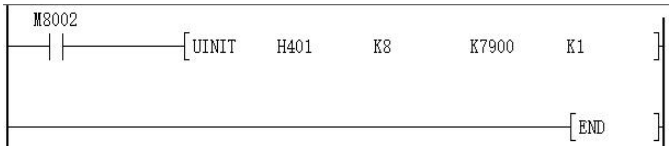
示例中所选择的通信端口及格式为：COM1、9600bps、无奇偶校验、8 数据位、1 停止位。

主站梯形图：



- 串口 1 配置为主站
- 每隔 1 秒钟 T0 触点吸合 1 次执行写多个线圈指令
- 梯形图扫描周期结束

从站梯形图：



- 串口 1 配置为 1 号从站
- 梯形图扫描周期结束

梯形图解释：

- 此梯形图的作用是将主站的 M100～M107 的状态送到从站 M0～M7 内，每隔 1 秒钟传送 1 次，1 次传送 8 个位状态。
- 从站位软元件的 K512 地址指向的是 M0。

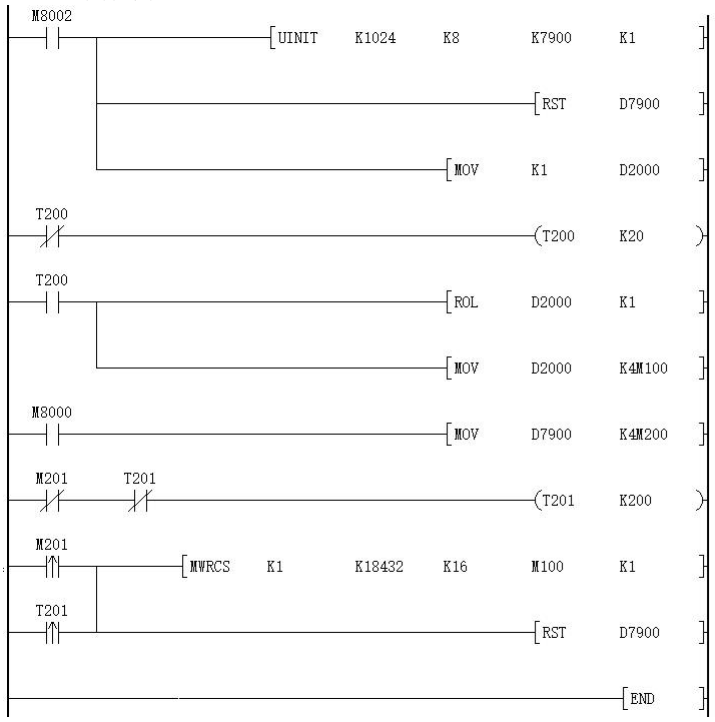
4.6.4 MC系列PLC与LP2 系列 PLC 进行一主一从通信（查询通信方式）

查询通信方式：上电后，主站初次启动与从站通信一次，后续过程中，每当接收报文成功标志位置 1 时，启动与从站通信。

主站：MC系列PLC

从站：LP2-14M12T

主站梯形图：

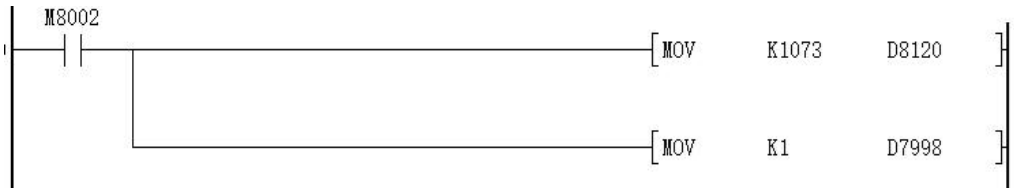


- 串口 1 配置为主站
- 通信状态寄存器清零
- 待发送数据初始化
- 每隔 200ms 待发送数据变化一次
- 更新通信状态寄存器的三个标志位

接收超时检测定时器计时

- 有两种情况启动 MODBUS 通信：接收成功标志位置 1；接收超时检测时间到，启动通信后， 立即将通信状态寄存器清零。

从站梯形图：



梯形图解释

- 此梯形图的作用是将主站的 M100～M115 的状态送到从站 Y0～Y17 内，发送间隔是相邻两次成功接收报文的时间。
- MC主站通过串口初始化指令将串口 1 配置为MODBUS(站号为 0 即代表主站)。
- LP2从站通过配置D8120和D7998，将串口 1 配置为MODBUS 从站，从站号设置为 1。
- LP2从站位软元件的 K18432 地址指向的是 Y0。

4.6.5 MC系列PLC与众辰TZ2400 变频器MODBUS 通信示例

1) 硬件接线：MC系列PLC的 RS485 端口，RSA 端口接变频器的 RS+端口，RSB 端口接变频器的 RS-端口。

2) 主站：MC系列PLC

从站：众辰变频器（众辰 TZ2400 系列矢量型变频器）

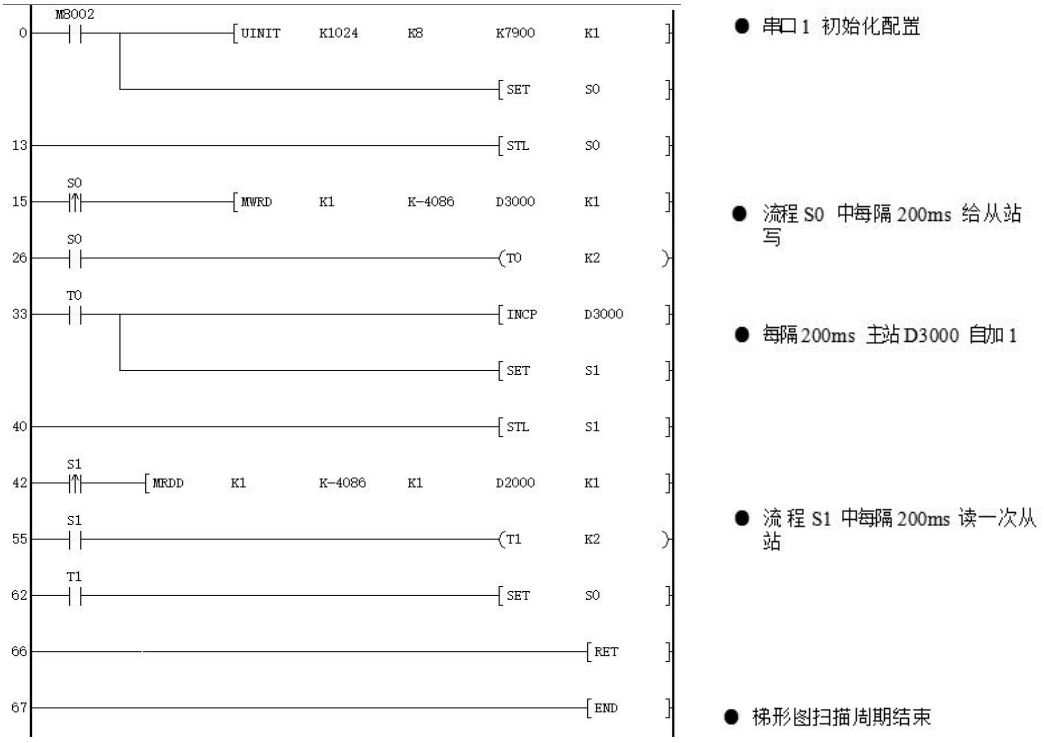
3) 从站 MODBUS 通信采用默认配置

参数	设置值	功能
PD.00	5	波特率为 9600BPS
PD.01	3	数据格式为：无校验、8 个数据位、1 个停止位
PD.02	1	本机地址设置为 1
PD.03	2	应答延时 2ms
PD.04	0	通信超时时间为 0
PD.05	1	选择标准的MODBUS 协议
PD.06	0	通信读取电流分辨率设置为 0.01A

注：从站 MODBUS 通信参数值其他设置请参考众辰变频器手册。

4) 主站使用串口初始化指令，将主站串口 1 配置成 MODBUS 通信协议，通信端口及格式为：COM1、9600bps、无奇偶校验、8 数据位、1 停止位。

5) 主站写入和读取从站预置频率示例主站梯形图：



注：K-4086 为变频器中预置频率的地址（变频器手册中该地址以十六进制表示为 HF00A）。

梯形图解释

- 此梯形图的作用是流程S0 中寄存器 D3000 中的数据每隔 200ms 加1，然后将数据写入到从站预置频率的地址中，修改从站的预置频率；在流程S1 中，每隔 200ms 读一次从站的预置频率，然后将数据存入到寄存器D2000 中。
- 主站通过串口初始化指令将串口 1 配置为MODBUS(站号为 0 即代表主站)。
- 变频器MODBUS 站号设置为 1。

使用注意事项

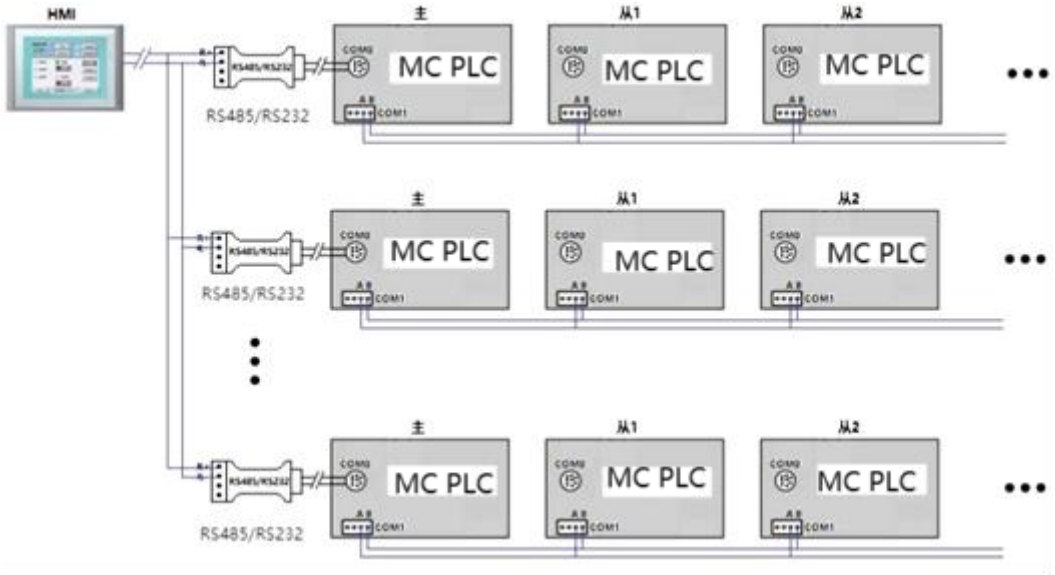
- 众辰变频器的软元件地址在手册中是用十六进制表示的，地址计算方法变频器手册中有说明。在主站 MODBUS 指令中，从站软元件地址可用十六进制表示，也可将十六进制转化为十进制进行表示。推荐用十六进制地址表示。
- 主站和从站的 MODBUS 通信格式必须保持一致，否则无法通信。
- 主站和从站硬件端口接线必须正确，否则无法通信。

4.6.6 使用MC系列 PLC组建两级 MODBUS 通信网络

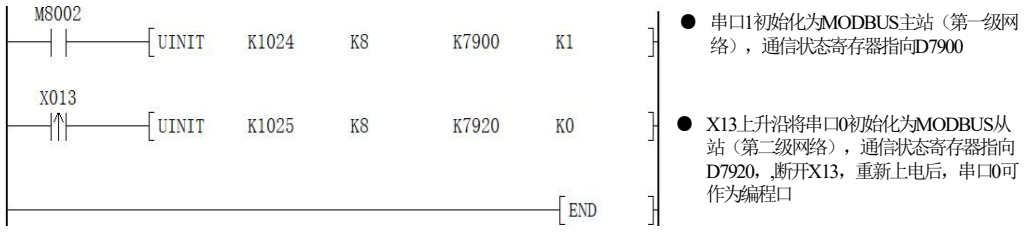
串口 1 配置为 MODBUS 主站，与其它从站组成第一级 MODBUS 网络；

串口 0 配置为 MODBUS 从站，与其它从站及主站一起组成第二级网络；

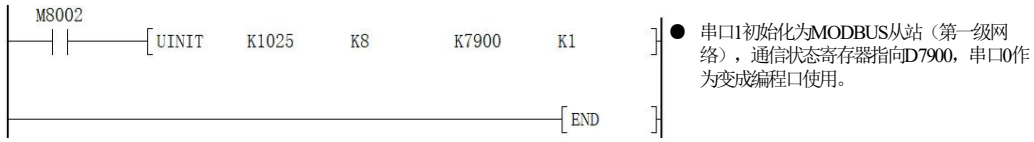
串口 0 也可以配置成第二级网络的主站，其它设备做从站。



主站串口初始化梯形图：



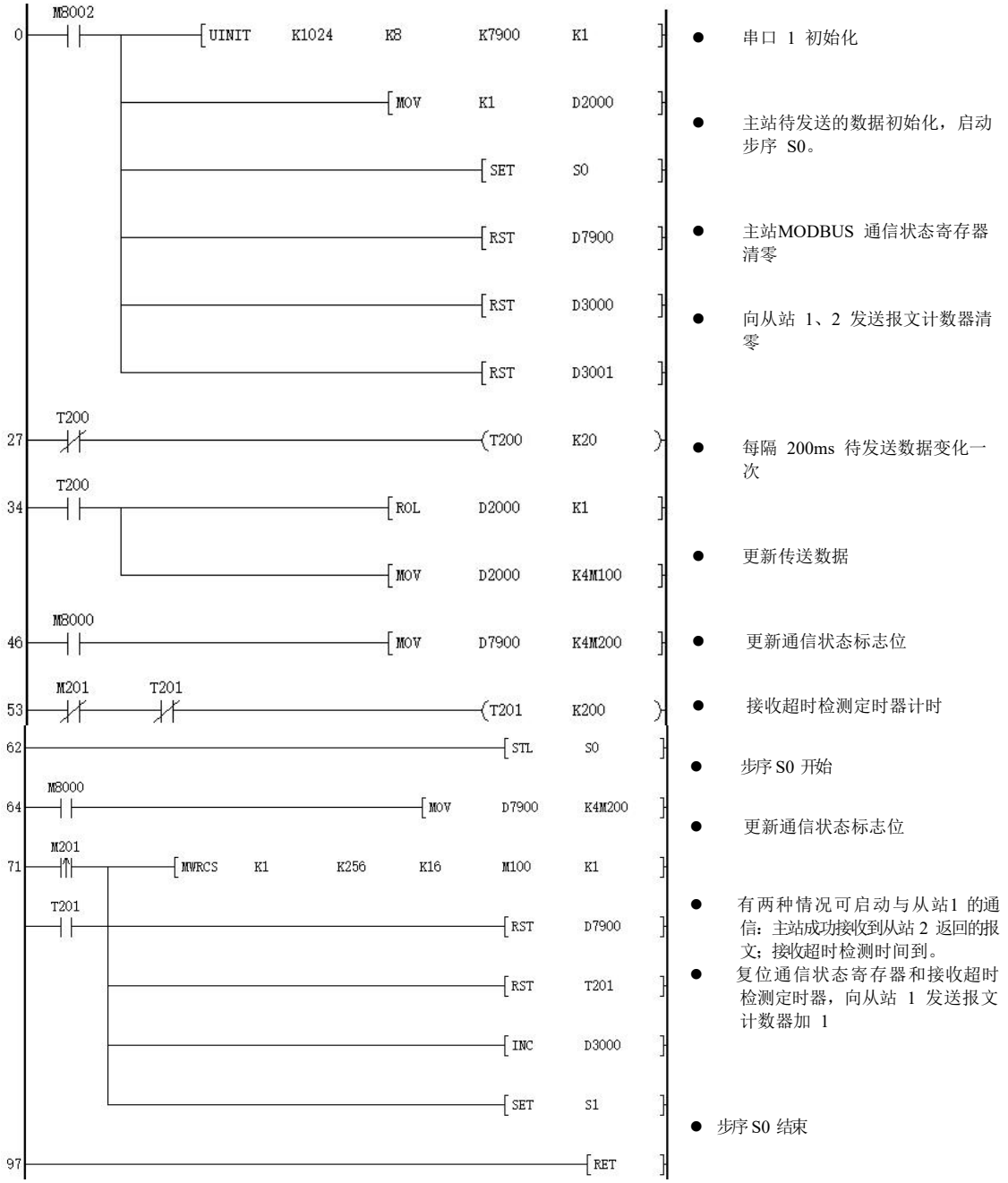
从站串口初始化梯形图：

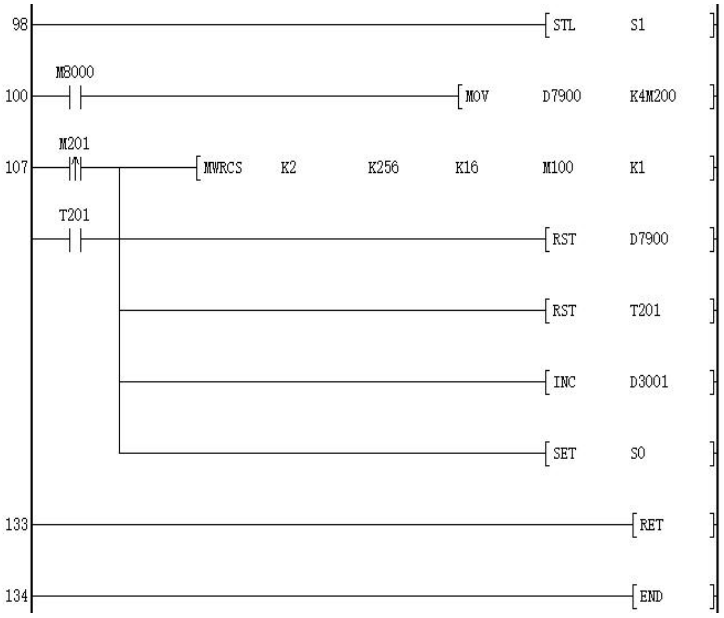


4.6.7 三台MC系列PLC组成多机 MODBUS 通信

功能说明：一台MC系列PLC做主站，使用 MODBUS 接收成功标志位（请参见 4.2.4 章 节 MODBUS 通信状态寄存器说明）启动 MODBUS 通信指令，在两个步进阶梯指令中分别将主站数据写到从站对应的Y 输出点，正常通信时，可以观测到两个从站的输出指示灯会跟随主站发送的数据从 Y0 到Y17 快速地循环右移。当其中一个从站断电后，另外一个从站也可以与主站保持通信，此时通信速度变慢， 从站指示灯变化不连续。

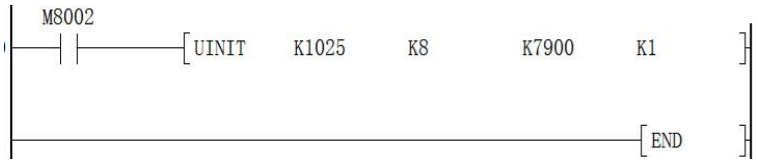
主站梯形图：





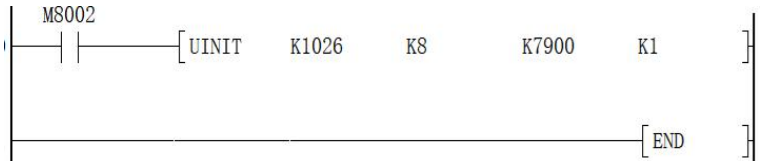
- 步序 S1 开始
- 更新通信状态标志位
- 有两种情况可启动与从站 2 的通信：主站成功接收到从站1 返回的报文； 接收超时检测时间到。
- 复位通信状态寄存器和接收超时检测定时器，
- 向从站2 发送报文计数器加 1 。
- 返回步序 S0，开始下一轮通信。
- 步序 S1 结束

从站 1 梯形图：



- 串口 1 初始化，从站地址为1

从站 2 梯形图：



- 串口 1 初始化，从站地址为2

5. 自由协议

5.1 协议介绍

控制器用的通信协议，尽管有标准协议可供执行，但在有些场合为了追求实时高效，有些产品在协议层并不遵循标准协议，此时按照其要求用梯形图编写通信程序，以满足各种通信情况的需要。由于自由协议的本质是发送和接收报文，就可以用自由协议对其报文格式和收发时序进行模拟。如此可通讯上第三方设备，对系统的组建起到关键的作用。

自由协议的主从信息地址信息是隐含在编程中，主从信息是在编程时设置主动发送信息还是被动接收信息， 地址信息包含在规划的报文格式中及解读报文的编程中。

5.2 协议格式

5.2.1 通信格式

自由协议支持的格式如下：

通信格式				梯形图设置
数据长度	停止位	奇偶校验	波特率	
8	2	无	9600	UINIT H500 H0 K7900 K1
8	2	无	19200	UINIT H500 H100 K7900 K1
8	2	无	38400	UINIT H500 H200 K7900 K1
8	2	奇	9600	UINIT H500 H2 K7900 K1
8	2	奇	19200	UINIT H500 H102 K7900 K1
8	2	奇	38400	UINIT H500 H202 K7900 K1
7	2	奇	9600	UINIT H500 H3 K7900 K1
7	2	奇	19200	UINIT H500 H103 K7900 K1
7	2	奇	38400	UINIT H500 H203 K7900 K1
8	2	偶	9600	UINIT H500 H6 K7900 K1
8	2	偶	19200	UINIT H500 H106 K7900 K1
8	2	偶	38400	UINIT H500 H206 K7900 K1
7	2	偶	9600	UINIT H500 H7 K7900 K1
7	2	偶	19200	UINIT H500 H107 K7900 K1
7	2	偶	38400	UINIT H500 H207 K7900 K1

8	1	无	9600	UINIT	H500	H8	K7900	K1	
8	1	无	19200	UINIT	H500	H108	K7900	K1	
8	1	无	38400	UINIT	H500	H208	K7900	K1	
8	1	奇	9600	UINIT	H500	HA	K7900	K1	
8	1	奇	19200	UINIT	H500	H10A	K7900	K1	
8	1	奇	38400	UINIT	H500	H20A	K7900	K1	
7	1	奇	9600	UINIT	H500	HB	K7900	K1	
7	1	奇	19200	UINIT	H500	H10B	K7900	K1	
7	1	奇	38400	UINIT	H500	H20B	K7900	K1	
8	1	偶	9600	UINIT	H500	HE	K7900	K1	
8	1	偶	19200	UINIT	H500	H10E	K7900	K1	
8	1	偶	38400	UINIT	H500	H20E	K7900	K1	
7	1	偶	9600	UINIT	H500	HF	K7900	K1	
7	1	偶	19200	UINIT	H500	H10F	K7900	K1	
7	1	偶	38400	UINIT	H500	H20F	K7900	K1	

注意：自由协议无主从站区分，推荐使用通信参数 9600 8 1 N。

5.2.2 自由协议任务调度

MC系列PLC配置为自由协议时，需要对通信协议调度区进行配置。

通信协议调度区起始地址由串口初始化指令 UINIT 的操作元 D1 指定，指定调度区 D 寄存器的起始号，调度区将占用从起始号开始的 20 个D 寄存器。

如 D1 = 7900，则 D7900～D7919 为该串口的任务调度数据区

地址单元	说明	推荐初始化数值
D7900	发送起始单元	激活发送指令 USEND 后由系统填写
D7901	待发送字节数	激活发送指令 USEND 后由系统填写
D7902	接收起始单元(发送自动转接收或手动置位接收请求标志位的情况下由用户自定义)	● 激活接收指令 URCV 后由系统填写 ● 用户自定义
D7903	待接收字节数(发送自动转接收或手动置位接收请求标志位的情况下由用户自定义)	● 激活接收指令 URCV 后由系统填写 ● 用户自定义，1～255
D7904	接收起始符，标示报文数据的开始，可视为报文的帧头	用户自定义，0，用 M8000 给定
D7905	接收终止符，标示报文数据的终止，可视为报文的帧尾	用户自定义，0，用 M8000 给定
D7906	接收等待间隔时间，用来判断接收报文结束.	用户自定义，单位毫秒,推荐 10
D7907	发送接收控制字(定义下述)，可修改。	用户自定义，1
D7908	自由协议标志位状态字(定义下述)，用来观测。	由系统填写，
D7909	实际已接收字节数	由系统填写,需手动清零
D7910～D7919	保留使用	0

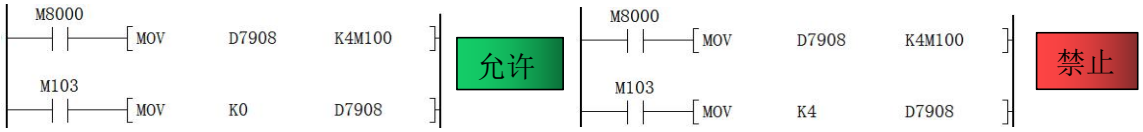
发送接收控制字

控制字(按位定义)	对应功能
Bit0	发送自动转接收 0：保持 1：自动转接收
Bit1	接收起始字符过滤 0：不过滤 1：过滤 设置过滤，接收方将过滤掉发送过来的接收起始符之前的所有内容。
Bit2	接收终止字符过滤 0：不过滤 1：过滤 设置过滤，接收方将过滤掉发送过来的接收终止符之后的所有内容。
Bit15～Bit3	保留使用

自由协议标志位状态字

控制字(按位定义)	对应功能
Bit0	发送请求标志 当执行发送指令 USEND 后被置位
Bit1	发送成功标志 当发送指令 USEND 执行完成后被置位
Bit2	接收请求标志 ● 当发送指令 USEND 执行完成后，且设置了发送转接收功能时被置位 ● 当接收指令 URCV 被激活时被置位
Bit3	接收成功标志 ● 当接收指令 URCV 执行完成后被置位 ● 当接收指令 URCV 被激活时被复位 ● 状态由系统给出，手动修改无效
Bit4	接收起始符标志 当接收到接收起始符后被置位
Bit5	接收终止符标志 当接收到接收终止符后被置位
Bit6	奇偶校验错标志
Bit7	保留使用
Bit8	D 寄存器设置错误 ● 执行串口初始化指令 UINIT 时，当配置区的起始单元和字符数设置超限时被置位 ● 当发送指令 USEND 或接收指令 URCV 被激活，且所填软元件的起始单元和字符数设置超限时被置位
Bit15～Bit9	保留使用

注意：发送接收控制字是用户可以修改的，修改后的值直接参与当前控制；自由协议标志位状态字为是给用户观测的，只能在标志位完成后进行清零操作，不能直接对其赋非 0 值参与控制，否则会造成通信异常。

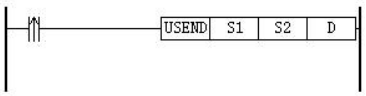


5.3 USEND 指令

5.3.1 USEND 指令介绍

通过指定串口发送 1 个或多个字节。

发送字节



16 位指令	USEND	连续执行型
7 步USEND	P	脉冲执行型

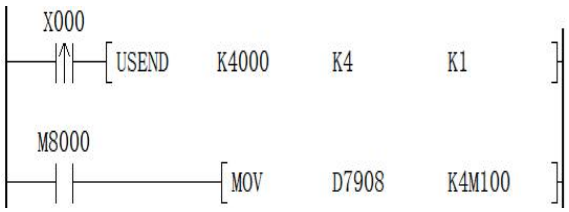
32 位指令
无

操作数	作用	类型
S1	指定本地发送的软元件首地址编号	16 位 BIN
S2	指定发送的字符个数	16 位 BIN
D	指定串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							

功能与动作

- 通过指定串口发送 1 个或多个字节(255≥S2≥1)。
- 指令执行时，发送请求标志位被置位；指令执行完毕，发送完成标志位被置位。
- S1 指定发送寄存器软元件的首地址编号，发送连续S2 个 D 寄存器的低字节数据。
- 若 S1 和 S2 设置后使得寄存器范围超限（比如 S1 设置为 K7990，S2 设置为 K20， 指向的地址区最终将超过 D7999），D 寄存器设置出错标志位被置位。
- 若发送自动转接收标志位被设置，则发送完毕接收请求标志位被置位，系统进入接收状态，这种情况下接收起始单元和字节个数由任务调度配置区指定。
- 发送指令 USEND 不能使用常开或常闭触点激活，必须使用边沿（上升/下降沿）指令触发，或者只能导通一个周期的触点。

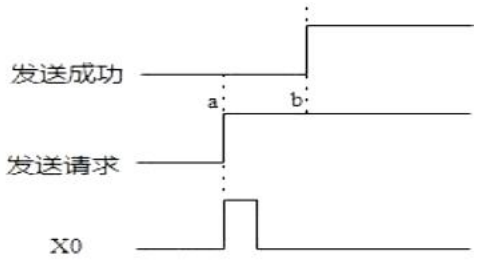


- 当 X000 触点闭合时，向串口 1 对象发送 D4000，D4001，D4002 D4003 的低字节（高字节被自动忽略）。
- 如 D4000=H1122，D4001=H3344，D4002=H5566，D4003=H7788，

那么发送到串口的数据为 H22，H44，H66 和 H88 这 4 个字节。

- 将自由协议状态放入M100~M115，便于观察

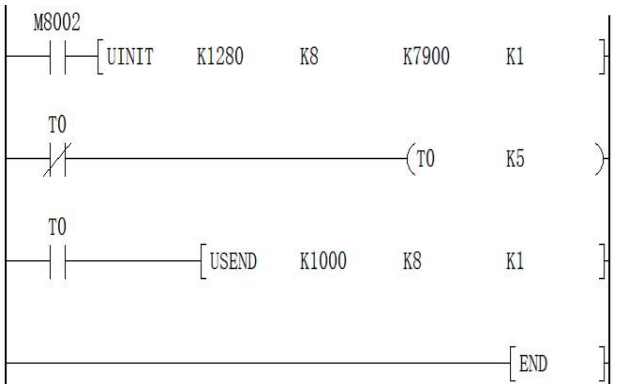
该梯形图动作时序如下：



- 当 X0 上升沿导通后，发送请求标志 M100 置 1
- 经过时间 ab 后，发送成功标志位置 1.
- ab 为发送字节所占用时间，假设本例波特率设置为 9600，发送4 个字节，则 ab 时间在 4ms 左右。

5.3.2 USEND 指令发送方法

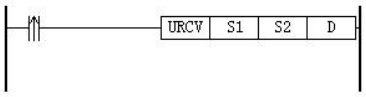
发送方法：USEND 指令发送可以分为随机发送和定时发送，一般应用场合选择定时发送，即利用计数器或者定时器在一定间隔时间后将指定内容发送出去。



- 上电后，每隔 0.5 秒向串口 1 对象发送 D1000，D1001，D1002， D1003，D1004，D1005，D1006，D1007 的低字节（高字节被自动忽略）。
- 如 D1000=H1122，D1001=H3344，D1002=H5566，D1003=H7788， D1004=H8877，D1005=H6655，D1006=H4433，D1007=H2211 那么发送到串口的数据为 H22，H44，H66，H88，H77，H55，H33，H11 这 8 个字节。
- 发送间隔时间不宜设置过短，必须大于发送字节传送所需时间， 本例中发送 8 个字节，波特率为 9600，则报文传输时间在 8MS 左右,考虑到接收方处理报文还需要时间，故发送时间最好不低于 8MS，并根据实际情况适当延长。

5.4 URCV 指令

5.4.1 URCV 指令介绍

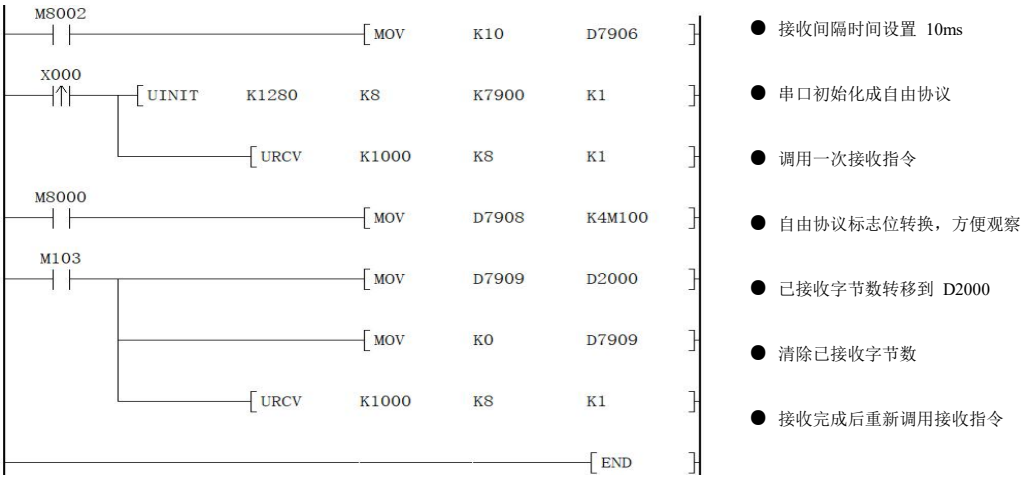
接收字节	
	
16位指令 URCV 连续执行型 7步 URCV P 脉冲执行型	32 位指令 无

操作数	作用	类型
S1	指定本地接收的软元件首地址编号	16 位 BIN
S2	指定接收的字符个数	16 位 BIN
D	指定串口号	16 位 BIN

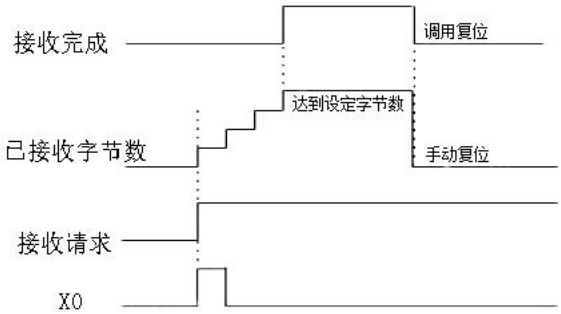
操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	•	•	•							

功能与动作

- 通过指定串口接收 1 个或多个字节(255≥S2≥1)。
- 指令执行时，接收请求标志位被置位，接收完成标志位被复位；指令执行完毕，接收完成标志位被置位。
- S1 指定接收寄存器软元件的首地址编号，存入连续 S2 个 D 寄存器的低字节数据。
- 若 S1 和S2 设置后使得寄存器范围超限（比如 S1 设置为 K7990，S2 设置为 K20，指向的地址区最终将超过 D7999），D 寄存器设置出错标志位被置位。
- 若接收到字节后一段时间内（3.5 个字节时间）再未接收到数据（这个时间由任务调度区的接收等待间隔时间设定），即使接收字节数小于设定值，仍然判定接收指令执行完毕，接收完成标志位被置位。
- 接收指令 URCV 不能使用常开或常闭触点激活，必须使用边沿（上升/下降沿）指令触发，或只导通一个扫描周期的触点。



该梯形图动作时序如下：

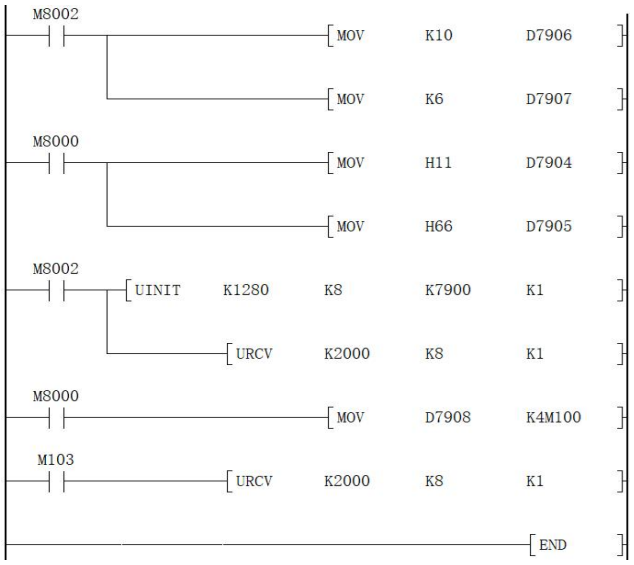


- 当 X0 上升沿导通时，接收请求标志 M103 置1
- 当已接收字节数 D7909 到达设定接收时，接收完成标志位置1
- D7909 是循环累加计数，每次接收完成后需利用梯形图手动清除，否则下次接收字节会累加。
- 接收完成后再次置位发送指令，则接收完成自动复位。
- 当接收区在接收 3.5 个字节后再未接收到数据，此时经过接收间隔时间 D7907 后，会把接收完成标志位置1
- 若一个数据都未接收到，则可以通过 D7909 来判定是否有数据接收过来。
-

5.4.2 接收起始符和接收终止符的应用

MC系列PLC在用自由协议和第三方进行通信作为接收方时，发果发送方发送过来的数据相对固定，则可以利用接收起始符和接收终止符来提高接收报文的正确率，设置接收起始符和终止符后，MC系列PLC只会接收报文数据接收起始符和接收终止符之间的报文内容（包含起始符和终止符）。

示例如下：



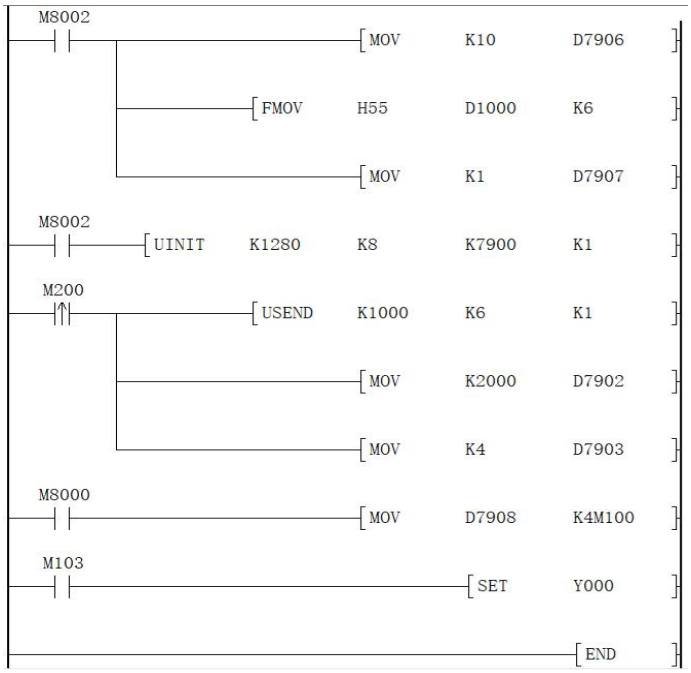
- 设置接收间隔等待时间 10ms
- 启用接收起始符和接收终止符
- 设置起始符值为 H11，表示报文开始
- 设置终止符值为 H66，表示报文结束
- 设置自由协议，波特率 9600 8 1 N
- 上电调用一次接收指令
- 将自由协议标志位用位表示，方便观察
- 接收完成后重新调用接收指令

功能说明

- 假设第三方发送数据为 8 个字节数据，波特率 9600，8，1，N，其值分别为 H11，H22，H33，H44， H55，H66，H77，H88；
- 运动控制接收起始符值为 H11，终止符值为H66，则每次接收过来的数据为H11，H22，H33，H44， H55，H66，分别存放到 D2000-D2005 6 个寄存器里面，由于 H77 和 H88 在终止符 H66 之后，故两个值被过滤掉了。

5.4.3 发送自动转接收

接收报文有两种方式，一种是利用接收指令来接收，参照 5.4.1，有些场合需要追求高效率通信，MC系列PLC可以设置发送报文后自动转为接收状态，只需要在 USEND 指令后指定接收区地址和接收字节数， 不需要重新调用接收指令，示例如下：



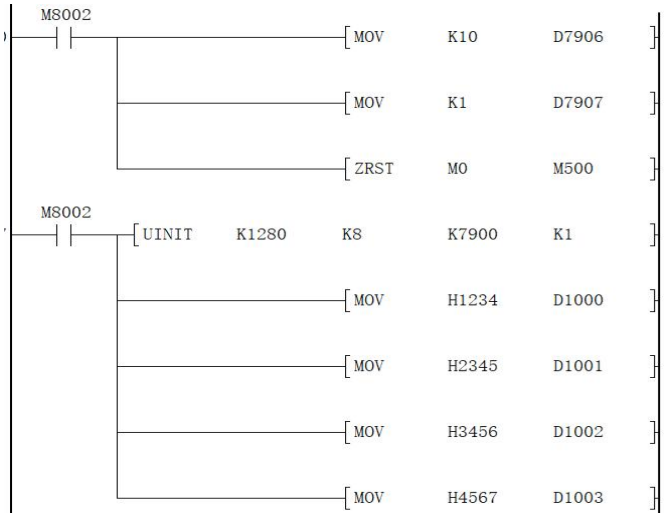
- 设置接收间隔等待时间 10ms
- 将 H55 传入 D1000-D1005 这 6 个寄存器中
- 设置为发送后自动转接收
- 设置为自由协议，9600 ，8 1 N
- M200 上升沿启动一次发送指令，转接收
- 接收区起始地址
- 接收字节数
- 将自由协议状态位转换成位，方便观察
- 接收成功后，点亮 Y0

5.5 应用案例

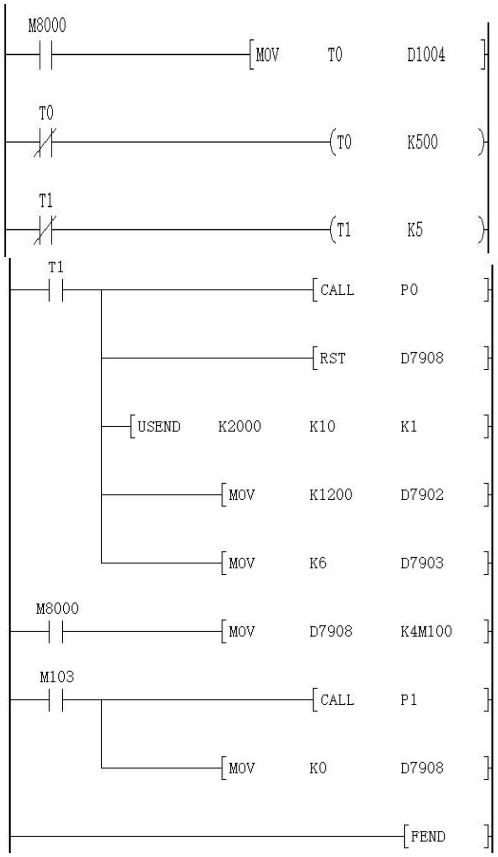
- 两台MC系列PLC利用自由协议传输数据

（1）MC A 和MC B 使用自由协议进行数据传输，MC A 将D1000-D1004 数据定时写入MC B 中， MC B 收到数据后，将 D1000-D1003 数据送入MC A 中，本例中数据均为梯形图模拟值，为随机发送值，不存在任何实际意义，用户实际应用中根据控制要求进行数据读写。

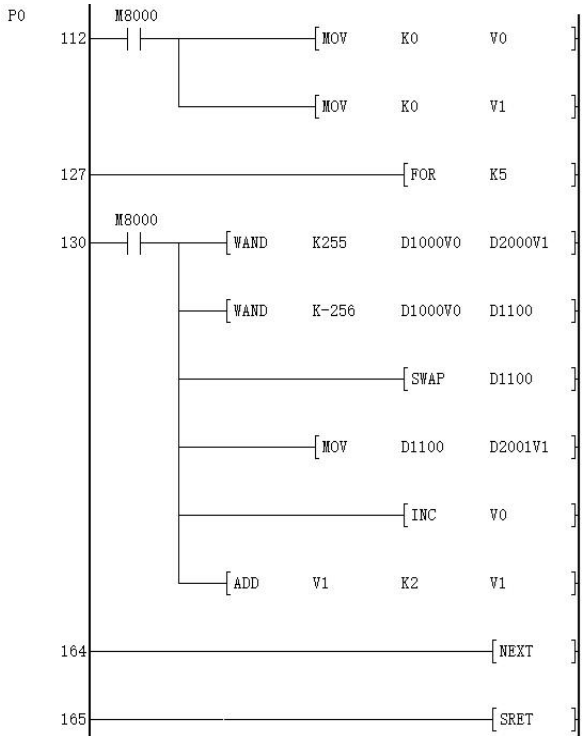
（2）MC A 梯形图如下：



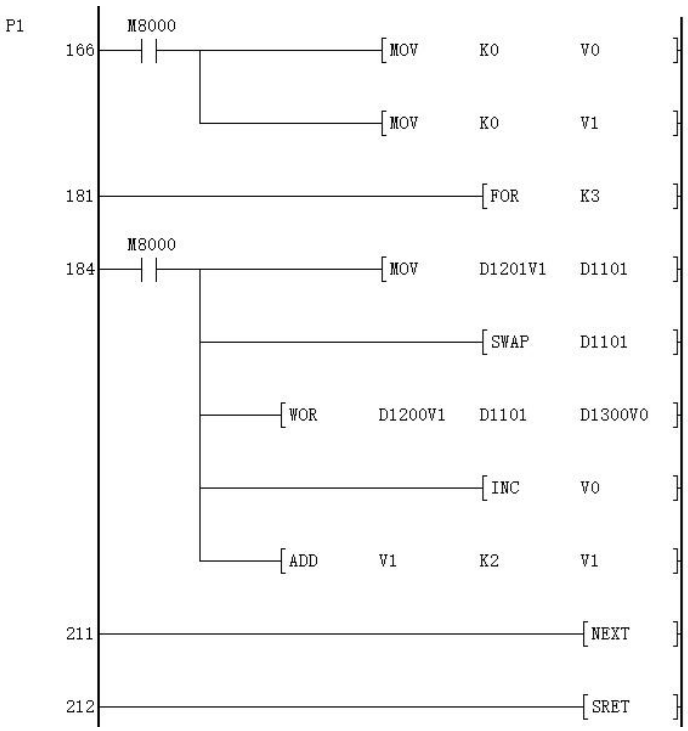
- 设置接收间隔等待时间 10ms
- 设置成发送自动转接收
- 上电清除 M0-M500
- 设置成自由协议 9600 8 1 N
- D1000-D1004 模拟给定数值，其中
- D1004 为 0-500 随机数



- P0 子程序将发送数据进行整理，然后发送
- 设置成 500MS 发送一次报文
- 清除自由协议状态位
- 发送从 D2000 开始 10 个字节
- 发送后自动转接收，从 D1200 开始接收，接收 6 个字节
- 将自由协议标志状态位转移成位， 方便观察
- P1 子程序将接收数据进行整理。
- 清除自由协议状态位标志
- 主程序结束标志

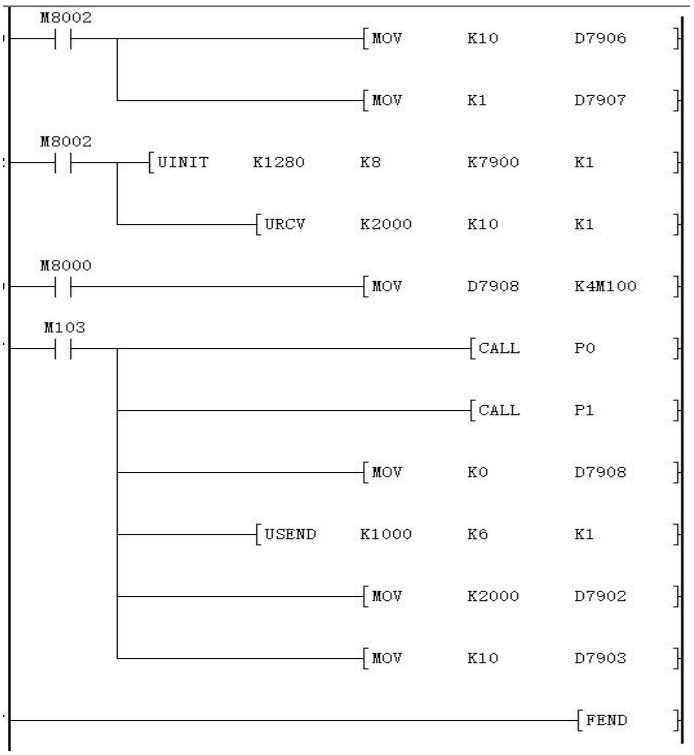


- 由于自由协议发送是以字节为单位，发送的是寄存器低字节数据，高字节自动忽略，P0 的作用是将模拟给定数值D1000-D1004 完整发送到LP3 B，此例中将 D1000-D1004 的低高位数据分别传入发送区D2000-D2009 中的 10 个寄存器低位。
- 将 D1000 低字节数据转移到 D2000
- 将 D1000 高字节数据转移到 D1100
- 将 D1100 高字节数据换到低字节
- 将 D1100 低字节数据转移到 D2001，此 3 步作用相当于将 D1000 高字节数据转移到 D2001 低字节
- 循环 5 次，将剩下的数据按照上面规律转移

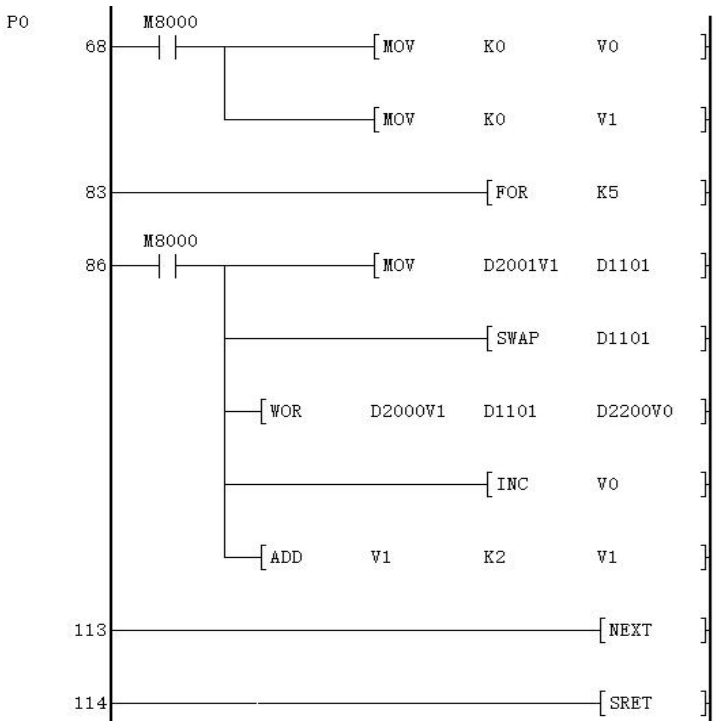


- 将接收区D1200-D1205 中6 个寄存器中的低字节数据转移到 D1300-D1302 三个寄存器中，其中D1300 高位存放 D1201 数据，D1300 低位存放 D1200 数据；D1301 高位存放 D1203 数据，D1301 低位存放 D1202 数据，D1302 中高位存放 D1205 数据，D1302 中低位存放 D1204 数据。

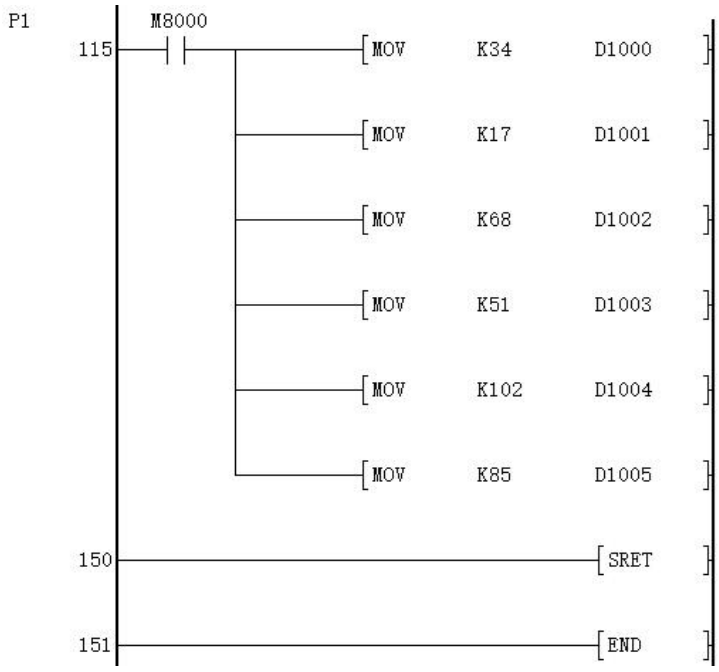
（3）MC B 梯形图如下：



- 设置接收间隔等待时间 10ms
- 设置成发送自动转接收
- 设置成自由协议 9600 8 1 N
- 上电进入接收状态
- 将自由协议标志转移成位，方便观察
- P0 程序整理接收过来的数据
- P1 程序整理需要发送的数据
- 清除自由协议状态位
- 从 D1000 开始，发送 6 个字节
- 发送完成后，接收起始地址 D2000
- 接收 10 个字节



- P0 子程序作用是将接收区 D1200-D1209 中的 10 个低 8 位字节数据转移到D2200-D2204的 5 个寄存器中。



- P1 子程序是填写发送区需要发送的数据。

6. KLink总线协议

6.1KLink总线协议介绍

本章节将要介绍的是科威专用高速通信总线协议KLink，KLink总线协议是面向主从站之间的高速数据交换，波特率为2.5Mbps。

KLink总线协议是一个主从协议，在同一时间，总线上只能有一个主站，和若干个（最多16个）从站。KLink总线协议由主站发起通信任务，从站接收到主站报文后回发响应报文。当从站没有接收到主站的通信任务时，从站不会发送数据。从站之间不能互相通信。

6.2 KLink总线协议格式

KLink总线主站包含10个任务列表，任务列表配置后主站依据任务列表调度通信。每个任务独立配置，配置与主站通信的从站站号、数据长度和相关地址，主站通信串口配置则通信建立。从站接收并处理报文后回发响应报文。每个从站必须有唯一的地址（0~15）。

主站可以向所有从站发送数据，所有从站可以接收主站发送的数据，但只有主站指定的从站可以回发响应报文。

KLink总线定义了四类报文，过程数据上传 RPDO、 过程数据下传 TPDO、 参数上传 RSDO和参数下传TSDO。过程数据上传 RPDO报文包含四种报文RPT_M0--RPT_M3，过程数据下传 TPDO包含四种报文RPT_M4--RPT_M7，RPT_M8，参数下传TSDO报文RPT_M9。过程数据上传RPDO报文主站实时读取从站的各种数据，如AD值、编码器位置、电流、力矩等实时数据。过程数据下传 TPDO报文主站实时向从站传送各种数据，如给定速度、脉冲位置、DA值、同步信号等命令数据。参数上传RSDO报文主站读取从站一个参数数据，参数下传TSDO报文主站向从站传送一个设定参数。参数上传RSDO和参数下传TSDO报文传送的数据实时性要求不高。

6.2.1 报文结构

KLink总线物理接口为驱动器的RJ45网口，通信波特率为2.5Mbps，报文长度为11bytes，字节按8位数据位和1位停止位传输，目前通信格式只支持这种，不能更改。

KLink总线报文结构如下：

报文字节数	byte0	byte1	byte2-byte9	byte10
报文内容	ID0	ID1	DATA0-DATA7	XRC
表达含义	报文索引0，含括报文类型和从站号	报文索引1，含括报文发送方设置	4字通信数据	异或校验值

ID0：报文索引0，低4位表示报文类型，目前定义10种报文0-9，RPT_M0--RPT_M9。高4位表示与主站通信的从站站号0-15。

报文类型	RPT_M0	RPT_M1	RPT_M2	RPT_M3	RPT_M4	RPT_M5	RPT_M6	RPT_M7	RPT_M8	RPT_M9
ID0低4位值	0	1	2	3	4	5	6	7	8	9

ID1：报文索引1，位0表示报文是主站还是从站发出，0为主站发出，1为从站发出。其余位暂未使用。

DATA0--DATA7：4字通信数据，不同的报文种类定义和传输的内容不同。

XRC：异或校验值。

6.2.2 RPT_M0--M3报文

过程数据上传 RPDO报文包含四种报文RPT_M0，RPT_M1，RPT_M2，RPT_M3，均可实现主站读取指定从站

的各种数据，并将该从站返回的数据存放在主站指定单元中。RPT_M0--RPT_M3号报文一次完整的通信为主站发起通信，从站接收到主站报文后处理并回发响应报文。

RPT_M0--RPT_M3号报文主站发出报文结构如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号 0-15	0-3，分别一一对应 RPT_M0--RPT_M3	0	读取的从站资源地址	异或校验值

ID0低4位为0，RPT_M0号报文通信，为1，RPT_M1号报文通信，以此类推。

DATA0-DATA7为主站读取的从站数据资源地址，两个DATA设置一个地址，DATA0-DATA1，DATA2-DATA3，DATA4-DATA5，DATA6-DATA7可设置4个地址，每个地址可独立设置，地址可以不连续。

从站接收到主站报文校验通过后根据主站报文回发响应报文，从站回发响应报文格式如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号 0-15	0-3，分别一一对应 RPT_M0--RPT_M3	1	从站回发的四字数据	异或校验值

从站回发的数据存入主站接收报文指定单元中，主站接收单元由主站接收报文定义字指定，后续章节讲述。

6.2.3 RPT_M4--M7报文

过程数据下传 TPDO报文包含四种报文RPT_M4，RPT_M5，RPT_M6，RPT_M7，主站实时向从站传送各种数据。从站可以回发响应报文或者不回发。

RPT_M4--RPT_M7号报文主站发出报文结构如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号0-15	4-7，分别一一对应 RPT_M4--RPT_M7	0	主站发出的四字数据	异或校验值

ID0低4位为4，RPT_M4号报文通信，为5，RPT_M5号报文通信，以此类推。

DATA0-DATA7为主站指定地址向从站传送的四字数据，数据也可以不连续。主站指定地址由主站发送报文控制字指定，后续会讲述。

主站可以同时向多个从站传送各种数据，各从站均可接收主站数据，并按约定使用报文中的部分或全部数据。但只有主站指定的从站（ID0高4位指定）可以回发响应报文，回发主站各种数据。从站也可以不回发。

主站指定从站回发响应报文格式如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号0-15	4-7，分别一一对应 RPT_M0--RPT_M3	1	从站回发的四字数据	异或校验值

从站回发的数据存入主站接收报文指定单元中，主站接收单元由主站接收报文定义字指定，后续章节讲述。

6.2.4 RPT_M8报文

参数上传RSDO报文RPT_M8，主站读取指定从站一个字参数。RPT_M8号报文一次完整的通信为主站发起通信，从站接收到主站报文后处理并回发一个字参数。

RPT_M8号报文主站发出报文结构如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号0-15	8，对应RPT_M8	0	读取的从站资源地址，1个字参数地址	异或校验值

ID0低4位为8，RPT_M8号报文通信。

DATA0-DATA7中DATA0-DATA1为主站读取的从站1个字参数资源地址，DATA2-DATA7不起作用。

从站接收到主站报文校验通过后根据主站报文回发响应报文，从站回发响应报文格式如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号0-15	8，对应RPT_M8	1	从站回发数据	异或校验值

DATA0-DATA7中DATA0-DATA1为从站字参数资源地址，DATA4-DATA5为从站回发的字参数数据。DATA2-DATA3和DATA6-DATA7不起作用。

从站回发的字参数存入主站接收报文指定单元中，主站接收单元由主站接收报文定义字指定，后续章节讲述。

6.2.5 RPT_M9报文

参数下传TSDO报文RPT_M9，主站向指定从站传送一个字参数。从站可以回发响应报文或者不回发。

RPT_M9号报文主站发出报文结构如下：

ID0		ID1	DATA0-DATA7	XRC
高4位	低4位			
从站站号0-15	9，对应RPT_M9	0	主站发出数据	异或校验值

ID0低4位为9，RPT_M9号报文通信。

DATA0-DATA1为主站设定字参数写入从站的资源地址（从站地址），DATA4-DATA5为主站向从站传送的该字参数，DATA2-DATA3，DATA6-DATA7不起作用。

从站可以回发响应报文，从站也可以不回发。

6.3 KLink总线配置资源

KLink总线配置了控制字和状态字，便于通信设置和监控。

6.3.1 主站配置资源

KLink总线主站配置资源为R2100--R2259共160个R，其中R2100--R2139 40个R为通信任务控制字，R2140--R2259 120个R为发送接收报文定义字。

主站通信任务控制字寄存器如下：

资源号	表示含义	备注

R2100	主站控制字	R2100=0停止通信； =1申请修改通信任务并通信； =2正常通信； =3通信出错；
R2101	主站通信计数	主站每发送从站一报文R2101加1，每正确接收从站一报文R2101减1
R2102	通信出错编号	暂时只有一种出错编号 当R2104<0或>10，通信出错，R2102=1 当R2110-R2119其中任何一个<0或>9，通信出错，R2102=1
R2103	通信时间刻度数	一次完整通信时间为R2103*50us。通信时间建议设置为250us--2.5ms，超出此范围可能会有错误报文。 支持在线修改，修改后将R2100赋值1修改生效
R2104	通信报文个数	通信的报文总数，R2104<=10 R2104=0，不通信不报错 R2104<0或>10，通信出错，R2100=3，R2102=1 支持在线修改，修改后将R2100赋值1修改生效
R2105-R2109	保留	
R2110-R2119	通信报文列表	R2110-R2119 为通信报文列表，指定系统报文的通信顺序，分别对应最大10个报文的报文类型设置。每个R可任意设置为RPT_M0--RPT_M9（0-9值）报文类型，且每种报文类型可以重复，顺序自由组合。最终通信几个报文由报文个数R2104限定。例如：R2104=3，R2110=1，R2111=4，R2112=1，则系统按RPT_M1--RPT_M4--RPT_M1号报文顺序通信，其后R2113-R2119所指定的报文类型并不参与通信。 R2110-R2119 的值必须设0-9范围内，否则通信出错，R2100=3，R2102=1 支持在线修改，修改后将R2100赋值1修改生效
R2120	从站0通信计数	主站和从站0成功通信一次R2120加1
R2121	从站1通信计数	主站和从站1成功通信一次R2121加1
R2122	从站2通信计数	主站和从站2成功通信一次R2122加1
R2123	从站3通信计数	主站和从站3成功通信一次R2123加1
R2124	从站4通信计数	主站和从站4成功通信一次R2124加1
R2125	从站5通信计数	主站和从站5成功通信一次R2125加1
R2126	从站6通信计数	主站和从站6成功通信一次R2126加1
R2127	从站7通信计数	主站和从站7成功通信一次R2127加1
R2128	从站8通信计数	主站和从站8成功通信一次R2128加1
R2129	从站9通信计数	主站和从站9成功通信一次R2129加1
R2130	从站10通信计数	主站和从站10成功通信一次R2130加1
R2131	从站11通信计数	主站和从站11成功通信一次R2131加1
R2132	从站12通信计数	主站和从站12成功通信一次R2132加1
R2133	从站13通信计数	主站和从站13成功通信一次R2133加1
R2134	从站14通信计数	主站和从站14成功通信一次R2134加1
R2135	从站15通信计数	主站和从站15成功通信一次R2135加1
R2136-R2139	保留	

主站发送从站报文和接收从站报文时，需设置从站站号、从站地址、主站地址、数据长度等参数，KLink总线配置了报文定义字设置这些参数，每种类型的报文发送和接收定义字配置资源数量和定义一致，只是占用不同的地址，便于理解使用。

主站发送接收报文定义字寄存器如下：

RPT_M0--RPT_M3号报文

报文种类	发送/接收	资源地址					
RPT_M0	发送地址	R2140	R2141	R2142	R2143	R2144	R2145
	接收地址	R2146	R2147	R2148	R2149	R2150	R2151
RPT_M1	发送地址	R2152	R2153	R2154	R2155	R2156	R2157
	接收地址	R2158	R2159	R2160	R2161	R2162	R2163
RPT_M2	发送地址	R2164	R2165	R2166	R2167	R2168	R2169
	接收地址	R2170	R2171	R2172	R2173	R2174	R2175
RPT_M3	发送地址	R2176	R2177	R2178	R2179	R2180	R2181
	接收地址	R2182	R2183	R2184	R2185	R2186	R2187
RPT_M0-RPT_M3	发送地址含义	从站站号	主站传送给从站读取从站数据资源地址的长度	读取的从站数据资源地址0	读取的从站数据资源地址1	读取的从站数据资源地址2	读取的从站数据资源地址3
RPT_M0-RPT_M3	接收地址含义	从站站号	主站接收从站数据，将数据刷新至主站对应资源地址的长度	主站接收到从站数据存放资源地址0	主站接收到从站数据存放资源地址1	主站接收到从站数据存放资源地址2	主站接收到从站数据存放资源地址3

主站传送给从站读取从站数据资源地址的长度：设定读取的从站数据资源地址0-3，几个地址传送给从站。例如设为2，则表示读取的从站数据资源地址0和读取的从站数据资源地址1地址传送给从站，读取的从站数据资源地址2和读取的从站数据资源地址3不传送给从站。

主站接收从站数据，将数据刷新至主站对应资源地址的长度：主站接收到从站数据，数据存放至主站资源地址0-3的长度。例如设为1，则表示刷新主站存放资源地址0的数据，而主站存放资源地址1-3数据不刷新。

编程时，建议主站传送给从站读取从站数据资源地址的长度（RPT_M0的R2141）、主站接收从站数据将数据刷新至主站对应资源地址的长度（RPT_M0的R2147）、从站RPT_M0的数据长度（从站资源）三者严格要求一致，设定不一致时，可能导致异常情况。RPT_M1-RPT_M3同理。

RPT_M4--RPT_M7号报文

报文种类	发送/接收	资源地址					
RPT_M4	发送地址	R2188	R2189	R2190	R2191	R2192	R2193
	接收地址	R2194	R2195	R2196	R2197	R2198	R2199
RPT_M5	发送地址	R2200	R2201	R2202	R2203	R2204	R2205
	接收地址	R2206	R2207	R2208	R2209	R2210	R2211
RPT_M6	发送地址	R2212	R2213	R2214	R2215	R2216	R2217
	接收地址	R2218	R2219	R2220	R2221	R2222	R2223
RPT_M7	发送地址	R2224	R2225	R2226	R2227	R2228	R2229
	接收地址	R2230	R2231	R2232	R2233	R2234	R2235
RPT_M4-RPT_M7	发送地址含义	指定回发报文的从站站号	主站发送至从站数据长度	主站发送数据资源地址0	主站发送数据资源地址1	主站发送数据资源地址2	主站发送数据资源地址3
RPT_M4-RPT_M7	接收地址含义	指定回发报文的从站站号	主站接收从站数据，将数据刷新至主站对应资源地址的长度	主站接收的从站数据存放资源地址0	主站接收的从站数据存放资源地址1	主站接收的从站数据存放资源地址2	主站接收的从站数据存放资源地址3

RPT_M4-RPT_M7号报文指定的从站可以回发响应报文，因此也定义了接收报文寄存器，设置主站接收从站数据的长度、存放资源地址等。主站接收从站数据，将数据刷新至主站对应资源地址的长度含义与RPT_M0-RPT_M3号报文一致。

编程时，建议主站接收从站数据将数据刷新至主站对应资源地址的长度（RPT_M4的R2195）、从站RPT_M4号报文回复主站的发送数据长度（从站资源R2136，并非主站资源）两者严格要求一致，设定不一致时，可能导致异常情况。RPT_M5-RPT_M7同理。

RPT_M8--RPT_M9号报文

报文种类	发送/接收	资源地址					
RPT_M8	发送地址	R2236	R2237	R2238	R2239	R2240	R2241
	含义	从站站号	×	×	读取的从站资源地址	×	×
	接收地址	R2242	R2243	R2244	R2245	R2246	R2247
	含义	从站站号	×	主站存放资源地址	×	×	×
RPT_M9	发送地址	R2248	R2249	R2250	R2251	R2252	R2253
	含义	从站站号	×	主站发送数据资源地址	从站接收到数据存放资源地址	×	×
	接收地址	R2254	R2255	R2256	R2257	R2258	R2259
	含义	从站站号	×	×	×	×	×

上表中×表示该寄存器不需要设置。

RPT_M8、RPT_M9号报文从站没有配置发送接收报文定义字寄存器，只配置了控制字寄存器。

主站发送接收报文定义字寄存器R2140--R2259支持在线修改，修改即生效。

6.3.2 从站配置资源

KLink总线从站配置资源为R2100--R2204共105个R，其中R2100--R2109 10个R为控制字，R2110--R2204 95个R为报文解析定义用。

从站控制字寄存器定义如下：

资源号	表示含义	备注
R2100	从站控制字	R2100=0停止通信； =2启动通信；
R2101	从站通信计数	从站每接收一正确报文R2101加1
R2102-R2109	保留	

从站接收主站报文和发送主站报文时，需设置从站站号、从站地址、数据长度等参数，KLink总线配置了报文定义字设置这些参数，每种类型的报文接收和发送定义字配置资源是不同的。RPT_M8、RPT_M9号报文从站没有配置发送接收报文定义字寄存器。

从站接收发送报文定义字寄存器如下：

RPT_M0--RPT_M3号报文

资源号	表示含义	备注
R2110	RPT_M0号报文从站回复主站数据的长度	
R2111	RPT_M1号报文从站回复主站数据的长度	
R2112	RPT_M2号报文从站回复主站数据的长度	
R2113	RPT_M3号报文从站回复主站数据的长度	
R2114-R2119	保留	

编程时，从站RPT_M0号报文回复主站的数据长度R2110与主站传送给从站读取从站数据资源地址的长度、主站接收从站数据将数据刷新至主站对应资源地址的长度（主站资源的RPT_M0的R2141、R2147，并非从站资源）三者严格要求一致，设定不一致时，可能导致异常情况。RPT_M1-RPT_M3同理。

RPT_M4--RPT_M7号报文

报文种类		资源地址						
RPT_M4	接收地址1	R2120	R2121	R2122	R2123	R2124	R2125	R2126
	接收地址2	R2127	R2128	R2129	R2130	R2131	R2132	R2133
	回发数据地址	R2134	R2135	R2136	R2137	R2138	R2139	R2140
RPT_M5	接收地址1	R2141	R2142	R2143	R2144	R2145	R2146	R2147
	接收地址2	R2148	R2149	R2150	R2151	R2152	R2153	R2154
	回发数据地址	R2155	R2156	R2157	R2158	R2159	R2160	R2161
RPT_M6	接收地址1	R2162	R2163	R2164	R2165	R2166	R2167	R2168
	接收地址2	R2169	R2170	R2171	R2172	R2173	R2174	R2175
	回发数据地址	R2176	R2177	R2178	R2179	R2180	R2181	R2182
RPT_M7	接收地址1	R2183	R2184	R2185	R2186	R2187	R2188	R2189
	接收地址2	R2190	R2191	R2192	R2193	R2194	R2195	R2196
	回发数据地址	R2197	R2198	R2199	R2200	R2201	R2202	R2203
RPT_M4-RPT_M7	接收地址1含义	站号	接收数据起始字	接收数据长度	接收数据存放资源地址0	接收数据存放资源地址1	接收数据存放资源地址2	接收数据存放资源地址3
RPT_M4-RPT_M7	接收地址2含义	站号	接收数据起始字	接收数据长度	接收数据存放资源地址0	接收数据存放资源地址1	接收数据存放资源地址2	接收数据存放资源地址3
RPT_M4-RPT_M7	回发数据地址含义	站号	×	发送数据长度	从站回复数据资源地址0	从站回复数据资源地址1	从站回复数据资源地址2	从站回复数据资源地址3

从站报文定义中的站号要和从站串口配置指令中的站号一致，否则无法通信。

RPT_M4-RPT_M7号报文从站可以接收主站发送报文中的部分或全部数据，通过设置接收数据起始字和接收数据长度实现。例如R2121=1，R2122=2，则表示接收到的主站第二个数据、第三个数据（长度为2）存放至接收地址1的接收数据存放资源地址0和接收数据存放资源地址1。R2121和R2122两者和不能超过4。

RPT_M4-RPT_M7号报文也可将接收到的数据存放到不同的地址应用，因此定义了第一接收数据和第二接收数据（分别对应接收地址1和接收地址2）。两个地址均是接收存放主站的数据。从站不能下载梯形图设置接收地址1和接收地址2的情形下，和主站约定好每个数据的意义和存放顺序，接收数据存放到不同的接收地址1和接收地址2有些场合具有工程意义。

RPT_M4-RPT_M7号报文指定的从站可以回发响应报文，因此也定义了从站发送报文寄存器，设置从站发送主站数据的长度、资源地址等。

RPT_M4-RPT_M7号报文从站回复主站数据时，发送数据长度可设。例如R2136=1，则表示从站回复数据资源地址0的数据回发给主站。

编程时，建议从站发送主站的数据长度（从站RPT_M4的R2136）、主站接收从站数据将数据刷新至主站对应资源地址的长度（主站RPT_M4的R2195）两者严格要求一致，设定不一致时，可能导致异常情况。RPT_M5-RPT_M7同理。

从站接收发送报文定义字寄存器R2100--R2204支持在线修改，修改即生效。

6.3.3 主从站资源地址

KLink总线主从站报文定义字中要设置主从站各种资源地址，不同资源地址设置如下：

数据	资源地址设置
D0--D7999	0000--7999
D8000--D8255	8000--8255
T0--T255	8256--8511
C0--C199	8512--8711
C200--C255	8712--8822
R0--R15999	8824--24823

例如RPT_M0号报文要读取1号从站D1000--D1002的数据存放至主站T0--T2中，则主站报文定义字设置如下：

报文种类	发送/接收	资源地址					
RPT_M0	发送地址	R2140	R2141	R2142	R2143	R2144	R2145
		1	3	1000	1001	1002	
	接收地址	R2146	R2147	R2148	R2149	R2150	R2151
		1	3	8256	8257	8258	
RPT_M0-RPT_M3	发送地址含义	从站站号	主站传送给从站读取从站数据资源地址的长度	读取的从站数据资源地址0	读取的从站数据资源地址1	读取的从站数据资源地址2	读取的从站数据资源地址3
RPT_M0-RPT_M3	接收地址含义	从站站号	主站接收从站数据，将数据刷新至主站对应资源地址的长度	主站接收到从站数据存放资源地址0	主站接收到从站数据存放资源地址1	主站接收到从站数据存放资源地址2	主站接收到从站数据存放资源地址3

6.4 KLink总线主从站互联

KLink总线主从站联接时，主站、从站均设置好控制字寄存器和报文定义字寄存器，然后通过串口初始化指令UINIT配置串口，则KLink总线通信启动（主站启动通信，从站是被动接收）。主从站均是先配置控制字寄存器和报文定义字寄存器，然后配置串口初始化指令。先配置串口初始化指令再配置控制字和报文定义字寄存器，主从站无法通信。

6.4.1 串口配置

主从站均是先配置控制字寄存器和报文定义字寄存器，然后配置串口初始化指令。串口配置一定要在控制字寄存器和报文定义字寄存器设置后配置，否则主从站无法通信。

主站串口配置如下：



H600即表示为KLink总线主站。KLink总线通信速度和数据格式暂时只有一种形式：2.5MBPS，8位数据，1停止位。通信缓存区首地址指向D7920（K7920），指定串口1为通信对象(K1)。

从站串口配置如下：

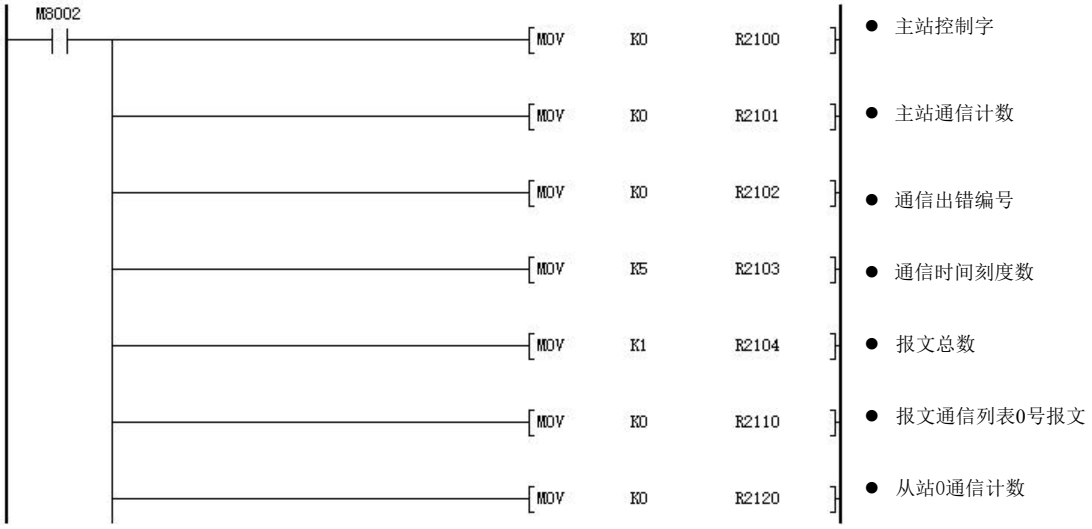


H700即表示为KLink总线0号从站。1号从站则设置为H701，2号从站设置为H702，以此类推。KLink总线通信速度和数据格式暂时只有一种形式：2.5MBPS，8位数据，1停止位。通信缓存区首地址指向D7920（K7920），指定串口1为通信对象(K1)。

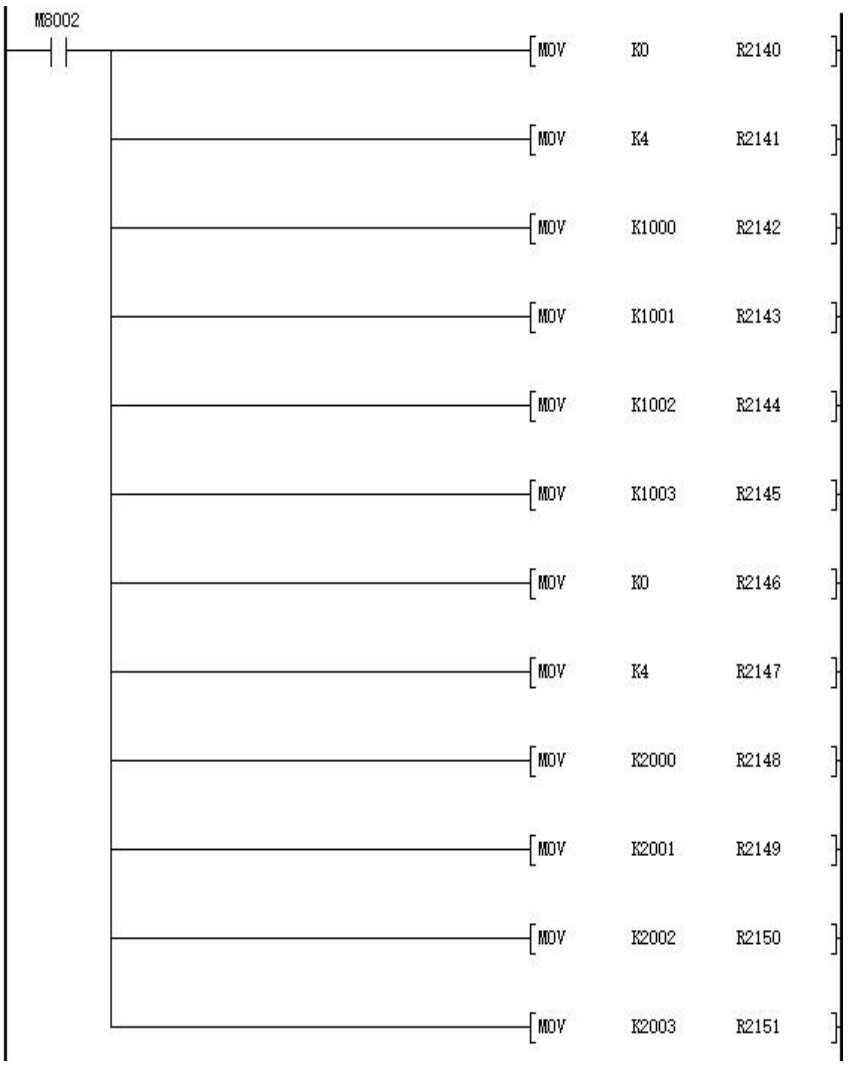
6.4.2 RPT_M0号报文主从站互联

KLink总线主站RPT_M0号报文读取0号从站D1000--D1003的数据存放至主站D2000--D2003。主从站梯形图按如下顺序配置。

主站通信控制字配置如下：



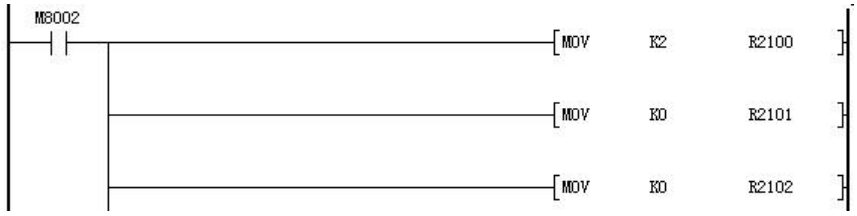
主站报文定义字配置如下：



主站串口初始化指令配置如下：(H600)



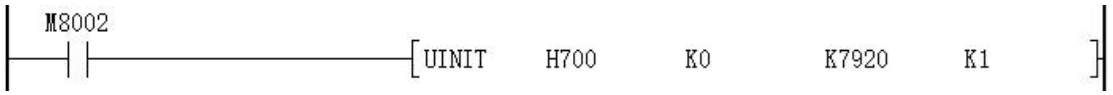
从站通信控制字配置如下：



从站报文定义字配置如下：



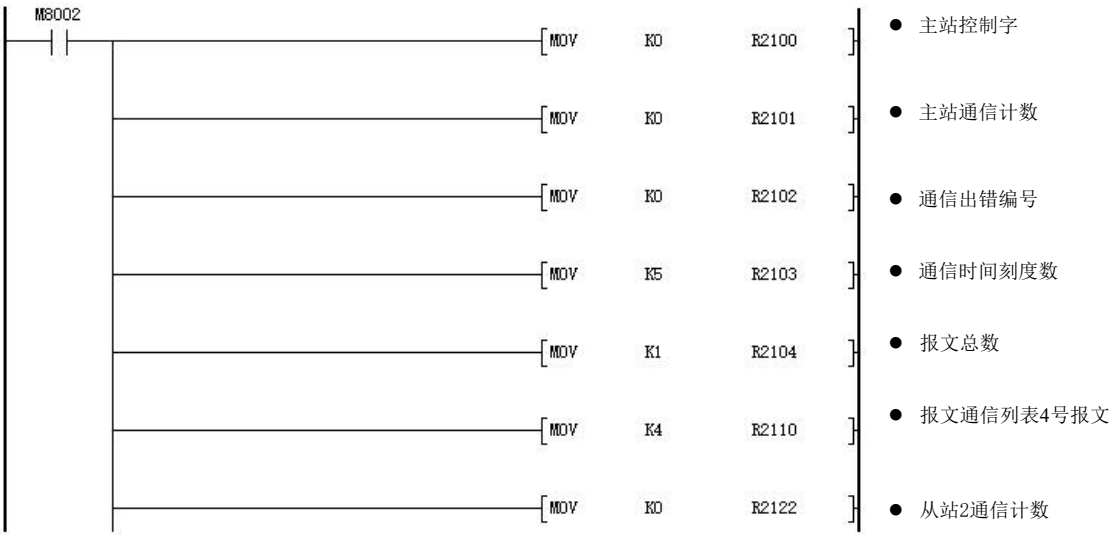
从站串口初始化指令配置如下：(H700)



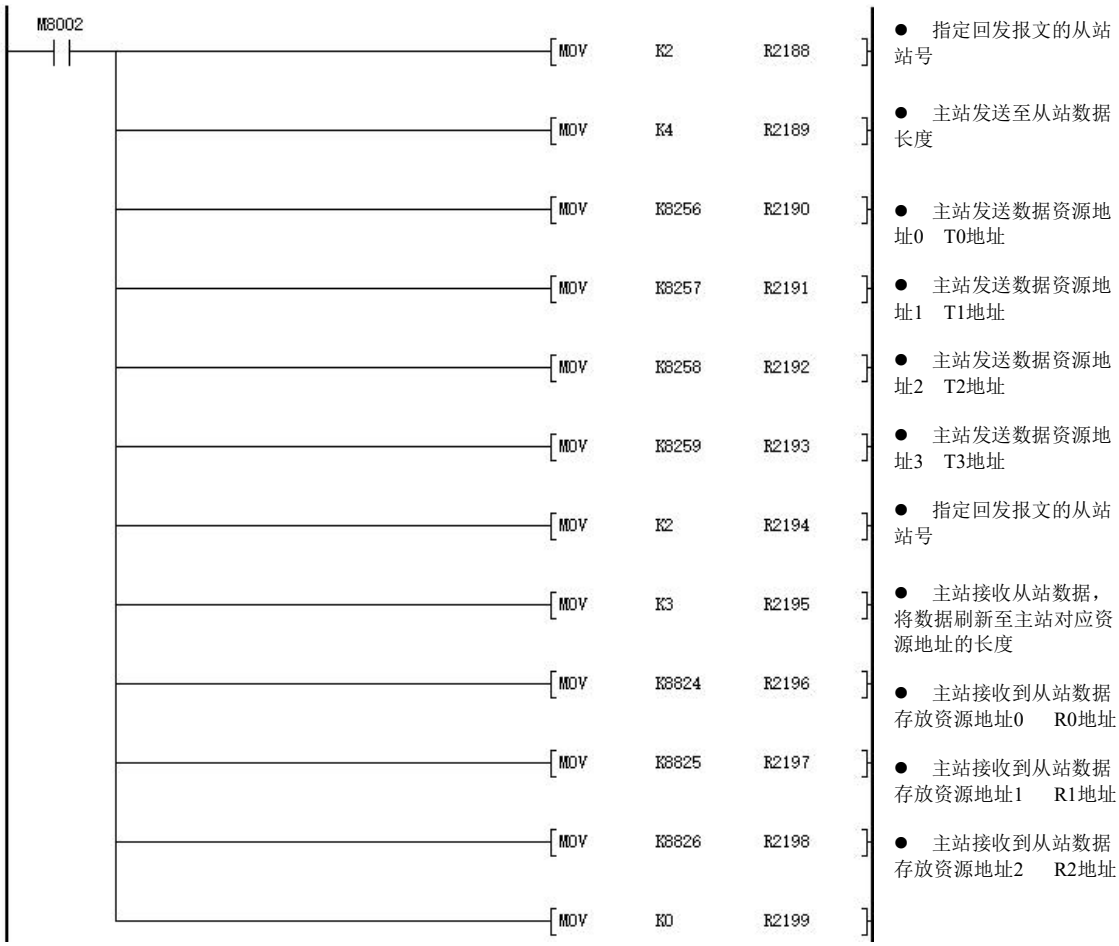
6.4.3 RPT_M4号报文主从站互联

KLink总线主站RPT_M4号报文T0--T3数据广播至所有从站，2号从站第一接收数据接收主站第二、第三个数据至从站C0--C1，2号从站第二接收数据接收主站第四个数据至从站C200。2号从站回复D8000--D8002数据至主站R0--R2。主从站梯形图按如下顺序配置。

主站通信控制字配置如下：



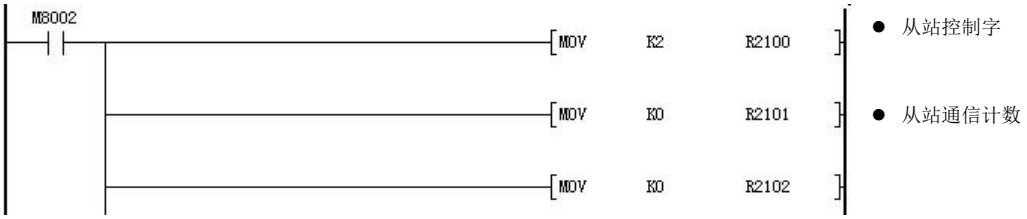
主站报文定义字配置如下：



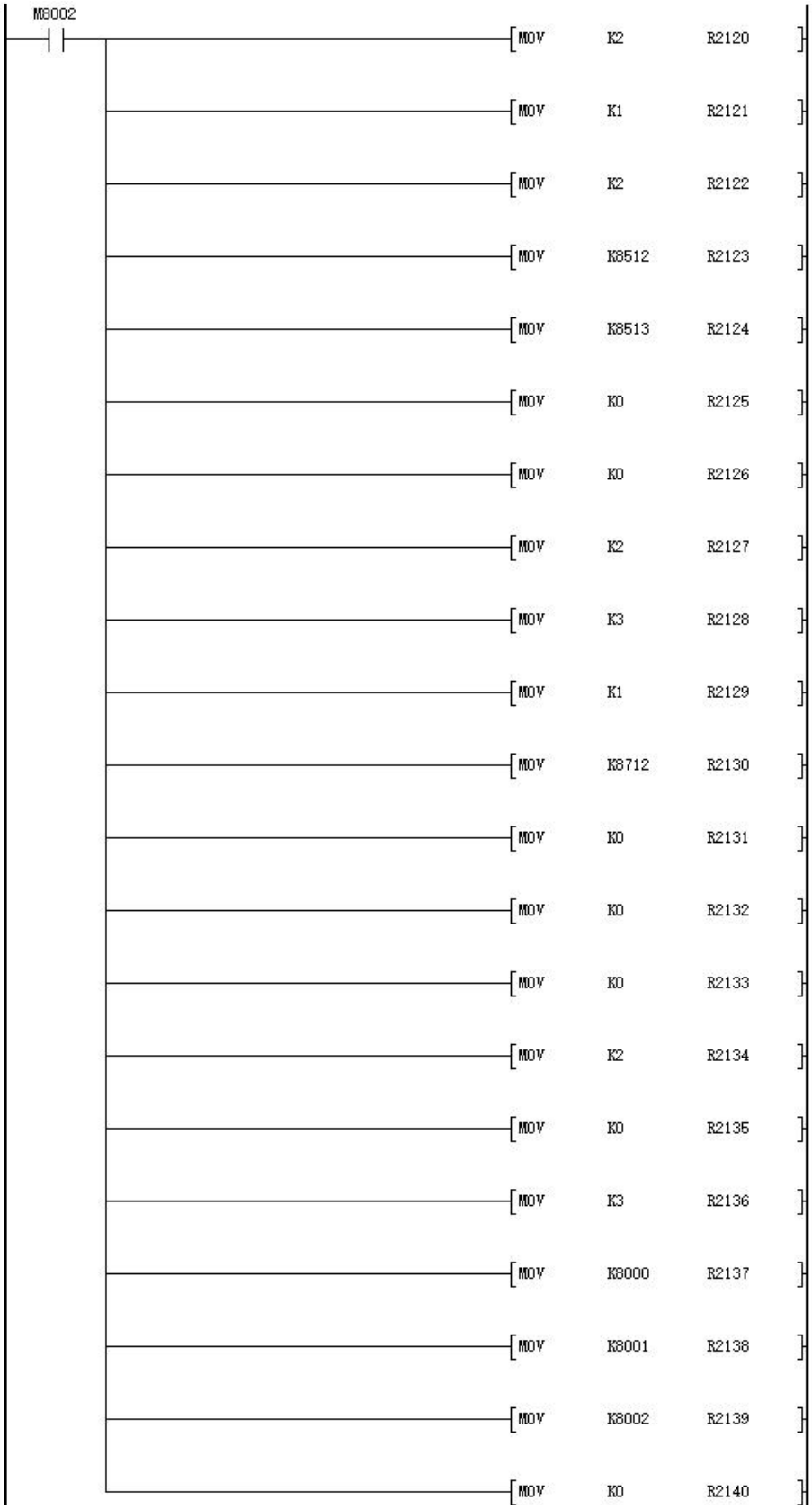
主站串口初始化指令配置如下：



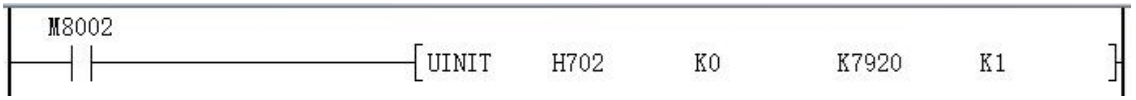
从站通信控制字配置如下：



从站报文定义字配置如下：



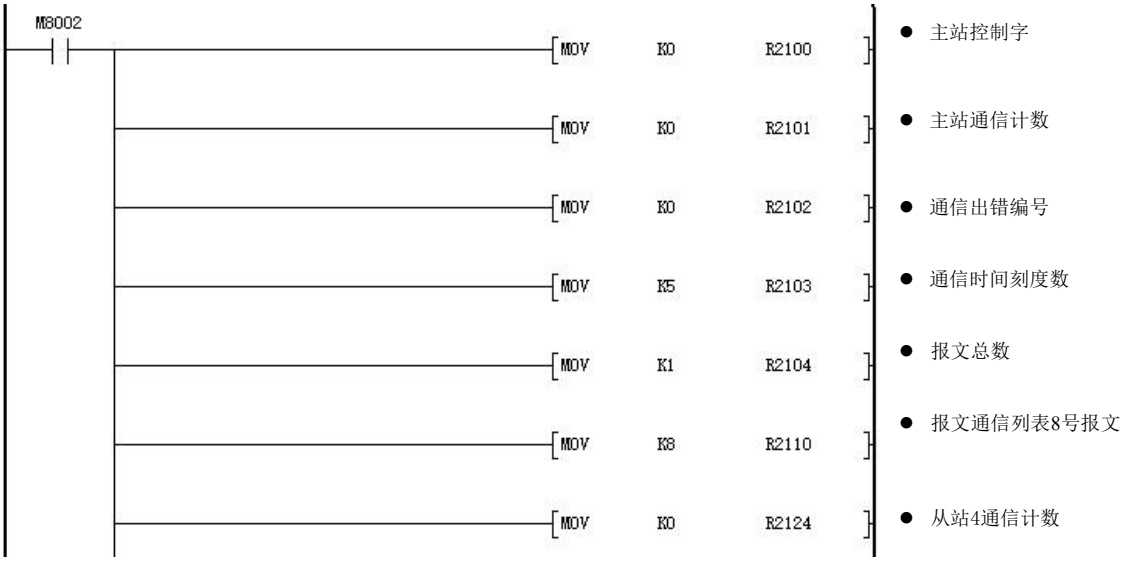
从站串口初始化指令配置如下：



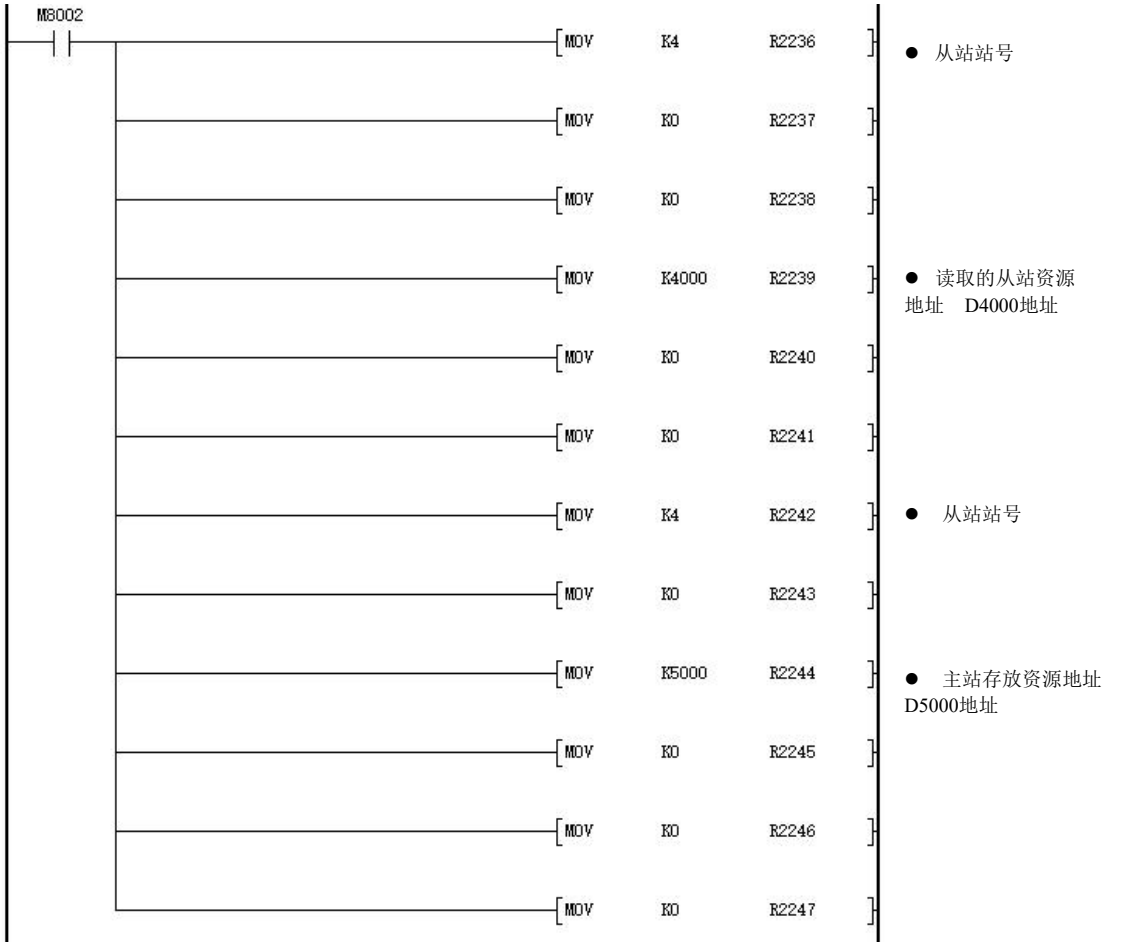
6.4.4 RPT_M8号报文主从站互联

KLink总线主站RPT_M8号报文读取4号从站D4000的数据存放至主站D5000。主从站梯形图按如下顺序配置。

主站通信控制字配置如下：



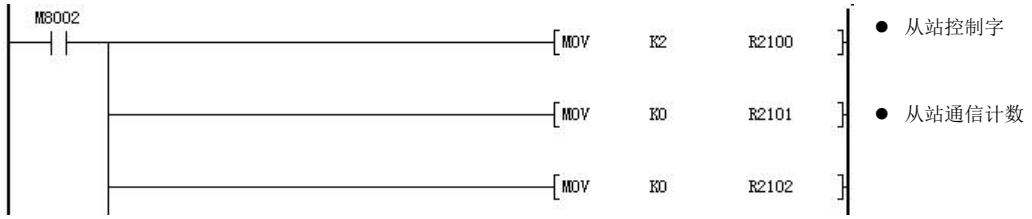
主站报文定义字配置如下：



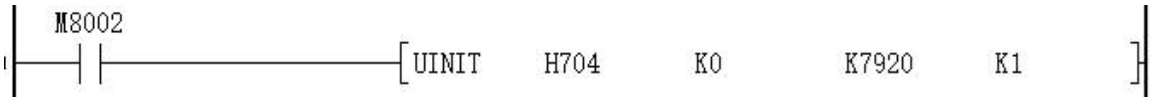
主站串口初始化指令配置如下：(H600)



从站通信控制字配置如下：



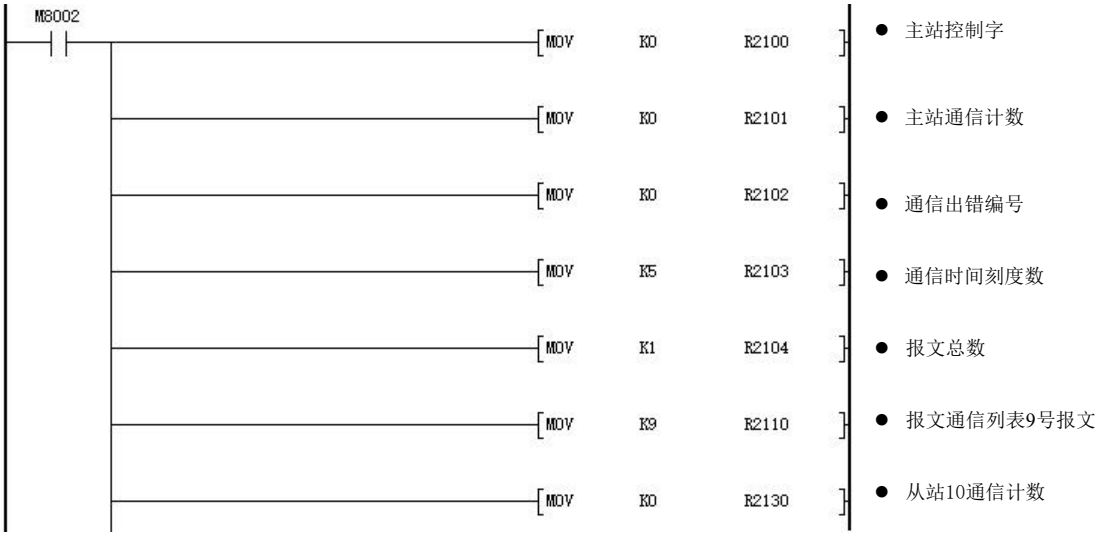
从站串口初始化指令配置如下：(H704)



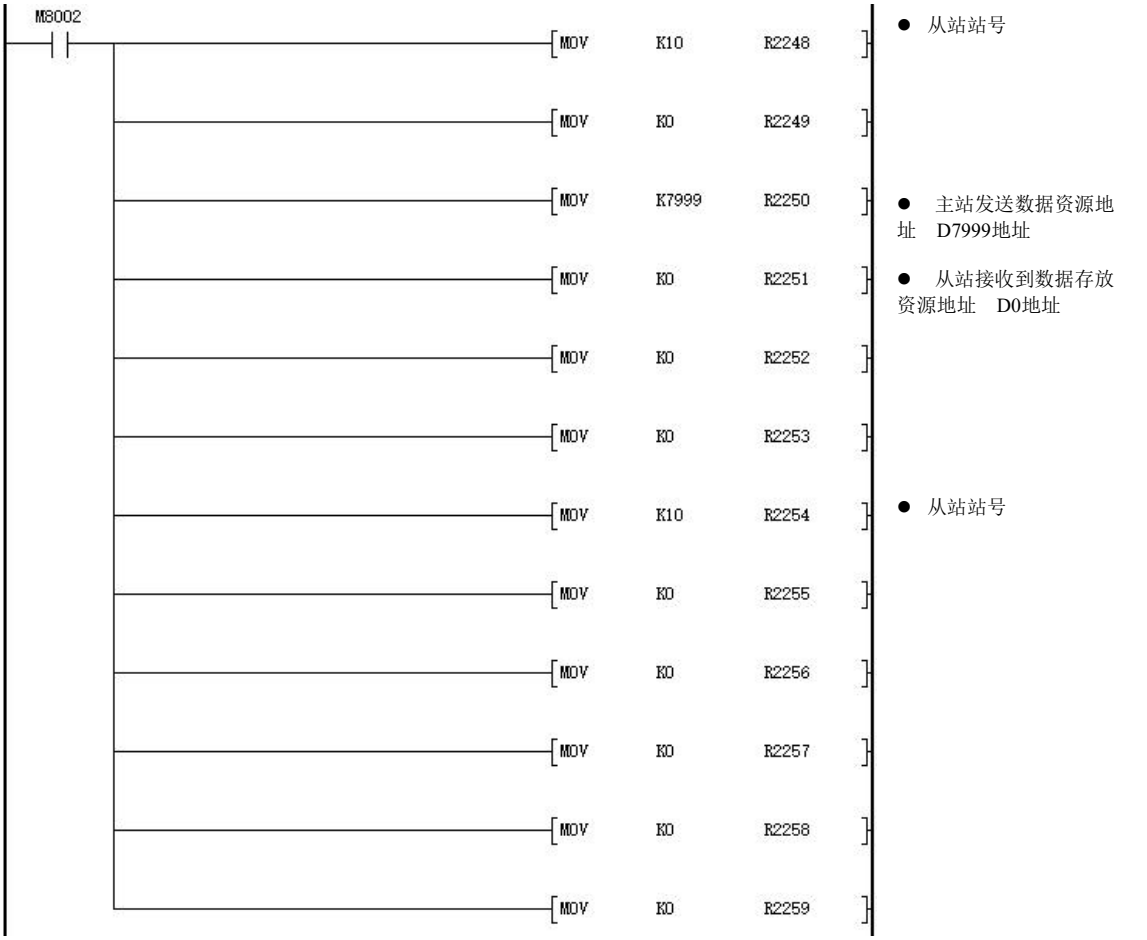
6.4.5 RPT_M9号报文主从站互联

KLink总线主站RPT_M9号报文D7999的数据发送至10号从站D0。主从站梯形图按如下顺序配置。

主站通信控制字配置如下：



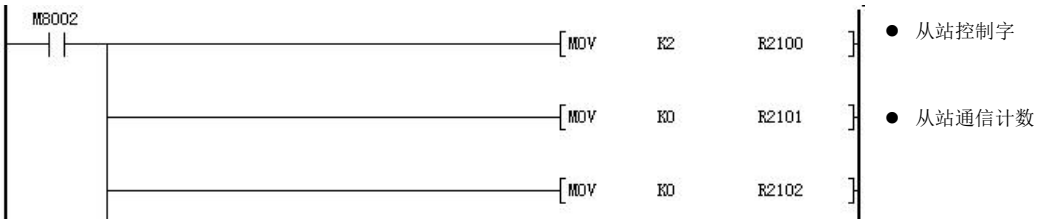
主站报文定义字配置如下：



主站串口初始化指令配置如下：(H600)



从站通信控制字配置如下：



从站串口初始化指令配置如下：(H70A)



6.5 KLink总线应用案列

KLink总线与脉冲控制比较，接线简洁方便，可高速地传输多项信息，比脉冲控制更加精准和实时，适用于多种场合。

6.5.1 自动铣刀磨刀系统

自动铣刀磨刀系统中KLink总线伺服通过KLink总线控制X、C、Z三轴伺服的位置、速度等动作流程实现。

自动铣刀磨刀的工作原理：设备复位状态启动后，磨刀砂轮根据设定好的磨削速度开始运转。磨刀头（Y轴）通

过传动系统将铣刀固定在轴上，并通过X轴匀速前进，与磨刀砂轮进行接触磨削。同时，固定装置会根据设定的参数自动调整磨刀头与磨刀砂轮之间的距离，以达到预设的精度要求。通过Z轴切换磨刀模式（对刀、主刃、侧刃、后刃），以达研磨要求。这种设备的优势在于高精度、高效率、自动化程度高，能够显著提高使用效益并节省成本，降低了操作人员的技能要求。通常适用于研磨2刃、3刃、4刃、6刃等通用铣刀。

自动铣刀磨刀机的结构包括磨刀轴（C轴）、磨刀头（Y轴）、X轴（左右调节）、Z轴（前后调节）、磨刀砂轮、及固定装置和传动系统等部分，如图1所示。

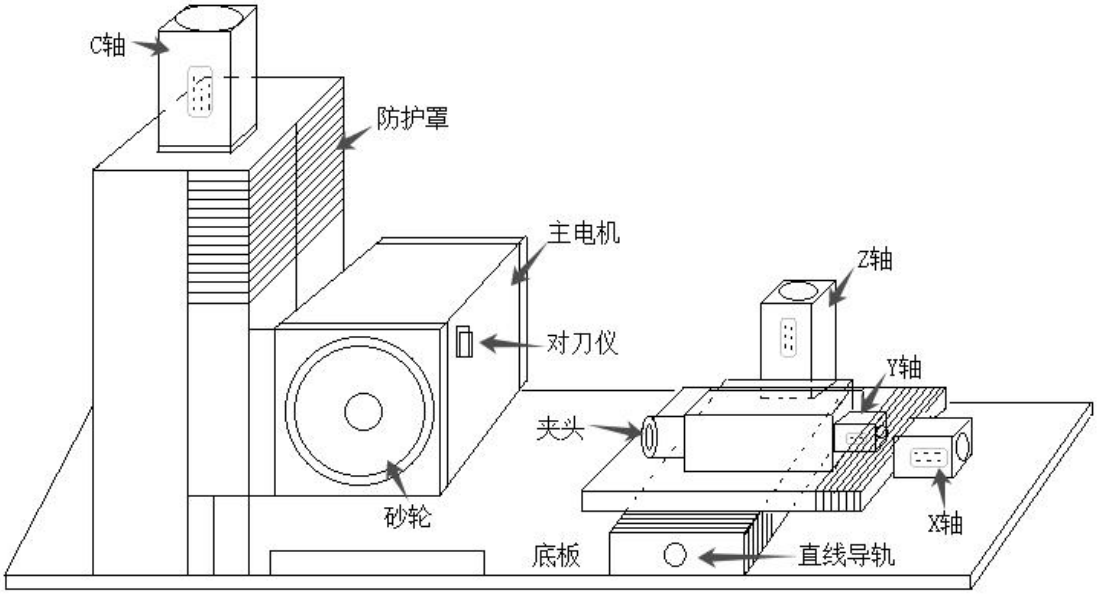


图1.铣刀研磨机械示意图

系统由触摸屏、1台MC-16M16T、1台步进、3台DPS4-075组成，如图2所示。MC-16M16T做主站，通过发脉冲给步进控制砂轮转速，通过KLink总线控制X、C、Z三轴的位置、速度等动作流程，并读取3台从站伺服状态。

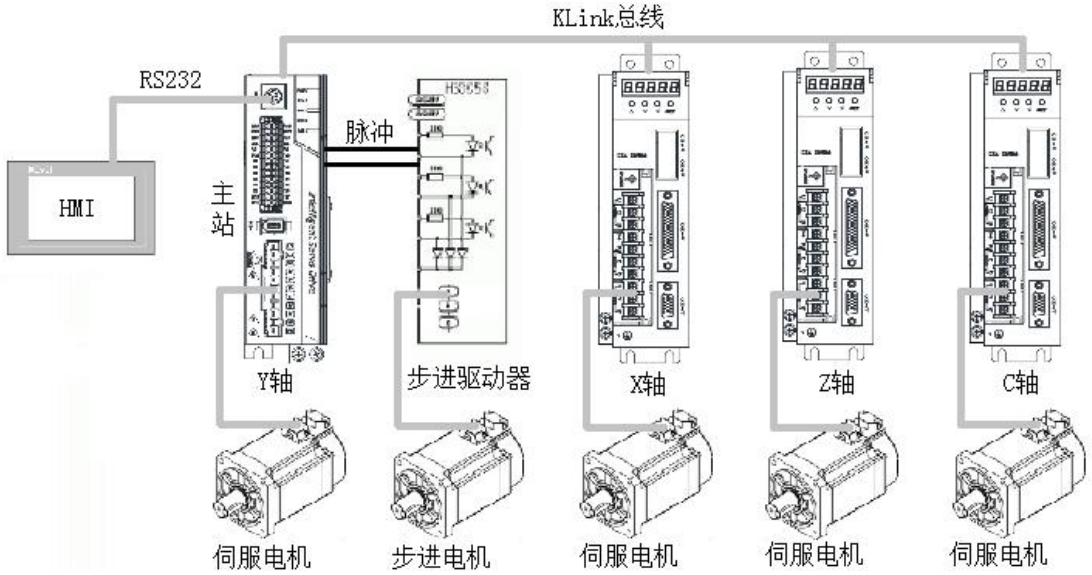


图2.系统组成结构图

磨刀流程如图3所示，主要分为：手动操作、轴回原点操作、对刀流程、研磨流程；手动操作：确认各部件是否异常；轴回原点操作：各轴在自动工作前回零操作；对刀流程：主要是确认铣刀当前位置及刀刃方向，以便确认加工时X轴进给量及Y轴旋转角度；研磨流程：根据刀刃选择及设定参数进行铣刀研磨修复，以达到修复铣刀的效果。

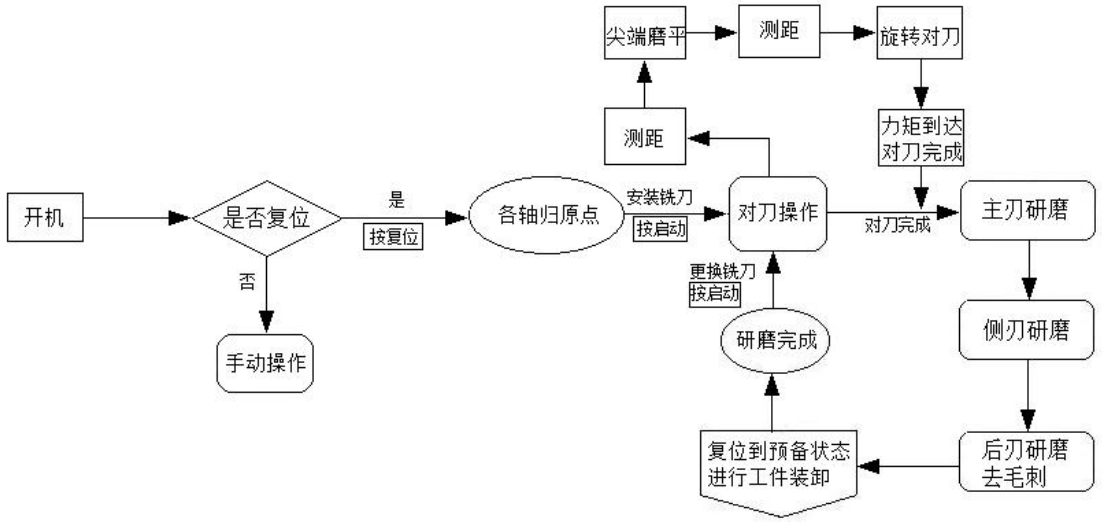


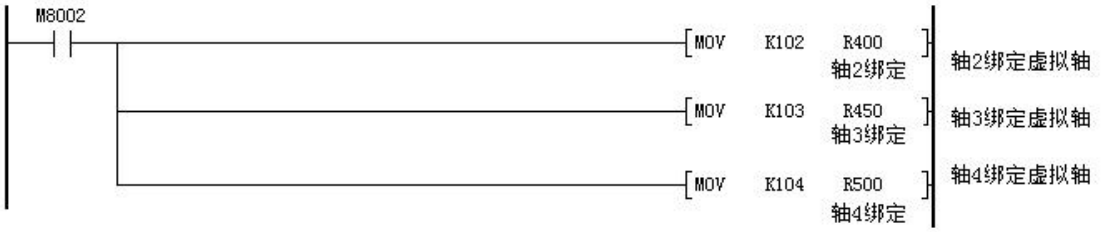
图3.磨刀系统流程图

整个系统主要基于KLink总线实现高速数据交换，所有数据都在主站运算完成，主站以虚轴方式发脉冲，从站通过KLink总线读取对应轴的脉冲数，再以电子齿轮方式（累积型变化）驱动本体伺服运行以达到工艺要求，主站与从站对应关系如表1所示。

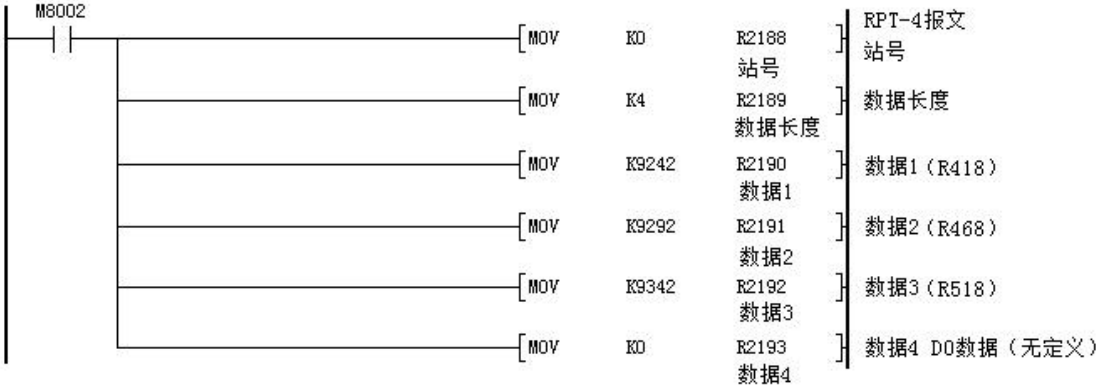
主站轴号	轴0	轴1	轴2	轴3	轴4
主站轴绑定	K0 本体伺服控制	K1 步进转速控制	K102	K103	K104
从站电子齿轮跟随主站地址			R418	R468	R518

表1.主站与从站轴对应关系表

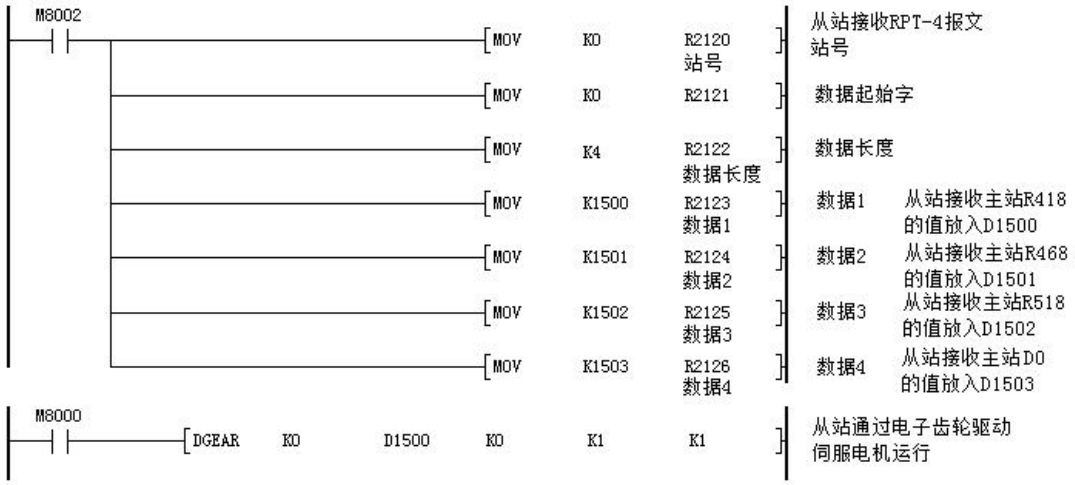
MC PLC默认轴0轴1分别脉冲Y0和脉冲Y1，轴2--轴4绑定虚轴，如下图：



主站按工艺流程要求使用脉冲指令发脉冲即可，轴2-轴4相对位置R418(L)、R468(L)、R518(L)通过KLink总线发送到从站，用RPT-M4号报文，数据可同时发送，数据发送16位即可，如下图所示：



从站接收本站所需要的数据，再以电子齿轮（累积型变化）方式驱动伺服电机运行，如下图所示，从站0以主轴D1500（累积型变化）驱动轴0，以达到伺服电机运行效果。0号从站伺服设置如下：



1号从站站号R2120设置为1，2号从站站号R2120设置为2，其余设置不变，不同从站取不同的数据用DGEAR指令跟随。

6.5.2 压力控制系统

客户压力控制器控制接触点的压力，使接触点的压力遵循一定曲线规律。 前期用脉冲速度方式实现，这种方式频率细分低，采样精度低，达不到控制要求。后续客户遵循科威KLink总线自行研发主控制器，通过KLink总线与从站科威伺服驱动器建立连接，从站伺服驱动器接收主控制器传送的数字给定，每250us通信一次，控制接触点的压力。数字给定准确、快速、无跳动，压力曲线达到控制精度要求。

系统由触摸屏、客户定制型KLink总线主控制器、1台DPS4-075组成，如图4所示。客户定制型KLink总线主控制器做主站，通过KLink总线传送DPS4-075数字给定值，控制从站的位置、速度和力矩，并读取从站伺服的状态。

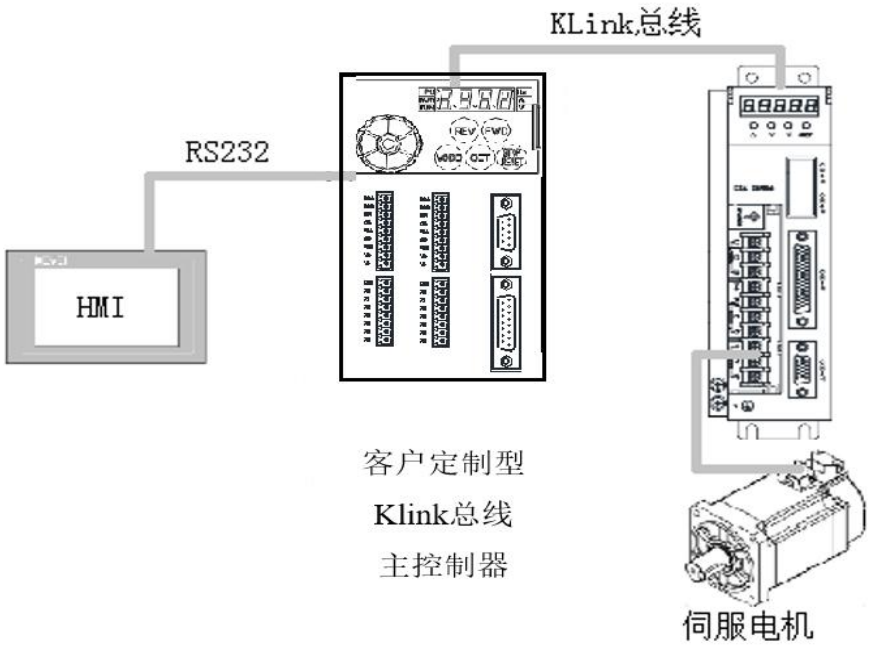


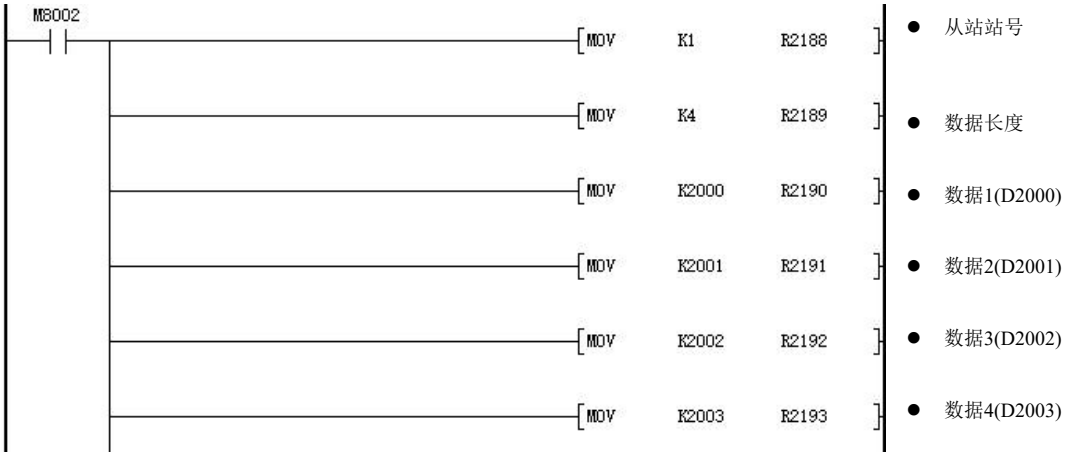
图4.系统组成结构图

整个系统主要基于KLink总线实现高速数据交换，每250us通信一次。从站伺服驱动器接收到数字给定值后驱动本体伺服运行以达到工艺要求。

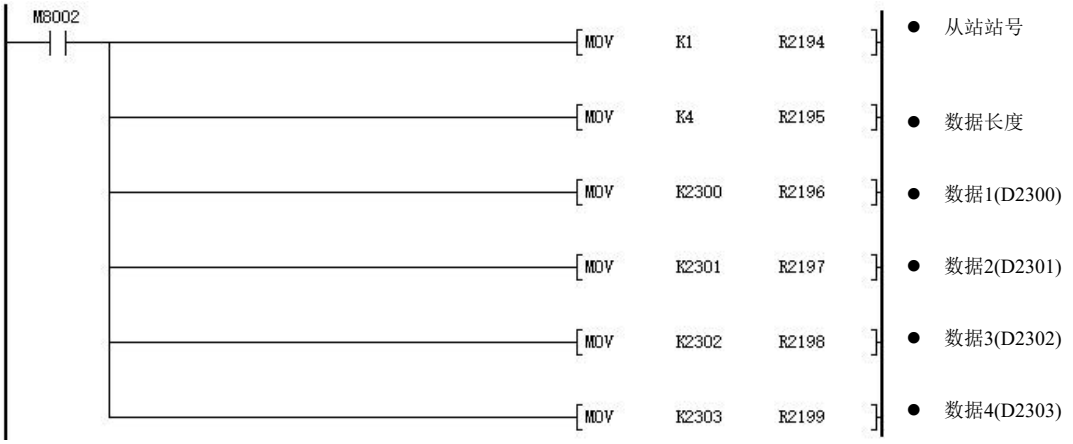
客户定制型KLink总线主控制器与从站科威伺服驱动器交换数据如下：

主控制器采用RPT_M4号报文，发送从站伺服驱动器速度、位置、力矩等给定值和接收从站伺服驱动器各种状态值。

R2188_站号	K1,允许从站K1回发指定的伺服参数
R2189_数据长度	K4,向从站传送四个字的数据
R2190_主站资源1	伺服模式控制（bit0_bit3）；伺服使能控制(bit7); 设定要访问的伺服参数组(bit8_bit15)→D2000
R2191_主站资源2	速度给定值_D2001
R2192_主站资源3	力矩给定值_D2002
R2193_主站资源4	位置给定值_D2003

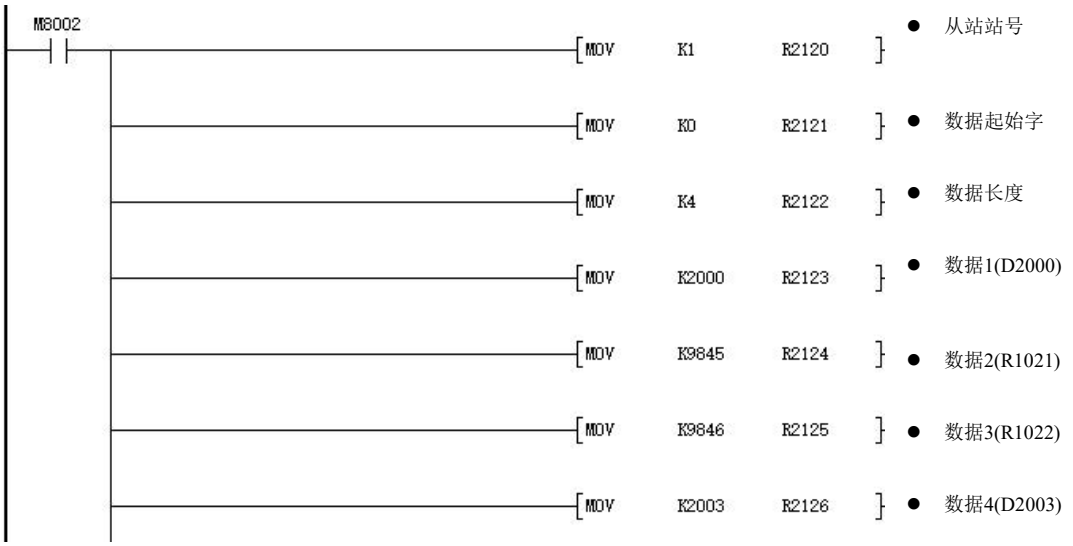


R2194_站号	K1,从站K1向主站返回指定索引组的伺服参数
R2195_数据长度	K4,接收从站传送四个字的数据
R2196_主站资源1	伺服参数组号（bit0—bit7）；伺服运行状态(bit8_bit15)→D2300。
R2197_主站资源2	对应索引号的参数一→D2301
R2198_主站资源3	对应索引号的参数二→D2302
R2199_主站资源4	对应索引号的参数三→D2303



从站伺服接收主站报文并向主站回发报文。从站伺服接收主站报文，接收数据一按主站发送的约定解析出对应命令。接收数据二不设置。

接收一	
R2120_站号	K1,从站K1向主站返回指定索引组的伺服参数
R2121_数据起始字	K0，报文首字开始
R2122_数据长度	K4,接收主站传送四个字的数据
R2123_从站资源1	伺服模式控制（bit0_bit3）；伺服使能(bit7)；伺服参数组号(bit8_bit15).→D2000。
R2124_从站资源2	速度给定值->R1021
R2125_从站资源3	力矩给定值->R1022
R2126_从站资源4	位置给定值->D2003 D2003数据与主站运动指令当前位置（如R318）相对应。从站按累积型电子齿轮引用该数据。



从站伺服回发RPT_M4报文，发送约定的状态或参数。

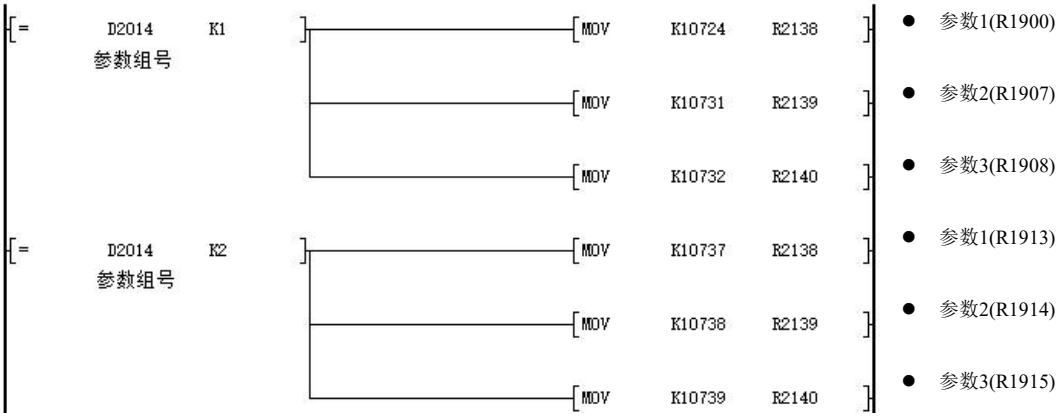
主站是通过发送参数组号来指定从站要返回的伺服参数。伺服参数组号是根据程序需要，将伺服参数编组，便于参数高效访问。以下是回发报文约定的内容：

发送	
R2134_站号	K1，从站K1

R2135_数据起始字	K0，报文首字开始
R2136_数据长度	K4,向主站传送四个字的数据
R2137_从站资源9	伺服参数组号（bit0—bit7）；伺服运行状态(bit8_bit15); 得到参数索引值转存D2014
R2138_从站资源10	对应索引号参数一
R2139_从站资源11	对应索引号参数二
R2140_从站资源12	对应索引号参数三



从站接收到主站给定参数组号后，回传对应的参数，示例如下：



下表参数组号与对应的参数对应表是用户自定义的：

参数组号_D2014	伺服参数一	伺服参数二	伺服参数三
1	电机速度_R1900	电机力矩_R1907	电机电流_R1908
2	位置指令脉冲频率_R1913	速度指令（给定速度）_R1914	力矩指令（给定电流）_R1915
3	散热器温度_R1918	驱动器出错对应报警号_R1919	模拟量通道电压_R1925

4	当前系统控制模式_R1912	1秒内电流最大有效值_R1924	电机瞬时速度_R1927
5	电机瞬时电流_R1928	A相电流AD值_R1929	C相电流AD值_R1930
6	位置指令脉冲数低字_R1932	位置指令脉冲数高字_R1933	位置跟踪误差低位_R1905
7	编码器反馈位置低字_R1934	编码器反馈位置高字_R1935	位置跟踪误差高位_R1906
8	绝对值编码器状态_R1917	编码器通信错误报文数_R1942	
9	16位数据，绝对值编码器圈数计数_R1944	绝对值编码器单圈相对位置_R1946（低字）	绝对值编码器单圈相对位置_R1947（高字）
10	母线电压_R1923	绝对值编码器实时位置_R1948（低字）	绝对值编码器实时位置_R1949（高字）

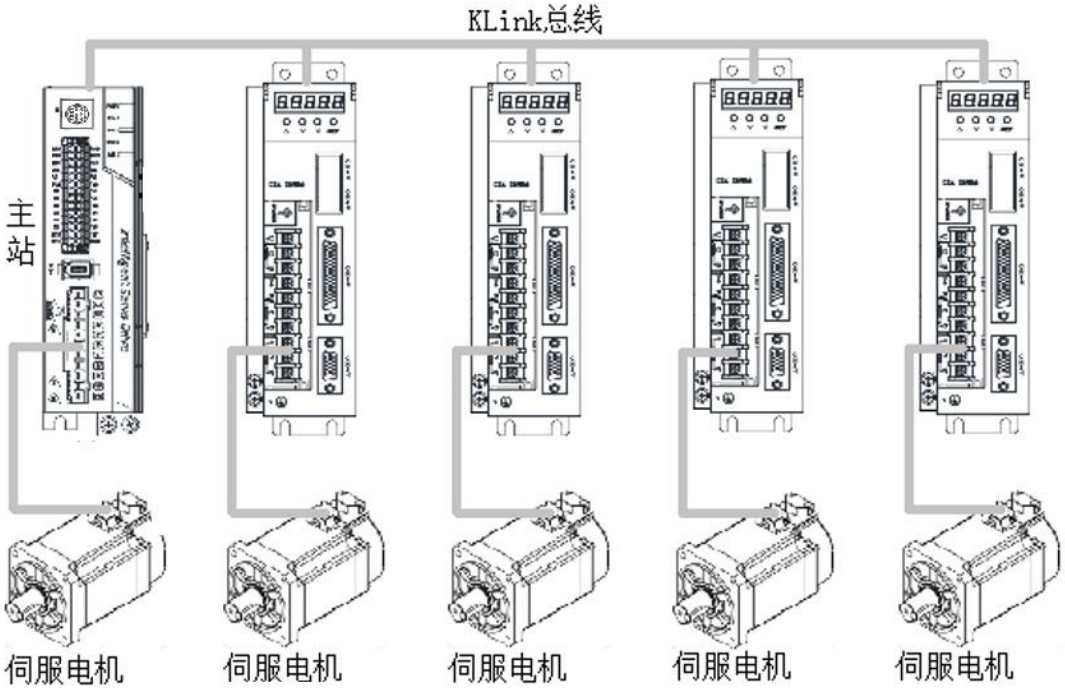
具体参数定义和对应资源地址参看相关说明书。

6.5.3 五轴伺服同步控制系统

同步控制是多轴伺服控制的一种典型应用。一轴为主轴，其他轴为从轴，从轴跟随主轴，不仅要求速度与主轴相同，且相位与主轴保持一致。

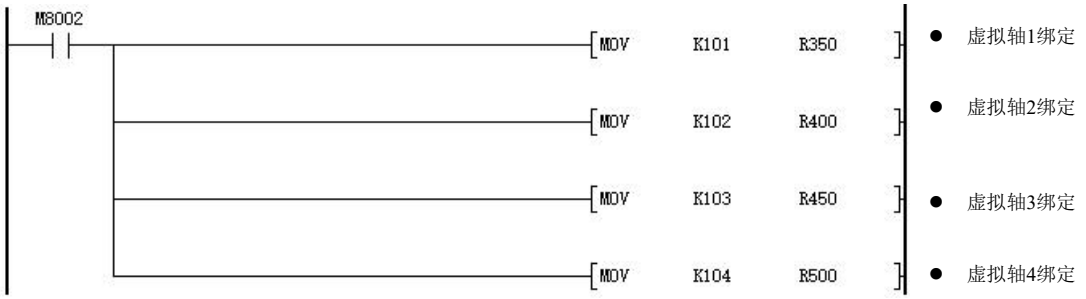
KLink总线型运动控制器或智能伺服，均具有多轴运动控制功能。本案例用总线控制方式替代原来的脉冲控制方式实现同步控制，接线简单，通信速度快，同步精度高。

系统由一台MPSD智能伺服，四台DPS4伺服及与之配套的五台750W磁编伺服电机组成。五台伺服驱动器通过KLink总线连接，一台MPSD作主站，另四台DPS4作从站。同步控制程序在主站编写，从站只编写通信程序或设置通信参数，从站的运行由主站以总线方式来控制，主站可以实时读取从站的伺服参数。



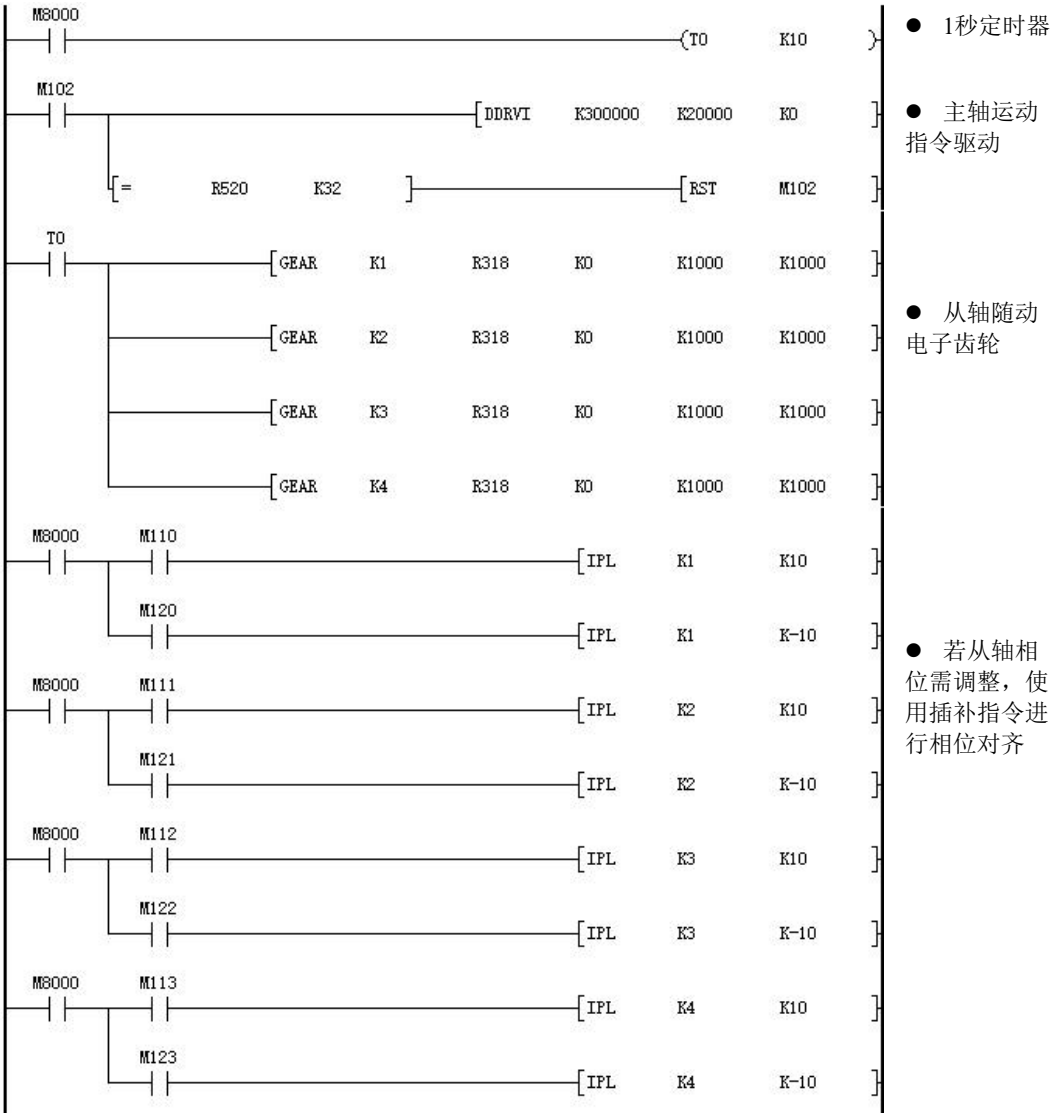
五台伺服驱动器均设为伺服位置内给定控制方式（R1004=8）。主站MPSD直接驱动所连电机，R300设置为0。

其余从轴绑定至到主站的其他轴。主站轴参数设置为R350=101，R400=102，R450=103，R500=104。



主轴根据工艺可使用任意运动指令驱动，示例中使用DDRVI指令驱动。从轴与主轴速度保持一致，采用电子齿轮跟随主轴的运行。如从轴相位需要调整，则使用插补指令进行相位对齐。程序中的M110---M113,M120---M123均为自恢复按键，按下插补，松开不插补。

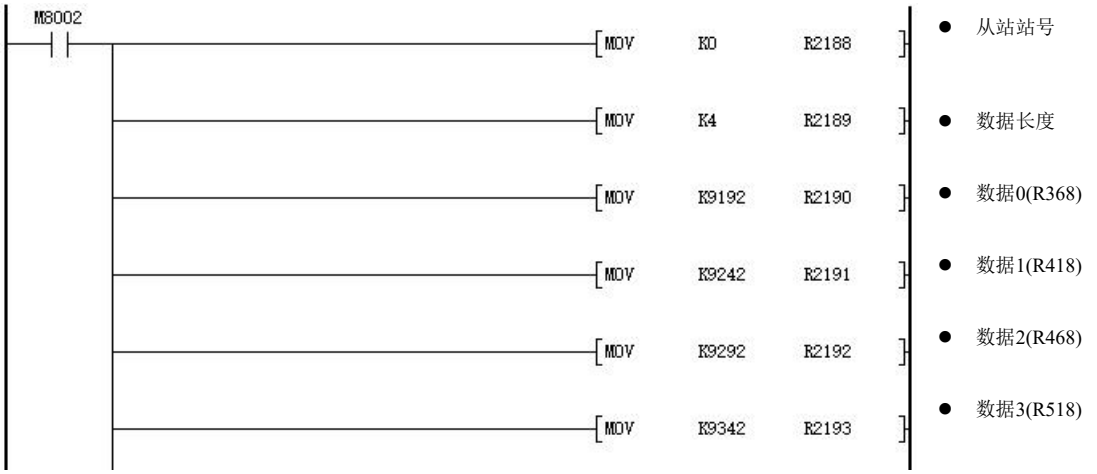
五轴运动控制理论位置分别为R318L，R368L，R418L，R468L，R518L。R318L是本体轴，直接控制所连电机位置。其他四轴作为从轴，通过KLINK将R368，R418，R468，R518传输到从站伺服驱动器（而非脉冲连接），伺服驱动器根据理论位置控制所连电机运转。



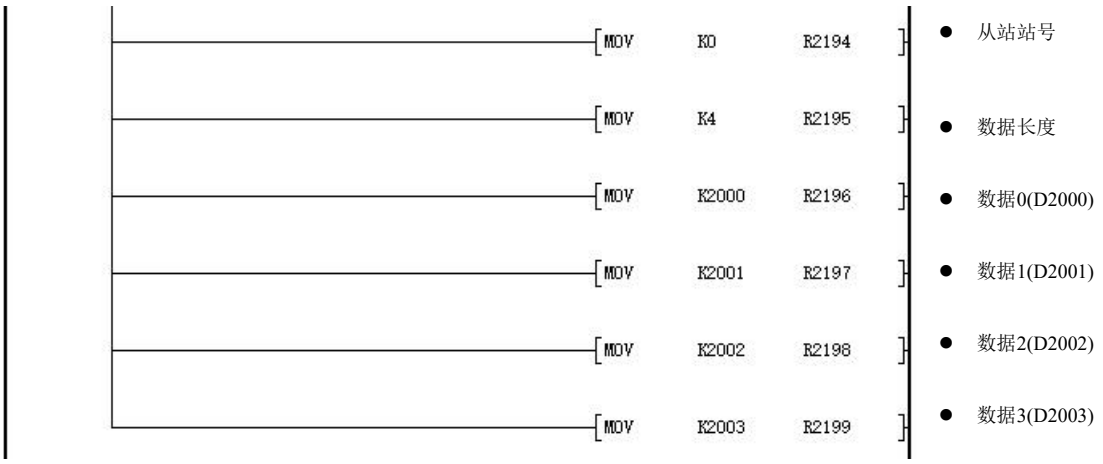
主站通过报文RPT_M4向从站进行广播发送。

通信监测

R2101	主站通信计数，每发一报文+1 每收一报文-1。正常时值较小。
R2104	报文总数。每种报文250us。K4表示通信周期1000 us=1.00 ms。 本例四个报文：RPT_M4广播四个从轴的命令位置，并返回从站K0伺服执行位置，RPT_M1---RPT_M3返回从站K1---K3伺服执行位置
RPT_M4发送报文：向从站K0---K3广播发送	
R2188_站号	K0,允许指定从站以RPT_M4回发报文
R2189_数据长度	K4,向从站传送广播四字数据
R2190_主站资源0	K9192对应R368,总线轴K101当前位置。
R2191_主站资源1	K9242对应R418,总线轴K102当前位置。
R2192_主站资源2	K9292对应R468,总线轴K103当前位置。
R2193_主站资源3	K9342对应R518,总线轴K104当前位置。



RPT_M4接收报文：接收从站K0回发的报文。	
R2194_站号	K0,从站K0向主站发送的RPT_M4报文
R2195_数据长度	K4, 从站K0向主站发送的数据长度=4字
R2196_主站资源0	从站K0回发的数据0存放在D2000。从站K0的当前位置
R2197_主站资源1	从站K0回发的数据1存放在D2001。
R2198_主站资源2	从站K0回发的数据2存放在D2002。
R2199_主站资源3	从站K0回发的数据3存放在D2003。



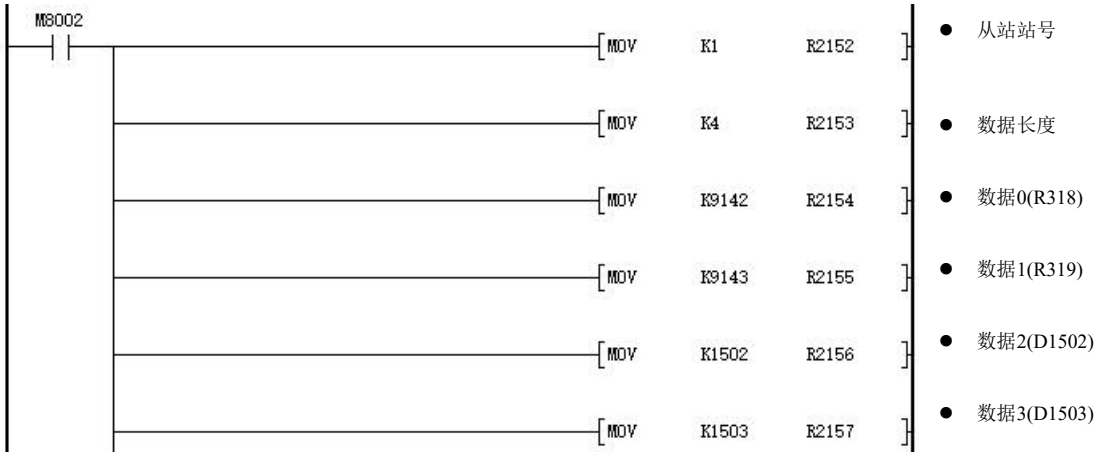
各从站将接收数据R368，R418，R468，R518分别存入各自的D3000,D3001,D3002,D3003。使用上0号从站采用D3000，1号从站采用D3001，2号从站采用D3002，3号从站采用D3003。各从轴通过从站电子齿轮运行来驱动从轴电机。采用16位位置数据传输是许可的，因为内部是循环计数的。

0号从站位置数据通过RPT_M4回发报文返回到时主站D2000外，其他从站（1号，2号，3号）位置数据是主站发起报文PRT_M1---PRT_M3分别与1-3号从站通信，从站回传到主站D2004、D2008、D2012。

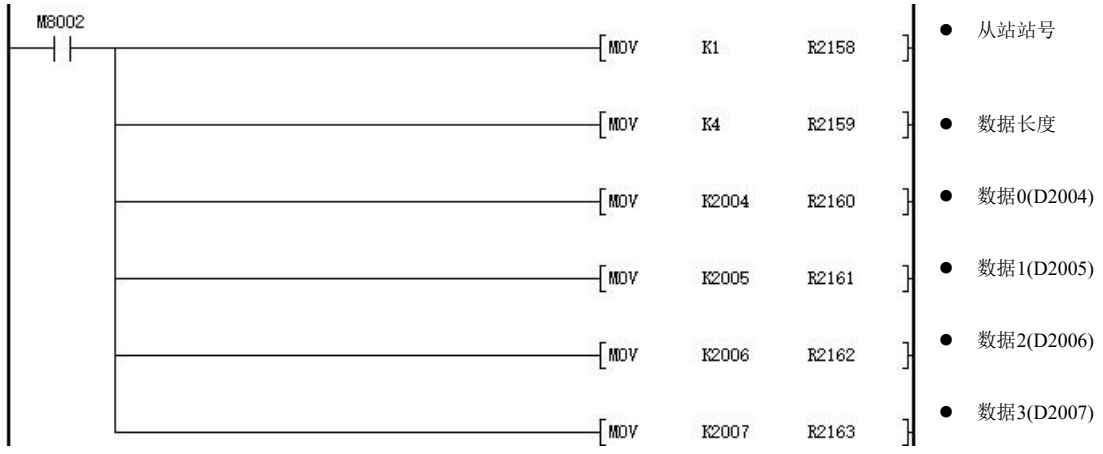
RRT_M1---RPT_M3只是主站发起读取操作，从站对应资源应答，从站并无RRT_M1---RPT_M3的处理程序。

主站RPT_M1报文程序示例如下。

RPT_M1发送报文:



RPT_M1接收报文

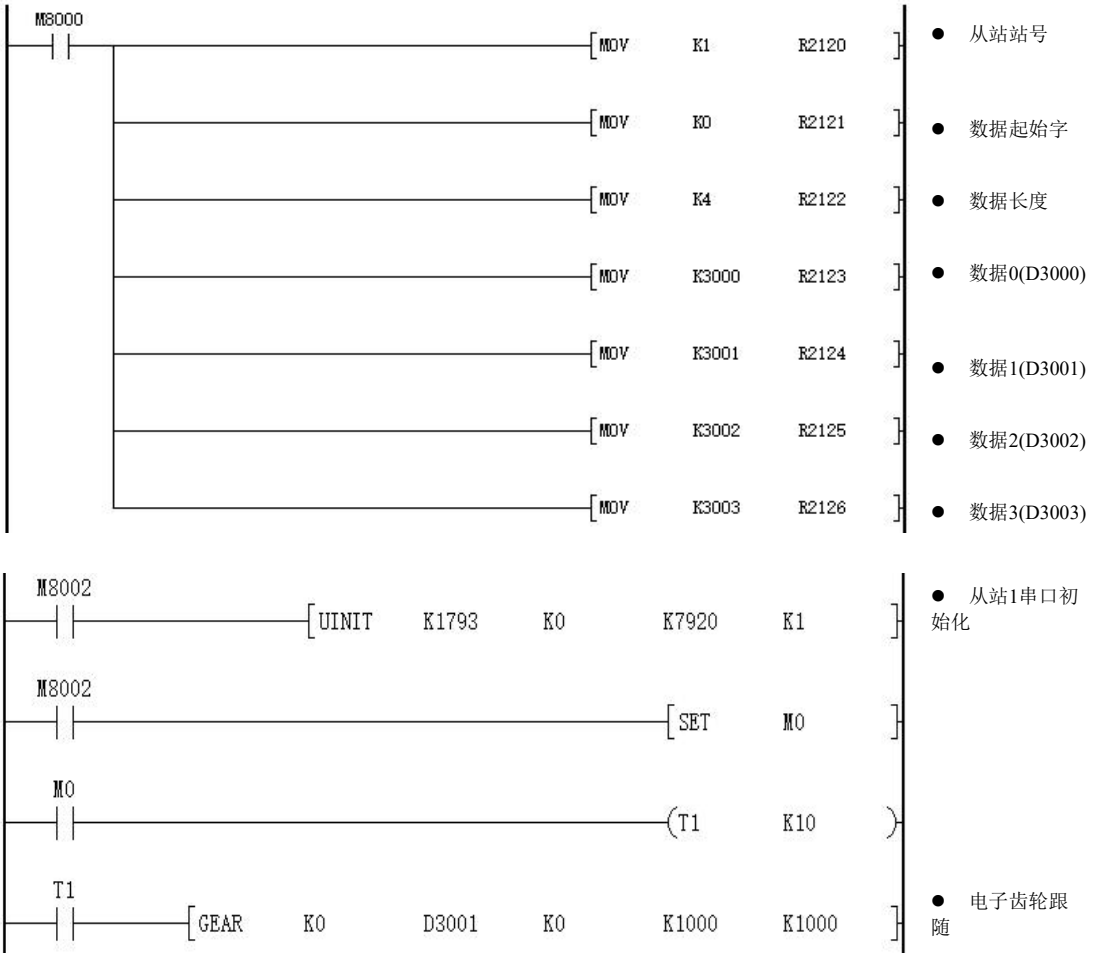


从站站号设置与从轴位置跟随

从站站号设置：从站站号通过串口初始化设置，H700=K1792(0从站）H701=K1793(1从站），H702=K1794(2从站）,H703=K1795(3从站）。

从轴位置跟随：各从站通过电子齿轮指令来跟随主轴的位置命令，主轴的位置命令是通过总线（而非脉冲）传送。从站并无同步工艺程序，同步工艺程序仅在主站上编写。

下图是1号从站的示意程序。其他从站程序除地址不同外，其他均相同。各从站将接收数据R368，R418，R468，R518分别存入各自的D3000，D3001，D3002，D3003。使用上0号从站采用D3000，1号从站采用D3001，2号从站采用D3002，3号从站采用D3003。



监控主轴定位完成后主站发送数据D368、D418、D468、D518与从站伺服返回的位置数据相同。在无插补的情形下，对应数据一致。

各从站通过电子齿轮指令来跟随主轴的位置命令时，建议主站上电时通过RPT_M9号报文初始化从轴的插补值和禁止从站电子齿轮指令使能。

6.6 KLink总线应用注意事项

KLink总线应用过程中，注意如下事项：

1、主从站的串口初始化配置指令必须在主从站控制字寄存器和报文定义字寄存器配置后面执行，否则无法通信。

2、主从站站号设置需一致。

3、RPT_M0--RPT_M3号报文主站传送给从站读取从站数据资源地址的长度（主站资源）、主站接收从站数据将数据刷新至主站对应资源地址的长度（主站资源）、从站RPT_M0--RPT_M3号报文的数据长度（从站资源）三者严格要求一致，设定不一致时，可能导致异常情况。

4、RPT_M4--RPT_M7号报文主站接收从站数据将数据刷新至主站对应资源地址的长度（主站资源）、从站RPT_M4--RPT_M7号报文回复主站的发送数据长度（从站资源）两者严格要求一致，设定不一致时，可能导致异常情况。

5、主站通信任务控制字寄存器R2103，R2104，R2110-R2119支持在线修改，修改后将R2100赋值1修改生效。

6、主站发送接收报文定义字寄存器R2140--R2259支持在线修改，修改即生效。

7、从站控制器寄存器和报文定义字寄存器R2100--R2204支持在线修改，修改即生效。

8、通信时间刻度数R2103建议设置范围5--50，超出此范围可能会有错误报文。

9、通信报文总数R2104<=10，设为0，不通信不报错；R2104<0或>10，通信报错。

10、通信报文列表R2110-R2119 的值必须设0-9范围内，否则通信报错。

11、主站通信计数R2101一个通信报文任务时，在线更改从站站号，此值会清零。

12、主站R2100设置为0停止通信，设置为1启动通信。从站R2100设置为0停止通信，设置为2启动通信。应用时可灵活配置。

7. 其他协议

本章节将要介绍的协议均属于点对点传输的内部协议，这些协议不能用于组建通信网络，所以它们将作为内部协议使用。用户推荐使用前述章节所介绍的协议进行构建通信网络。

HEX传输协议

- 通常应用于下载批量数据，驱动代码或是应用文件时使用。
- 以页为单位进行传输，每帧报文包含1页数据（128 bytes）。
- 以字符型进行传输，一种高速传输的通信方案。通信流程包含下载→比较→固化这3个传输过程。
- HEX传输协议是一个内部的协议，用户无需了解协议结构。

编程口协议

- 梯形图上传/下载以及元件监视的专用协议。
- 以ASCII码型进行传输，每帧报文包含数据可变，但一般不超过64个字节。
- 编程口协议是一个内部的协议，用户无需了解协议结构。

编程口协议兼容三菱FX2N的编程口协议，在使用第三方设备与MC系列PLC进行通信时(比如HMI)，也可选择该协议模式对元件进行监视。

六、扩展模块应用手册

MC系列PLC通过SPI总线方式连接扩展模块，有开关量扩展和模拟量扩展模块，最多可连接六个扩展模块。开关量扩展模块增加I/O点数，模拟量模块在温度、流量、液位、压力等过程控制系统有着广泛应用。

1、 扩展寄存器

MC系列PLC的D8129为扩展寄存器的设置首地址，初始默认值为7800，即默认D7800开始的区域是扩展管理寄存器区。用户可以更改D8129修改缺省的起始地址，PLC重新运行设置生效。

每个扫描周期结束，如有通信任务完成，则进行扩展的数据交换。以D7800为首地址进行示例说明，扩展寄存器区定义如下所示：

D7800～D7806	从站1分配地址 ● D7800：从站的X端子在主站上的起始地址 ● D7801：从站的Y端子在主站上的起始地址 ● D7802～D7803：内部使用 ● D7804：从站的AD值在主站上的起始地址，默认D7000 ● D7805：从站的PGA设置值以及DA值在主站上的起始地址，默认D7256 ● D7806：内部缓存区起始地址
D7807～D7813	从站2分配地址，分配地址参考从站1分配地址
D7814～D7820	从站3分配地址，分配地址参考从站1分配地址
D7821～D7827	从站4分配地址，分配地址参考从站1分配地址
D7828～D7834	从站5分配地址，分配地址参考从站1分配地址
D7835～D7841	从站6分配地址，分配地址参考从站1分配地址
D7842	接入的从站个数
D7843	开关量扩展刷新时间设置(默认60ms)，可在线修改。 ● 定时小于扫描周期时，以扫描周期通信 ● 定时大于扫描周期时，以扫描周期整数倍通信 ● 开关量扩展通信优先级高于模拟量扩展，故该参数不能设置太小，否则将影响模拟量模块的通信
D7844	模拟量扩展刷新时间设置(默认60ms)，可在线修改。 ● 定时小于扫描周期时，以扫描周期通信 ● 定时大于扫描周期时，以扫描周期整数倍通信
D7845	开关量扩展刷新时间实测
D7846	模拟量扩展刷新时间实测
D7847	扩展错误序号 ● =0 正确 ● =1 接入的扩展分配超限 X扩展输入最大限制：128个点 Y扩展输出最大限制：128个点 模拟量接入最大限制：AD/DA数值及其自动分配占用的D寄存器不得超过384个。 ● =2 手动分配的地址超限
D7000～D7767	模拟量扩展数据区

扩展接续起始地址

- 开关量扩展XY地址为主机已占用的地址的下一个8位地址开始，如
  - 16点：开关量扩展X输入从X10开始，开关量扩展Y输出从Y10开始
  - 26点：开关量扩展X输入从X20开始，开关量扩展Y输出从Y20开始
  - 32点：开关量扩展X输入从X30开始，开关量扩展Y输出从Y20开始
  - 40点：开关量扩展X输入从X30开始，开关量扩展Y输出从Y20开始
- 模拟量扩展地址从D7000开始

2、 模拟量模块

模拟量模块的供电电源为DC24V。模拟量模块总数不超过2个时，可用MC系列PLC DC24V给模块供电。

型号	通道数	范围	分辨率	功能
电压・电流输入				
EX1-4AD	4通道	电压：DC 0V~10V	0.6 mV (14位)	电压・电流输入 可混合使用
		电流：DC 0~20mA	1.3uA (14位)	
电压・电流输出				
EX1-4DA	4通道	电压：DC 0V~10V	2.5mV (12位)	电压・电流输出 可混合使用
		电流：DC 0~20mA	5uA (12位)	
电压・电流输入 / 输出混合				
EX1-2AD/2DA	输入 2通道	电压：DC 0V~10V	0.6mV (14位)	电压・电流输入 可混合使用
		电流：DC 0~20mA	1.3uA (14位)	
	输出 2通道	电压：DC 0V~10V	2.5mV (12位)	电压・电流输出 可混合使用
		电流：DC 0~20mA	5uA (12位)	
温度传感器输入				
EX1-4PT	4通道	-30 ℃~500 ℃	0.03 ℃ (14位)	支持铂电阻Pt100
EX1-4TC	4通道	K型：0~1350 ℃	0.1 ℃ (14位)	支持K型、E型、S型、B型热电偶。
		E型：0~600 ℃	0.03 ℃ (14位)	
		S型：0~1600 ℃	0.1 ℃ (14位)	
		B型：60~1800 ℃	0.12 ℃ (14位)	

2.1 模块规格

EX1-4AD（含EX1-2AD2DA的AD输入）产品规格：

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D类接地(接地电阻: 100Ω以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体；导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 200mA；EX1-4AD启动电流1A。（需要在端子排上连接DC24V电源供电）。	
模拟量输入范围	电压输入：DC 0V~+10V； (输入电阻200kΩ)	电流输入：DC 0mA~+20mA； (输入电阻250Ω)
最大绝对输入	15V	30mA
AD值范围	无符号16位 ，稳定14位	无符号16位 ，稳定14位
AD转换时间	80ms/通道（含滤波）；4通道320ms。每通道间存在数字开关切换，不适用于快速测量。	

EX1-4DA（含EX1-2AD2DA的DA输出）产品规格：

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D类接地(接地电阻: 100Ω以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体；导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 200mA；EX1-4DA启动电流1A。(需要在端子排上连接DC24V电源供电) 内部5V电源由DC/DC转换而来。	
模拟量输出范围	DC 0V~10V (外部负载≥2KΩ)	DC 0mA~20mA (外部负载≤500Ω)
DA值范围	12位 二进制	
分辨率	2.5mV(10V/4000)	5uA(20mA/4000)
DA转换时间	<2ms/通道，4通道< 8ms。	

EX1-4TC产品规格：

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D类接地(接地电阻: 100Ω以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体；导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 50mA；EX1-4TC启动电流400mA(需在端子排上连接DC24V电源供电)	
模拟量输入范围	0-55 mV	-

最大绝对输入		-
AD值范围	无符号16位 ，稳定14位	无符号16位 ，稳定14位
AD转换时间	80ms/通道（含滤波）；4通道400ms。每通道间存在数字开关切换，不适用于快速测量。	

EX1-4PT产品规格：

项目	规格
环境温度	0~50℃……工作时 -25~75℃……保存时
相对湿度	5~85%RH(无结露)……工作时
接地	D类接地(接地电阻: 100Ω以下)<不可以和强电系统共同接地>
使用环境	无腐蚀性、可燃性气体；导电性尘埃(灰尘)不严重
工作电源	DC24V±10% 50mA；EX-4PT启动电流400mA。（需要外部24V供电）
输入信号	3线式Pt100
测温范围	-30~500℃
AD值范围	无符号16位 ，稳定14位
AD转换时间	80ms/通道（含滤波）；4通道320ms。每通道间存在数字开关切换，不适用于快速测量。

● 用户可通过修改D7843 的值来改变读取模拟量模块输入转换值的时间，设置范围为1～255，默认值为60ms，即60ms读取一次模拟量输入转换值，设置时应考虑模拟量输入模块的转换速度，设置值应大于模块的转换速度。

2.2 模块读写控制

模拟量模块地址是自动顺序逻辑分配的。

设置值包括扩展模块的PGA设置值(不包括PT100扩展模块)和模拟量输出DA设置值(含DA输出的扩展模块)，另外每个设置值在内存里还分配了相应单元的内存作为读写控制缓存区。

输入PGA设置

模拟量输入信号有电流信号输入(DC 0~20mA)，也有电压信号输入(DC 0V~10V)，还有不同分度号的热电偶输入，因此必须选择输入类型，从而获取对应类型的转换值。一个通道输入类型选择占用一个数据寄存器。

向寄存器写入输入信号的设置类型，和模拟量输出通道设置输出值，同属写入操作，因此输入类型设置和输出值设置，同一顺序编号。

输入信号类型设置：

输入信号类型	K分度	E分度	S分度	B分度	电压	电流
PGA	0	1	2	3	5	6

当输入信号为PT100时，无需设置信号类型，不占用设置数据寄存器。

以下例说明PGA的输入信号设置。序号是从主机向外进行编号的。

型号	输入寄存器		输出寄存器	
EX1-4AD	D7000	AD1标准化值0~20000	D7256	AD1类型选择：电压或电流
	D7001	AD2标准化值0~20000	D7257	AD2类型选择：电压或电流
	D7002	AD3标准化值0~20000	D7258	AD3类型选择：电压或电流
	D7003	AD4标准化值0~20000	D7259	AD4类型选择：电压或电流
EX1-4PT (0.1 ℃)	D7004	PT1温度值	不占用输出寄存器	无需设置
	D7005	PT2温度值		
	D7006	PT3温度值		
	D7007	PT4温度值		
EX1-4TC (0.1 ℃)	D7008	TC1温度值	D7260	TC1设置（0，1，2，3）
	D7009	TC2温度值	D7261	TC2设置（0，1，2，3）
	D7010	TC3温度值	D7262	TC3设置（0，1，2，3）
	D7011	TC4温度值	D7263	TC4设置（0，1，2，3）
EX1-2AD/2DA	D7012	AD1标准化值0~20000	D7264	AD1设置(5，6)
	D7013	AD2标准化值0~20000	D7265	AD2设置(5，6)
			D7266	DA1设置(5，6)
			D7267	DA2设置(5，6)
			D7268	DA1的DA输出值
			D7269	DA2的DA输出值

输出PGA设置

模拟量输出有电流输出(DC 0~20mA)和电压输出(DC 0V~10V)；EX1-4DA或EX1-2AD/2DA均支持电压/电流输出，每路DA按设置的信号类型进行输出，因此每路DA既有类型选择寄存器，也有DA输出寄存器。

输出信号类型	电压	电流
PGA	5	6

以下说明PGA的输出信号设置。序号是从主机向外进行编号的。

型号	输入寄存器		输出寄存器	
EX1-4DA			D7256	DA1类型设置(5，6)
			D7257	DA2类型设置(5，6)
			D7258	DA3类型设置(5，6)
			D7259	DA4类型设置(5，6)
			D7260	DA1的DA输出值(0~4000)
			D7261	DA2的DA输出值(0~4000)
			D7262	DA3的DA输出值(0~4000)
			D7263	DA4的DA输出值(0~4000)
EX1-2AD/2DA	D7000	AD1标准化值0~20000	D7264	AD1类型设置(5，6)
	D7001	AD2标准化值0~20000	D7265	AD2类型设置(5，6)
			D7266	DA1类型设置(5，6)
			D7267	DA2类型设置(5，6)
			D7268	DA1的DA输出值(0~4000)
			D7269	DA2的DA输出值(0~4000)

2.3 输入模拟量

模拟量输入模块的标准化值或温度值，经模块转换后，分别对应到D7200开始的数据寄存器中。由主机向外读取AD标准化值或温度值。在读取输入值前，必须设置输入类型（输入PGA设置）。

如系统有三个模拟量输入模块，由主机向外的顺序是EX1-4AD（电流型输入）、EX1-4PT（蒸汽温度测量）、EX1-4TC（S型电偶、炉膛温度测量）。则相关通道的输入值对应在下列数据寄存器中。

型号	输入信号转换值	输入类型设置
EX1-4AD	D7000 =(0~20000)	D7256=6
	D7001 =(0~20000)	D7257=6
	D7002 =(0~20000)	D7258=6
	D7003 =(0~20000)	D7259=6
EX1-4PT (0.1 ℃)	D7004 =(-300~5000)	PT100无类型选择
	D7005 =(-300~5000)	
	D7006 =(-300~5000)	
	D7007 =(-300~5000)	
EX1-4TC (0.1 ℃)	D7008 =(0~16000)	D7260=2
	D7009 =(0~16000)	D7261=2
	D7010 =(0~16000)	D7262=2
	D7011 =(0~16000)	D7263=2

2.4 输出模拟量

输出模拟量模块可设置成电压输出，也可设置成电流输出，因此，每一个通道的DA输出之前，也必须进行输出信号类型设置（输出PGA设置）。PGA设置和DA值输出均是用户向PLC规定的数据寄存器写数据，规划到以D7256开始的连续区域。对输出而言，先设置类型，后设置输出。对混合模块（有输入和输出）而言，先分配输入类型寄存器，次分配输出类型寄存器，最后分配DA输出值寄存器。

假设系统有一个EX1-4DA（两路电压输出，两路电流输出），则信号类型设置寄存器和DA值输出寄存器分配如下：

型号	输出信号类型设置	DA值输出寄存器
EX1-4DA	D7256=5（电压）	D7260=(0~4000)
	D7257=5（电压）	D7261 =(0~4000)
	D7258=6（电流）	D7262 =(0~4000)
	D7259=6（电流）	D7263 =(0~4000)

输入输出混合

模拟量输出有电流输出(DC 0~20mA)和电压输出(DC 0V~10V)；EX1-4DA或EX1-2AD/2DA均支持电压/电流输出，每路DA按设置的信号类型进行输出，因此每路DA既有类型选择寄存器，也有DA输出寄存器。

假设系统挂有EX1-4AD（2路电压输入、2路电流输入）、EX1-4TC（2路K分度、2路S分度）、EX1-2AD2DA（2路电压输入、2路电流输出）、EX1-4DA（4路均为电流输出）共四个模拟量模块。

数据寄存器分配应符合以下三条：

- 对每个模块（无论输入输出）而言，必须先设置信号类型（先输入后输出），再读取AD值或输出DA值。
- 当输入信号是PT100时，不进行信号类型设置。
- AD值的读取是从D7000开始，顺次向后排序。

信号类型设置和DA值输出是从D7256开始，顺次向后排序。

对上述系统进行数据寄存器分配。

型号	输入类型设置	输出类型设置	AD值寄存器	DA值寄存器
EX1-4AD	D7256=5（电压）		D7000=(0~20000)	
	D7257=5（电压）		D7001=(0~20000)	
	D7258=6（电流）		D7002=(0~20000)	
	D7259=6（电流）		D7003=(0~20000)	
EX1-4TC	D7260=0（K）		D7004=(0~13500)	
	D7261=0（K）		D7005=(0~13500)	
	D7262=2（S）		D7006=(0~16000)	
	D7263=2（S）		D7007=(0~16000)	
EX1-2AD2DA	D7264=5（电压）	D7266=6（电流）	D7008=(0~20000)	D7268=(0~4000)
	D7265=5（电压）	D7267=6（电流）	D7009=(0~20000)	D7269=(0~4000)
EX1-4DA		D7270=6（电流）		D7274=(0~4000)
		D7271=6（电流）		D7275=(0~4000)
		D7272=6（电流）		D7276=(0~4000)
		D7273=6（电流）		D7277=(0~4000)

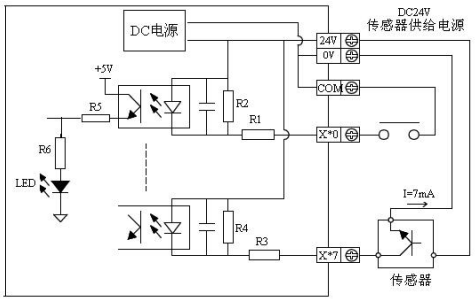
3、 开关量模块

MC系列PLC最多可接受总和184个X输入扩展点和184个Y输出扩展点。

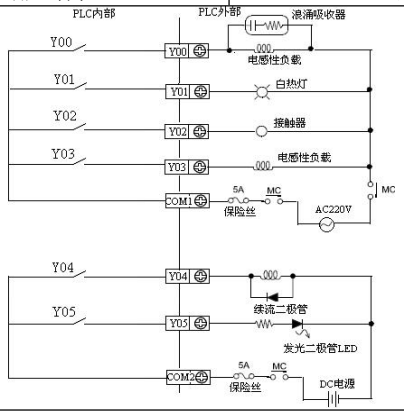
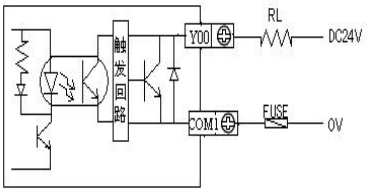
型号	供电电源	合计点数	输入输出特性			
			普通输入	输入电压	普通输出	输出方式
EX1-08M	DC24V	8	8	DC24V	/	/
EX1-08R	总线供电	8	/	/	8	继电器
EX1-08T	总线供电	8	/	/	8	晶体管
EX1-16M	DC24V	16	16	DC24V	/	/
EX1-16R	总线供电	16	/	/	16	继电器
EX1-16T	总线供电	16	/	/	16	晶体管
EX1-16M16R	AC220V	32	16	DC24V	16	继电器
EX1-16M16T	AC220V	32	16	DC24V	16	晶体管

3.1 模块规格

开关量输入模块：

项目	规格
	X020~X267
输入电压	DC24V±10%
输入电流	4.5mA
输入阻抗	5.1 k Ω
输入ON电流	2.3mA以上
输入OFF电流	2.2mA以下
输入响应时间	约16 ms
输入信号形式	无电压触点，或NPN集电极开路晶体管
输入电路隔离	光电耦合器隔离
输入状态显示	光电耦合器被驱动时面板LED灯亮
	

开关量输出模块：

项目	继电器输出 Y20~Y267	晶体管输出 Y20~Y267
外部电源	AC250V；DC30V 以下	DC5~30V
电路绝缘	机械绝缘	光耦绝缘
动作显示	继电器线圈通电时面板LED灯亮	光耦驱动时面板LED灯亮
最大电阻负载	2A/点 8A/4点公共端 8A/8点公共端	0.5A/点； 0.8A/4点公共 1.6A/8点公共
最大感性负载	80VA （注1）	12W/DC24V
最大指示灯负载	100W （注2）	1.5W/DC24V
开路漏电流	-	0.1 mA /DC30V
最小负载	DC5V 2 mA	-
响应时间OFF->ON	约10ms	0.2ms以下
响应时间ON->OFF	约10ms	0.2ms以下
	负载容量越大，继电器输出接点寿命越短，负载容量＜30VA，接点寿命大于20万次。 因输出接点OFF时，没有电流，所以可以直接驱动氖灯等。 当继电器输出型驱动直流回路的负载为感性（如继电器线圈）时，用户电路需并联续流二极管。 当继电器输出驱动交流回路的负载为感性时，用户电路需并联RC浪涌吸收电路，以保护输出继电器触点。 原则上继电器输出端口不宜接入容性负载，若有必要，需保证其冲击浪涌电流小于规格说明中的最大电流。
	晶体管输出端口须遵守允许最大电流限制，以保证输出端口的发热限制在允许范围。若有多个晶体管端口输出电流大于100mA，则应均匀分布于输出端口，不宜安排在相邻的输出端口，以便散热；建议同时为ON（导通）状态的输出点数不要长时间超过总输出点数的70%。

3.2 模块读写控制

开关量模块和模拟量模块配置在一起时，开关量和模拟量的地址分配相互不影响，各自独立分配。

MC系列PLC接入开关量模块时，地址是按照8进制递推顺序自动分配的，以字节为单位，主机或模块上未满8位的按8位算。

比如顺序接入EX1-08M，EX1-08R，EX1-16M，EX1-16R，那么主机和各模块对应的地址分配如下：

连接顺序	产品型号	输入地址	输出地址
主机	MC-6M16T	X000~X007 X010~X017	Y000~Y007 Y010~Y017
模块一	EX1-08M	X020~X027	/
模块二	EX1-08R	/	Y020~Y027
模块三	EX1-16M	X030~X037 X040~X047	/
模块四	EX1-16R	/	Y030~Y037 Y040~Y047

七、运动指令使用手册

1、绪论

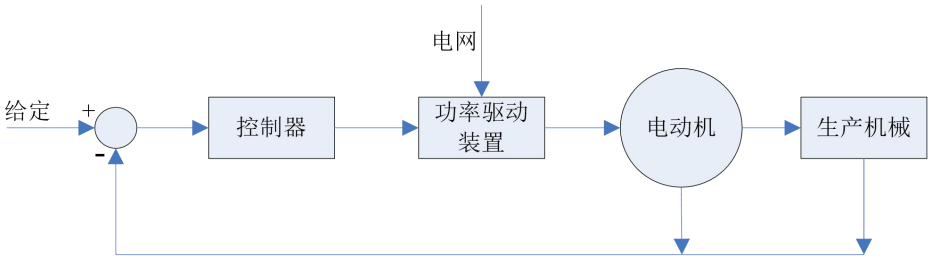
MC系列PLC是以 PLCOpen 为背景进行的模块化研发，以梯形图为编程语言，以指令的形式进行编程的。这些指令的应用适用于大多数非数控类的运动系统。

1.1 运动系统类型

1.1.1 运动控制系统简述

运动控制系统是以电动机及其拖动的机械设备为控制对象，控制器为核心，电力电子功率变换装置为执行机构，组成的电气传动系统。这类系统通过对电机的转矩、转速和转角的控制，将电能转化为机械能，实现机械的各种运动要求，如定位、调速、随动等。

运动控制系统基本组成部分包括：控制器、功率驱动装置和电动机。



功率驱动装置包括变频器、步进驱动器、伺服驱动器。

步进驱动器、伺服驱动器都能接收脉冲信号作为命令信号。

1.1.2 运动控制系统类型

运动控制系统含括以下两种：

轨迹规划型：以工件为对象，进行轨迹和形状加工。标准数控（如数控机床通用加工设备）和开放式的非标数控（如线切割、行业机械手等专业加工设备）属于该类型，其外接信号少。

工位型：在生产线上，依据前后设备状态和周边信号状况，以同步、定长等方式实现控制，满足工艺要求，通常要求效率高。如印刷同步（同频同相），定长裁切（含追剪、滚切）。

本书介绍的MC系列PLC运动控制是工位型的，主要是输出脉冲信号给伺服驱动器或步进驱动器，从而满足机械设备的工艺控制要求。

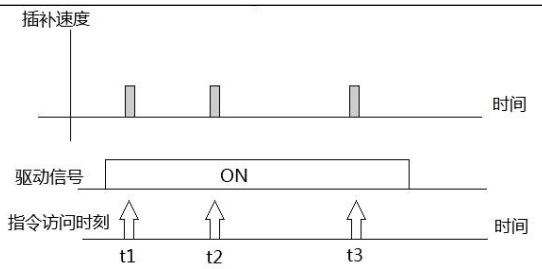
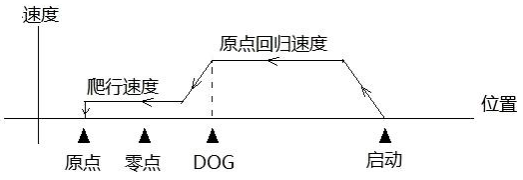
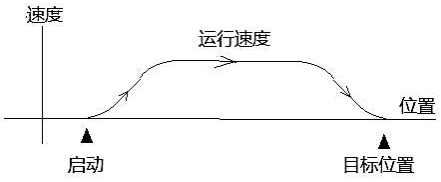
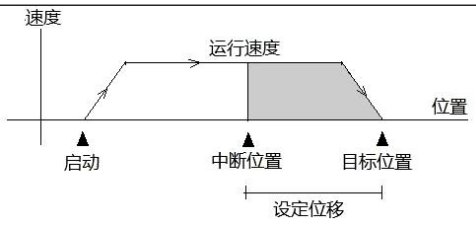
1.2 MC系列PLC运动功能简介

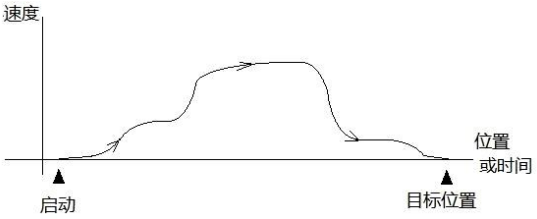
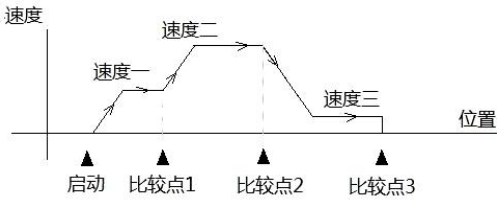
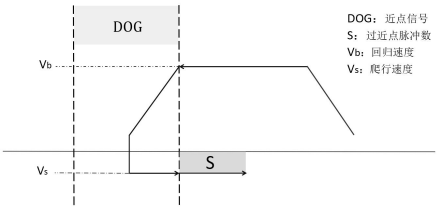
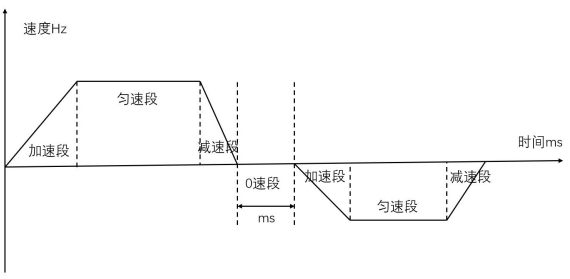
采用梯形图语言编程，以轴指令和轴参数方式向伺服驱动器、步进驱动器等功率驱动装置输出脉冲（最多四路）信号，完成定位控制。以脉冲频率控制对象的移动速度，脉冲数控制对象的移动量。单轴脉冲频率最高 200KHz，四轴同时运行各轴频率可达 200KHz。

下面对MC系列PLC的运动功能作简要介绍。

1.2.1 单轴定位

（一）、单轴定位
运动指令（插补指令IPL 除外）驱动触点接通后，运动脉冲的演算周期为 1ms，不受梯形图扫描时间影响。

指令名称	功能图示	简要说明
插补运行IPL		IPL 指令受访问时间影响，访问一次（驱动触点闭合），插补一次，其插补量为 1ms 的脉冲数。 常用于相位修正。
机械原点回零 HOME		机械原点是系统工作的绝对参考点。 在此基础上，用户建立工作原点或工作坐标系。 从机械原点返回到工作原点，使用其他快速定位指令。
梯形定位 DDRVI/DDRVA/DPLSR		带加减速的定位
四次多项式曲线 定位PSOUT		改善启动和停止的平稳性，减少机械振动和噪声，有助于延长机械寿命。
中断定位 DDVIT		运行过程中，检测到中断信号时，移动一段设定位移后停止。

表格输出定位 PTOUT		表格数据来源： 由PTAB指令制作。
任意多段速定位DPLSV /DPLSY		可任意变更速度。
简易机械原点回零 HOMA		回机械原点后，反向出原点。
多段速定位 DPTO		多段以起始速度和终止速度结合每段脉冲数计算出速度均匀变化的多段定位。 而且多段速中支持存在一个匀速中断段，可实现中断定位方式。

1.2.2 轴关联定位

（二）、轴关联定位

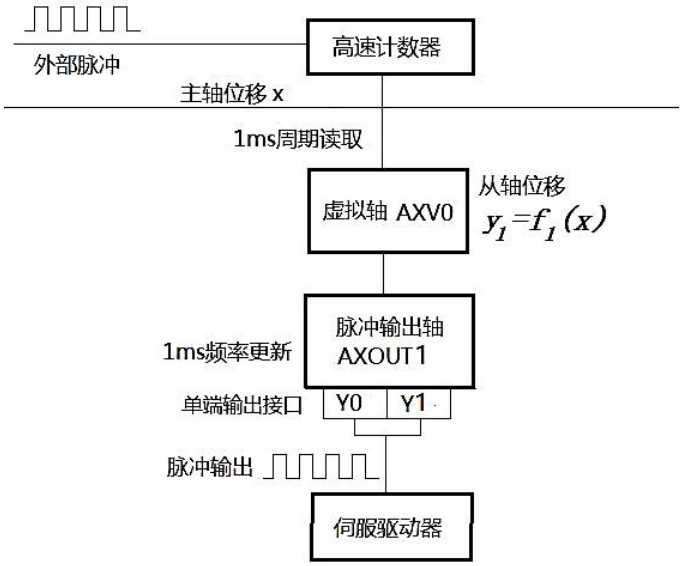
以一个轴（主轴）的位移作为参考（x），另外的轴（从轴）按设定的关系进行位置移动（y），用关系式表示 y = f (x) 。

主轴可以是：①外部脉冲输入端口（如高速计数器）；②控制器的脉冲输出端口（包含实际脉冲输出端口 AXOUT0~AXOUT3 和存储器脉冲输出端口 AXOUT4~AXOUT11）；③程序设置的数字输出（如每ms 输出脉冲数）。

从轴是控制器的脉冲输出端口。

一个主轴可以连接多个从轴，但一个从轴只能有一个主轴。

一个从轴又可以成为下一级的主轴（级联）。



虚拟轴 AXV0~AXV11：系统以每ms的节拍读取输入数据，按指令功能计算，输出计算结果。计算结果可由①实输出轴 AXOUT0~AXOUT3转化成脉冲输出；②数字输出轴 AXOUT4~AXOUT11 作为中间引用值存储；③作为次级运动主轴的运行数据。

脉冲输出轴AXOUT0~AXOUT3：将虚拟轴的计数结果转化成脉冲，并经过脉冲端口对外输出并进行计数。

脉冲端口在本控制器设计成单端输出型式。

主从关系 y = f (x) 的具体表达：

1、数学表达式：

$y=\frac{A}{B}\cdot x$； 电子齿轮指令 GEAR 是实现比例控制的运动控制指令。
$y = A \cdot \sin x$
$y = B \cdot \cos x$
.....

表格表达：

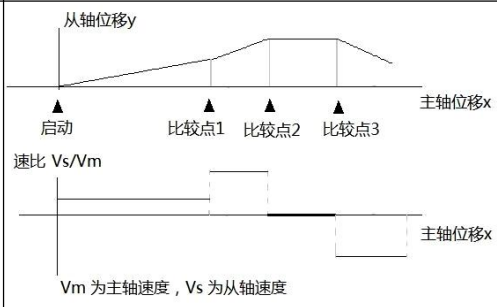
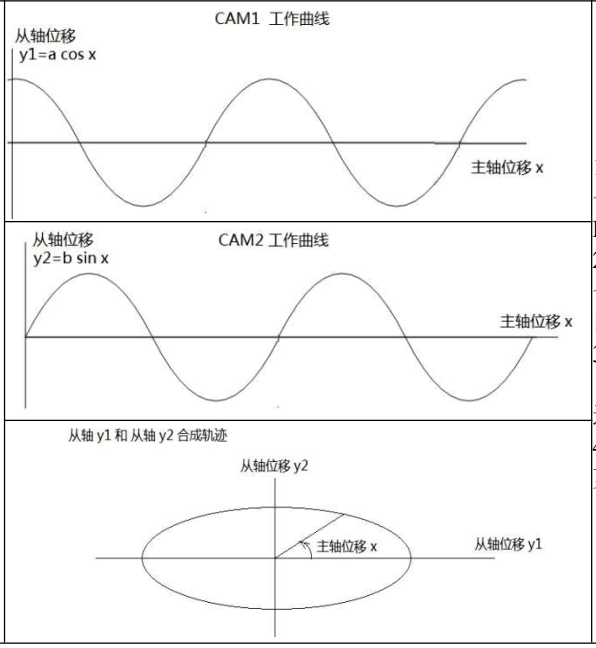
表格点数：N+1

主轴 x	0	1dx	2dx	.....	(N-1)dx	N dx
从轴 y	Y0=Y0	Y1=Y0+dy1	Y2=Y1+dy2	.....	Y(N-1)=Y(N-2)+dy(N-1)	Y(N)=Y(N-1)+dy(N)

一般情况下，从轴初始位置坐标 Y0=0；但也有初始坐标为最大的情况，如 $y = B \cdot \cos x$ 。

表格可以用来描述包含数学表达式在内的任意周期性单值对应关系。凸轮指令 CAM 是以表格为基础的运动控制指令。

与关联运行的运动指令有电子齿轮指令 GEAR 和凸轮指令 CAM。为方便使用提供了相关辅助指令。

指令名称	功能图示	简要说明
电子齿轮指令 GEAR		通常所指的两轴直线插补。 电子齿轮可以在线修改。 与表格比较指令 DHSCT 配合使用，可以在定点处修改电子齿轮比。
电子凸轮指令 CAM		1、用表格来描述主从运动关系的指令。制表辅助指令有PCTAB，PBTAB，PTAB。 2、凸轮指令是主轴位移的周期指令，一个周期内可以设置<=128 个比较点的 16 个位软元件输出。 3、以主轴为媒介，多个从轴位移可以合成圆（或椭圆），螺旋线等轨迹。 4、单一凸轮指令多应用于滚切和追剪等定长裁切。

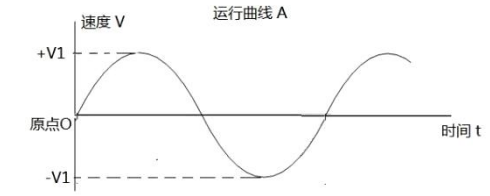
1.2.3 运动的复合

（三）、运动的复合

在不同轴上分别描述的运动过程，在一个轴上进行叠加输出或者是接续输出。

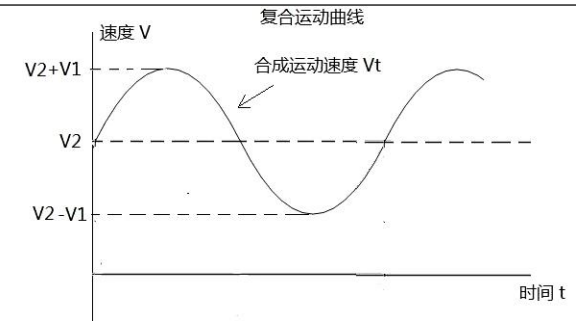
1、运动的叠加

独立运动过程





复合运动过程



说明：

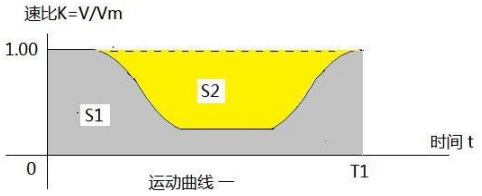
主轴速度通常是从轴在交会点处应该达到的速度。从轴在一个周期节拍内要求的平均速度是V2。有时需要升速至同步速（V=V2+V1），有时需要降速至同步速（V=V2-V1）。

2、运动的接续

运动过程一

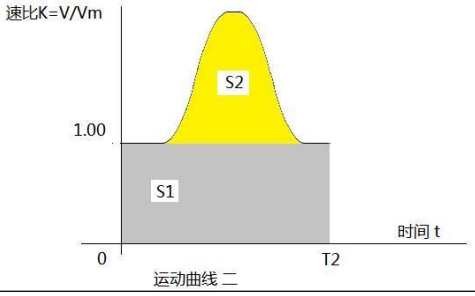
Vm 为主轴速度，对应速比线恒为 1。

曲线描述的是从轴速度与主轴速度的速比。该过程从轴脉冲数小于主轴脉冲数，对应滚切过程中是长料。

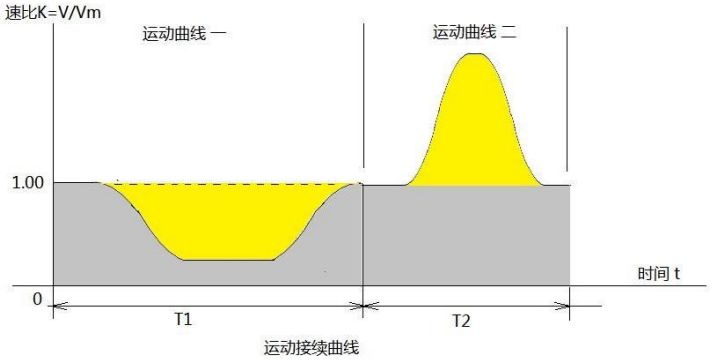


运动过程二

该过程从轴脉冲数大于主轴脉冲数，对应滚切过程中是短料。



运动一和运动二的接续运行过程



在生产过程中，经常出现长、短料在一条包装线上进行包装。因此系统根据物料的测量长度，更换与之对应的包装凸轮，两个不同长度的物料之间的运行过程切换，就是运行接续的例子。

3、实现复合运动的参数配置指令 AXPO

虚拟轴：

共有12个虚拟轴（AXV0-----AXV11）， 负责运动指令的解析计算，按 1 ms 节拍输出计算结果，其输出结果只能连接到一个输出轴。将一个虚拟轴重新连接到一个新的实轴上，原有的连接即被断开。一般情况下， 在梯形图首个扫描周期，一次性连接固定，即完成端口分配。

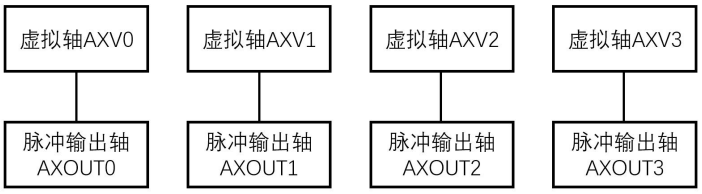
输出轴：

共有4个外部端子接口的脉冲输出轴（AXOUT0----- AXOUT3）和8个内存接口的数字输出轴（AXOUT4------AXOUT11）。

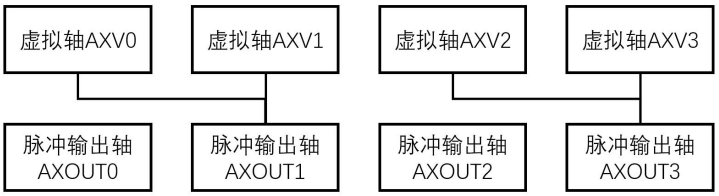
数字输出轴作中间轴使用。

外接脉冲输出轴将虚拟轴的计算值按 1 ms 节拍转化成脉冲，供伺服驱动器位置（或速度）控制使用。一个输出轴可以同时连接多个虚拟轴。正因为如此，MC系列PLC才具备复合运动的功能。

默认连接示意图：



配置连接示意图：



图示中 AXOUT1（外接端口 Y1）复合了AXV0+AXV1 的运动功能。

AXOUT3（外接端口Y3）复合了 AXV2+AXV3 的运动功能。

未连接 AXV 的脉冲输出轴可作为通用的输出端口供梯形图编程使用，如AXOUT0 等效于 Y0。

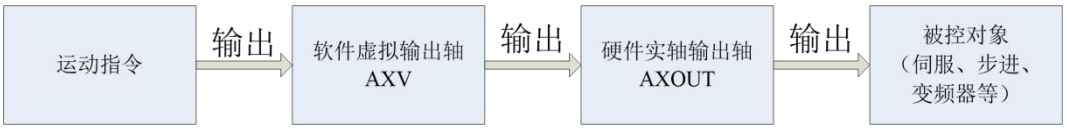
2、运动控制模型

MC系列PLC是以“运动指令”为方法、“轴”为载体、“PLC 梯形图语言”为手段三者结合实现一个或者多个运动对象的控制。

一个运动对象的控制涉及输入到输出，输入是对运动指令的配置以及调用，输出是调用的运动指令通过 控制器硬件端口以脉冲的形式输出到被控的执行机构，运动指令的输入配置以及输出指定是以“轴”为中间载体进行连接的，以 PLC 梯形图语言实现的。

载体“轴”包含输入轴（软件输入虚拟轴和硬件输入实轴）、输出轴（软件虚拟输出轴和硬件输出实轴），如果将输出轴的软件虚拟轴同硬件输出轴进行绑定，那么软件虚拟输出轴所调用的运动指令最终将以脉冲的形式通过硬件输出轴输出到被控的执行机构。

MC系列PLC控制外部执行机构的流程如下：

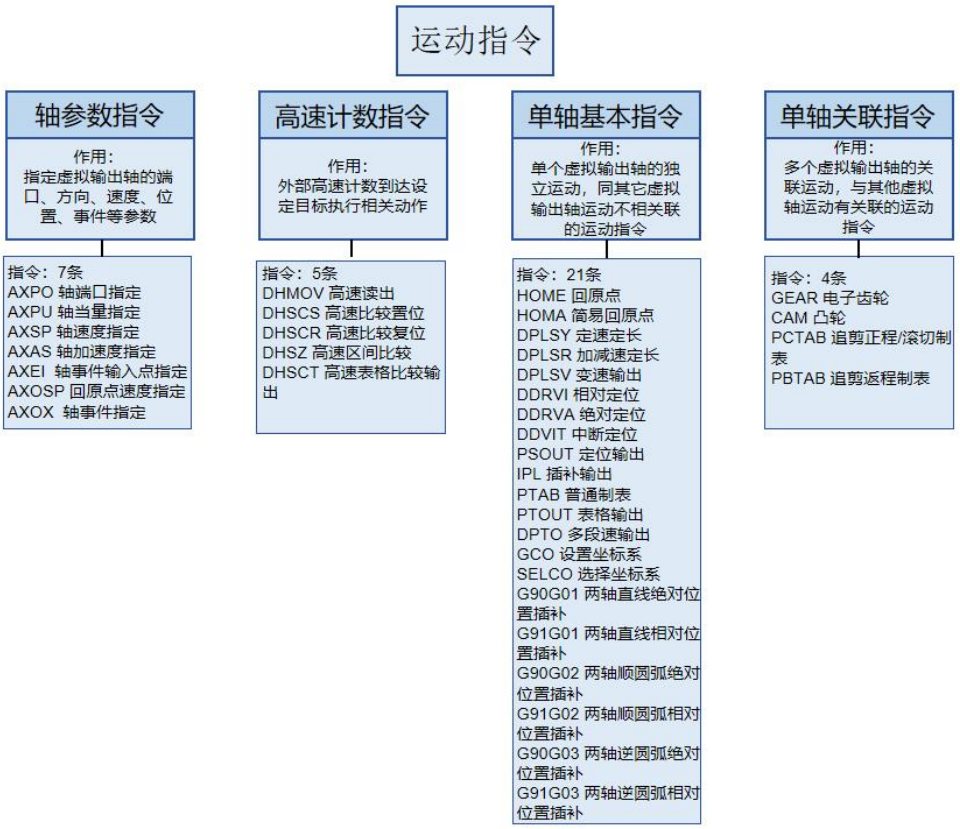


2.1 运动参数概念

主要介绍和定义运动指令涉及到的相关概念，包括运动指令、输入轴、输出轴、虚拟轴、数字轴、实轴、事件等相关概念。

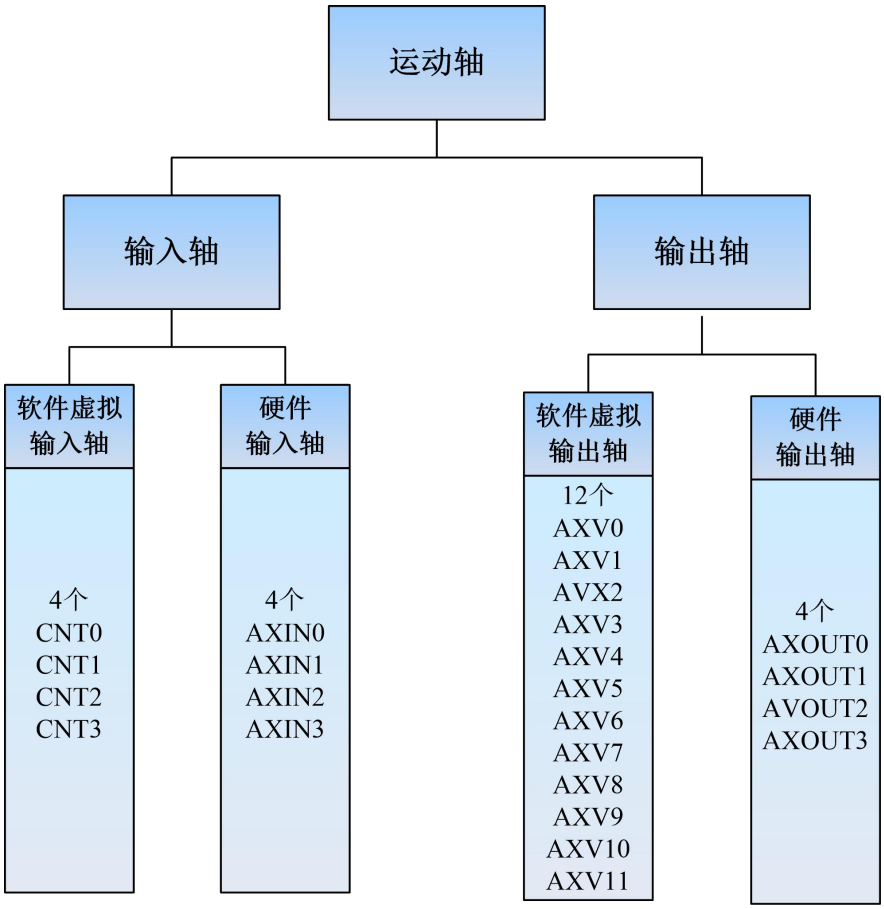
2.1.1.运动指令

MC系列PLC的运动指令分为四大部分，分别为轴参数指令、高速计数指令、单轴基本指令以及单轴关联指令，详细指令和作用如下图：



2.1.2.运动轴

MC系列PLC的运动轴分为两大部分，分别为输入轴、输出轴， MC系列PLC相关的轴定义如下图。

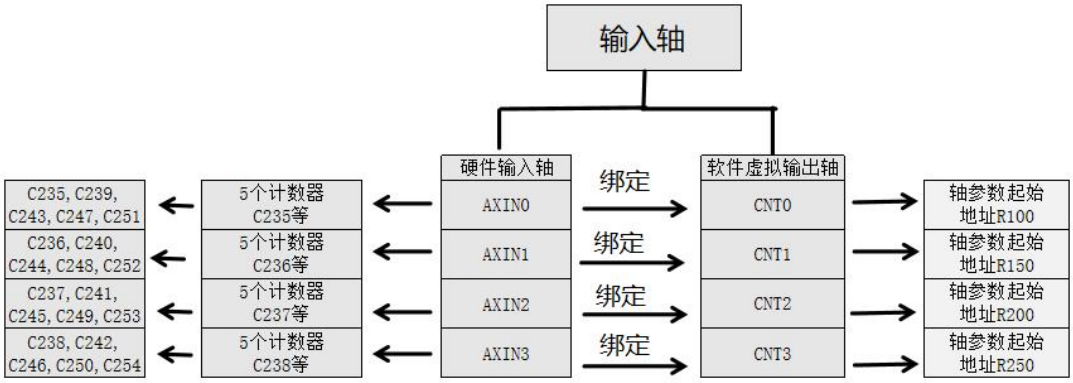


- 软件虚拟输入轴：MC系列PLC将相应的 C 寄存器作为软件虚拟输入轴的输入源。
- 硬件输入轴：针对控制器四路高速计数硬件端口 X0 -X7 而言，MC系列PLC系统内部自动将这四路硬件输入轴绑定到四个软件虚拟轴。
- 软件虚拟输出轴：运动指令所执行的运动曲线输出的对象。
- 硬件输出轴：将控制器内部的运动曲线输出到控制器的硬件端口。
- 数字输出轴：将控制器内部的运动曲线输出到控制器系统内部，不进行对外输出。

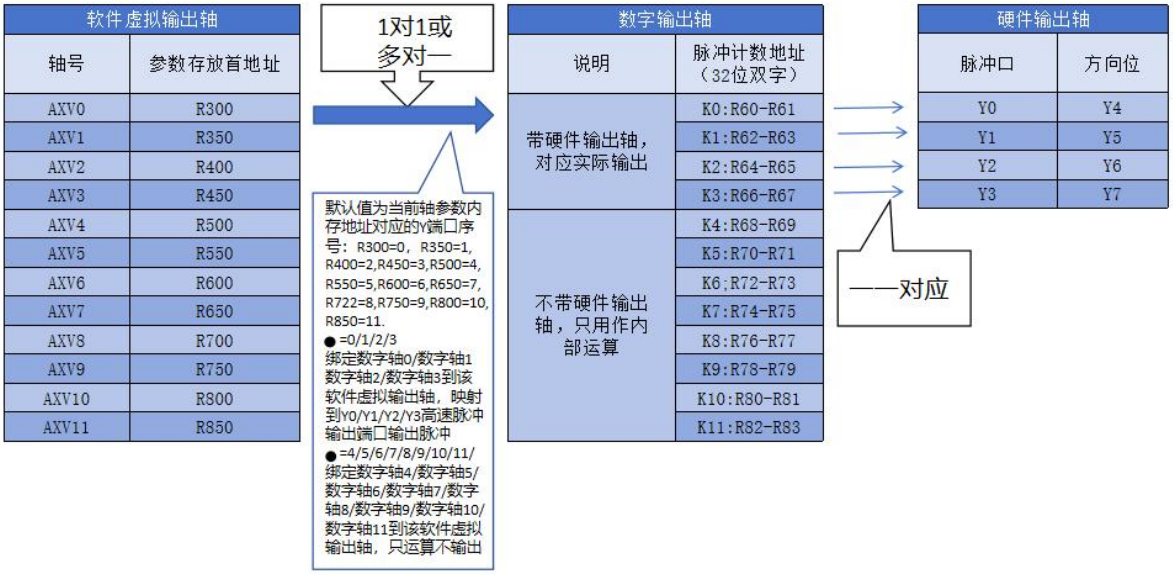
2.1.3.运动轴之间的绑定关系

本小节主要介绍 MC系列PLC的运动轴之间的绑定关系。

- 输入轴：硬件输入轴和软件虚拟输入轴是一一对应的固定绑定的关系。
- 硬件输入 AXIN0 绑定到软件虚拟输入轴 CNT0，软件虚拟轴 CNT0 的轴参数地址是R100 起始的连续 50 个R 寄存器
- 硬件输入 AXIN1 绑定到软件虚拟输入轴 CNT1，软件虚拟轴 CNT1 的轴参数地址是R150 起始的连续 50 个R 寄存器
- 硬件输入 AXIN2 绑定到软件虚拟输入轴 CNT2，软件虚拟轴 CNT2 的轴参数地址是R200 起始的连续 50 个R 寄存器
- 硬件输入 AXIN3 绑定到软件虚拟输入轴 CNT3，软件虚拟轴 CNT3 的轴参数地址是R250 起始的连续 50 个R 寄存器
- 输入轴之间的关系如下图：



- 输出轴之间的相互绑定关系：软件虚拟输出轴可以一对一或者多对一的绑定到硬件输出端口或者数字输出轴，但是软件虚拟输出轴不能够一对多的绑定到硬件输出端口或者数字轴，数字输出轴和硬件输出轴之间不能够相互绑定。
- 每个软件虚拟输出轴（12个）都对应的有确定轴参数配置地址，轴参数地址是轴起始地址连续的 50个 R 寄存器
- 每个数字输出轴（12个）都有一个双字的 R 寄存器来存放端口累积输出脉冲数值（带正负号，正脉冲数值增加，负脉冲数值减小）
- 每个数字输出轴（12 个）都有一个双字的 R 寄存器来存放轴的累积脉冲数值（带正负号，正脉冲数值增加，负脉冲数值减小）
- 输出轴之间的关系如下图：

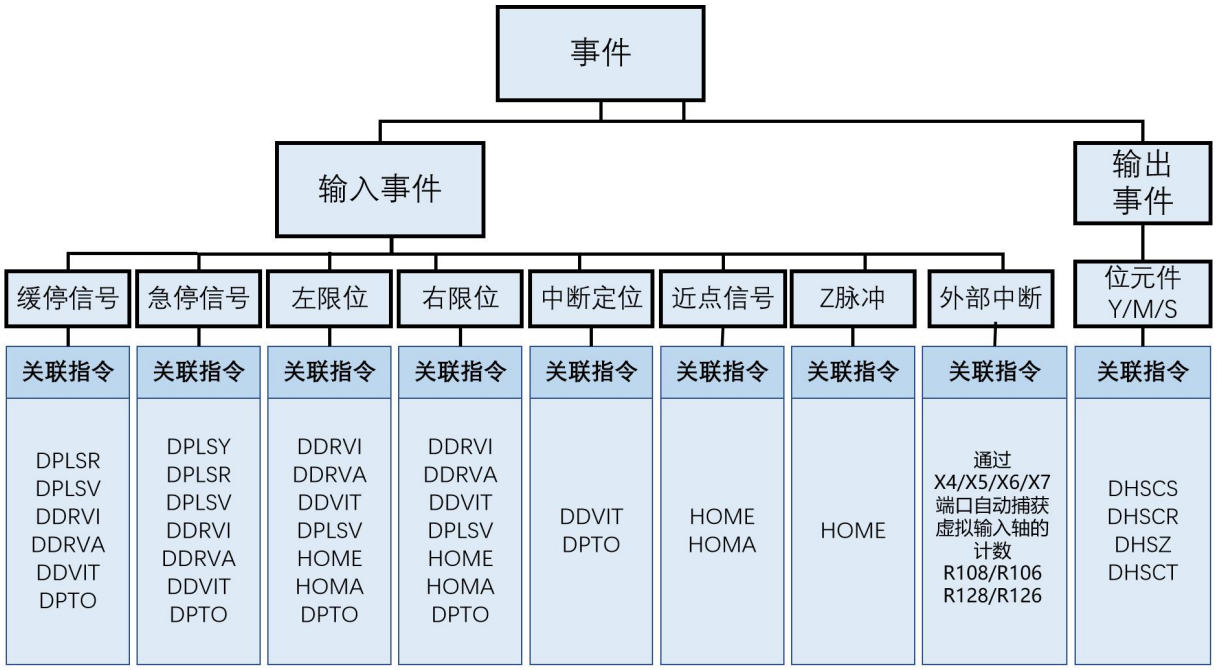


2.1.4.运动指令事件

控制器虚拟输出轴在执行运动过程中可以被外部输入事件中断或者输出高速事件控制外部执行机构。

- 输入事件：外部中断信号、急停信号、左限位信号、右限位信号、中断信号、近点信号、Z 脉冲信号

- 输出事件：比较事件输出，一般为控制器的位资源，常用为控制器输出 Y 点，主要在高速比较指令中应用。



正转：轴插补增量为正值，轴相对位置轴位置增加，此情形定义为轴正转。正转时若轴脉冲输出方向使能，则脉冲方向口ON。
反转：轴插补增量为负值，轴相对位置轴位置减小，此情形定义为轴反转。反转时若轴脉冲输出方向使能，则脉冲方向口OFF。
左限位：反转极限，左限位信号有效时，轴只能正转，不能反转。
右限位：正转极限，右限位信号有效时，轴只能反转，不能正转。

2.1.5.硬件轴输入输出频率

MC系列 PLC 4 路硬件输入轴和 4 路硬件输出轴都有对应的最高频率限制。

- 4 路硬件输入轴的最高频率如下表：

说明：4 路硬件输入轴的计数方式有四种，分别为 AB 相计数、脉冲+方向计数、增加数、减计数。				
硬件端口	AB 相计数		脉冲+方向计数/增加数/减计数	
	一路	四路	一路	四路
X0/X1 X2/X3	200K	200K	200K	200K
X4/X5 X6/X7		200K		200K

- 4路硬件输出轴的最高频率如下表：

说明：4 路硬件输出轴的频率为 200K，如果有 N（N=1,2,3,4）路硬件输出轴同时配置了脉冲输出功能，不管有没有执行运动指令，每路的最高脉冲频率为200K，例如配置了 Y0/Y1/Y2 这三路为脉冲输出功能端口，但是在实际使用时只有两路或者一路同时执行发脉冲功能，这三个输出端口的最高频率依旧是200K。

<table>
<tr><th>硬件端口</th><th>一路输出</th><th>两路输出</th><th>三路输出</th><th>四路输出</th></tr>
<tr><td rowspan="4">Y0/COM1
Y1/COM2
Y2/COM3
Y3/COM3</td><td rowspan="4">200K</td><td rowspan="2">200K</td><td>200K</td><td>200K</td></tr>
<tr><td rowspan="2">200K</td><td>200K</td></tr>
<tr><td rowspan="2">200K</td><td>200K</td><td>200K</td></tr>
</table>

2.2.轴参数定义

本小节主要介绍软件虚拟输入轴参数定义以及软件虚拟输出轴的参数定义。

2.2.1. 输入轴参数

输入轴参数配置指令对软件虚拟输入轴进行配置。轴参数在 PLC 的内存分配地址如下表所示(每个轴分配 50 个 R 寄存器)：

轴号	对应参数存放的内存首地址
虚拟输入轴0	R100
虚拟输入轴1	R150
虚拟输入轴2	R200
虚拟输入轴3	R250

轴参数配置的方式：**梯形图使用 MOV 指令对其寄存器直接进行填写**

- 轴参数定义

轴首地址+偏移	名称	参数描述	单位
轴首地址+0(L)	高速计数绝对值	该虚拟输入轴接入的脉冲计数绝对值。 单位为脉冲。该参数为 32 位数值，占用 2 个 R 寄存器。该参数无需设置，为显示值。	脉冲
轴首地址+2(L)	高速计数绝对值高 32 位	该虚拟输入轴接入的脉冲计数绝对值的高 32 位。 该参数无需设置，为显示值。	脉冲
轴首地址+4(L)	保留使用	保留使用。	-
轴首地址+6(L)	X11/X13/X15/X17 外部中断时刻计数值	R106为当X11产生外部中断时，虚拟输入轴0接入的脉冲计数值。 R156为当X13产生外部中断时，虚拟输入轴1接入的脉冲计数值。 R206为当X15产生外部中断时，虚拟输入轴0接入的脉冲计数值。 R256为当X17产生外部中断时，虚拟输入轴1接入的脉冲计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+8(L)	X10/X12/X14/X16 外部中断时刻计数值	R108为当X10产生外部中断时，虚拟输入轴0接入的脉冲计数值。 R158为当X12产生外部中断时，虚拟输入轴1接入的脉冲计数值。 R208为当X14产生外部中断时，虚拟输入轴2接入的脉冲计数值。 R258为当X16产生外部中断时，虚拟输入轴3接入的脉冲计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+10	脉冲计数方式	该虚拟输入轴的脉冲计数方式。在首次输出(OUT)或切换高速计数器时更新 ● = 0，AB 相计数 ● = 1，脉冲+方向计数 ● = 2，增计数 ● = 3，减计数	-
轴首地址+11	脉冲捕获方式	设置该虚拟输入轴的脉冲捕获方式。（注：下例中HX0中X代表0或1均可） ● HX0：R111 对应X10上升沿，R161 对应X12上升沿。 R211 对应X14上升沿，R261 对应X16上升沿。 ● HX1：R111 对应X10下降沿，R261 对应X12下降沿。 R111 对应X14下降沿，R261 对应X16下降沿。 ● H0X：R111 对应X11上升沿，R261 对应X13上升沿。 R211 对应X15下降沿，R261 对应X17下降沿。 ● H1X：R111 对应X11下降沿，R261 对应X13下降沿。 R211 对应X15下降沿，R261 对应X17下降沿。	-

轴首地址+12	周期 1MS 计数值	该虚拟输入轴接入的脉冲 1MS 周期计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+13	保留使用	保留使用。	
轴首地址+14(L)	DHSCT输出时刻计数值	R114为当DHSCT输出时，虚拟输入轴0接入的脉冲计数值。 R164为当DHSCT输出时，虚拟输入轴1接入的脉冲计数值。 R214为当DHSCT输出时，虚拟输入轴2接入的脉冲计数值。 R264为当DHSCT输出时，虚拟输入轴3接入的脉冲计数值。 该参数无需设置，为显示值。	
轴首地址+16 ～ 轴首地址+49	保留，用户无需配置。		

2.2.2. 输出轴参数

输出轴参数配置指令对软件虚拟输出轴进行配置，其配置对象为 AXV0～AXV11。轴参数在 PLC 的内存分配地址如下表所示(每个轴分配 50 个 R寄存器)：

轴号	对应参数存放的内存首地址
AXV0	R300
AXV1	R350
AXV2	R400
AXV3	R450
AXV4	R500
AXV5	R550
AXV6	R600
AXV7	R650
AXV8	R700
AXV9	R750
AXV10	R800
AXV11	R850

轴参数配置的方式：

① 梯形图使用 MOV 指令对其寄存器直接进行填写

② 梯形图使用轴参数指令进行设置(下述 3.2 章节介绍)

- 轴参数定义

轴首地址+偏移	名称	参数描述	单位
轴首地址+0	轴输出绑定	指定轴脉冲输出端口 AXOUT0～AXOUT11，对应数值 0～ 11。默认值为当前轴参数内存地址对应的 Y 端口序号： R300=0，R350=1，R400=2，R450=3，R500=4，R550=5， R600=6，R650=7，R700=8，R750=9，R800=10，R850=11。 ● = 0/1/2/3/4/5/6/7，绑定 Y00/Y01/Y02/Y03/Y04/Y05/Y06/Y07高速脉冲输出端口到该轴 ● = 8/9/10/11，绑定数字轴 8/数字轴 9/数字轴 10 /数字轴 11 到该轴	-

<table>
<tr><td>轴首地址+1</td><td>轴脉冲输出方向使能</td><td>使能轴脉冲输出方向端口。默认值为 0。该参数配置方式如下：
● = 0，不绑定方向端口到该虚拟输出轴
● = 1，方向端口绑定情况如下表所示：
<table>
<tr><th>轴输出绑定 =</th><th>输出端口</th><th>脉冲方式</th><th>脉冲方向口</th><th>方向方式</th></tr>
<tr><td>0</td><td>Y0/COM1</td><td>集电极</td><td>Y4</td><td>集电极</td></tr>
<tr><td>1</td><td>Y1/COM2</td><td>集电极</td><td>Y5</td><td>集电极</td></tr>
<tr><td>2</td><td>Y2/COM3</td><td>集电极</td><td>Y6</td><td>集电极</td></tr>
<tr><td>3</td><td>Y3/COM3</td><td>集电极</td><td>Y7</td><td>集电极</td></tr>
</table>
* 如果该虚拟轴绑定的是数字轴，则轴脉冲输出方向使能填 0 即可。</td><td>-</td></tr>
<tr><td>轴首地址+2</td><td>轴时间当量</td><td>● 轴时间当量设置，默认值为1000，不建议修改，是用来衡量后述速度/加速度参数的计量标准。
● 例：脉冲指令频率设置为1000时，实际脉冲频率为1000Hz。
● 设置为负数时，则取 16 位无符号数作为计算。比如设置-1，则认为为 65535。</td><td>ms</td></tr>
<tr><td>轴首地址+3</td><td>轴脉冲当量</td><td>该参数保留使用，可随意填写该值，默认值 0。</td><td>脉冲 / 轴长度当量</td></tr>
<tr><td>轴首地址+4</td><td>轴长度当量</td><td>该参数保留使用，可随意填写该值，默认值 0。</td><td>mm</td></tr>
</table>

轴首地址+偏移	名称	参数描述	单位
轴首地址+5	轴最高限速	轴运行最高速度限制。默认值为20000（200kHz）。该值必须大于 0。 计算公式：（以0轴为例） 设置轴最高限速（R305）= $\frac{\text{实际轴最高速度（Khz）}\times\text{轴时间当量（R302）}}{10}$ 例：需设置轴最高速度为1kHz，即每ms发送1个脉冲，在默认轴时间当量（1000）下，轴起始速度设置值应为= $\frac{1\text{（Khz）}\times 1000}{10}$=100。	脉冲 / 轴时间当量
轴首地址+6	轴起始速度	轴运行起始速度(起跳速度)。默认值0，该值必须大于等于0。 计算公式：（以0轴为例） 设置轴起始速度（R306）= $\frac{\text{实际轴起始速度（Khz）}\times\text{轴时间当量（R302）}}{10}$ 例：需设置轴起始速度为1kHz，即每ms发送1个脉冲，在默认轴时间当量（1000）下，轴起始速度设置值应为= $\frac{1\text{（Khz）}\times 1000}{10}$=100。 为保证控制对象的平稳启动，该参数推荐填0。	脉冲 / 轴时间当量
轴首地址+7	轴最大加减速	轴运行最大加/减速度。默认值为1000（10pls/ms² ） 计算公式：（以0轴为例） 设置轴最大加减速（R307）= $\frac{\text{实际轴最大加减速（pls/}ms^2\text{）}\times\text{轴时间当量（R302）}^2}{10000}$ 例：设置加速度为1脉冲/ms²，默认轴时间当量=1000时，那么轴最大加减速应设置为$\frac{1\times 1000^2}{10000}$=100。 应结合负载变化设置该参数数值，对较轻负载，较大加减速能获得良好的加减速性能；对较重负载，加减速不合理（较大）会导致实际输出始终滞后。	脉冲 / 轴时间当量 ^{2}

<table>
<tr><td>轴首地址+8</td><td>近点信号端口</td><td>近点信号输入端口指定（按位设置，数值为 X 端口掩码），回原点需配置该参数。默认值为 0。
<table>
<tr><th>近点信号端口 =</th><th>选定端口</th><th>近点信号端口 =</th><th>选定端口</th></tr>
<tr><td>0</td><td>不选定端口</td><td colspan="2"></td></tr>
<tr><td>1</td><td>X0/M8096</td><td>256</td><td>X10/M8104</td></tr>
<tr><td>2</td><td>X1/M8097</td><td>512</td><td>X11/M8105</td></tr>
<tr><td>4</td><td>X2/M8098</td><td>1024</td><td>X12/M8106</td></tr>
<tr><td>8</td><td>X3/M8099</td><td>2048</td><td>X13/M8107</td></tr>
<tr><td>16</td><td>X4/M8100</td><td>4096</td><td>X14/M8108</td></tr>
<tr><td>32</td><td>X5/M8101</td><td>8192</td><td>X15/M8109</td></tr>
<tr><td>64</td><td>X6/M8102</td><td>16384</td><td>X16/M8110</td></tr>
<tr><td>128</td><td>X7/M8103</td><td>-32768</td><td>X17/M8111</td></tr>
</table>
● 通过轴事件输入点.Bit13 可使用M8096～M8111 代替X00～X17 的控制。
● 端口响应频率为 2KHz。</td><td>-</td></tr>
<tr><td>轴首地址+9</td><td>Z 脉冲信号端口</td><td>反馈编码器 Z 脉冲信号输入端口指定，回原点需要配置该参数。默认值为 0。
端口选定设置值参考近点信号端口。</td><td>-</td></tr>
<tr><td>轴首地址+10</td><td>左限位端口</td><td>运动轴左限位端口指定，回原点需要配置该参数。默认值为 0。
端口选定设置值参考近点信号端口。</td><td>-</td></tr>
<tr><td>轴首地址+11</td><td>右限位端口</td><td>右限位端口指定，回原点需要配置该参数。默认值为 0。
端口选定设置值参考近点信号端口。</td><td>-</td></tr>
<tr><td>轴首地址+12</td><td>原点回归速度</td><td>原点回归速度指定，回原点需要配置该参数。默认值为 0。
计算方法参考轴最高限速参数说明。</td><td>脉冲 / 轴时间当量</td></tr>
<tr><td>轴首地址+13</td><td>原点爬行速度</td><td>原点爬行速度指定，回原点需要配置该参数。默认值为 0。
计算方法参考轴最高限速参数说明。</td><td>脉冲 / 轴时间当量</td></tr>
</table>

<table>
<tr><th>轴首地址+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>轴首地址+14</td><td>轴事件输入点</td><td>
轴输入点属性指定。默认值为0。按位设置模式功能，其定义如下：
<table>
<tr><th>回原点模式.Bit</th><th>定义</th></tr>
<tr><td>Bit0</td><td>= 0：近点信号端口不使能
= 1：近点信号端口使能</td></tr>
<tr><td>Bit1</td><td>= 0：Z脉冲信号端口不使能
= 1：Z脉冲信号端口使能</td></tr>
<tr><td>Bit2</td><td>= 0：左限位端口不使能 = 1：左限位端口使能</td></tr>
<tr><td>Bit3</td><td>= 0：右限位端口不使能 = 1：右限位端口使能</td></tr>
<tr><td>Bit4</td><td>近点信号端口极性设置。
= 0：近点信号端口信号进入为ON，离开为OFF
= 1：近点信号端口信号进入为OFF，离开为ON</td></tr>
<tr><td>Bit5</td><td>Z脉冲信号端口极性设置。
参考近点信号端口极性设置。</td></tr>
<tr><td>Bit6</td><td>左限位信号端口极性设置。
参考近点信号端口极性设置。</td></tr>
<tr><td>Bit7</td><td>右限位信号端口极性设置。
参考近点信号端口极性设置。</td></tr>
<tr><td>Bit8</td><td>中断信号端口极性设置。
参考近点信号端口极性设置。</td></tr>
<tr><td>Bit9</td><td>= 0：中断信号端口不使能
= 1：中断信号端口使能</td></tr>
<tr><td>Bit10</td><td>急停信号端口极性设置。
参考近点信号端口极性设置。</td></tr>
<tr><td>Bit11</td><td>= 0：急停信号端口不使能
= 1：急停信号端口使能</td></tr>
<tr><td>Bit12</td><td>= 0：急停产生时立即停止
= 1：急停产生时减速停止
● 配置为1时，必须对应的运动指令支持减速停止
● 指令前置条件断开停车方式与此设置关联</td></tr>
<tr><td>Bit13</td><td>输入信号源选择。
= 0：指向M8096～M8111
= 1：指向X00～X17</td></tr>
<tr><td>Bit14</td><td>回原点时，近点信号到开始脉冲计数时的边沿选择。
=0：下降沿 =1：上升沿</td></tr>
<tr><td>Bit15</td><td>回原点（HOME）停止方式(过近点信号后生效)：
=0：遇到近点信号后先计数过近点Z脉冲数个Z脉冲，然后再计数过近点脉冲数个脉冲停止回原点流程。
=1：遇到近点信号后计数过近点 Z 脉冲数个 Z 脉冲，或计数过近点脉冲数个脉冲回原点流程。计数过近点脉冲数个脉冲停止回原点流程。
简易回原点（HOMA）停止方式：
=0：停止在近点信号内，详细见简易回原点模式一。
=1：停止在近点信号外，详细见简易回原点模式二。</td></tr>
</table>
● 若端口不使能，则其极性设置和端口指定均不生效。
例如轴事件输入点.Bit2 = 0，则轴事件输入点.Bit6和左限位端口设置均无效。
</td><td></td></tr>
</table>

<table>
<tr><td>轴首地址+15</td><td>过近点脉冲数</td><td>设置离开近点信号后需要计数的脉冲数再执行停止回原点流程。默认值为0。
该参数为16位无符号数，超过32767将以负数形式设置/显示。</td><td>脉冲</td></tr>
<tr><td>轴首地址+16</td><td>过近点 Z 脉冲数</td><td>设置离开近点信号后需要计数的 Z 脉冲数再执行停止回原点流程。默认值为 0。
该参数为 16 位无符号数，超过 32767 将以负数形式设置/显示。</td><td>脉冲</td></tr>
<tr><td>轴首地址+17</td><td colspan="3">保留，用户无需配置。</td></tr>
<tr><td>轴首地址+18(L)</td><td>轴相对位置</td><td>轴当前的相对位置，为当前轴位置插补增量的累积值。每个插补周期（1ms）更新 1 次。单位为脉冲。该参数为 32 位数值，占用 2 个 R 寄存器。
该参数无需设置，为显示值。</td><td>脉冲</td></tr>
<tr><td>轴首地址+20</td><td>轴状态</td><td>显示轴当前的运行状态。
<table>
<tr><th>轴状态.Bit</th><th>定义</th></tr>
<tr><td>Bit0</td><td>/</td></tr>
<tr><td>Bit1</td><td>/</td></tr>
<tr><td>Bit2</td><td>/</td></tr>
<tr><td>Bit3</td><td>/</td></tr>
<tr><td>Bit4</td><td>当前轴的运行状态
= 0：当前轴没有输出脉冲
= 1：当前轴运动指令正在执行，正在输出中</td></tr>
<tr><td>Bit5</td><td>当前轴指令的完成状态
= 0：当前轴未执行或正在执行运动指令
= 1：当前轴运动指令已执行完成</td></tr>
<tr><td>Bit6</td><td>显示当前轴的报警状态
= 0：当前轴无报警
= 1：当前轴错误报警</td></tr>
<tr><td>Bit7</td><td>当前轴急停事件状态
= 0：未发生轴急停事件
= 1：发生轴急停事件</td></tr>
<tr><td>Bit8</td><td>指令执行时（脉冲输出中）驱动触点是否断开
= 0：驱动触点未断开
= 1：驱动触点断开</td></tr>
<tr><td>Bit9</td><td>/</td></tr>
<tr><td>Bit10</td><td>/</td></tr>
<tr><td>Bit11</td><td>/</td></tr>
<tr><td>Bit12</td><td>/</td></tr>
<tr><td>Bit13</td><td>/</td></tr>
<tr><td>Bit14</td><td>/</td></tr>
<tr><td>Bit15</td><td>/</td></tr>
</table>
可以通过当前寄存器中的显示值判断当前轴的状态，例如：
= 0 ：当前轴未被占用，可对该轴执行运动指令，其他数值情况下执行该轴的运动指令将被阻塞，不会生效，IPL 指令除外。
= 16：当前轴有运动指令正在执行</td><td>-</td></tr>
</table>

<table>
<tr><td></td><td></td><td>= 32：当前轴的运动指令执行完成，但该轴仍被当前指令所占用
= 64：该轴参数设置/运行出错，出错原因见轴出错代码
= 144：该轴正在执行缓停
= 160：该轴处于急停状态
= 400：该轴脉冲输出中前置触点断开，正在减速停车（急停方式配置为缓停），停车完成后，自动将轴状态清 0。
在使用该寄存器判断当前运动指令轴状态时，建议将该寄存器转换为位元件判断例如Bit5是否置位。
当轴状态中指令的完成状态（bit5）为1时，说明针对该轴的某条运动指令或者急停事件已经执行完毕，此时可断开运动指令的前置触点，断开前置触点后，该指令对当前轴的占用状态被清除，此时可再次对该轴执行运动指令。
当梯形图扫描到该运动指令断开后，自动将轴状态清 0。
该参数无需设置，为显示值。</td><td></td></tr>
<tr><td>轴首地址+21</td><td>轴出错代码</td><td>轴参数设置/运行出错代码。
<table>
<tr><th>轴出错代码 =</th><th>定义</th></tr>
<tr><td>0</td><td>无错</td></tr>
<tr><td>1</td><td>当前轴速度超硬件支持的极限速度</td></tr>
<tr><td>2</td><td>未绑定输出轴</td></tr>
<tr><td>3</td><td>轴最高限速超硬件支持的极限速度</td></tr>
<tr><td>4</td><td>当前轴速度低于轴起始速度</td></tr>
<tr><td>5</td><td>当前轴速度超过原点回归速度，回原点时检测</td></tr>
<tr><td>6</td><td>当前轴速度低于原点爬行速度，回原点时检测</td></tr>
<tr><td>7</td><td>轴起始速度大于轴最高限速</td></tr>
</table>
该参数无需设置，为显示值。</td><td>-</td></tr>
<tr><td>轴首地址+22</td><td>轴位置插补增量</td><td>轴上一个插补周期(1ms)的位置增量。每个插补周期更新 1 次。
该参数无需设置，为显示值。</td><td>脉冲</td></tr>
<tr><td>轴首地址+23</td><td>轴急停事件</td><td>轴急停事件端口指定，按位进行设置。默认值为 0。
端口选定设置值参考近点信号端口。</td><td>-</td></tr>
<tr><td>轴首地址+24</td><td>轴中断事件</td><td>轴中断事件端口指定，按位进行设置。默认值为 0。
端口选定设置值参考近点信号端口。</td><td></td></tr>
<tr><td>轴首地址+25</td><td colspan="3">保留，用户无需配置。</td></tr>
<tr><td>轴首地址+26(L)</td><td>轴中断位置</td><td>轴产生中断事件时的相对位置。
该参数无需设置，为显示值。</td><td>脉冲</td></tr>
<tr><td>轴首地址+28~49</td><td colspan="3">保留，用户无需配置。</td></tr>
</table>

2.3.虚拟轴初始化

MC系列PLC有 16个软件虚拟轴，分为两种类型：12 个软件虚拟输出轴和 4 个软件虚拟输入轴。

MC系列PLC运动指令总共有 38条，具体指令请参见运动章节第 3章，其中有 15 条运动指令的操作数包含软件虚拟轴。在使用这些运动指令之前必须对软件虚拟轴进行初始化。

2.3.1 软件虚拟输入轴初始化

通过启动对应的高速计数器，MC系列PLC自动将硬件输入轴与软件虚拟输入轴进行绑定，软件虚拟输入轴对应的寄存器的数值每 1ms 更新一次，这样就实现了两个软件虚拟输入轴的初始化。

虚拟输入轴号	对应参数存放的寄存器空间	高速计数器编号	硬件输入对应端口
轴0	R100~R149	C235、C239、C243、C247、C251	X0、X1
轴1	R150~R199	C236、C240、C244、C248、C252	X2、X3
轴2	R200~R249	C235、C241、C245、C249、C253	X4、X5
轴3	R250~R299	C238、C242、C246、C250、C254	X6、X7

2.3.2 软件虚拟输出轴初始化

软件虚拟输出轴初始化的主要任务是对硬件输出轴号、轴脉冲输出方向端口、轴时间当量、轴运行最高速度、轴运行起始速度、轴运行最大加/减速度等参数进行配置，其中轴运行最高速度和最大加速度要低于输出执行机构（如伺服电机、步进电机）和机械设备能够承受的范围。

例如：伺服电机额定转速 3000rpm，电机编码器 2500 线，电机转一圈对应 10000 个脉冲，如果伺服驱动器设置电子齿轮比为 5:1，则输入脉冲频率为 100Khz 时，电机转速即达到 3000rpm。所以配置硬件输出轴最高运行速度不能超过 100Khz。

使用轴参数配置指令（AXPO、AXPU、AXSP、AXAS）对软件虚拟输出轴相关参数进行初始化，具体指令用法请参见运动章节第 3 章，也可以直接对轴参数对应的寄存器进行赋值，完成初始化过程。软件虚拟输出轴初始化必须在梯形图第一个运行周期之内完成。

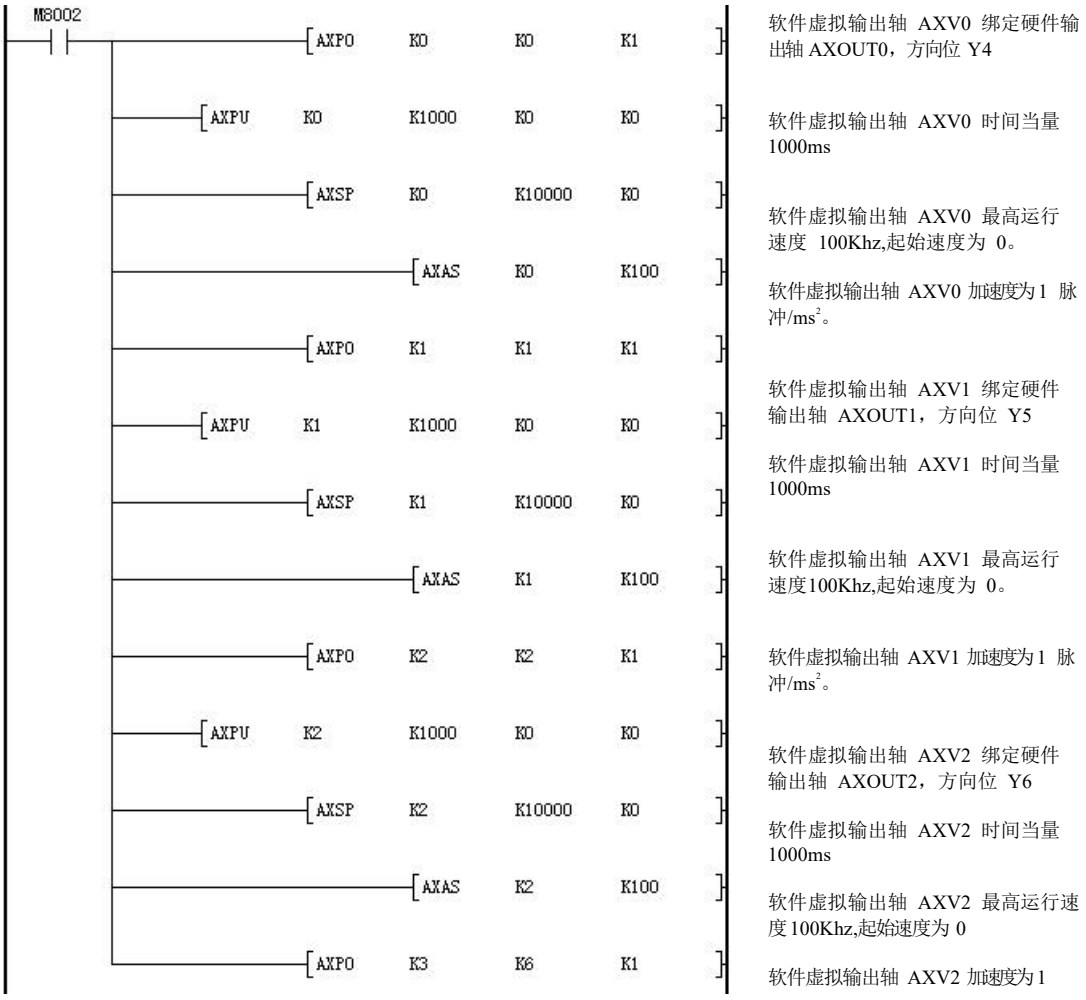
如果用户程序没有对 4个软件虚拟输出轴进行初始化，则使用系统默认的初始化配置参数。默认配置参数如下表所示：

<table>
<tr><th>软件虚拟输出轴轴号</th><th>硬件输出轴轴号</th><th colspan="2">轴脉冲输出方向端口</th><th>轴时间当量(ms)</th><th colspan="2">轴运行最高速度</th><th colspan="2">轴运行起始速度</th><th colspan="2">轴运行最大加/减速度</th></tr>
<tr><td>AXV0</td><td>AXOUT0 (R300=0)</td><td>R301=0</td><td rowspan="8">不配置脉冲输出方向端口</td><td>R302=1000</td><td>R305=20000</td><td rowspan="8">对应脉冲输出频率 200Khz</td><td>R306=0</td><td rowspan="8">对应脉冲输出频率 0hz</td><td>R307=1000</td><td rowspan="8">对应输出脉冲加速度为 10脉冲/ms²</td></tr>
<tr><td>AXV1</td><td>AXOUT1 (R350=1)</td><td>R351=0</td><td>R352=1000</td><td>R355=20000</td><td>R356=0</td><td>R357=1000</td></tr>
<tr><td>AXV2</td><td>AXOUT2 (R400=2)</td><td>R401=0</td><td>R402=1000</td><td>R405=20000</td><td>R406=0</td><td>R407=1000</td></tr>
<tr><td>AXV3</td><td>AXOUT3 (R450=3)</td><td>R451=0</td><td>R452=1000</td><td>R455=20000</td><td>R456=0</td><td>R457=1000</td></tr>
<tr><td>AXV4</td><td>AXOUT4 (R500=4)</td><td>R501=0</td><td>R502=1000</td><td>R505=20000</td><td>R506=0</td><td>R507=1000</td></tr>
<tr><td>AXV5</td><td>AXOUT5 (R550=5)</td><td>R551=0</td><td>R552=1000</td><td>R555=20000</td><td>R556=0</td><td>R557=1000</td></tr>
<tr><td>AXV6</td><td>AXOUT6 (R600=6)</td><td>R601=0</td><td>R602=1000</td><td>R605=20000</td><td>R606=0</td><td>R607=1000</td></tr>
<tr><td>AXV7</td><td>AXOUT7 (R650=7)</td><td>R651=0</td><td>R652=1000</td><td>R655=20000</td><td>R656=0</td><td>R657=1000</td></tr>
<tr><td>AXV8</td><td>AXOUT8 (R700=8)</td><td>R701=0</td><td></td><td>R702=1000</td><td>R705=20000</td><td></td><td>R706=0</td><td></td><td>R707=1000</td><td></td></tr>
</table>

AXV9	AXOUT9 (R750=9)	R751=0		R752=1000	R755=20000		R756=0		R757=1000	
AXV10	AXOUT2 (R800=10)	R801=0		R802=1000	R805=20000		R806=0		R807=1000	
AXV11	AXOUT3 (R850=11)	R851=0		R852=1000	R855=20000		R856=0		R857=1000	

以上参数如果满足用户对 12个软件虚拟输出轴的配置要求，则用户程序无需再进行初始化。其它情况下，用户要根据实际要用到的硬件输出轴个数、轴运行最高速度、轴运行起始速度、轴运行最大加/减速度等参数进行配置。由于默认配置参数的存在，对于不需要输出脉冲的硬件输出轴，用户可以通过绑定对应的软件虚拟输出轴到其他数字输出轴或其他硬件输出实轴，从而释放该硬件输出轴，使其可以作为普通输出口使用。

举例说明：以下梯形图程序中，只绑定 AXOUT0~AXOUT2 三个硬件输出轴，AXOUT3作为普通输出口使用，AXOUT0~AXOUT2 的输出频率最高可达 200Khz。



2.3.3 程序运行时如何修改轴配置参数

程序运行时，停止相对应的高速计数器，软件虚拟输入轴立即释放，对应的轴参数寄存器不再更新。程序运行时，软件虚拟输出轴与硬件输出轴的绑定关系不能修改，轴运行最高速度、轴运行起始速度、轴运行最大加/减速度等参数可以修改，但必须在对应的轴运动指令处于空闲状态下才能修改。

2.3.4 使用回原点指令时如何对软件虚拟输出轴进行初始化

先按照上述 2.2.1~2.2.3 章节的方法对软件虚拟输出轴进行初始化，再使用轴配置指令 AXEI、AXOSP、AXOX 指令在梯形图第一个运行周期之内对相关参数进行配置。这三条指令的具体用法请参见第 3.2.4~3.2.7章节。

2.4.运动指令执行过程

一条运动指令的执行过程：启动条件，缓冲方式，1ms 的运算。

所有运动指令均使用常开或常闭触点作为指令的启动条件，但 IPL 指令一般建议用边沿触发类触点作为指令的启动条件，需要持续进行脉冲插补时，IPL 指令可使用常开或常闭触点。

运动指令执行过程主要是对配置的软件虚拟输出轴进行数字插补运算，插补周期 1ms，不同的指令对应不同的插补算法。当两个软件虚拟输出轴绑定同一个硬件输出轴时，实质上是将这两个软件虚拟输出轴在 1ms 内的脉冲插补增量进行数学叠加，再输出到对应的硬件输出轴上。

2.4.1. 运动指令的执行时延

从运动指令输入触发触点闭合的时刻开始计时，到对应的脉冲输出口发出第一个脉冲的边沿停止计时， 这段时间称为运动指令的执行时延，时间不会超过 1ms。软件虚拟输出轴对应的寄存器的数据每 1ms 由系统自动刷新一次。断开运动指令的输入触点时，相关轴运行状态并不是立即清零，最多会延时 1ms。

2.4.2. 运动指令的启动条件

执行运动指令前，必须查询软件虚拟输出轴的运行状态，其数值为 0 表示该轴处于空闲状态，此时方可执行轴运行指令，其他数值情况下执行该轴的运动指令将被阻塞，不会生效，但 IPL 指令不受限制。

2.4.3. 运动指令的停止条件

运动指令执行过程中，当检测到轴状态等于 32 或160 时，说明该轴的某条运动指令已经执行完毕， 此时可断开运动指令。

2.4.4. 运动指令的运行状态

轴运行状态	状态说明
=0	当前无该轴的运动指令运行，可执行该轴的运动指令
=16	该轴的运动指令正在执行
=32	该轴的运动指令正常执行完成（无急停等特殊情况）
=64	该轴设置/运行出错，出错代码可查看轴出错代码
=144	该轴正在执行缓停
=160	该轴处于急停状态

可通过查询轴运行状态字，了解轴目前所处的运行状态。

2.5.运动指令报警信息

当轴参数配置不合理或运动指令运行异常时，系统将停止对应的软件虚拟输出轴的执行过程，对应的硬件输出口停止脉冲输出。当进行多个软件虚拟输出轴的运动叠加时（即将多个软件虚拟输出轴与一个硬件输出轴进行绑定），很容易出现轴超速报警，此时应降低每个软件虚拟输出轴的最高限速和最大加速度， 确保多个轴叠加后轴插补增量对应的速度低于硬件输出轴的最大输出速度。

轴参数设置/运行出错代码表：

轴出错代码 =	定义
0	无错
1	当前轴速度超硬件支持的极限速度
2	未绑定输出轴
3	轴最高限速超硬件支持的极限速度
4	当前轴速度低于轴起始速度
5	当前轴速度超过原点回归速度，回原点时检测
6	当前轴速度低于原点爬行速度，回原点时检测
7	轴起始速度大于轴最高限速

当运动指令执行出现异常时，请根据以上出错代码调整软件虚拟输出轴的相关参数。

3 运动指令

3.1 运动指令索引

MC系列PLC运动指令包括虚拟输入轴的轴参数配置指令 7 条，高数计数指令 6 条，单轴基本运动指令 10 条。

下表为 MC系列PLC的运动指令索引。

3.1.1 运动指令一览表

助记符、名称	功能
轴参数配置	**3.2 章节**
AXPO	轴端口指定
AXPU	轴当量指定
AXSP	轴速指定
AXAS	轴加减速指定
AXEI	轴事件输入点指定
AXOSP	回原点速度指定
AXOX	轴事件指定
单轴基本运动	**3.4 章节**
HOME	回原点
HOMA	简易回原点
IPL	插补
DPLSY	定速定长输出
DPLSR	加减速定长输出
DDRVI	相对定位
DDRVA	绝对定位
DDVIT	中断定位
DPLSV	变速输出
DPTO	多段速输出
PSOUT	定位输出
PTAB	普通制表
PTOUT	制表输出

助记符、名称	功能
高数计数	**3.3 章节**
DHMOV	高速读出
DHSCS	高速比较置位
DHSCR	高速比较复位
DHSZ	高速区间比较
DHSCT	高速表格比较输出
REF	输入输出刷新
单轴关联运动	**3.5 章节**
GEAR	电子齿轮
PCTAB	追剪正程/滚切制表
PBTAB	追剪返程制表
CAM	电子凸轮输出
两轴插补指令	**3.6 章节**
GCO	设置坐标系
SELCO	选择坐标系
G90G01	两轴直线绝对位置插补
G91G01	两轴直线相对位置插补
G90G02	两轴顺圆弧绝对位置插补
G91G02	两轴顺圆弧相对位置插补
G90G03	两轴逆圆弧绝对位置插补
G91G03	两轴逆圆弧相对位置插补

3.1.2 输入端口掩码

控制器在指定轴事件端口时，通过掩码的方式指定，在不同指令应用时，需按下面表格，指定相应的控制端口。

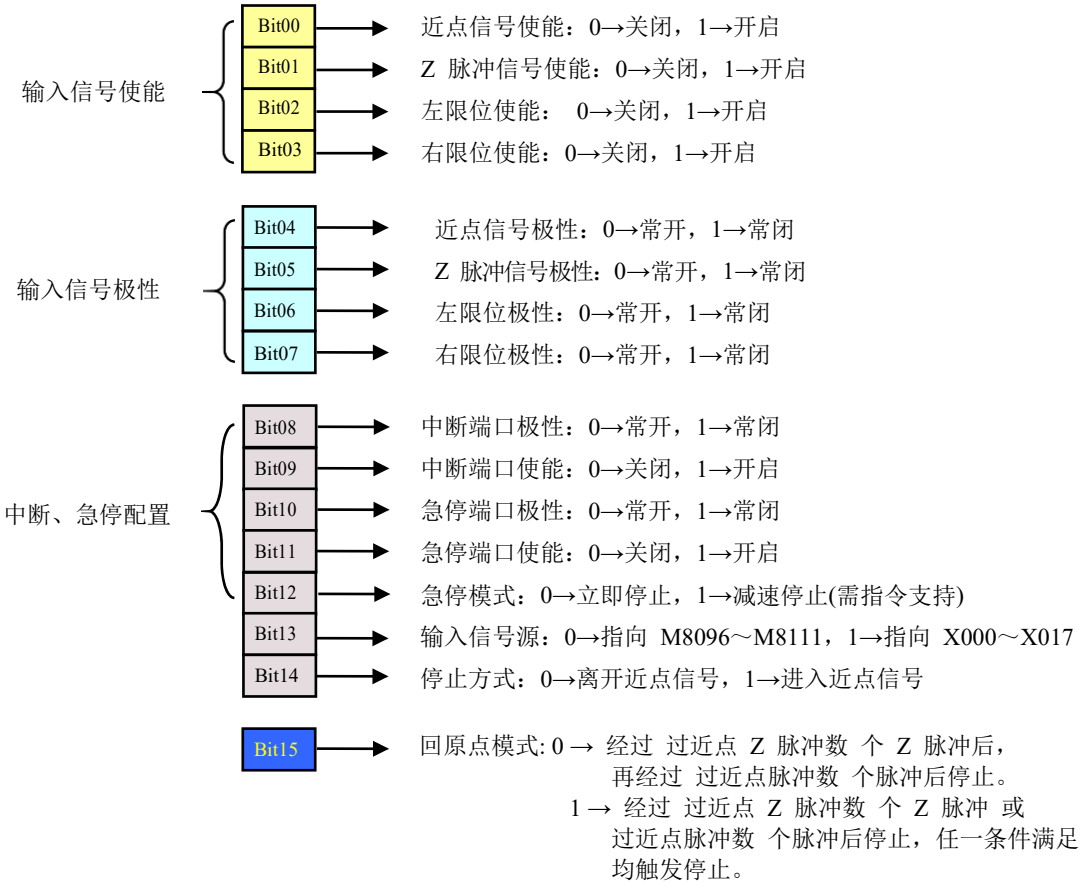
输入端口掩码	选定端口	输入端口掩码	选定端口
0	不选定	256	X10/M8104
1	X0/M8096	512	X11/M8105
2	X1/M8097	1024	X12/M8106
4	X2/M8098	2048	X13/M8107
8	X3/M8099	4096	X14/M8108
16	X4/M8100	8192	X15/M8109
32	X5/M8101	16384	X16/M8110
64	X6/M8102	-32768	X17/M8111
128	X7/M8103		

3.1.3 轴输入事件定义

1) 在运动指令中，若需用到外部输入事件，需按下表进行设定。

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3 轴			
R308	R358	R408	R458	近点信号输入点掩码	16 位	0
R309	R359	R409	R459	编码器 Z 脉冲输入点掩码	16 位	0
R310	R360	R410	R460	左限位信号输入点掩码	16 位	0
R311	R361	R411	R461	右限位信号输入点掩码	16 位	0
R314	R364	R414	R464	轴事件输入点	16 位	0
R323	R373	R423	R473	轴急停事件	16 位	0
R324	R374	R424	R474	轴中断事件	16 位	0

2) 轴参数R+14（轴参数输入）参数说明



3) 配置轴事件主要事项

➢ 回原点指令（HOME）的外部事件可以通过指令或给相应的寄存器赋值实现；但轴急停事件及中断事件只能通过MOV赋值相应的寄存器实现，指定端口通过给相应的寄存器赋值相应的掩码实现；

➢ 所有的轴事件都可以关闭或开启，若需要用到相应的轴事件，需在轴参数的R+14号寄存器中将相应的控制位置1；反之，若不需相应的事件产生影响，可以清0关闭；

➢ 所有的轴事件都可以选择极性，配置轴事件端口时，需在轴参数的R+14号寄存器中配置相应控制位，选择端口的极性；

➢ 所有的轴事件均可用物理端口或相应的M寄存器来控制，应用时可以通过轴参数R+14号参数的Bit13位选择。

3.2 轴配置指令

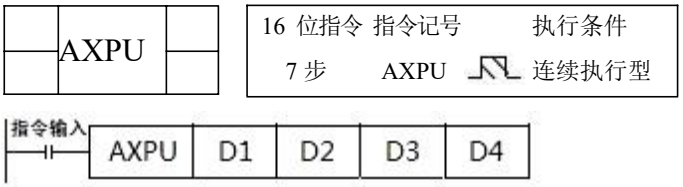
本小节主要介绍虚拟轴配置指令。

3.2.1 轴当量指定[AXPU]

对指定软件虚拟输出轴(AXV)的时间当量、脉冲当量和长度当量设置。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	时间当量	BIN 16 位
D3	脉冲当量	BIN 16 位
D4	长度当量	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							

- **相关软元件**

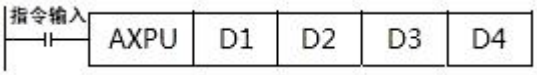
可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2轴	虚拟3轴			
R302	R352	R402	R452	时间当量	16 位	1000
R303	R353	R403	R453	脉冲当量	16 位	0
R304	R354	R404	R454	长度当量	16 位	0

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的时间当量、脉冲当量和长度当量。



1) 在D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

2) 在D2 中指定时间当量

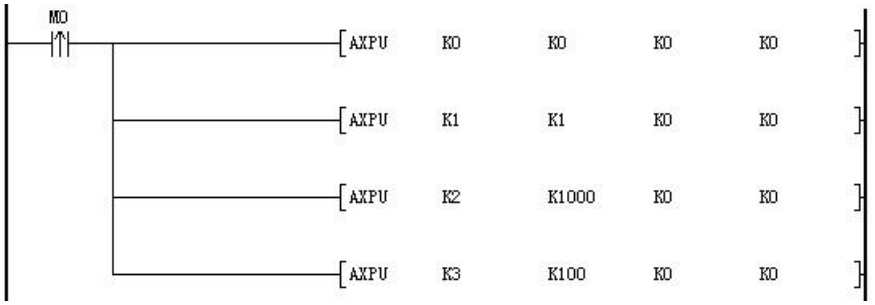
	设定范围
16 位运算	1~32000

3) 在D3 中指定脉冲当量

	设定范围
16 位运算	缺省，不需设置

4) 在D4 中指定长度当量

	设定范围
16 位运算	缺省，不需设置



M0 上升沿指令执行后：

虚拟轴	轴参数			效果
0	R302=0	R303=0	R304=0	速度类参数无效，错误配置
1	R352=1	R353=0	R354=0	速度类参数过小，不建议配置
2	R402=1000	R403=0	R404=0	系统默认配置
3	R452=100	R453=0	R454=0	速度类参数比默认放大 10 倍

- **注意要点**

1) 此指令建议不要使用保持触点执行，即 M8002 中执行一次即可；
2) 脉冲当量和长度当量目前是缺省值，无需修改；
3) 配置时间当量后，对应轴参数中的最大速度、起始速度、加速度和脉冲指令中的频率均有影响；
4) 程序中可不写此指令，默认配置就可以正常使用。

3.2.2 轴速度指定[AXSP]

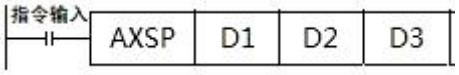
对指定软件虚拟输出轴(AXV)的最大速度和起始速度设置。

- **指令格式**

1) 指令格式



16 位指令	指令记号	执行条件
7 步	AXSP	连续执行型



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	最大速度	BIN 16 位
D3	起始速度	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							

- **相关软元件**

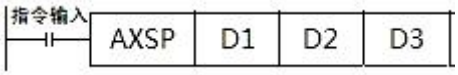
可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3 轴			
R305	R355	R405	R455	最大速度	16 位	20000
R306	R356	R406	R456	起始速度	16 位	0

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的最大速度和起始速度。



1. 在 D1 中指定软件虚拟输出轴

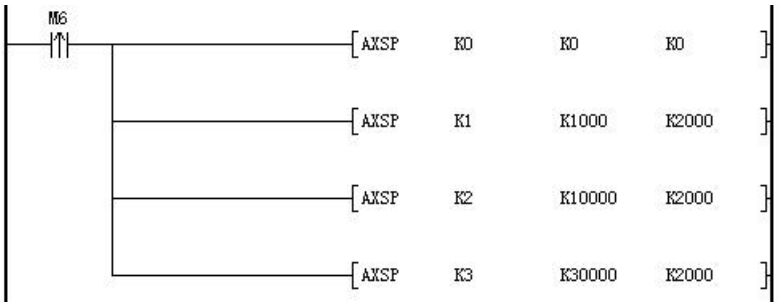
	设定范围
16 位运算	0~3

2. 在 D2 中 指 定 最 大 速 度

	设定范围
16 位运算	0~20000（默认1000ms时间当量）

3. 在 D3 中指定起始速度

	设定范围
16 位运算	0~20000（默认1000ms时间当量）



M6 上升沿指令执行后：

虚拟轴	轴参数		效果（默认时间当量 1000，每路 200KHZ）
0	R305=0	R306=0	轴无法输出，配置错误
1	R355=1000	R356=2000	最大速度小于起始速度，配置错误
2	R405=10000	R406=2000	最大速度 100K，起始速度 20K，配置正确
3	R455=30000	R456=2000	大于硬件速度，配置错误

- **注意要点**

1) 此指令建议不要使用保持触点执行，即需修改轴速度时执行一次即可；
2) 根据时间当量算出实际设定的轴最大速度和起始速度；
3) 如时间当量默认，轴端口任意配置值最大速度表格举例如下：

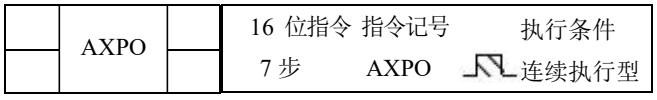
虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3 轴	最大速度（HZ）	开通脉冲口
R300	R350	R400	R450		
0	1	2	3	200K	Y0/Y1/Y2/Y3
0	0	1	2	200K	Y0/Y1/Y2
0	0	0	1	200K	Y0/Y1
0	0	0	0	200K	Y0

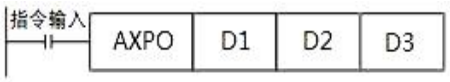
3.2.3 轴端口指定[AXPO]

将软件虚拟输出轴(AXV)与实际硬件输出轴(AXOUT)绑定。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	实际输出轴号	BIN 16 位
D3	轴方向使能	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●	●	●							
D2	●	●	●							
D3	●	●	●							

- **相关软元件**

可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：

1) 该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2轴	虚拟3轴			
R300	R350	R400	R450	实际输出轴号	16 位	0/1/2/3
R301	R351	R401	R451	脉冲输出方向使能	16 位	0

2) 软件虚拟输出轴绑定列表：

软件虚拟输出轴	轴端口指定寄存器	实际输出轴号			
		脉冲输出轴	脉冲输出轴	脉冲输出轴	脉冲输出轴

轴0	R300	0	1	2	3
轴1	R350	0	1	2	3
轴0	R300	0	1	2	3
轴1	R350	0	1	2	3

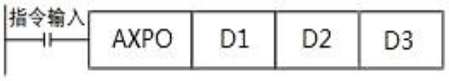
3) 当软件虚拟输出轴绑定硬件输出轴后，脉冲输出方向列表

硬件输出轴号	脉冲输出端口	方向输出端口 （默认绑定无法更改）	轴方向使能 R301/R351/R401/R451	
0	Y0	Y4	0：不使能	1：使能
1	Y1	Y5	0：不使能	1：使能
2	Y2	Y6	0：不使能	1：使能
3	Y3	Y7	0：不使能	1：使能

*数字输出轴 4-11，无实际方向输出。

- **功能和举例**

该指令主要用于配置脉冲输出端口和方向功能。



1） 在 D1 中指定软件虚拟输出轴

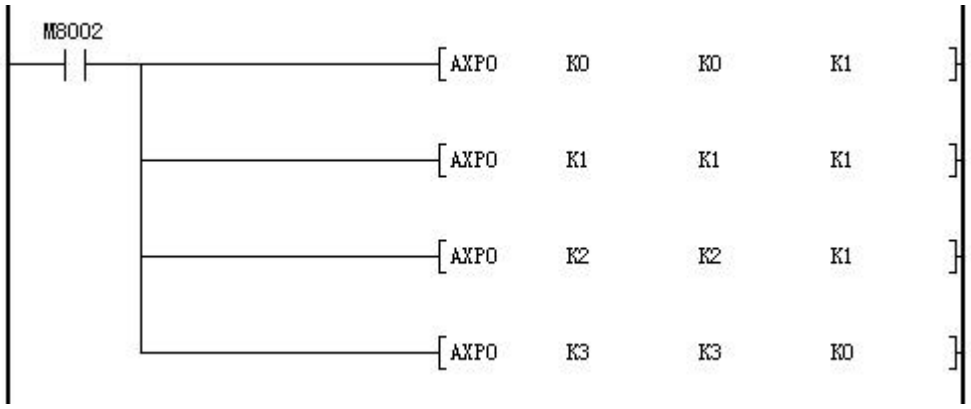
	设定范围
16 位运算	0~11

2） 在 D2 中指定实际输出轴

	设定范围
16 位运算	0~3

3） 在 D3 中指定实际输出轴的方向使能

	设定范围
16 位运算	0~1



上电 M8002 指令执行后：

虚拟轴	轴参数		效果
0	R300=0	R301=1	脉冲输出轴Y0，方向Y4 使能

1	R350=1	R351=1	脉冲输出轴Y1，方向Y5使能
2	R400=2	R401=1	脉冲输出轴Y2，方向Y6使能
3	R450=3	R451=0	脉冲输出轴Y3，无方向

每路脉冲频率最高 100KHZ。

- **注意要点**

1) 此指令必须在上电第一个扫描周期配置脉冲端口和方向端口；

2) 脉冲和方向端口配置完成后，禁止修改对应寄存器（如 R300/R301 等）；

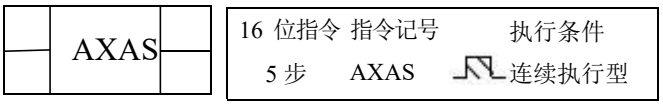
3) 程序中未对轴端口进行指定，则默认配置虚拟轴 0~3分别对应 Y0~Y3 脉冲输出端口，但方向位不使能，每路最高脉冲频率200KHZ；

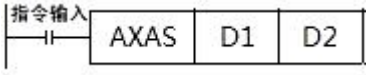
3.2.4 轴加减速指定[AXAS]

对指定软件虚拟输出轴(AXV)的最大加减速度设置。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	最大加减速度	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							

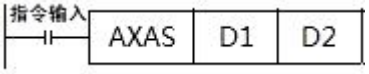
- **相关软元件**

可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：
该指令作用于下表寄存器

软元件编号				名称	数据类型	初始值
虚拟0轴	虚拟1轴	虚拟2轴	虚拟3轴			
R307	R357	R407	R457	最大加减速度	16位	1000

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的最大加减速度。

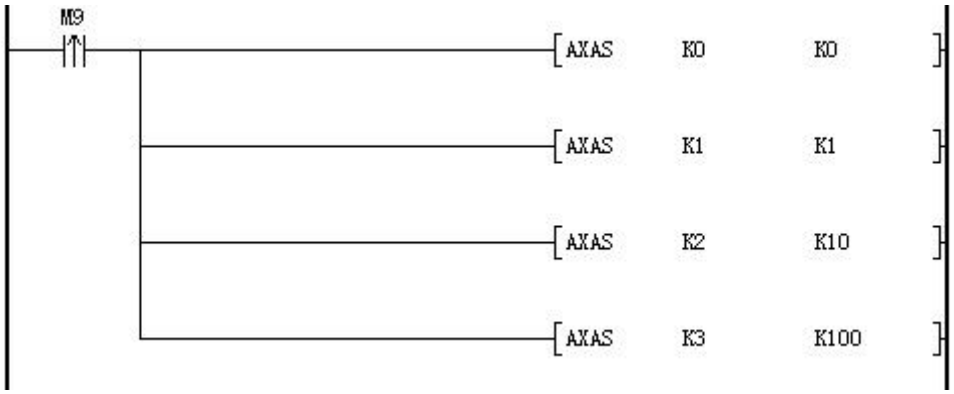


1) 在D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

2) 在 D2 中指定最大加减速度

	设定范围
16 位运算	0~32767



M9 上升沿指令执行后：

虚拟轴	轴参数	效果（默认时间当量 1000）
0	R307=0	有加减速过程的轴无法输出，配置错误
1	R357=1	加减速度为 0.01 脉冲/ms^2，配置正确
2	R407=10	加减速度为 0.1 脉冲/ms^2，配置正确
3	R457=100	加减速度为 1 脉冲/ms^2，配置正确

- **注意要点**

1) 此指令建议不要使用保持触点执行，即需修改轴加减速度时执行一次即可；
2) 根据时间当量算出实际设定的轴加减速度；
3) 这一参数会影响DPLSR 中的加减速时间，具体关系参看DPLSR 指令详解；
4) 设置时注意对应的硬件端口频率上限，防止过大导致超速。

3.2.5 轴事件输入点指定[AXEI]

对指定软件虚拟输出轴(AXV)的事件输入点设置。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	近点信号输入点掩码	BIN 16 位
D3	编码器 Z 脉冲输入点掩码	BIN 16 位
D4	左限位信号输入点掩码	BIN 16 位
D5	右限位信号输入点掩码	BIN 16 位

3) .对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							
D5	•	•	•							

- **相关软元件**

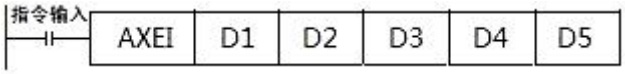
可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R308	R358	R408	R458	近点信号输入点掩码	16 位	0
R309	R359	R409	R459	编码器 Z 脉冲输入点掩码	16 位	0
R310	R360	R410	R460	左限位信号输入点掩码	16 位	0
R311	R361	R411	R461	右限位信号输入点掩码	16 位	0
R314	R364	R414	R464	轴事件输入点	16 位	0

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的事件输入点。



1) 在D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

2) 在D2 中指定近点信号输入点掩码

	设定范围
16 位运算	-32768~16384

3) 在D3 中指定编码器 Z 脉冲输入点掩码

	设定范围
16 位运算	0~128

4) 在D4 中指定左限位信号输入点掩码

	设定范围
16 位运算	-32768~16384

5) 在D5 中指定右限位信号输入点掩码

	设定范围
16 位运算	-32768~16384

6) 输入点掩码对应表格：

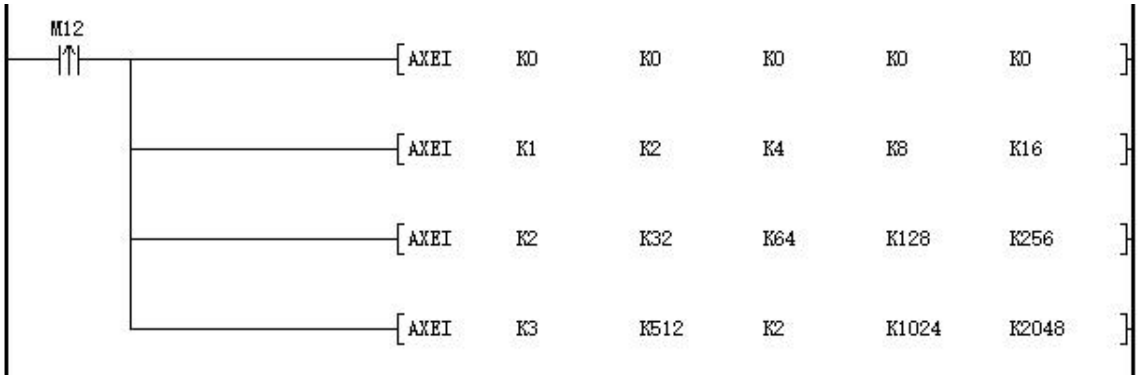
D2/D3/D4/D5	选定端口	D2/D3/D4/D5	选定端口
0	不选定	256	X10/M8104
1	X0/M8096	512	X11/M8105
2	X1/M8097	1024	X12/M8106
4	X2/M8098	2048	X13/M8107
8	X3/M8099	4096	X14/M8108
16	X4/M8100	8192	X15/M8109
32	X5/M8101	16384	X16/M8110
64	X6/M8102	-32768	X17/M8111
128	X7/M8103		

端口选定 X 或 M，需对应设定轴参数中的“轴事件输入点”中的第 13 位进行设置。

例如：虚拟 0 轴中的 R314 中的第 13 位为 0 指定 M，为 1 指定 X。

R314	0	0	1	0		0	0	0	0		0	0	0	0		0	0	0	0

上图中 R314=8192，端口定义 X0~X17 有效。



M12 上升沿指令执行后：

虚拟轴	轴参数				近点信号	Z 脉冲信号	左限位	右限位
0	R308=0	R309=0	R310=0	R311=0	无	无	无	无
1	R358=2	R359=4	R360=8	R361=16	X1	X2	X3	X4
2	R408=32	R409=64	R410=128	R411=256	X5	X6	X7	X10
3	R458=512	R459=2	R460=1024	R461=2048	X11	X1	X12	X13

这一配置中 X 上升沿有效，则需：R314/R364/R414/R464=8207

- **注意要点**

1)此指令建议不要使用保持触点执行，即M8002 中执行一次即可；

2)定义事件输入点后一定要设定对应虚拟输出轴的轴参数“轴事件输入点”寄存器中的位数据才能有效。

3.2.6 回原点速度指定[AXOSP]

对指定软件虚拟输出轴(AXV)的原点回归速度和原点爬行速度设置。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	原点回归速度	BIN 16 位
D3	原点爬行速度	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							

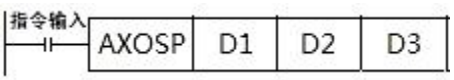
- **相关软元件**

可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：
该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R312	R362	R412	R462	原点回归速度	16 位	0
R313	R363	R413	R463	原点爬行速度	16 位	0

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的原点回归速度和原点爬行速度。



1) 在D1 中指定软件虚拟输出轴

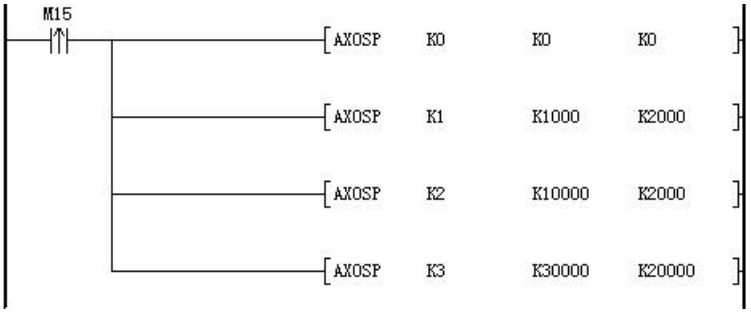
	设定范围
16 位运算	0~3

2) 在D2 中指定原点回归速度

	设定范围
16 位运算	0~10000（默认1000ms时间当量）

3) 在D3 中指定原点爬行速度

	设定范围
16 位运算	0~10000（默认1000ms时间当量）



M15 上升沿指令执行后：

虚拟轴	轴参数		效果（默认时间当量 1000，每路 100KHZ）
0	R312=0	R313=0	轴无法输出，配置错误
1	R362=1000	R363=2000	回原点速度 10K，爬行速度 20K，配置错误
2	R412=10000	R413=2000	回原点速度 100K，爬行速度 20K，配置正确
3	R462=30000	R463=20000	大于硬件速度，配置错误

- **注意要点**

1) 此指令建议不要使用保持触点执行，即需修改轴速度时执行一次即可；

2) 根据时间当量算出实际设定的轴回原点速度和爬行速度；

3) 回原点速度应该设置合理以保证回原点功能的正常运行，详细使用方法参见 HOME 指令。

3.2.7 回原点参数指定[AXOX]

对指定软件虚拟输出轴(AXV)的回原点方式设置。

- **指令格式**

1) **指令格式**



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	回原点模式	BIN 16 位
D3	PG(Z 脉冲)信号脉冲数	BIN 16 位
D4	DOG(近点)信号脉冲数	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							

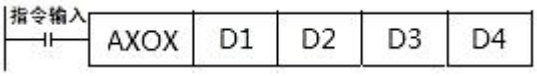
- **相关软元件**

可使用指令对相关软元件进行配置，也可直接对相关软元件赋值完成配置。相关软元件及配置如下所示：
该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R314	R364	R414	R464	回原点模式	16 位	10
R315	R365	R415	R465	Z 脉冲信号脉冲数	16 位	0
R316	R366	R416	R466	近点信号脉冲数	16 位	0

- **功能和举例**

该指令主要用于配置软件虚拟输出轴的回原点方式。



1) 在 D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

2) 在 D2 中指定回原点模式（回原点方式、输入信号的使能和极性）

	设定范围
16 位运算	-32767~32767

输入信号使能
- Bit00 → 近点信号使能：0→关闭，1→开启
- Bit01 → Z 脉冲信号使能：0→关闭，1→开启
- Bit02 → 左限位使能： 0→关闭，1→开启
- Bit03 → 右限位使能：0→关闭，1→开启

输入信号极性
- Bit04 → 近点信号极性：0→常开，1→常闭
- Bit05 → Z 脉冲信号极性：0→常开，1→常闭
- Bit06 → 左限位极性：0→常开，1→常闭
- Bit07 → 右限位极性：0→常开，1→常闭

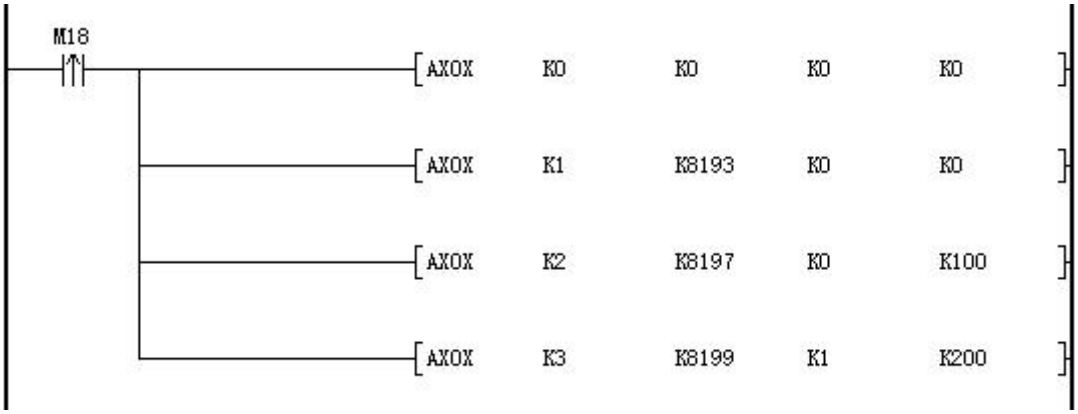
中断、急停配置
- Bit08 → 中断端口极性：0→常开，1→常闭
- Bit09 → 中断端口使能：0→关闭，1→开启
- Bit10 → 急停端口极性：0→常开，1→常闭
- Bit11 → 急停端口使能：0→关闭，1→开启
- Bit12 → 急停模式：0→立即停止，1→减速停止(需指令支持)
- Bit13 → 输入信号源：0→指向 M8096～M8111，1→指向 X000～X017
- Bit14 → 停止方式：0→离开近点信号，1→进入近点信号
- Bit15 → 回原点模式: 0 → 经过 过近点 Z 脉冲数 个 Z 脉冲后，再经过 过近点脉冲数 个脉冲后停止。
  1 → 经过 过近点 Z 脉冲数 个 Z 脉冲 或 过近点脉冲数 个脉冲后停止，任一条件满足均触发停止。

3) 在 D3 中指定 Z 脉冲信号脉冲数

	设定范围
16 位运算	0~32767

4) 在 D4 中指定近点信号脉冲数

	设定范围
16 位运算	0~32767



M18 上升沿指令执行后：

虚拟轴	轴参数			效果
0	R314=0	R315=0	R316=0	参数均为 0，回原点设置错误
1	R364=8193	R365=0	R366=0	有近点信号，离开近点后立即停止
2	R414=8197	R415=0	R416=100	有近点信号，左限位，离开近点后运行 100 个脉冲停止
3	R464=8199	R465=1	R466=200	有近点信号，左限位，Z 脉冲信号，离开近点后遇到1 个Z 脉冲信号再运行200 个脉冲停止

- **注意要点**

1) 此指令建议不要使用保持触点执行，即配置回原点功能时执行一次即可；
2) D2 操作数的设置是按位来设定近点信号，左限位，右限位，Z 脉冲信号的使能和极性，以及回原点最后停止的方式；
3) 端口极性设置为 0：进入信号 X 端口为 ON，离开为OFF； 设置为 1：进入信号 X 端口为 OFF，离开为 ON。

3.3 高速计数指令

本小节主要介绍LP3系列PLC高速计数的相关指令和用法。

3.3.1 高速读出[DHMOV]

对高速计数器或高速输出脉冲的寄存器值即时读取。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	读出数据地址	BIN 32 位
D2	高速计数器当前值存入的目标寄存器地址	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		•	•	•						
D2		•	•	•						

- **功能和举例**

1) D1 指定需要高速读出的软元件，一般为高速计数器 C235～C255 或脉冲输出的D 寄存器。

2) D2 指定需要存入的目标寄存器地址[D+1,D]中。

比如 DHMOV C235 D1000，执行结果将当前 C235 的数值存入D1001，D1000(32 位)内。

3) 常规情况下高速输入数值是0.1ms实时更新，梯形图程序处理时，读取的高速输入数值高低字节可能不是同一时刻的完整字节。本指令可避免此问题，实时读取双字数值，但计数值不一定是实时值，可能是上一个0.1ms的数据。

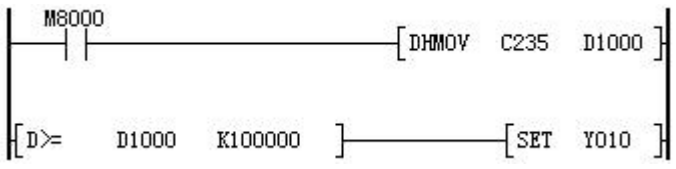


图.高速读出示例

- DHMOV 指令执行，读取 C235 高速计数器双字，
- 然后将计数器值送入到 D1000(32 位)。
- 比较 D1000 和一个预设值的大小(这里为100000)，如果≥该值则置位 Y10。
- 该控制流程与 DHSCS 高速比较置位指令类似，但DHSCS 不受扫描周期的影响，处理结果更为实时。

DHMOV与 DMOV 的区别：

1) DHMOV 指令执行，将D1 指定的计数器双字读取，然后将计数器当前值传入到 D2。

2) DMOV 指令执行，将 D1 指定的计数器当前值传入到D2，但读取到的当前值高低字节可能不是同一时刻的，从而导致结果错误。

3.3.2 高速比较置位[DHSCS]

对指定的高速计数器当前值与设定的比较值比较，由<状态变为≥状态，或由≥状态变为 < 状态，则置位指定的位元件。

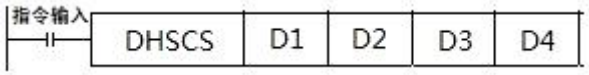
- **指令格式**

1）指令格式



32 位指令 指令记号 执行条件

13步DHSCS 连续执行型



2）数据格式

操作数种类	内容	数据类型
D1	待比较的脉冲计数值	BIN 32 位
D2	比较计数值对应的寄存器	BIN 32 位
D3	置位的位元件	位
D4	高速数据运算区号	BIN 32 位

3）对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1	•	•	•							
D2		•	•	•						
D3							•	•	•	
D4	•	•	•							

- **功能和举例**

1) D1 指定高速计数器待比较的数值。

2) D2 通常指定高速计数器 C235～C255 或者高速脉冲输出寄存器。

3) D3 指定比较置位的位元件。

◆ 当高速计数器当前值由 < 变为≥D1 值，或由≥变为 <D1 值时，比较置位输出的位元件立即置位，不受扫描周期的影响。即使后续条件不满足，该指令也不会执行软元件的复位操作，需要手动复位。

4) D4 指定指令所占用的高速数据运算区号 0～5。

5) 若 D2 为高速计数器，则在执行高速比较置位指令之前，需要先 OUT D2 指定高速计数器，如下图。



启动高速计数器 C235

- 注意要点

1) 高速计数器分辨率

◆ 高速计数器 C235～C255 的计数分辨率为 0.1ms。

◆ 数据源为 200KHz 频率脉冲，计数值比较误差 10 个脉冲以内。

◆ 数据源小于 10KHz 频率脉冲，计数值比较误差 1 个脉冲以内。

- 高速计数器C235~C255值可实时更改。

2) 高速数据运算区

- 高速计数指令 DHSCS，DHSCR，DHSZ 共享6 个内部的高速数据运算区。
- 高速数据运算区号范围为 0～5。
- 每个高速数据运算区同一时间仅能执行 1 条上述高速计数指令，使用高速计数指令时需要注意灵活分配运算区。

指令示例一

高速比较置位指令置位输出（Y）应用，高速计数器 C255 进行脉冲 AB 相计数，当计数值到 100 时输出Y10， 当计数到 150 时输出Y11。



- 要使用 C255 进行高速计数，必须先 OUT 输出，设置数值任意。
- 配C255 计数到100 时输出Y10，占用高速数据运算区 0。
- 配置C255 计数到150 时输出Y11，占用高速数据运算区 1。

图.高速比较置位示例 1

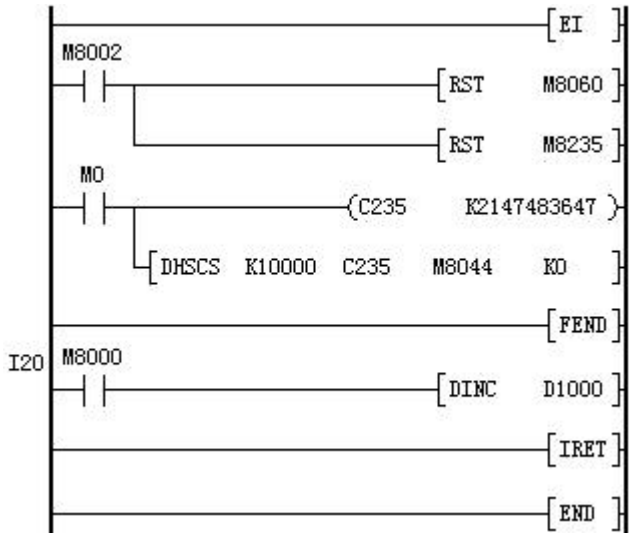


图.高速比较置位示例 1 执行示意图

1) 当 C255 计数超过 100 时，Y10 由 OFF 变为 ON 状态，若本身为ON 状态则保持不变。
2) 当 C255 计数超过 150 时，Y11 由 OFF 变为 ON 状态，若本身为ON 状态则保持不变。

指令示例二

高速比较置位指令启动中断，高速计数器 C235 进行增计数，当计数值到 10000 时，进入软中断 I20 处理用户事件。



- EI 打开总中断。
- 复位 M8060，允许 I20；复位 M8235，C235 增计数。（设置参考高速计数器章节）
- OUTC235 打开高速计数。高速比较置位计数值超过10000 时置位I20 的中断申请位 M8044。
- 主程序结束
- I20 中断服务程序， 当 C235 计数值超过 10000 时，进入该中断。
- 中断程序返回
- 程序结束

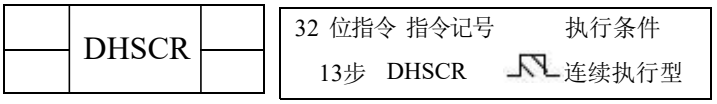
图.高速比较置位示例 2

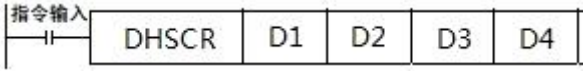
3.3.3 高速比较复位[DHSCR]

对指定的高速计数器当前值与设定的比较值比较， 由<状态变为≥状态，或由≥状态变为 < 状态，则复位指定的位元件。

- **指令格式**

1) 指令格式





2) 数据格式

操作数种类	内容	数据类型
D1	待比较的脉冲计数值	BIN 32 位
D2	比较计数值对应的寄存器	BIN 32 位
D3	复位的位元件	位
D4	高速数据运算区号	BIN 32 位

3) 数据软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1	•	•	•							
D2		•	•	•						
D3							•	•	•	
D4	•	•	•							

- **功能和举例**

1) D1 指定高速计数器待比较的数值。

2) D2 通常指定高速计数器 C235～C255 或者高速脉冲输出寄存器。

3) D3 指定比较复位的位元件。

4) 当高速计数器当前值由<变为≥D1 值，或由≥变为 ＜D1 值时，比较复位输出的位元件立即复位，不受扫描周期的影响。即使后续条件不满足，该指令也不会执行软元件的置位操作。

5) D4 指定指令所占用的高速数据运算区号 0～5。

6) 在执行高速比较复位指令之前，需要先 OUT D2 指定高速计数器，如下图所示。



启动高速计数器 C235

- **注意要点**

1) 高速计数器分辨率

◆ 高速计数器 C235～C255 的计数分辨率为 0.1ms。

◆ 数据源为 200KHz 频率脉冲，计数值比较误差 10 个脉冲以内。

◆ 数据源小于 10KHz 频率脉冲，计数值比较误差 1 个脉冲以内。

◆ 高速计数器C235~C255值可实时更改。

2) 高速数据运算区

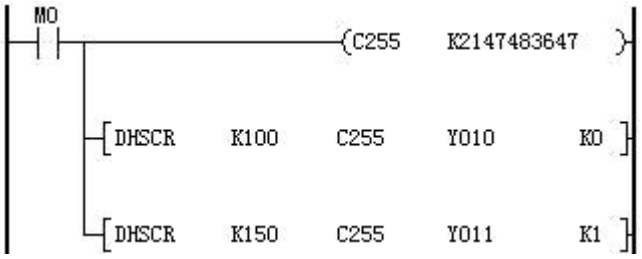
◆ 高速计数指令 DHSCS，DHSCR，DHSZ 共享 6 个内部的高速数据运算区。

◆ 高速数据运算区号范围为 0～5。

◆ 每个高速数据运算区同一时间仅能执行 1 条上述高速计数指令，使用高速计数指令时需要注意灵活分配运算区。

- **指令示例**

高速计数器 C255 进行脉冲 AB 相计数，当计数值到 100 时复位 Y10，当计数到 150 时复位 Y11。



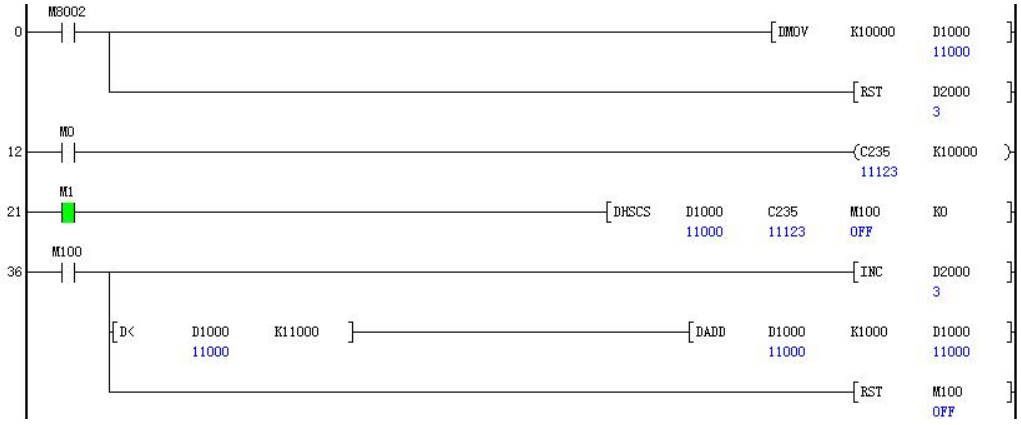
- 要使用 C255 进行高速计数，必须先 OUT 输出，设置数值任意。
- C255 计数到 100 时复位 Y10，占用高速数据运算区 0。
- C255 计数到 150 时复位 Y11，占用高速数据运算区 1。

高速比较复位示例

1) 当 C255 计数超过 100 时，Y10 由 ON 变为OFF 状态，若本身为OFF 状态则保持不变。

2) 当 C255 计数超过 150 时，Y11 由 ON 变为OFF 状态，若本身为OFF 状态则保持不变。

DHSCS/R D1 D2 D3 D4 ， D1 指定高速计数器待比较的数值，可以在线修改，但需要注意由<状态变为≥状态，或由≥状态变为 < 状态两个方向均会置位/复位D3，在线修改D1可能产生额外的动作。如：



C235增计数不断增加，

D1000	C235	比较关系	M100
10000	0	C235<D1000	OFF
10000	10000	C235=D1000	ON->OFF D2000+1
10000	10100	C235>D1000	OFF
11000	10150	C235<D1000	ON->OFF D2000+1
11000	10200	C235<D1000	OFF
11000	11000	C235=D1000	ON->OFF D2000+1

①C235从0开始增计数，M100初始状态OFF，D1000初始值10000，C235<D1000

②C235计数到10000，C235<D1000变为C235>=D1000，M100置位输出1次，程序立即复位M100

③D1000加1000，C235>D1000变为C235<D1000，M100置位输出1次，程序立即复位M100

④C235计数到11000，C235<D1000变为C235>=D1000，M100置位输出1次，程序立即复位M100

即在10000比较点10000时输出一次，修改D1000时又输出了一次，在11000比较点10000时输出一次，两个比较点共输出了3次，应用时需注意。

3.3.4 高速区间比较[DHSZ]

对指定的高速计数器当前值与设定的 2 个比较值(区间)进行数值的比较，并将比较结果输出至位软元件(占用连续 3 个位元件)中。

- **指令格式**

1) 指令格式



2) 数据格式

操作数种类	内容	数据类型
D1	区间比较值下限	BIN 32 位
D2	区间比较值上限	BIN 32 位
D3	比较计数值对应的寄存器	BIN 32 位
D4	比较输出结果	位
D5	高速数据运算区号	BIN 32 位

3) 数据软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1	•	•	•							
D2	•	•	•							
D3		•	•	•						
D4							•	•	•	
D5	•	•	•							

- **功能和举例**

1) D1 指定高速计数器待比较的计数值区间下限值。

2) D2 指定高速计数器待比较的计数值区间上限值。

◆ D2 的设定值必须大于D1 的设定值。

3) D3 通常指定高速计数器 C235～C255 或者高速脉冲输出寄存器。

4) D4 指定比较输出结果的位元件。

◆ 比较输出的位元件满足条件后立即置位，不受扫描周期的影响。

◆ 指定的 3 个位元件必须处于相同的 16 位地址内（Y、M、S 设置相同）

例如：从M0 开始，每16 个 M 组成一个字，指定的位元件必须在同一个字内，比如指定 M15 是不正确的，因为M15，M16，M17 分别处于不同的字内(M15 处于 M0～M15 内，M16，M17 处于M16～M23 内)。

5) D5 指定指令所占用的高速数据运算区号 0～5。

比较输出动作以指令 DHSZ K3000 K4000 C235 Y10 K1 举例：

1) 区间下限为 3000，区间上限为 4000
2) 指定的高速计数器为 C235
3) 指令驱动的输出端口是 Y10，Y11，Y12
4) 指令占用高速数据运算区号 1

C235 计数值与输出端口的电平状态关系如下表所示：

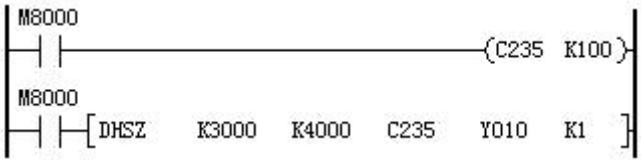


图.高速区间比较示例图

比较条件	Y10状态	Y11状态	Y12状态
C235<3000	ON	OFF	OFF
3000≤C235≤4000	OFF	ON	OFF
C235>4000	OFF	OFF	ON

表.C235计数值与输出端口状态

比较置位/比较复位/区间比较区别

1) DHSZ 区间比较对输出点完全管理其状态，包括置位和复位两种状态。
2) DHSCS 高速比较置位和DHSCR 高速比较复位仅单一管理输出点的状态。置位指令仅执行置位操作，复位指令仅执行复位操作。当不满足该条件时，不会对输出点进行反向操作。

3.3.5 高速表格比较输出[DHSCT]

对指定的高速计数器当前值与设定的比较表格内当前行的比较值进行比较，满足条件后输出对应的位元件，表格行下移。该过程依次执行，直到表格内所有行的数据比较结束。

- **指令格式**

1) 指令格式



2) 数据格式

操作数种类	内容	数据类型
D1	表格起始单元	BIN 32 位
D2	表格行数	BIN 32 位
D3	比较计数值对应的寄存器	BIN 32 位
D4	输出单元起始地址	位
D5	输出单元使能位	BIN 32 位

3) 数据软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1		•	•							
D2	•	•	•							
D3		•	•	•						
D4							•	•	•	
D5	•	•	•							

- **功能和举例**

1) D1 指定比较表格的起始软元件地址。如 D1 指定D1000，则 D1000 为表格起始单元。

2) D2 指定比较表格的行数，D2 的取值范围为 1～128。

◆ 每个行占用三个数据寄存器。其中比较数据 2 个，比较输出值 1 个，如下图。

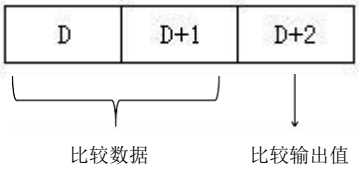


图.每行数据结构

3) D3 通常指定高速计数器 C235～C255。

4) D4 指定比较输出结果的位元件起始地址。从起始地址开始指定连续的 16 个位元件。当计数器当前值大于等于表格某行的比较值时，这 16 个位元件输出该行的输出值。

◆ 比较输出的位元件满足条件后立即置位，不受扫描周期的影响。

◆ 指定的位元件起始地址必须被 16 整除。比如 Y0 合法，Y3 不合法，M、S 用法一样。

5) D5 指定比较输出位元件的输出使能位，按位进行设置。

◆ 设置为 1 时，若表格输出值对应位为 1，对应位元件输出。

◆ 设置为 0 时，无论表格输出值是否为 1，对应位元件始终不输出。

◆ 假定比较输出结果的位元件起始地址指定 M0 ， 输出使能位设置为 15 ， 转换成 2 进制为0000000000001111，即 M0～M3 输出使能位有效，其他 M 不输出。

◆ 输出的位元件立即输出，不受扫描周期的影响。即使后续条件不满足，该指令也不会执行软元件的复位操作，需要手动复位。

6) 表格输出每次满足输出条件输出时，触发I050中断，同时捕获高速计数器当前计数值。

◆ 比较输出的位元件满足条件后立即置位，同时触发中断I050（须I050中断允许状态），不受扫描周期的影响。

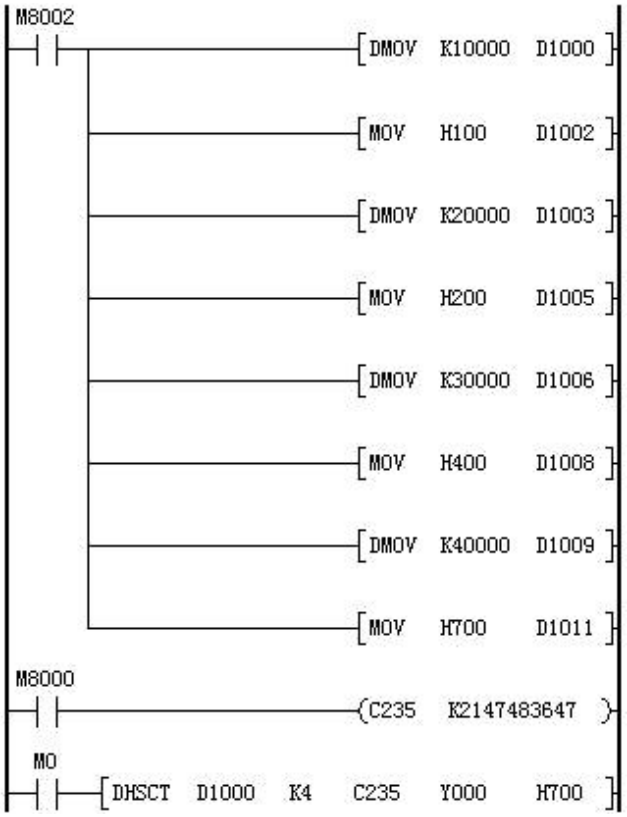
◆ AXV6虚拟轴的当前高速计数实时值传送到R214（L）。AXV7虚拟轴的当前高速计数实时值传送到R264（L）。

- 表格格式

假定 D1 指定 1000，D2 指定为 5（行）。则其地址分配如下所示：

表格行	比较值	输出值	输出条件
1	D1000(L)	D1002	计数器当前值≥D1000(L)
2	D1003(L)	D1005	计数器当前值≥D1003(L)
3	D1006(L)	D1008	计数器当前值≥D1006(L)
4	D1009(L)	D1011	计数器当前值≥D1009(L)
5	D1012(L)	D1014	计数器当前值≥D1012(L)

使用计数器 C235，输出元件指定为 Y10～Y12。表格首地址指定为 D1000。梯形图如下:



配置高速计数表格输出的参数

- 配配置表格第 1 行参数，计数比较值为 10000，输出状态为H100，即对应 Y10 输出。
- 配配置表格第 2 行参数，计数比较值为 20000，输出状态为H200，即对应 Y11 输出。
- 配配置表格第 3 行参数，计数比较值为 30000，输出状态为H400，即对应 Y12 输出。
- 配配置表格第 4 行参数，计数比较值为 40000，输出状态为H700，即对应 Y10～Y12 输出。
- 使使能输出高速计数器 C235
- 表格首地址=D1000行数=K4 高速计数器=C235 比较输出单元首地址=Y0比较输出使能位=H700
- （Y10-Y12 输出使能位）

图.高速表格比较输出示例

C235 的计数值与 Y 端口的输出关系如下表所示：

比较值	输出值
C235≥1000	Y10=ON, Y11=OFF, Y12=OFF
C235≥2000	Y10=OFF, Y11=ON, Y12=OFF
C235≥3000	Y10=OFF, Y11=OFF, Y12=ON
C235≥4000	Y10=ON, Y11=ON, Y12=ON

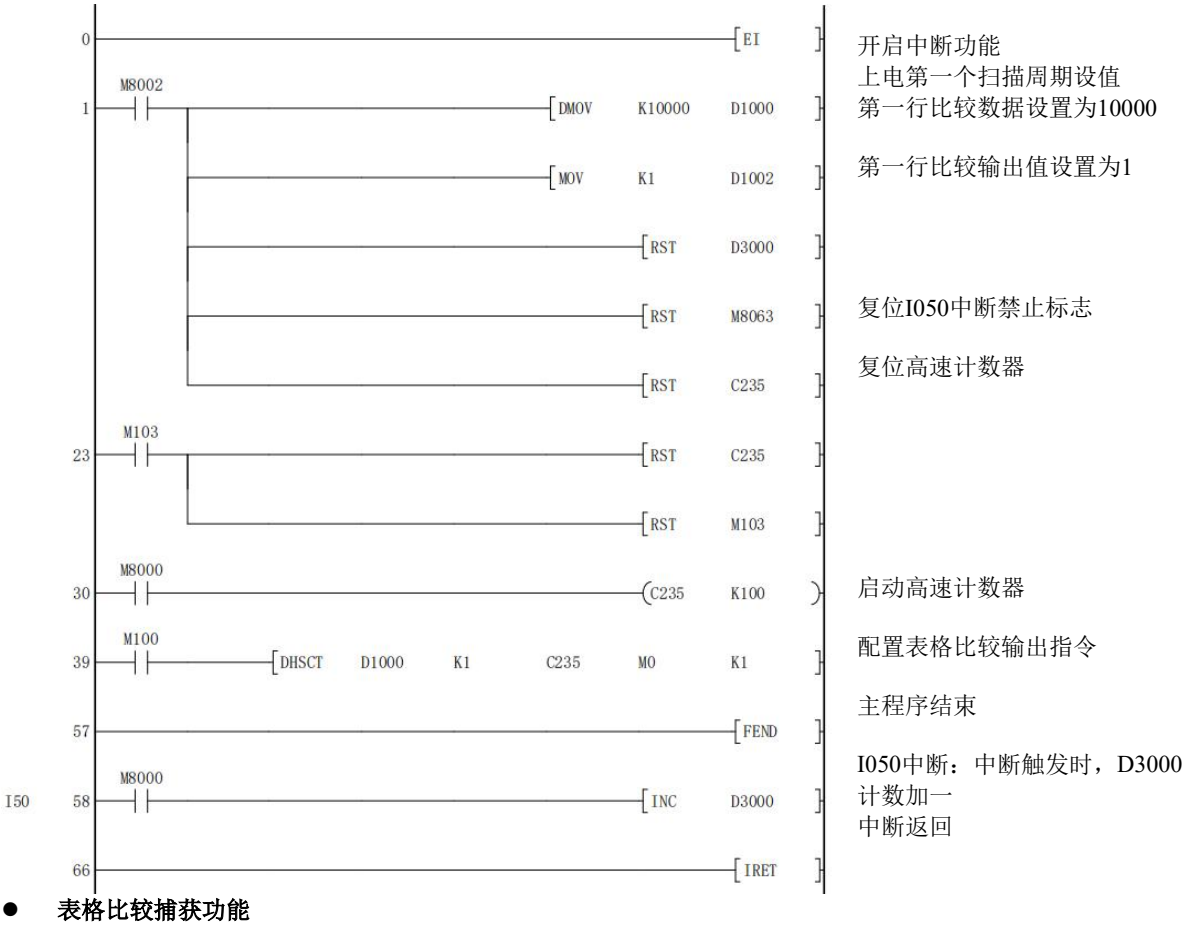
上表中共有 4 种计数比较值需要输出，故表格行数设计为 4 行。

输出 Y10～Y12，那么首地址应填写Y0，因为 Y10～Y12 在Y00～Y17 范围之内（Y 为8 进制）。输出使能的 2 进制数值应为 00000111 00000000，对应 16 进制数为H0700。

- **表格比较输出触发中断功能**

梯形图中配置有软中断I050时，当表格比较输出指令满足表格中的条件，软中断I050即触发一次。

例：当高速计数值（C235）大于10000时，表格比较输出M0，同时中断I050触发一次



- **表格比较捕获功能**

梯形图中配置有高速计数指令时，每当表格比较输出指令输出时，系统会立即将当前的高速计数值传送到寄存器R114（L）中。

- **错误示例**



● 置位M1,复位D2000，再置位M100，给D2000赋值20000，发现第一行输出Y10，第二行表格没有输出Y11,给D2000赋值30000,发现第三行表格没有输出Y12，给D2000赋值40000,发现第四行表格没有输出Y10~Y12。重新运行程序，D2000从0给值10000再给20000再给30000再给40000，发现表格每一行依次输出相应Y。

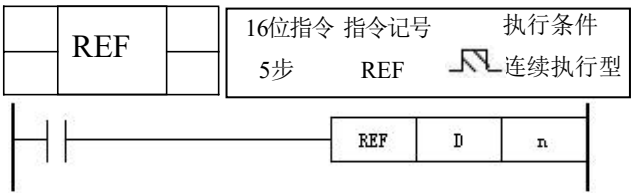
由此可知，表格比较输出待比较的寄存器必须在每一行表格区间内依次比较输出，若待比较的寄存器值直接大于表格第二行比较值，则除第一行外剩下每一行的表格满足输出条件都不会输出，指令运行异常。

3.3.6 输入输出刷新[REF]

需要实时更新输入输出的信息，可使用输入输出刷新指令REF，输入输出不受扫描周期的影响。

- **指令格式**

1.指令格式



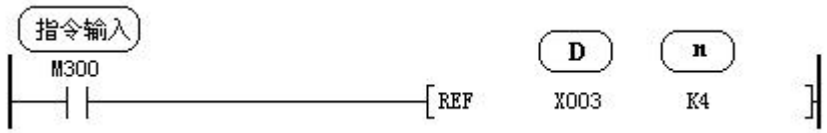
2.数据格式

操作数种类	内容	数据类型
D	刷新起始单元	位
n	刷新个数	BIN 16 位

3.数据软元件

操作数	位/字软元件									
种类	K/H	D	R	C	T	X	Y	M	S	V/Z
D						•	•			
n	•									

- **功能和举例**



D ：输入X或输出Y的起始软元件号。X000~X037；Y000~Y027；仅对主机有效。

n ：输入或输出的刷新点数。K1~K32；通常用十进制；H1~H20。

应用

★ REF指令只适用于主机输入输出，不适用于扩展输入输出。

★ 在FOR~NEXT指令之间、标号(新步号)~CJ(老步号)之间，一般需要REF指令。

★ 在有输入输出动作的中断子程序中，需进行输入输出刷新。

★ 输出刷新中的输出接点将在输出继电器或晶体管应答时间后动作。

继电器输出型的应答滞后时间为10ms，晶体管输出型约为0.2ms以下。

3.4 单轴基本运动指令

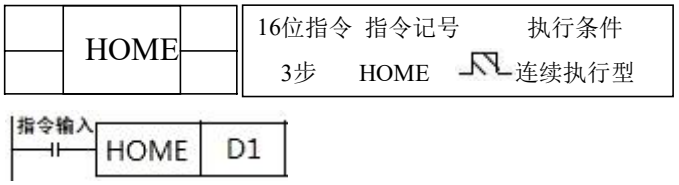
单轴基本运动指令涵盖了回原点、定位、运动规划制表、表格输出和插补输出指令。在执行某个软件虚拟输出轴的运动指令前，请先确认该轴的相关轴参数已经配置完成。

3.4.1 回原点[HOME]

指定对应的软件虚拟输出轴执行回原点功能。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位

3) 对象软元件

操作数	位/字软元件									
种类	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1	•									

- **相关软元件**

该指令相关寄存器或寄存器的位如下表：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R308	R358	R408	R458	近点信号输入端口指定	16 位	0
R314 bit0	R364 bit0	R414 bit0	R464 Bit0	近点信号使能	位	0
R314 Bit4	R364 Bit4	R414 Bit4	R464 Bit4	近点信号极性	位	0
R309	R359	R409	R459	Z 脉冲信号输入端口指定	16 位	0
R314 Bit1	R364 Bit1	R414 Bit1	R464 Bit1	Z 脉冲信号使能	位	0
R314 Bit5	R364 Bit5	R414 Bit5	R464 Bit5	Z 脉冲信号极性	位	0

R310	R360	R410	R460	左限位信号输入端口指定	16 位	0
R314 Bit2	R364 Bit2	R414 Bit2	R464 Bit2	左限位信号使能	位	0
R314 Bit6	R364 Bit6	R414 Bit6	R464 Bit6	左限位信号极性	位	0
R312	R362	R412	R462	原点回归速度	16 位	0
R313	R363	R413	R463	原点爬行速度	16 位	0
R315	R365	R415	R465	过近点脉冲数	16 位	0
R316	R366	R416	R466	过近点 Z 脉冲数	16 位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源类型选择	位	0
R314 Bit14	R364 Bit14	R414 Bit14	R464 Bit14	近点信号边沿选择	位	0
R314 Bit15	R364 Bit15	R414 Bit15	R464 Bit15	停止方式选择	位	0

- **功能和举例**

该指令主要用于机械上电后根据外部信号寻找机械零点位置，通常仅在机械开始工作前运行一次。所以指令配置以准确为目的而不过多考虑速度。

首先展示基础的过近点信号停止功能，以轴0 为例，使用近点信号相关的软元件有：

1) R308——近点信号端口指定

2) R314bit0——近点信号使能

3) R314bit4——近点信号极性

近点信号来源大多为机台上的光电开关或磁性开关，接在控制器的 X 点作为回原点指令开始减速的判断条件。根据近点信号接入的 X 点配置 R308指定该 X 点为近点信号端口，配置方法如下：

信号来源有两种：
应用于外部信号和内
近点信号事件。只能
见R314bit13 说明。

配置完近点信号
设置为 1，使能近点

根据接近开关实
常开信号设为 0，常

近点信号端口 =	选定端口	近点信号端口 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

X与特殊M，特殊M可
部逻辑运算并用触发
选择其中一种，具体

端口后，将R314bit0
信号。

际极性配置R314bit4，
闭信号设为 1。

4) R309——Z 脉冲信号端口指定

5) R314bit1——Z 脉冲信号使能

6) R314bit5——Z 脉冲信号极性

Z 脉冲信号一般来自于电机的编码器，此信号准确且快速，用于二次判断原点位置信息。因为端口信号处理始终存在延时，即便每次经过近点信号后再走固定的脉冲数依旧存在误差，回原点速度越快该误差越大。所以需要高精度的情况下推荐

使用过近点Z 脉冲+脉冲数的方式来停止。即遇到近点信号后开始减速至爬行速度，遇到Z 脉冲信号后走固定脉冲数停止。

Z 脉冲信号指定方法同近点信号，但只支持 X 方式，且只能指定 X0 到X7 高速信号端口。当 Z 脉冲信号来源于电机时，该信号的持续时间长度只有一个脉冲时间，所以此时爬行速度不宜设置过快。X0~X7 端口的采集频率为 2KHZ，即最多可采集到不低于 1/2000s 的信号。假设编码器为每圈2500 脉冲，则爬行速度低于每秒 2000 个编码器脉冲的速度以内，即 48RPM。

7) R314bit13——输入信号源类型选择

设为 0：端口信号源指向 M8096～M8111 设为 1：端口信号源指向 X0~X17

8) R314bit14——近点信号边沿选择

设为0：离开近点信号的后边沿后开始计算过近点脉冲数与过近点 Z 脉冲数。设为1：碰到近点信号前边沿后开始计算过近点脉冲数与过近点 Z 脉冲数。

9) R314bit15——停止方式选择

10) R315 过近点脉冲数

11) R316 过近点 Z 脉冲数

R314bit15 设为0：先计数过近点 Z 脉冲数个 Z 脉冲，然后再计数过近点脉冲数个脉冲停止回原点流程。

设为 1：计数过近点 Z 脉冲数个 Z 脉冲，或计数过近点脉冲数个脉冲回原点流程。

12) R312 原点回归速度

13) R313 原点爬行速度

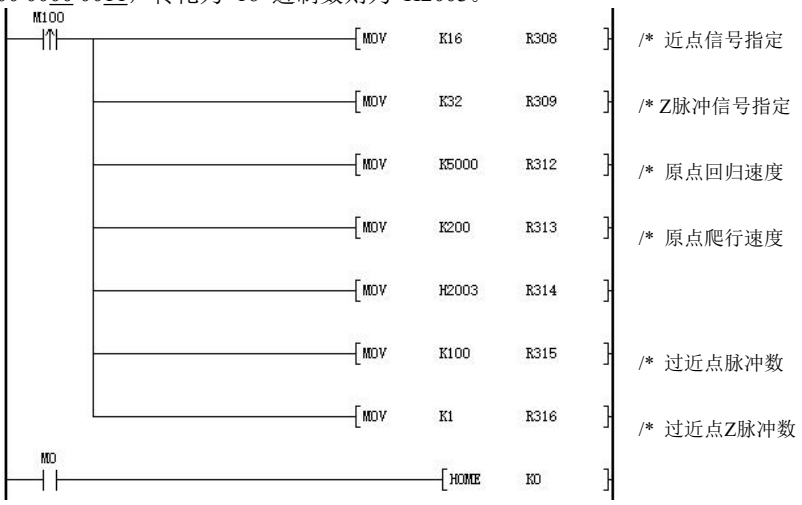
使用时应遵守原点爬行速度＜原点回归速度＜轴最大速度。配置方法与轴最大速度相同。

当轴时间当量设为 10 时，R312 设为1000 时原点回归速度对应的脉冲频率为 100KHZ。有 Z 脉冲信号时爬行速度根据以上说明，应该低于 2KHZ。

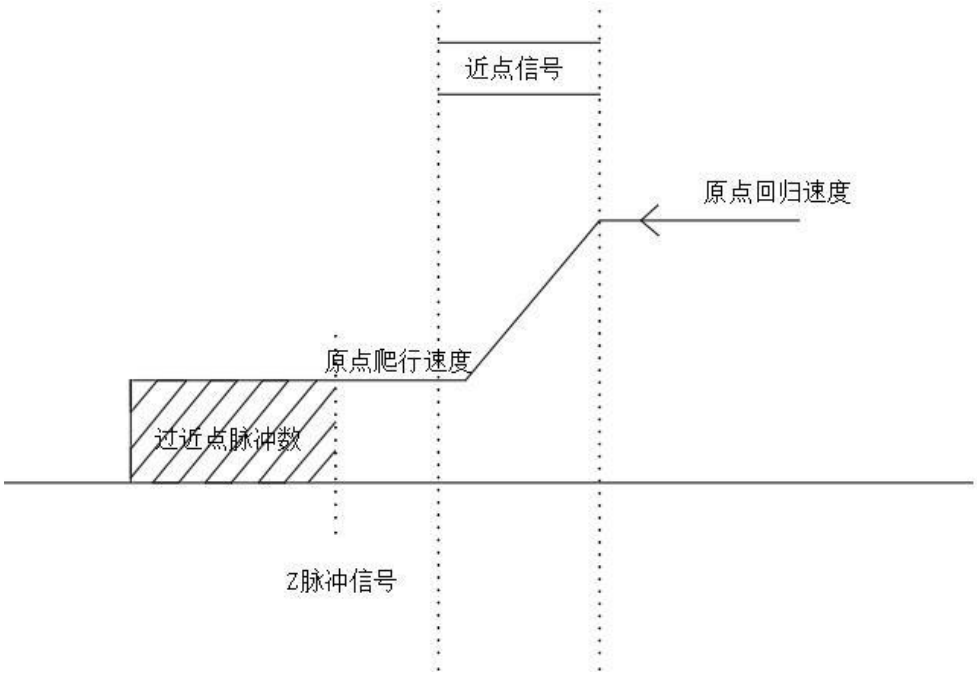
使用案例 1：

将X4 指定为近点信号，X5 指定为Z 脉冲信号，开关皆为常开型。原点回归速度设为 50KHZ， 原点爬行速度设为 2KHZ。离开近点信号后边沿 1 个Z 脉冲后再过 100 个脉冲停止。

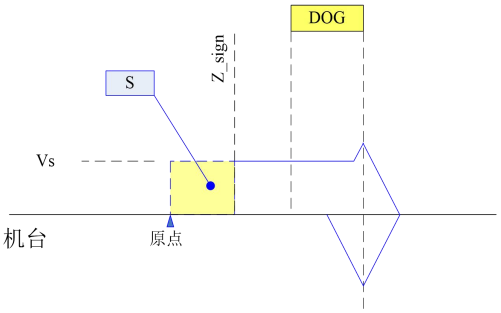
R314 为 0010 0000 0000 0011，转化为 16 进制数则为 H2003。



当起点位于近点信号右侧时：

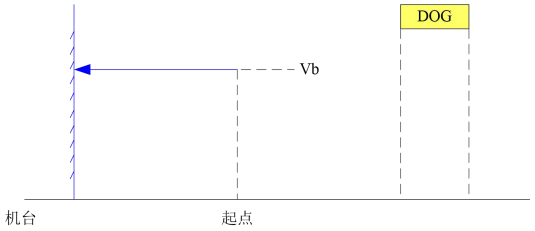


当恰好位于原点位置时，先自动以反向的回原点速度离开原点，在转换方向执行回原点。



符号
DOG：近点信号
Z_sign：Z脉冲信号
Vs：回归爬行速度
S：过近点脉冲数

当起点位于近点信号的左侧时



符号
DOG：近点信号
Vb：原点回归速度

此时会直接撞在机械左侧上。

如果需要解决这种情况的话有以下两种方案：

使用案例 2：

在案例 1 的基础上增加左限位信号。

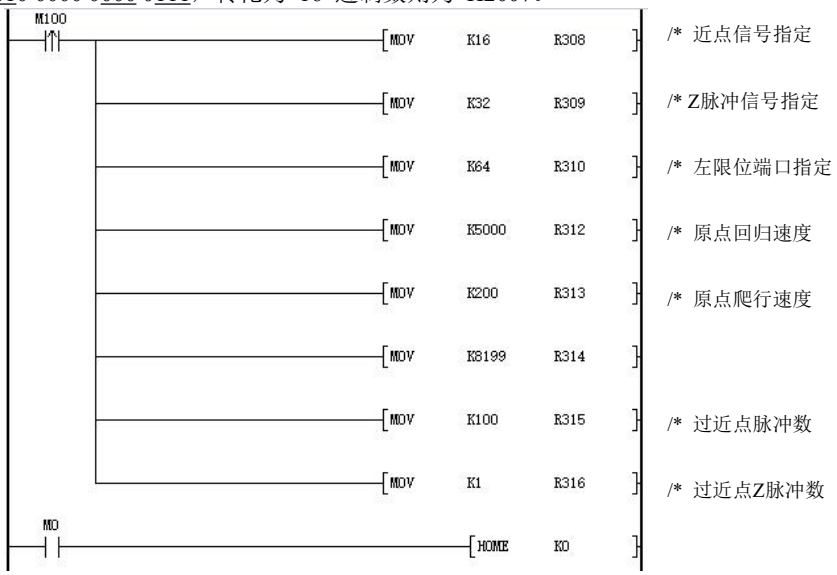
1) R310——左限位信号端口指定

2) R314bit2——左限位信号使能

3) R314bit6——左限位信号极性

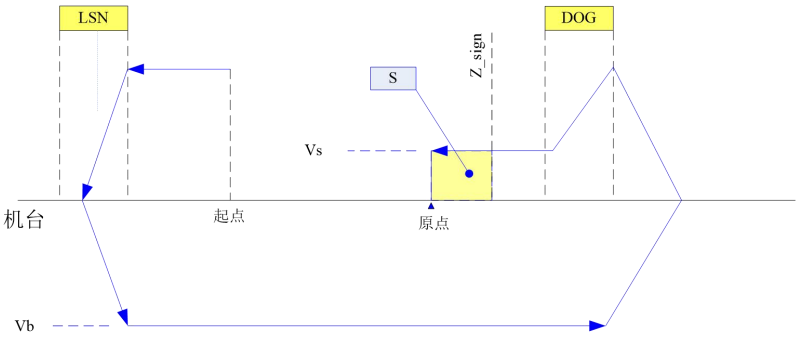
将X4 指定为近点信号，X5 指定为Z 脉冲信号，X6 指定为左限位信号。开关皆为常开型。原点回归速度设为50KHZ，原点爬行速度设为 2KHZ。离开近点信号后边沿 1 个 Z 脉冲后再过 100 个脉冲停止。

R314 为 0010 0000 0000 0111，转化为 16 进制数则为 H2007。



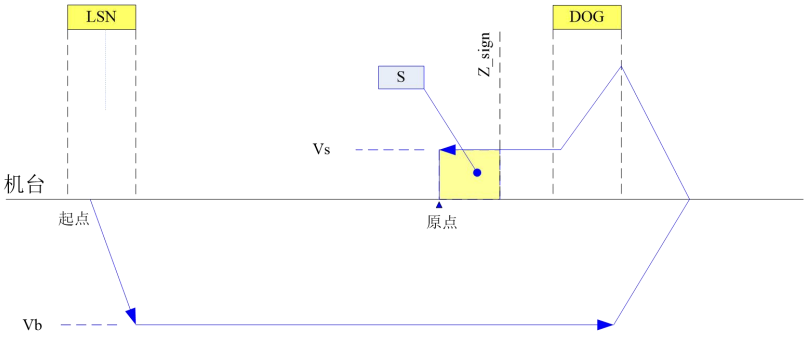
此时运行情况为：

遇到左限位信号后开始减速并反向，反向过程离开近点信号后再次反向并减速然后完成回原点过程。



符号
LSN：左限位
DOG：近点信号
Z_sign：Z脉冲信号
Vs：回归爬行速度
Vb：原点回归速度
S：过近点脉冲数

如果起点刚好在左限位内，则直接反向。



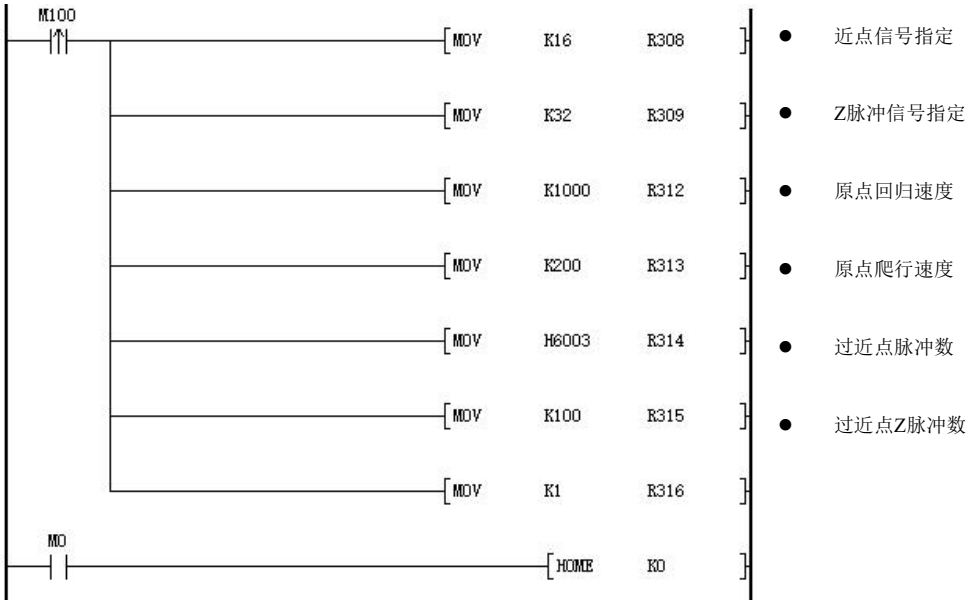
符号
LSN：左限位
DOG：近点信号
Z_sign：Z脉冲信号
Vs：回归爬行速度
Vb：原点回归速度
S：过近点脉冲数

使用案例 3：

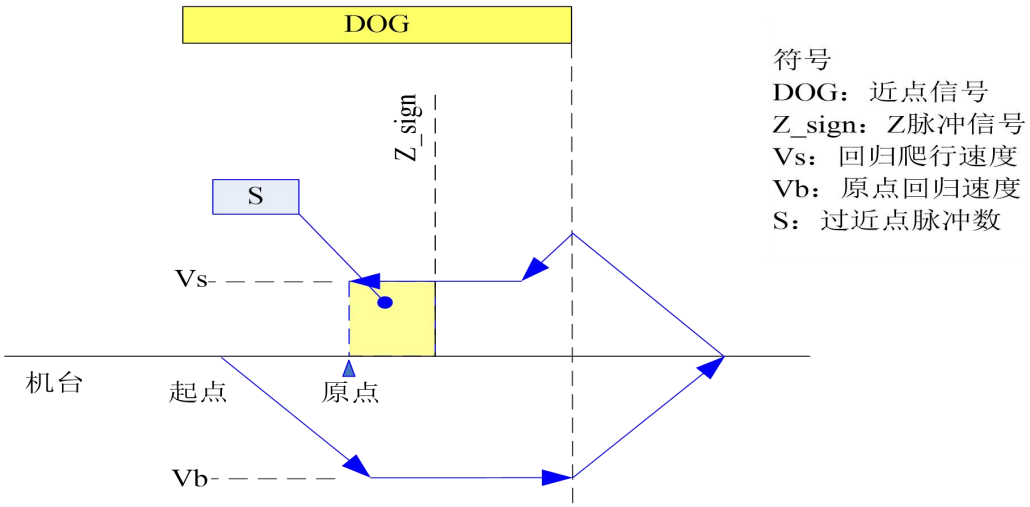
有时现场因为工艺原因机台无法安装左限位开关。可以将近点信号移至左侧限位上，使起点不可能在近点信号左侧。此时碰到原点后立即开始执行定位停止，并且留有的减速距离可能较短。故R314bit14 设为1：碰到近点信号前边沿后开始计算过近点脉冲数与过近点 Z 脉冲数。且原点回归速度不得设置过快。

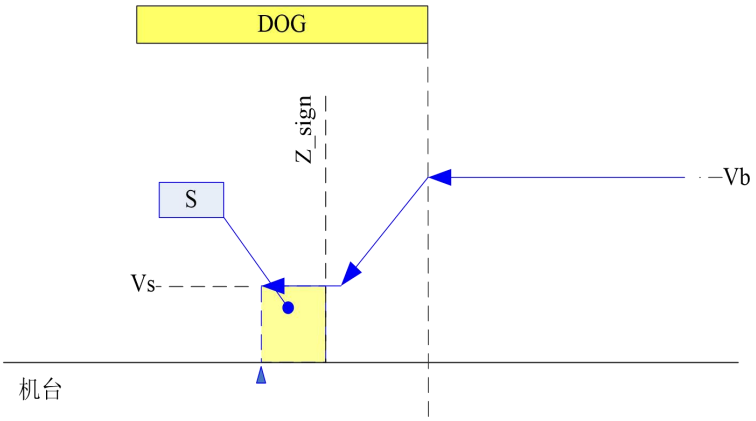
将X4 指定为近点信号，X5 指定为Z 脉冲信号，开关皆为常开型。原点回归速度设为 10KHZ，原点爬行速度设为 2KHZ。离开近点型号后边沿 1 个Z 脉冲后再过 100 个脉冲停止。

R314 为 0110 0000 0000 0011，转化为 16 进制数则为 H6003。



当起点在近点信号内时





符号
DOG：近点信号
Z_sign：Z脉冲信号
Vs：回归爬行速度
Vb：原点回归速度
S：过近点脉冲数

该方案因为受到原点回归速度不得设置太大的影响，如果起点较远，回原点执行时间会比较长。如果在经常需要用到回原点的情况下要使用该方案则需要再进一步修改。

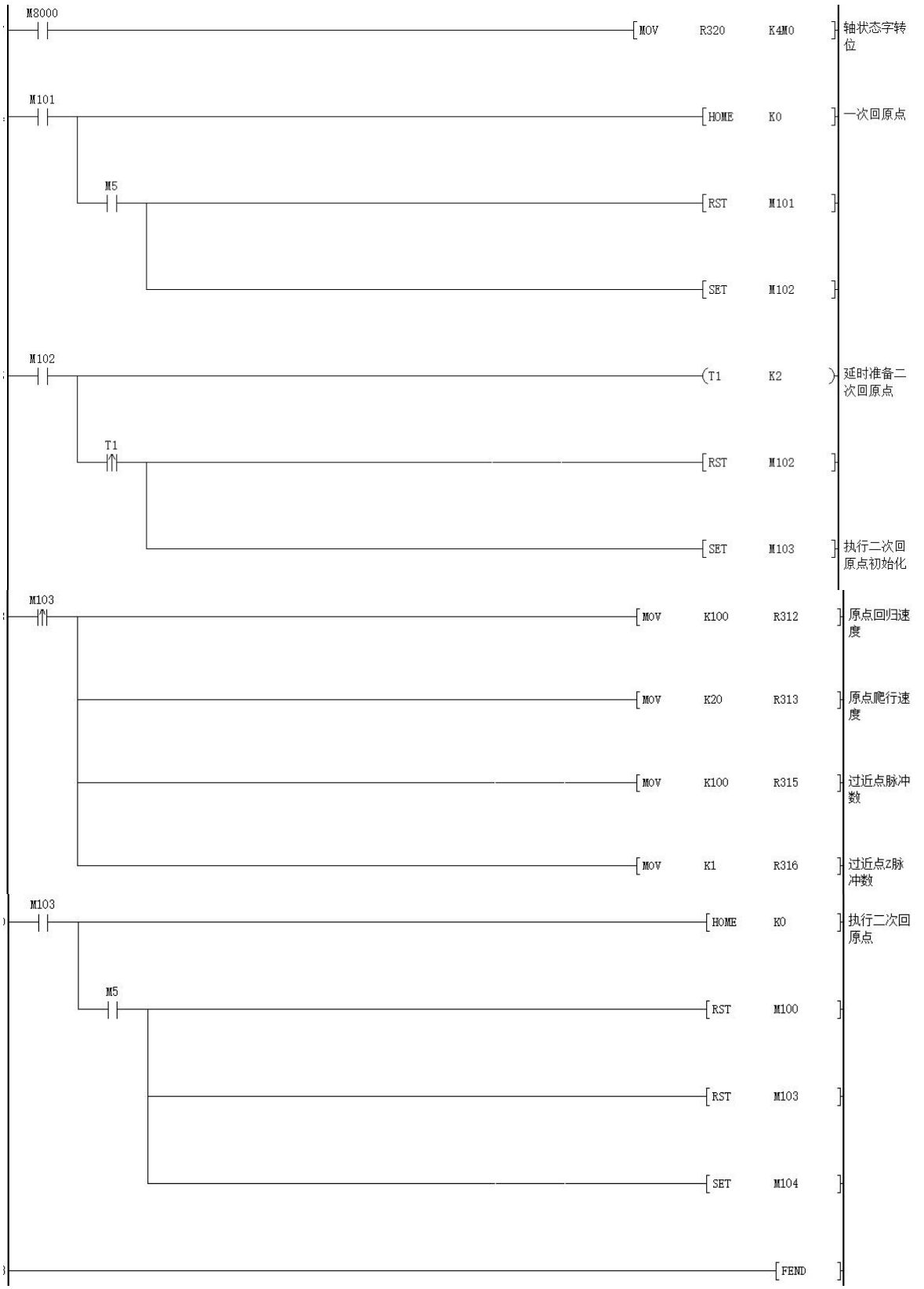
使用案例 4：

整个回原点定位过程一共采用两次回原点指令，第一次回原点速度可以设置较快，采用前边沿停止的方式，并将爬行速度与过近点脉冲数、过近点 Z 脉冲数皆设为 0。则执行情况为遇到近点信号后以最大加速度停止。判断执行完毕后，执行第二次回原点指令，使回原点速度慢，并设置正常的过近点脉冲数与过近点 Z 脉冲数。

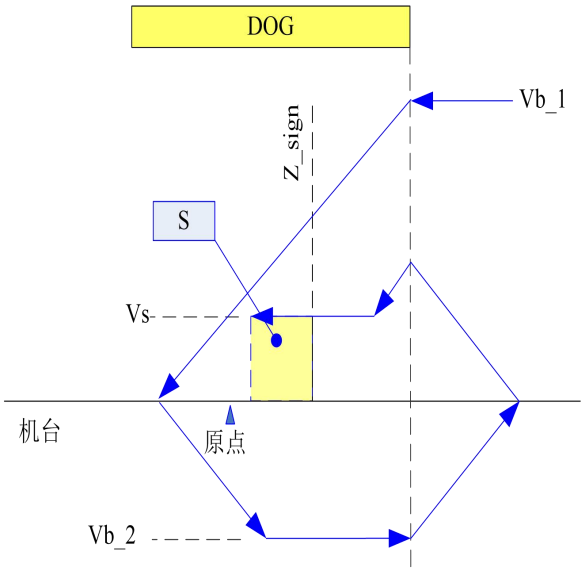
将X4 指定为近点信号，X5 指定为Z 脉冲信号，开关皆为常开型。第一次原点回归速度设为 5KHZ，原点爬行速度设为0KHZ。离开近点信号后边沿 0 个Z 脉冲后再过 0 个脉冲停止。第二次原点回归速度设为1KHZ，原点爬行速度设为200HZ。离开近点信号后边沿 1 个 Z 脉冲后再过 100 个脉冲停止。

R314 为 0110 0000 0000 0011，转化为 16 进制数则为 H6003。





该程序中启动 M101 后自动开始执行两次回原点指令。并两次执行完成后置位M104 作为回原点的完成标志位。具体运行情况如下图：



符号
DOG：近点信号
Z_sign：Z脉冲信号
Vs：回归爬行速度
Vb_1：一次回归速度
Vb_2：二次回归速度
S：过近点脉冲数

- **注意要点**

1) 速度设置中必须按照：原点爬行速度＜原点回归速度＜轴最高限速来设置；
2) Z 脉冲信号端口配置只能指定为X 类型信号，不能使用M 类型。且 R314bit13 统一指定所有信号端口类型。即如果用到 Z 脉冲的情况下，近点信号，左限位信号，右限位信号，急停信号，中断信号①端口均只能指定为 X 类型；
3) Z 脉冲信号端口并非只能接编码器 Z 脉冲。如果在近点信号后再加一个磁性开关接到指定的 Z 脉冲信号端口上也可以大幅提高回原点精度。回原点误差受到运行速度与端口处理延时影响，经过近点信号后运行速度由原点回归速度降至原点爬行速度，此时再记录脉冲数产生的误差必然小于直接采用过近点脉冲数方式的误差；
4) 回原点执行方向固定为向脉冲的负方向执行。如果受到电机安装方向的影响，实际运行并非需要的方向，可以将回原点指令在数字轴上执行，然后通过电子齿轮指令反向输出到脉冲轴上。不可以通过改变近点信号的极性设置来改变运行方向，如果端口信号极性与实际不符会造成指令执行逻辑混乱。

附：①右限位信号，急停信号，中断信号具体定义与用法见说明书其他部分介绍，与回原点指令的执行无关

3.4.2 定速定长[DPLSY]

指定软件虚拟输出轴(AXV) 按照指定频率输出一定个数脉冲数(带方向)。

- 指令格式

1) 指令格式



2) **设定数据**

操作数种类	内容	数据类型
D1	脉冲频率	BIN 32 位
D2	脉冲数	BIN 32 位
D3	软件虚拟输出轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•									

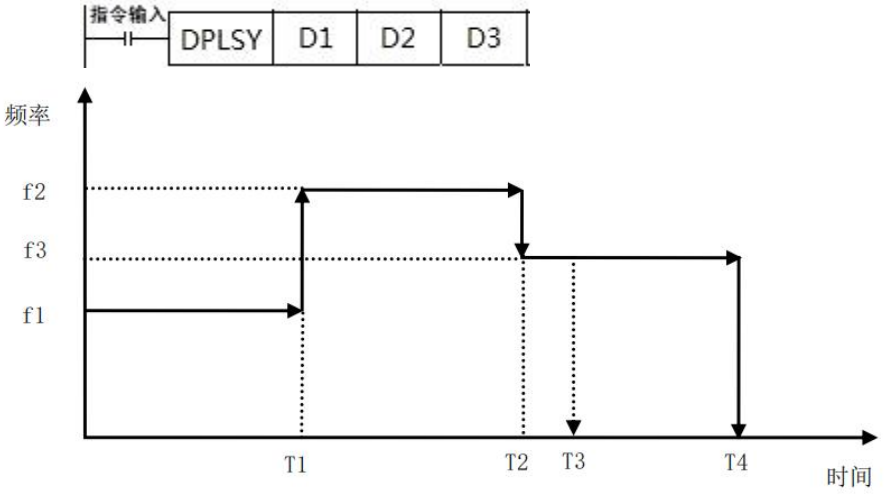
- **相关软元件**

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴				
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R302	R352	R402	R452	时间当量	16 位	1000	默认
R305	R355	R405	R455	最大速度	16 位	20000	读写
R314	R364	R414	R464	轴事件输入点	16 位	0	读写
R318	R368	R418	R468	相对位置脉冲数	32 位	0	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读
R322	R372	R422	R472	1ms 插补增量	16 位	0	只读
R323	R373	R423	R473	轴急停事件	16 位	0	读写

- **功能和举例**

该指令可实现指定软件虚拟输出轴(AXV)按照指定频率输出一定个数脉冲数(带方向)。



- 以初始的f1 频率发送脉冲，f1 为指令指定的频率
- T1 处在线修改了输出频率 f2，以该频率输出脉冲
- T2 处在线修改了输出频率 f3，以该频率输出脉冲
- T3 处若设置了急停事件使能，且相应的事件产生，则立即停止
- T4 处若设置了有限个脉冲输出，且此时输出完毕，则立即停止

1) 在D1 中指定脉冲频率

	设定范围
32 位运算	0~200000（默认1000ms时间当量）

注：指令运行中修改有效；

频率设定值受轴最大速度限制，以及硬件端口频率限制。

2) 在D2 中指定脉冲数

	设定范围
32 位运算	-16000000~16000000

注：配置方向端口后，负值为反转，端口 0FF，正值为正转，端口 ON。

3) 在 D3 中指定软件虚拟输出轴

	设定范围
32 位运算	0~3

4) 输入点掩码对应表格：

D4/D5	选定端口	D4/D5	选定端口
0	不选定	256	X10/M8104
1	X0/M8096	512	X11/M8105
2	X1/M8097	1024	X12/M8106
4	X2/M8098	2048	X13/M8107
8	X3/M8099	4096	X14/M8108
16	X4/M8100	8192	X15/M8109
32	X5/M8101	16384	X16/M8110
64	X6/M8102	-32768	X17/M8111
128	X7/M8103		

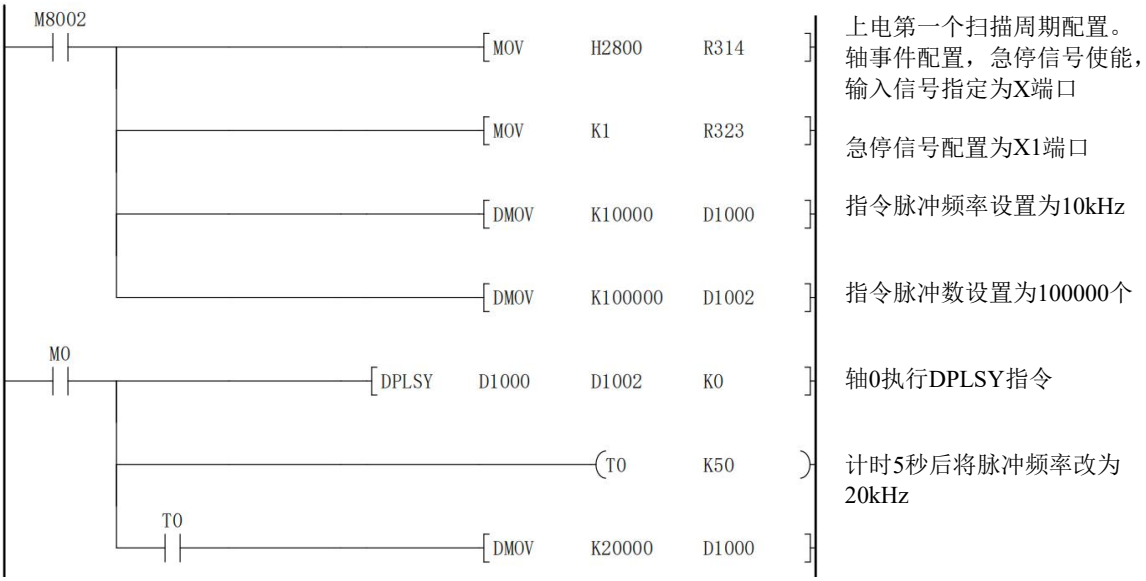
另：端口选定 X 或 M，需对对应设定轴参数中的“轴事件输入点”中的第 13 位设置，例如：虚拟 0 轴中的 R314 中的第 13 位为 0 指定 M，为 1 指定 X。

R314	0	0	1	0		0	0	0	0		0	0	0	0		0	0	0	0

上图中 R314=8192，端口定义 X0~X17 有效。

5) 案例

当需要控制伺服电机正转 1000000 个脉冲，开始运行频率 10KHz，运行后 5 秒要改变频率为 20KHz， 同时支持 X0 端子信号来急停，梯形图如下：



- **注意要点**

1) 必须使用常开触点导通执行指令，断开触点则轴运动立即停止；

2) 指令的运行速度受指定虚拟轴最大速度限制；

3) 配置急停功能时需设置“轴事件输入点”参数和“轴急停事件”参数。

3.4.3 加减速定长[DPLSR]

指定软件虚拟输出轴(AXV)按照指定频率(带加减速)输出一定个数或无限个数的脉冲数(带方向)。可指定从启动速度到最高速度的加速时间。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	脉冲频率	BIN 32 位
D2	脉冲数	BIN 32 位
D3	加减速时间	BIN 32 位
D4	软件虚拟输出轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•									

- **相关软元件**

1) D1 指定输出的脉冲频率。

实际输出的脉冲频率通过轴参数轴时间当量计算得出。计算公式 =D1÷(轴时间当量÷1000)。脉冲频率必须填写>0 的数值。

D1 的选择范围应该大于轴起始速度，小于轴最高限速。

D1 的设置如果小于轴参数轴起始速度 。最终速度为轴参数轴起始速度。

且当 D1 的设置大于轴最高限速时，实际的目标速度不会超过轴最高限速 。无论是正向还是负向发脉冲，该值皆为正数，是速度的绝对值。

2) D2 指定输出脉冲数。

D2 的取值范围是-16000000～16000000。

当 D2 填写为负值时，且配置了方向端口，则为反方向脉冲输出。没有指定方向端口， 则负值不起作用(按正值输出)。

3) D3 指定加减速时间

单位毫秒，只可以填入正值。

该值为从起始速度加速至实际指定脉冲频率的时间。当设置D3换成轴加速度值小于默认轴加速度时，则实际运行曲线按照设置的加减速时间运行，当设置D3换算成轴加速度大于默认轴加速度时，则按照默认轴加速度曲线运行。

换算公式如下：

$$a = \frac{V设定 - V起始}{t}$$ （速度单位为kHz，时间t单位为ms）

注意：当设定速度大于轴默认最高速时，则V设定按照轴默认最高速计算，轴最高速为200K。

4) **D4 指定软件虚拟输出轴号(AXV0～AXV3)**对应数值 0～3。

相关轴参数列表：

软元件编号				名称	数据类型	初始值
0轴	1轴	2轴	3轴			
R302	R352	R402	R452	轴时间当量	16 位	1000
R305	R355	R405	R455	轴最高限速	16 位	20000
R306	R356	R406	R456	轴起始速度	16 位	0
R307	R357	R407	R457	轴最大加速度	16 位	1000
R323	R373	R423	R473	轴急停事件	16 位	0
R314 Bit10	R364 Bit10	R414 Bit10	R464 Bit10	急停信号极性 =0 常开 =1 常闭	位	0
R314 Bit11	R364 Bit11	R414 Bit11	R464 Bit11	急停信号使能 =0 不使能 =1 使能	位	0
R314 Bit12	R364 Bit12	R414 Bit12	R464 Bit12	急停类型选择 =0 立即停车 =1 减速停车	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源选择 =0 指向M =1 指向X	位	0

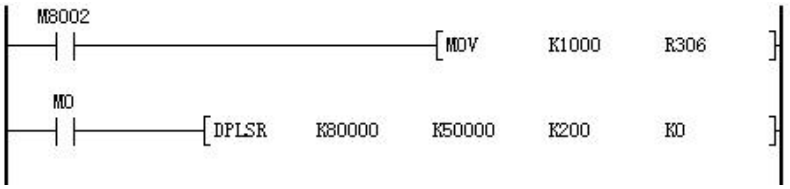
- **功能和举例**

说明案例 1

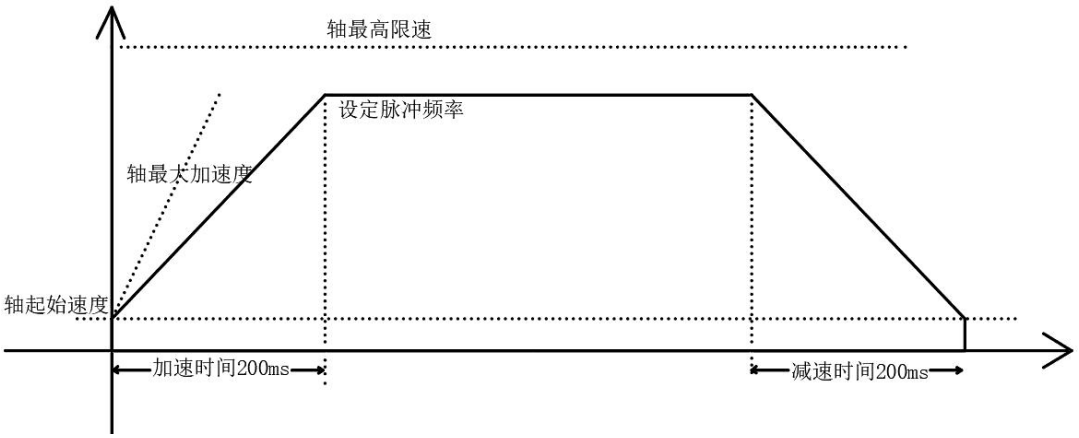
使用 DPLSR 指令设定从 10KHZ 加速至 80KHZ，使虚拟轴 0 发出 50000 个脉冲，加减速过程为 200ms。

需要配置的参数	定义	配置值
D1	脉冲频率	80000
D2	脉冲数	50000
D3	加减速时间	200
D4	软件虚拟输出轴号	0
R306	起始速度	1000

示例梯形图：



效果图解：



轴最大加速度默认 1000，时间当量默认为 1000，即轴加速度为 10P/ms²。

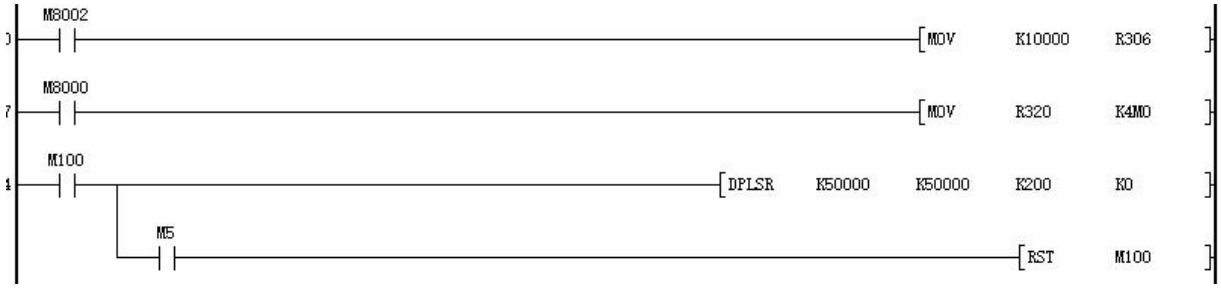
可算出该运动过程加速度为 0.35P/ms²小于轴最大加速度。则以设定的加减速时间运行。

说明案例 2

使用 DPLSR 指令设定从 100KHZ 减速至 50KHZ，使虚拟轴 0 发出 50000 个脉冲，加减速过程为 200ms。

需要配置的参数	定义	配置值
D1	脉冲频率	50000
D2	脉冲数	50000
D3	加减速时间	200
D4	软件虚拟输出轴号	0
R306	起始速度	10000

示例梯形图：



效果图解：



如果轴起始速度大于设定脉冲频率，则直接以轴起始速度运行至脉冲发送完毕，不会出现负加速情况。

说明案例 3：

使用 DPLSR 指令设定从10KHZ加速至80KHZ，使虚拟轴0发出50000个脉冲，加减速过程为200ms。并采用急停信号端口，绑定X3，急停方式为缓停。

事件指定方法如下表：

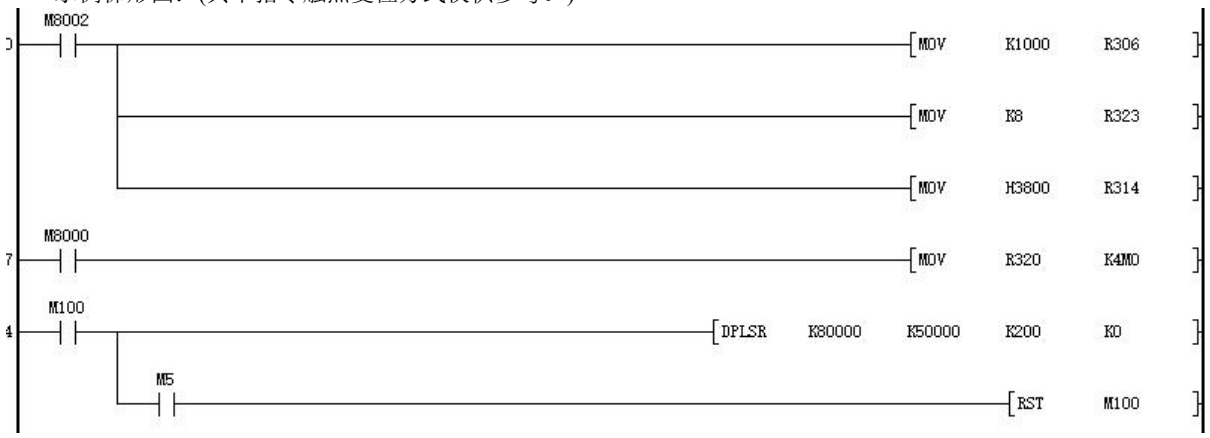
信号端口 =	选定端口	信号端口 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

需要配置的参数	定义	配置值

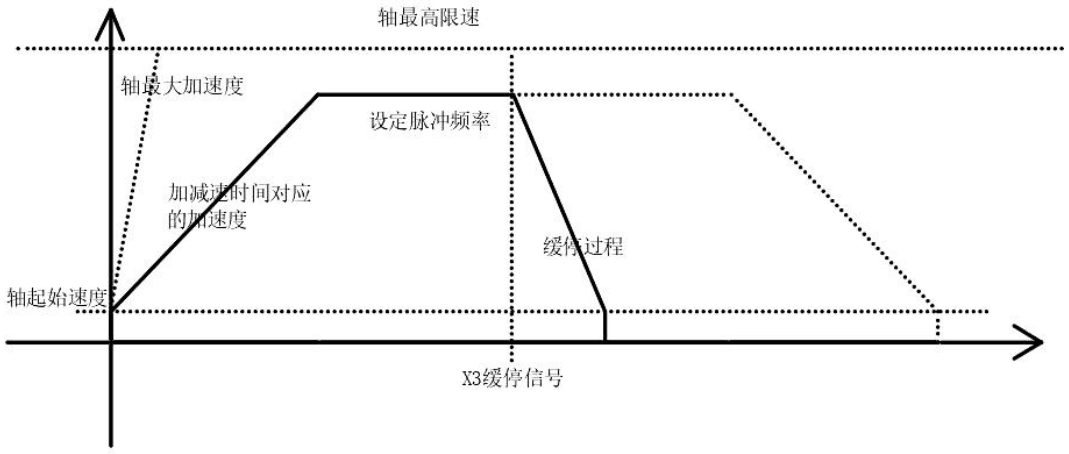
D1	脉冲频率	80000
D2	脉冲数	50000
D3	加减速时间	200
D4	软件虚拟输出轴号	0
R306	起始速度	1000
R323	轴急停事件	8
R314Bit10	急停信号极性	0
R314Bit11	急停信号使能	1
R314Bit12	急停类型选择	1
R314Bit13	输入信号源选择	1

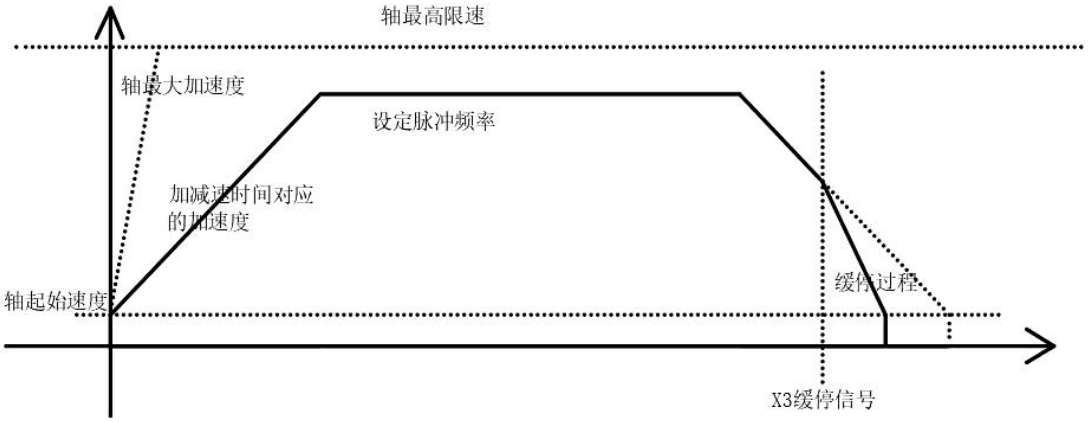
R314 为 0011 10000000 0000化为16进制，H3800

示例梯形图：(其中指令触点复位方式仅供参考。)



效果图解：





①升速段遇到缓停信号时，直接减速，按照轴最大加速度减速至起始速度然后停止。

②在匀速段遇到缓停信号时，则立即减速，按照轴最大加速度减速至起始速度然后停止。

③减速段遇到缓停信号时，按照轴最大加速度减速至起始速度然后停止。如果设置为急停信号则全部立即停止。

- **注意要点**

1) 速度参数配置时，最好遵守轴起始速度＜设定脉冲频率＜轴最高限速；
2) 如果设定脉冲频率小于轴起始速度，则指令以轴起始速度匀速发完设定脉冲数，无加减速段；
3) 设定频率大于轴最高限速时以轴最高限速运行；
4) 当设定的加减速时间对应的加速度大于轴最大加速度时，以轴最大加速度运行；
5) 只能在轴状态为 0 时开始运行；
6) 增加缓停或急停信号时，轴状态在指令停止发脉冲后不会变为完成状态（32），急停状态为 160， 缓停状态为 144（缓停结束后变为160）。若要运动完成后断开触点，则要判断轴状态为完成状态（32）或急停/缓停状态（160），具体梯形图见说明案例 3。否则该脉冲轴将始终处于被占用状态；
7) 指定脉冲数不得大于 16000000，否则影响定位精度；

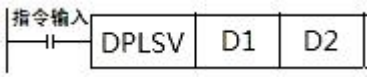
3.4.4 变速输出[DPLSV]

指定软件虚拟输出轴(AXV)按指定频率输出无限脉冲数(带方向)。

- **指令格式**

1、指令格式





2.设定数据

操作数种类	内容	数据类型
D1	脉冲频率	BIN 32 位
D2	软件虚拟输出轴号	BIN 32 位

3、对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•									

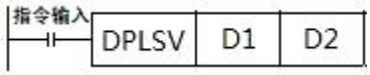
- **相关软元件**

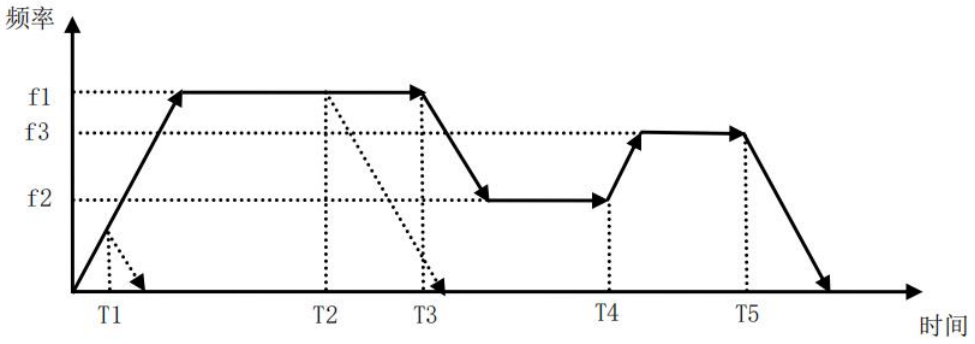
该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴				
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R302	R352	R402	R452	时间当量	16 位	1000	默认
R305	R355	R405	R455	最大速度	16 位	20000	读写
R307	R357	R407	R457	最大加减速度	16 位	1000	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读
R322	R372	R422	R472	1ms 插补增量	16 位	0	只读

- **功能和举例**

指定软件虚拟输出轴(AXV)按指定频率输出无限脉冲数(带方向)。





- 启动指令，以轴参数轴最大加减速加速到 f1 进入匀速段。f1 为指令操作数D1 指定。
  - 若使能急停，且T1 处产生了该事件，则根据急停模式减速/立即停止。
  - 若使能正转/反转限位，且T1 处产生了该事件，则减速停止。
- f1 匀速段时的事件处理
  - 若使能急停，且T2 处产生了该事件，则根据急停模式减速/立即停止。
  - 若使能正转/反转限位，且 T2 处产生了该事件，则减速停止。
  - T3 处用户在线修改操作数D1 指定 f2，以轴参数轴最大加减速减速到 f2 再次进入匀速段
- f2 匀速段时的事件处理
  - T4 处用户在线修改操作数D1 指定 f3，以轴参数轴最大加减速加速到 f3 再次进入匀速段
- f3 匀速段时的事件处理
  - 若使能急停，且 T5 处产生了该事件，则根据急停模式减速/立即停止。
  - 若使能正转/反转限位，且 T5 处产生了该事件，则减速停止。

1) 在D1 中指定脉冲频率

	设定范围
32 位运算	-200000~200000（默认1000ms时间当量）

注：该参数在指令运行过程中修改有效；

配置方向端口后，负值为反转，端口 0FF，正值为正转，端口 ON

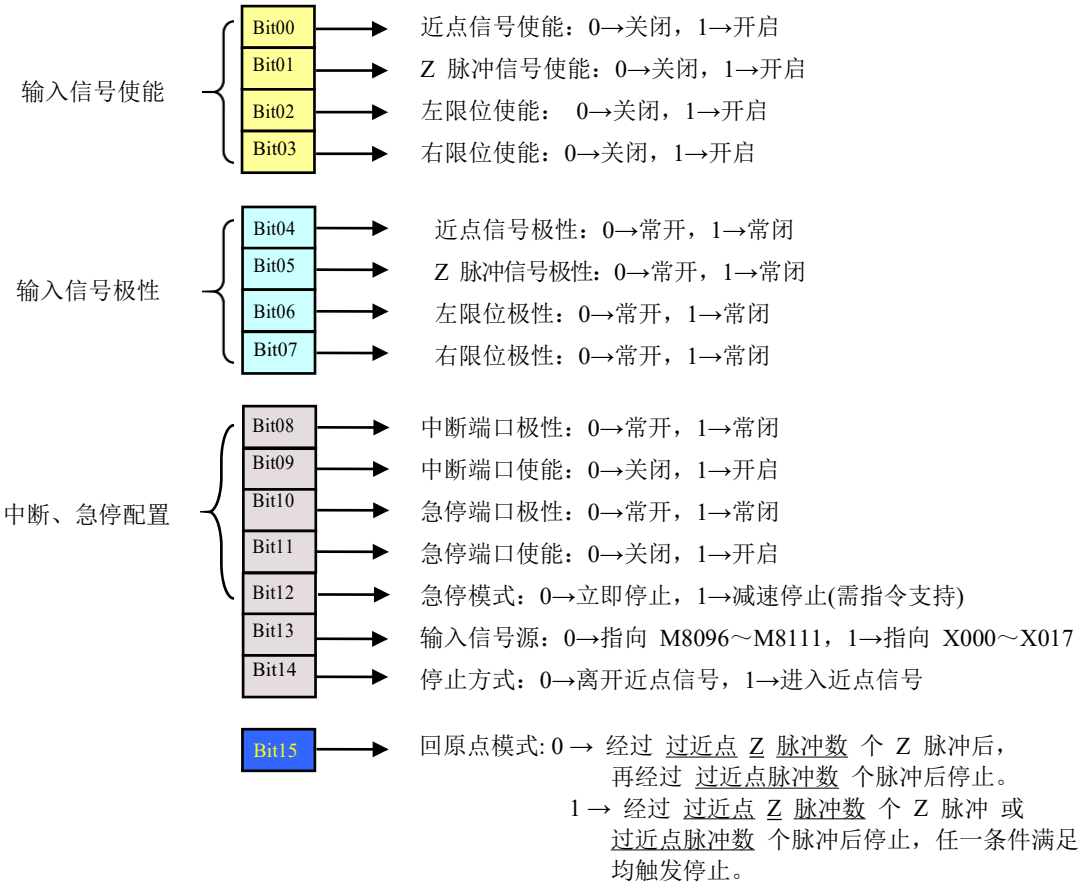
2) 在 D2 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

3) 输入点掩码对应表格：

信号端口 =	选定端口	信号端口 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

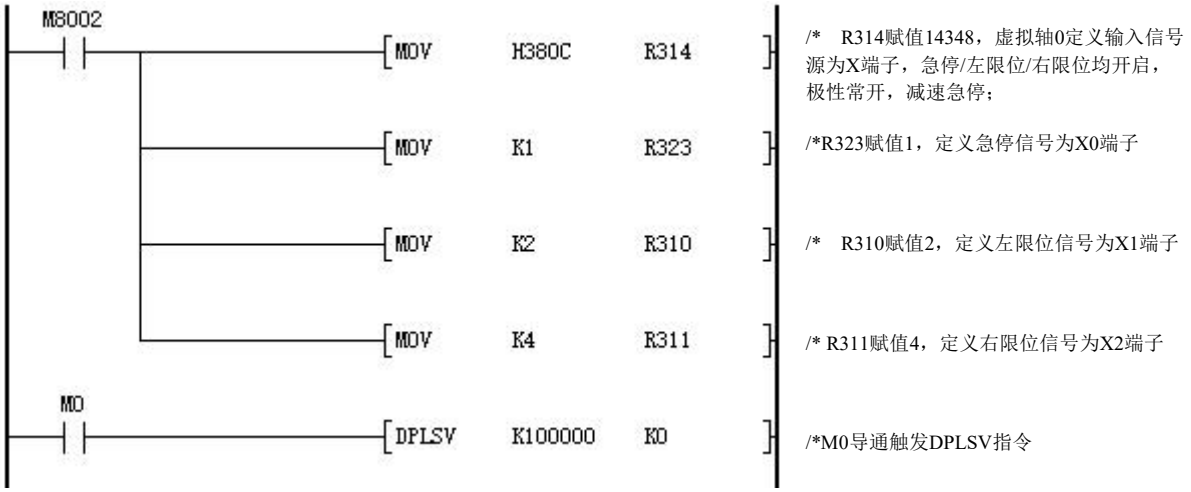
4) 轴事件输入点定义



指令中左限位、右限位、急停信号的定义表格如下：

R314	Bit02	Bit03	Bit06	Bit07	Bit10	Bit11	Bit12	Bit13
急停					0：常开 1：常闭	0：关闭 1：开启	0：立即停止 1：减速停止	
左限位	0：关闭 1：开启		0：常开 1：常闭					
右限位		0：关闭 1：开启		0：常开 1：常闭				
信号源								0：指M 1：指X

5) 当需要控制伺服电机以100Khz 的频率一直发脉冲，同时能修改频率，最大加减速度10p/ms^2，同时X0 端子信号急停（减速停止），X1 端子信号左限位，X2 端子信号右限位，则梯形图如下：



监控虚拟 0 轴参数：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
0	0	0

置位 M0 为 ON，正转发脉冲，运行时轴参数：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
16	0	100

赋值 D1000=800（频率 80KHz），继续正转发脉冲，此时轴参数：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
16	0	80

赋值 D1000=-500（频率 50KHz），由正转减速到 0 再反转发脉冲，此时轴参数：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
16	0	-50

另：运行过程中导通反转限位 X1 为 ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
32	0	-1

运行过程中导通正转限位 X2 为ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
32	0	1

运行过程中导通急停信号 X0 为ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
160	0	1/-1

- **注意要点**

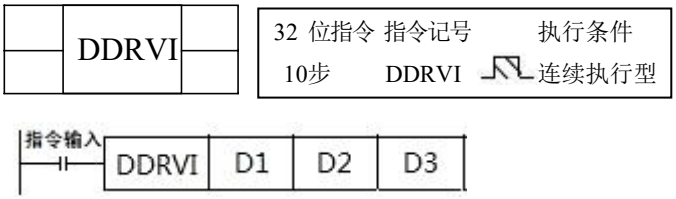
1) 必须使用常开触点导通执行指令，断开触点则轴运动立即停止；
2) 指令的运行速度受指定虚拟轴最大速度限制；
3) 配置急停功能/反转限位/正转限位时需设置“轴事件输入点”参数，“轴急停事件”参数，“反转限位端口”参数，“正转限位端口”参数；
4) 对应虚拟轴的“轴状态”参数必须为 0 才能执行指令；
5) 指令第一个操作数 D1 为 32 位，请注意反向频率是负值。

3.4.5 相对定位[DDRVI]

指定软件虚拟输出轴(AXV)按照指定频率(带加减速)输出一定个数的脉冲数(带方向)。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	脉冲数	BIN 32 位
D2	脉冲频率	BIN 32 位
D3	软件虚拟输出轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•									

- **相关软元件**

1) **D1 指定输出脉冲数**。

D1 的取值范围是-16000000～16000000。

当 D1 填写为负值时，且配置了方向端口，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) D2 指定输出的脉冲频率。

实际输出的脉冲频率通过轴参数轴时间当量计算得出。计算公式 = D2÷(轴时间当量÷1000)。D2 的设置如果小于轴参数轴起始速度 。最终速度为轴参数轴起始速度。

且当 D2 的设置大于轴最高限速 时，实际的目标速度不会超过轴最高限速 。无论是正向还是负向发脉冲，该值皆为正数，是速度的绝对值。

3) **D3 指定软件虚拟输出轴号**(AXV0～AXV3)，对应数值 0～3。

相关轴参数列表：

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3 轴			
R302	R352	R402	R452	轴时间当量	16 位	1000
R305	R355	R405	R455	轴最高限速	16 位	20000
R306	R356	R406	R456	轴起始速度	16 位	0
R307	R357	R407	R457	轴最大加速度	16 位	1000
R323	R373	R423	R473	轴急停事件	16 位	0
R314 Bit10	R364 Bit10	R414 Bit10	R464 Bit10	急停信号极性 =0 常开 =1 常闭	位	0
R314 Bit11	R364 Bit11	R414 Bit11	R464 Bit11	急停信号使能 =0 不使能 =1 使能	位	0
R314 Bit12	R364 Bit12	R414 Bit12	R464 Bit12	急停类型选择 =0 立即停车 =1 减速停车	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源选择 =0 指向M =1 指向X	位	0
R310	R360	R410	R460	左限位信号输入端口指定	16 位	0
R314 Bit2	R364 Bit2	R414 Bit2	R464 Bit2	左限位信号使能	位	0
R314 Bit6	R364 Bit6	R414 Bit6	R464 Bit6	左限位信号极性	位	0
R311	R361	R411	R461	右限位信号输入端口指定	16 位	0
R314 Bit3	R364 Bit3	R414 Bit3	R464 Bit3	右限位信号使能	位	0
R314 Bit7	R364 Bit7	R414 Bit7	R464 Bit7	右限位信号极性	位	0

起始速度由轴参数 轴起始速度 指定，加速度采用 轴最大加减速 所设定的数值。

支持急停事件，急停事件产生时根据轴参数 轴事件输入点.Bit12 决定是立即停止还是减速停止。支持左右限位事件，左右限位事件产生时运动轴减速停止。

当指令停止时，其停止速度为轴起始速度。

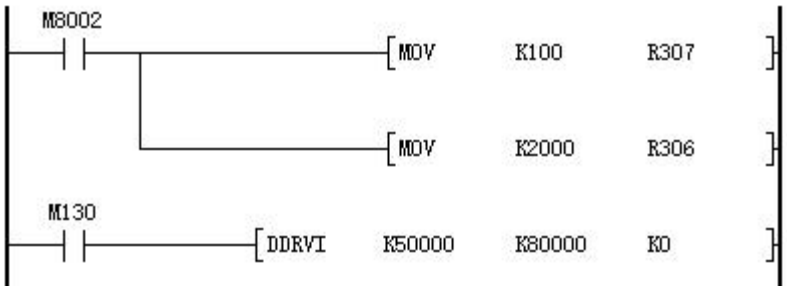
- **功能和举例**

说明案例 1

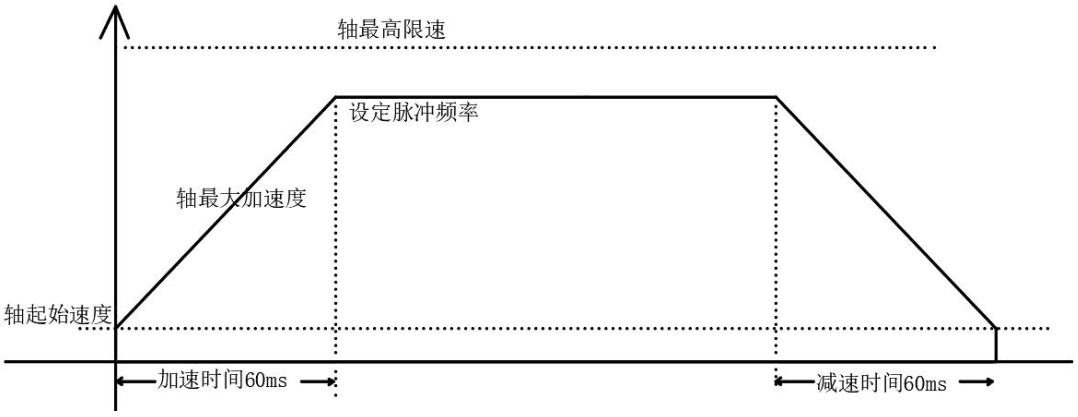
使用 DDRVI 指令设定从 20KHZ 加速至 80KHZ，使虚拟轴 0 发出 50000 个脉冲。默认轴最大加速度， 轴最高限速。

需要配置的参数	定义	配置值
D1	脉冲数	50000
D2	脉冲频率	80000
D3	软件虚拟输出轴号	0
R306	起始速度	2000
R307	轴最大加速度	100（默认1000）

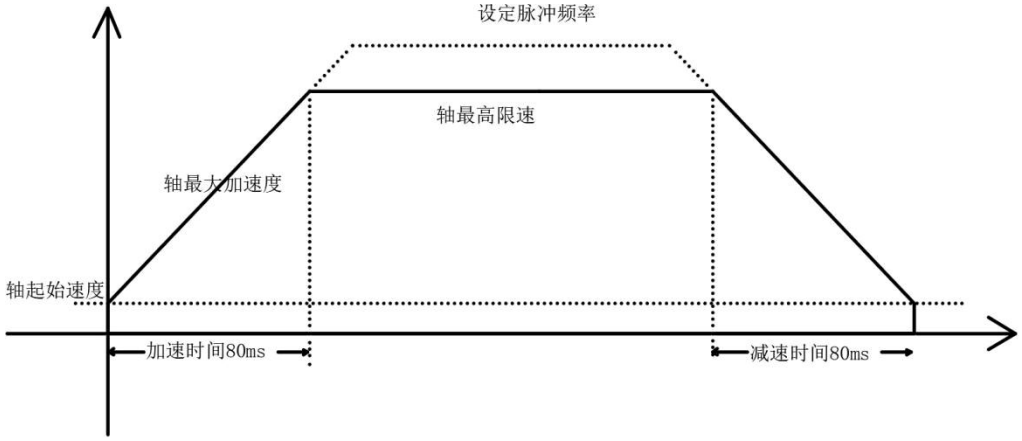
示例梯形图：



效果图解：



从起始速度开始，以轴最大加速度加速至设定脉冲频率，实际发送脉冲数即设定脉冲数。



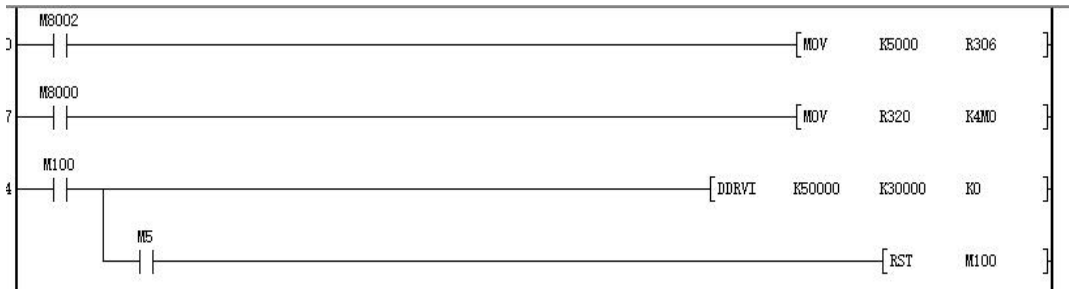
如果设定脉冲频率大于轴最高限速，则加速至轴最高限速。

说明案例 2：

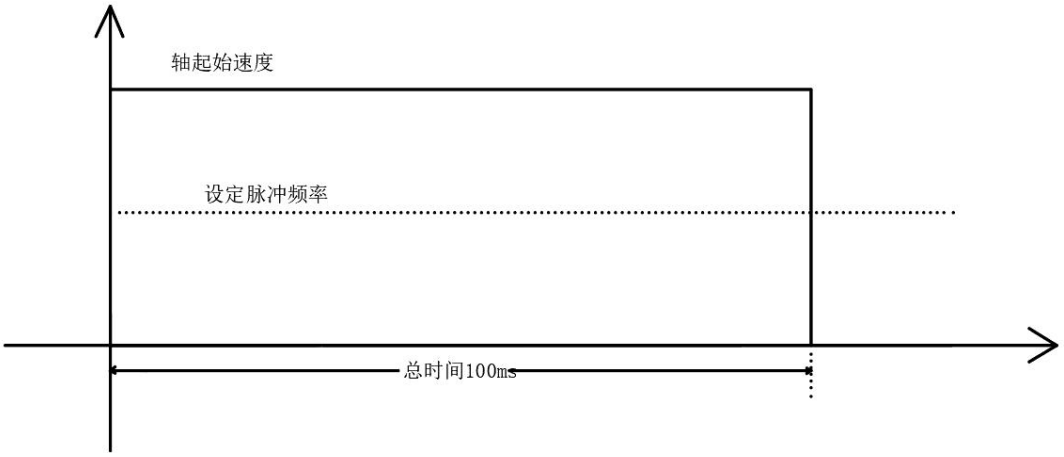
使用 DDRVI 指令设定从 50KHz 负加速（减速）至 30KHz，使虚拟轴 0 发出 50000 个脉冲。默认轴最大加速度，轴最高限速。

需要配置的参数	定义	配置值
D1	脉冲数	50000
D2	脉冲频率	30000
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	5000
R307	轴最大加速度	1000（默认）

示例梯形图：



效果图解：



如果轴起始速度大于设定脉冲频率，则直接以轴起始速度运行至脉冲发送完毕，不会出现负加速（减速）情况。

说明案例 3：

使用 DDRVI 指令设定从 20KHZ 加速至 60KHZ，使虚拟轴 0 发出50000 个脉冲。轴最大加速度设置1P/ms²，方便观察，默认轴最高限速。加入急停信号 X3，左限位信号X4，右限位信号 X5。急停信号方式选为减速停车。所有信号皆为常开信号。

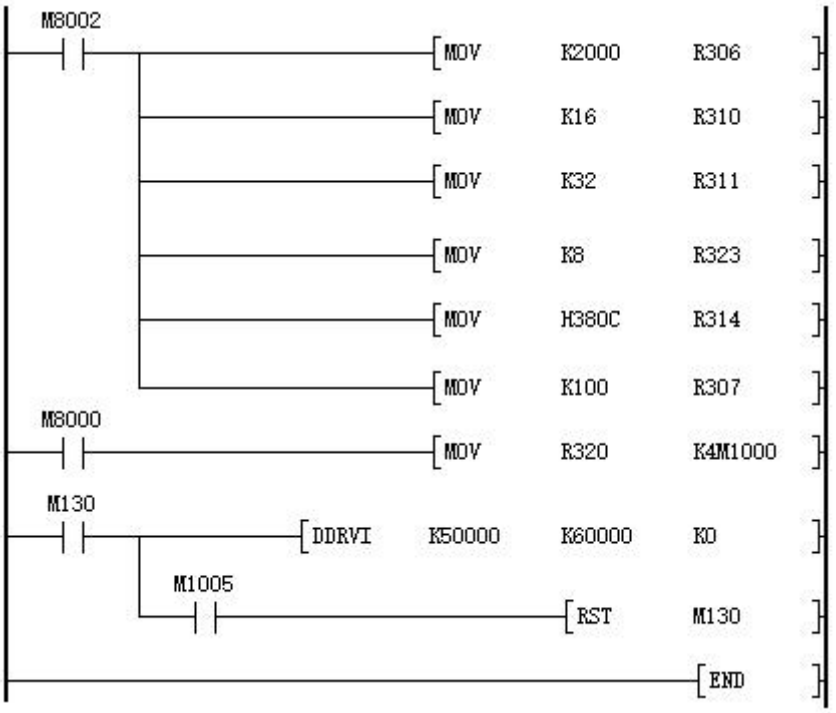
事件指定方法如下表：

信号端口 =	选定端口	信号端口 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

需要配置的参数	定义	配置值
D1	脉冲数	50000
D2	脉冲频率	60000
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	2000
R307	轴最大加速度	1000
R323	轴急停事件	8
R314Bit10	急停信号极性	0
R314Bit11	急停信号使能	1
R314Bit12	急停类型选择	1
R314Bit13	输入信号源选择	1
R310	左限位信号输入端口指定	16
R314Bit2	左限位信号使能	1
R314Bit6	左限位信号极性	0
R311	右限位信号输入端口指定	32
R314Bit3	右限位信号使能	1
R314Bit7	右限位信号极性	0

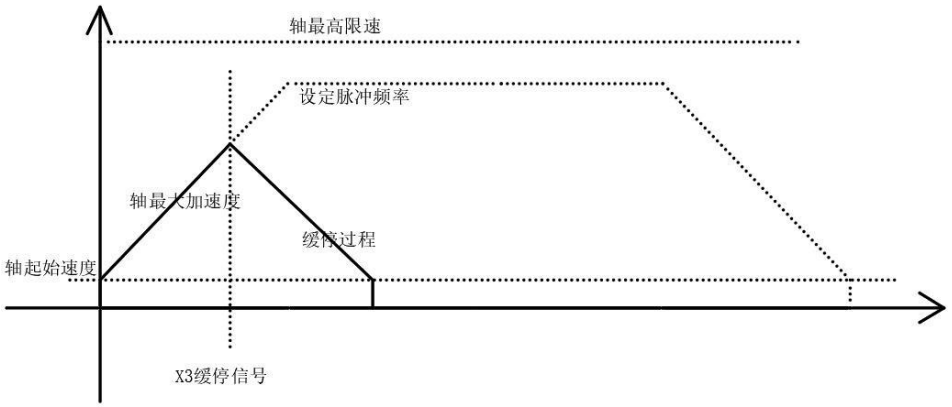
R314 为 0011 1000 0000 1100 化为16 进制，H380C

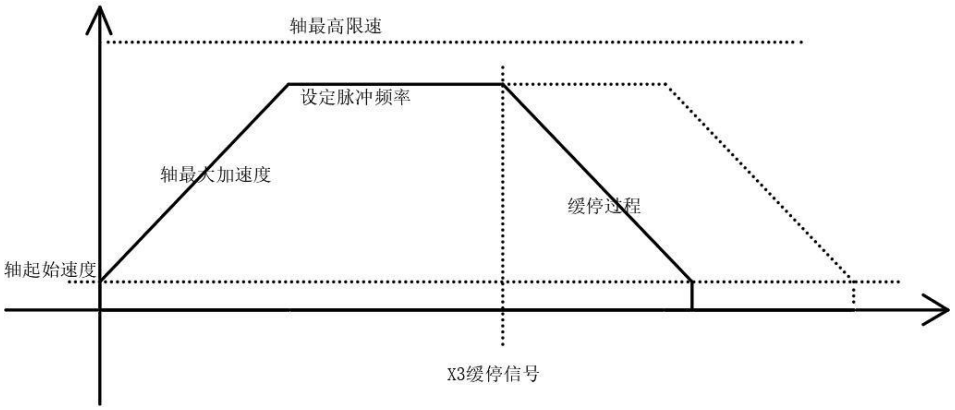
示例梯形图：(其中指令触点复位方式仅供参考。)

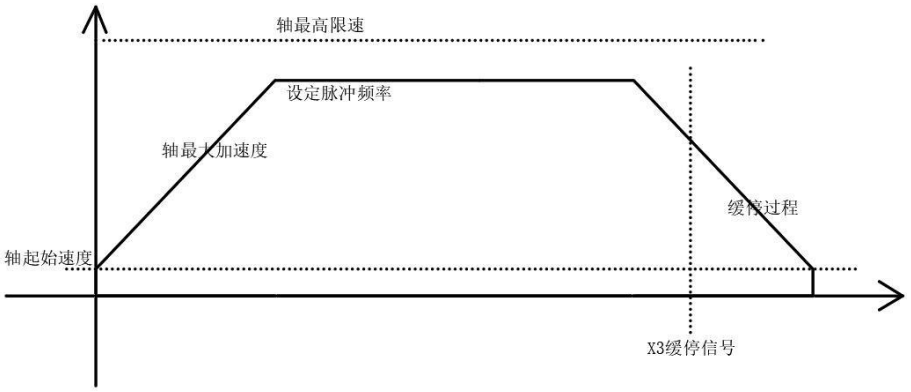


效果图解：

分别在升速，匀速，减速时给出急停信号：





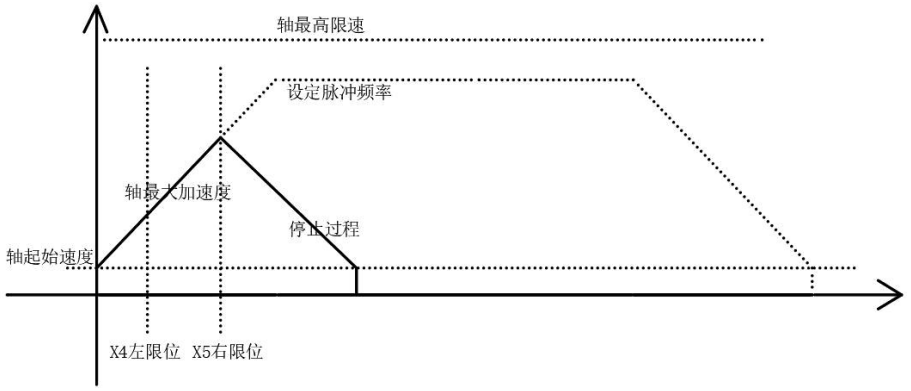


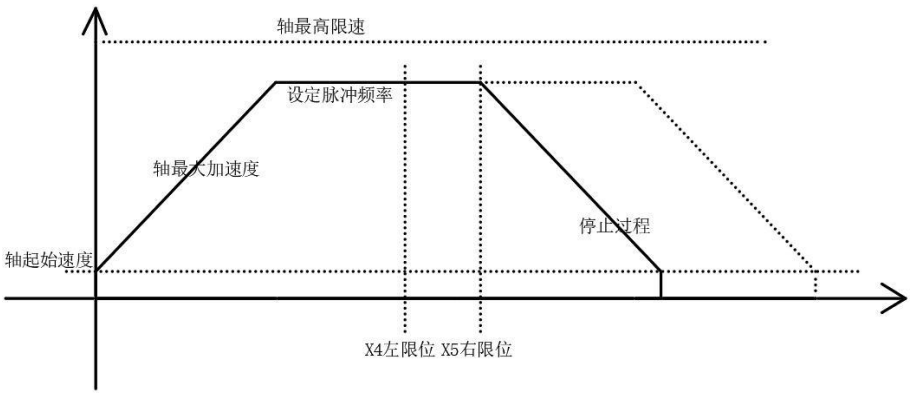
①在升速段遇到缓停信号时，直接减速，按照轴最大加速度减速至起始速度然后停止。

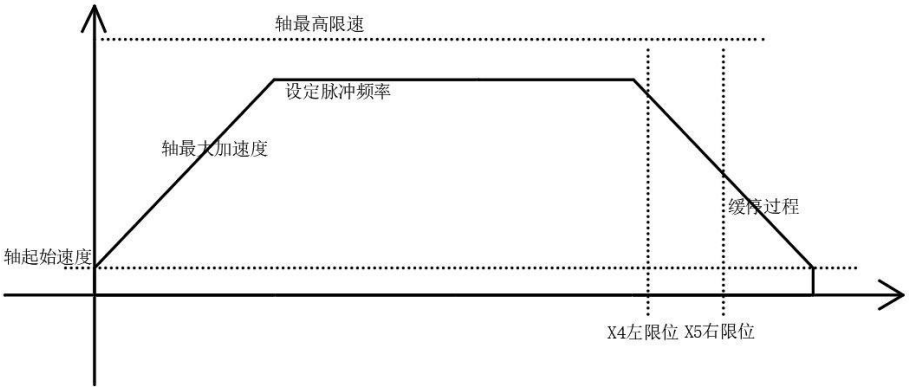
②在匀速段遇到缓停信号时，则立即减速，按照轴最大加速度减速至起始速度然后停止。

③在减速段遇到缓停信号时，按照轴最大加速度减速至起始速度然后停止。如果设置为急停信号则全部立即停止。

分别在升速，匀速，减速时给出左限位和右限位信号：



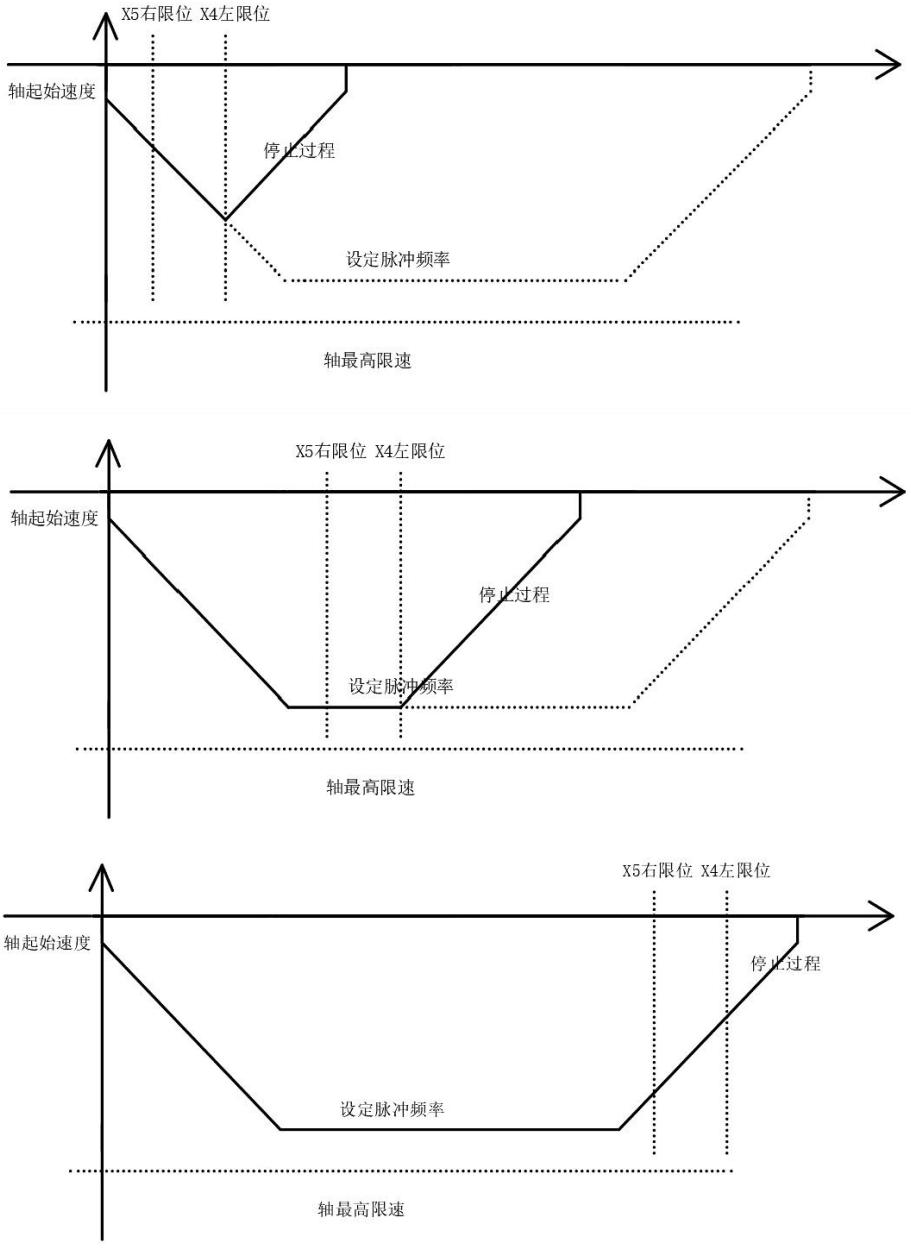




在升速，匀速，减速时给出左限位信号无效，给出右限位信号进入缓停。减速过程均以轴最大加速度缓停。

左限位信号也叫负限位信号，为实际物理轴脉冲负向极限位置上的信号。只对负方向脉冲发送进行限制。同理，右限位只对正方向脉冲发送进行限制。

将脉冲方向设置为负向，即 D1 改为-50000。再次运行：



负方向发脉冲是只有左限位生效，右限位不生效。

当指令未启动时给出左限位信号。指令只能发出正方向脉冲，如果设定负方向脉冲，指令触点闭合后无脉冲输出。给出右限位信号后，指令再闭合只能发负方向脉冲。

- **注意要点**

1) 速度参数配置时，最好遵守轴起始速度＜设定脉冲频率＜轴最高限速；
2) 如果设定脉冲频率小于轴起始速度，则指令以轴起始速度匀速发完设定脉冲数，无加减速段；
3) 设定脉冲频率大于轴最高限速时以轴最高限速运行；
4) 加减速过程以轴最大加速度运行；
5) 只能在轴状态为 0 时开始运行；
6) 如果增加缓停或急停信号时，轴状态在指令停止发脉冲后不会变为完成状态（32），急停状态为160，缓停状态为 144（缓停结束后变为160）。若要运动完成后断开触点，则要判断轴状态为完成状态（32）或急停/缓停状态（160），具体梯形图见说明案例3。否则该脉冲轴将始终处于被占用状态；
7) 指定脉冲数不得大于 16000000，否则影响定位精度。

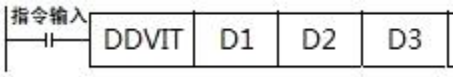
3.4.6 中断定位[DDVIT]

指定软件虚拟输出轴(AXV)按照指定频率(带加减速)输出无限个数的脉冲数(带方向)。遇到中断事件后，输出一定数量的脉冲后减速停止。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	脉冲数	BIN 32 位
D2	脉冲频率	BIN 32 位
D3	软件虚拟输出轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•									

- **相关软元件**

1) **D1 指定输出脉冲数**。

D1 的取值范围是-16000000～16000000。

当 D1 填写为负值时，且配置了方向端口，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) D2 指定输出的脉冲频率。

实际输出的脉冲频率通过轴参数轴时间当量计算得出。计算公式 ＝D2÷(轴时间当量÷1000)。D2 的设置如果小于轴参数轴起始速度，最终速度为轴参数轴起始速度。

且当 D2 的设置大于轴最高限速时，实际的目标速度不会超过轴最高限速 。无论是正向还是负向发脉冲，该值皆为正数，是速度的绝对值。

3) **D3 指定软件虚拟输出轴号**(AXV0～AXV3)，对应数值 0～3。

4) 相关轴参数列表：

软元件编号	名称	数据类型	初始值

虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3 轴			
R302	R352	R402	R452	轴时间当量	16 位	1000
R305	R355	R405	R455	轴最高限速	16 位	10000
R306	R356	R406	R456	轴起始速度	16 位	0
R307	R357	R407	R457	轴最大加速度	16 位	2000
R323	R373	R423	R473	轴急停事件	16 位	0
R314 Bit10	R364 Bit10	R414 Bit10	R464 Bit10	急停信号极性 =0 常开 =1 常闭	位	0
R314 Bit11	R364 Bit11	R414 Bit11	R464 Bit11	急停信号使能 =0 不使能 =1 使能	位	0
R314 Bit12	R364 Bit12	R414 Bit12	R464 Bit12	急停类型选择 =0 立即停车 =1 减速停车	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源选择 =0 指向M =1 指向X	位	0
R310	R360	R410	R460	左限位信号输入端口指定	16 位	0
R314 Bit2	R364 Bit2	R414 Bit2	R464 Bit2	左限位信号使能	位	0
R314 Bit6	R364 Bit6	R414 Bit6	R464 Bit6	左限位信号极性	位	0
R311	R361	R411	R461	右限位信号输入端口指定	16 位	0
R314 Bit3	R364 Bit3	R414 Bit3	R464 Bit3	右限位信号使能	位	0
R314 Bit7	R364 Bit7	R414 Bit7	R464 Bit7	右限位信号极性	位	0
R324	R374	R424	R474	轴中断信号指定	16位	0
R314 Bit8	R364 Bit8	R414 Bit8	R464 Bit8	中断信号极性	位	0
R314 Bit9	R364 Bit9	R414 Bit9	R464 Bit9	中断信号使能	位	0

起始速度由轴参数轴起始速度指定，加速度采用轴最大加减速所设定的数值。

支持急停事件，急停事件产生时根据轴参数轴事件输入点.Bit12 决定是立即停止还是减速停止。支持左右限位事件，左右限位事件产生时运动轴减速停止。

当指令停止时，其停止速度为轴起始速度。

- **功能和举例**

说明案例 1

使用DDVIT 指令设定从 20KHZ 加速至 80KHZ，在接到中断信号后使虚拟轴 0 发出 50000 个脉冲后停止。默认轴最大加速度，轴最高限速。中断信号指定 X6，常开信号。

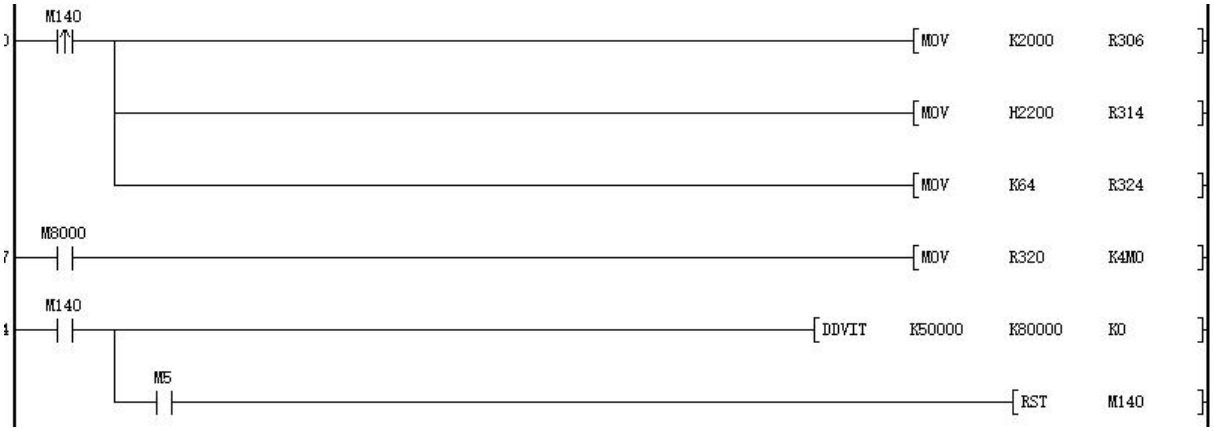
事件指定方法如下表：

信号端口 =	选定端口	信号端口 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

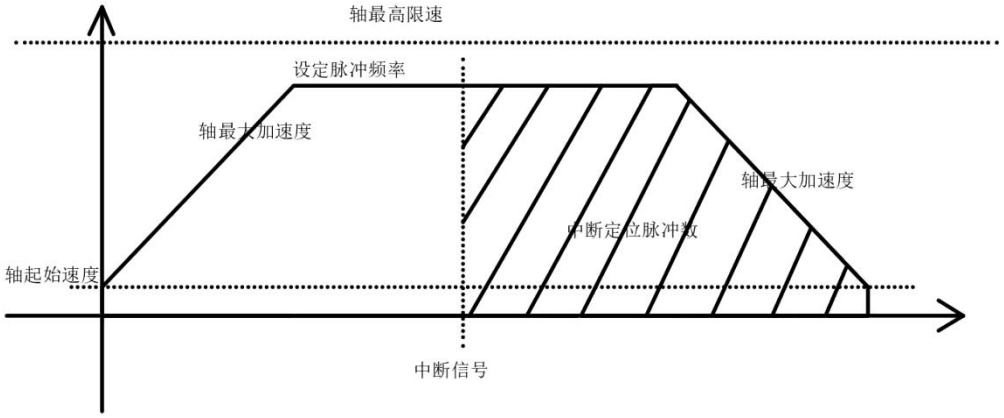
需要配置的参数	定义	配置值
D1	脉冲数	50000
D2	脉冲频率	80000
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	2000
R307	轴最大加速度	1000（默认）
R324	轴中断信号指定	64
R314Bit8	中断信号极性	0
R314Bit9	中断信号使能	1
R314Bit13	输入信号源选择	1

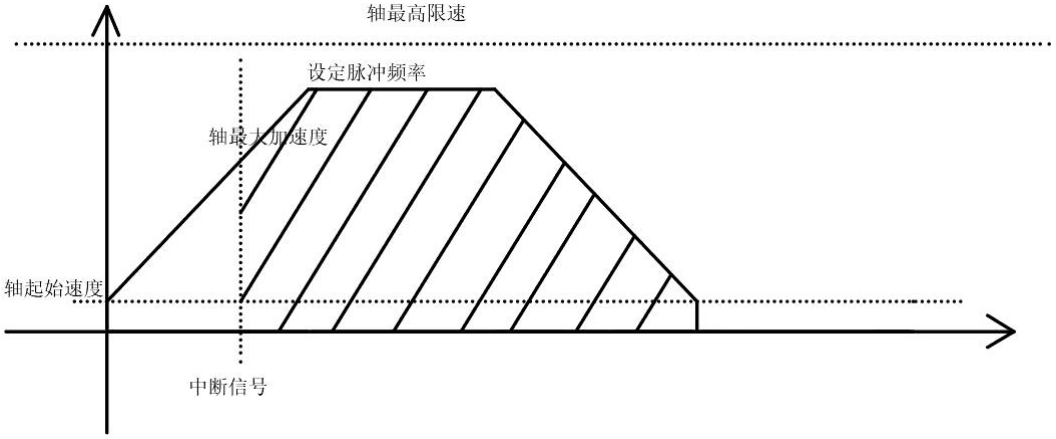
R314 0010 0010 0000 0000，转化为 16 进制，H2200

示例梯形图：



效果图解：





从起始速度开始，以轴最大加速度加速至设定脉冲频率，接收到中断信号后发送固定脉冲数并减速至起始速度停止。

如果在加速过程收到中断信号，当中断脉冲数足够时，依旧升到设定频率然后减速至起始速度停止。

说明案例 2

使用 DDVIT 指令设定从 20KHZ 加速至 80KHZ，在接到中断信号后使虚拟轴 0 发出 1600 个脉冲后停止。轴最大加速度设置为1P/ms²，轴最高限速。中断信号指定 X6，常开信号。

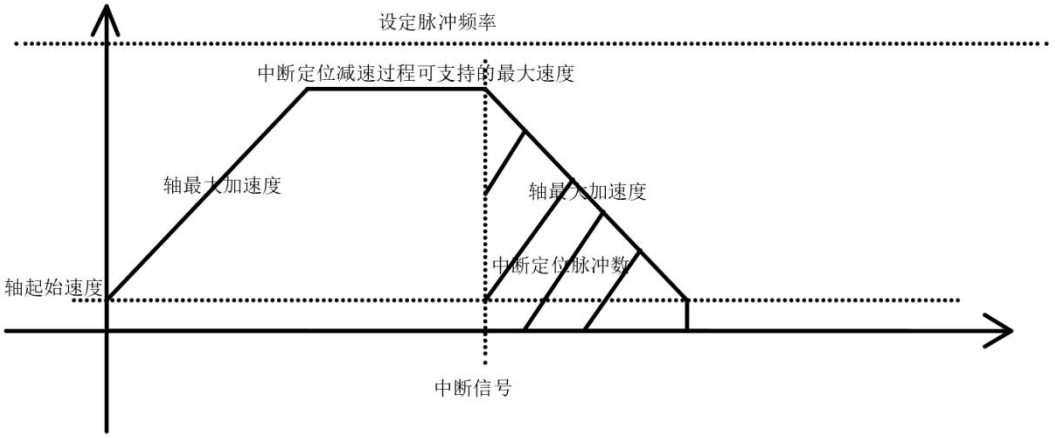
需要配置的参数	定义	配置值
D1	脉冲数	1600
D2	脉冲频率	80000
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	2000
R307	轴最大加速度	100
R324	轴中断信号指定	64
R314Bit8	中断信号极性	0
R314Bit9	中断信号使能	1
R314Bit13	输入信号源选择	1

R314 0010 0010 0000 0000，转化为 16 进制，H2200

示例梯形图：



效果图解：



如果中断定位脉冲数设置过小，匀速段脉冲无法达到设定的脉冲频率。

以本组数据为例：中断定位脉冲数为 1600，起始速度 20KHZ 轴加速度 1P/ms²，易得可支持的最大速度为 60KHZ。（60p/ms+20p/ms）×40ms÷2=1600p

所以设定脉冲频率 80KHZ，实际最大速度只会达到 60KHZ（实际可能存在±1 的小数取整偏差）

如果以 80KHZ 运行，收到中断信号后以 1P/ms²的加速度立即减速至 20KHZ 停止，可算得至少需要 3000p，即设定的中断脉冲数大于 3000 时，最大速度才会达到设定的 80KHZ。

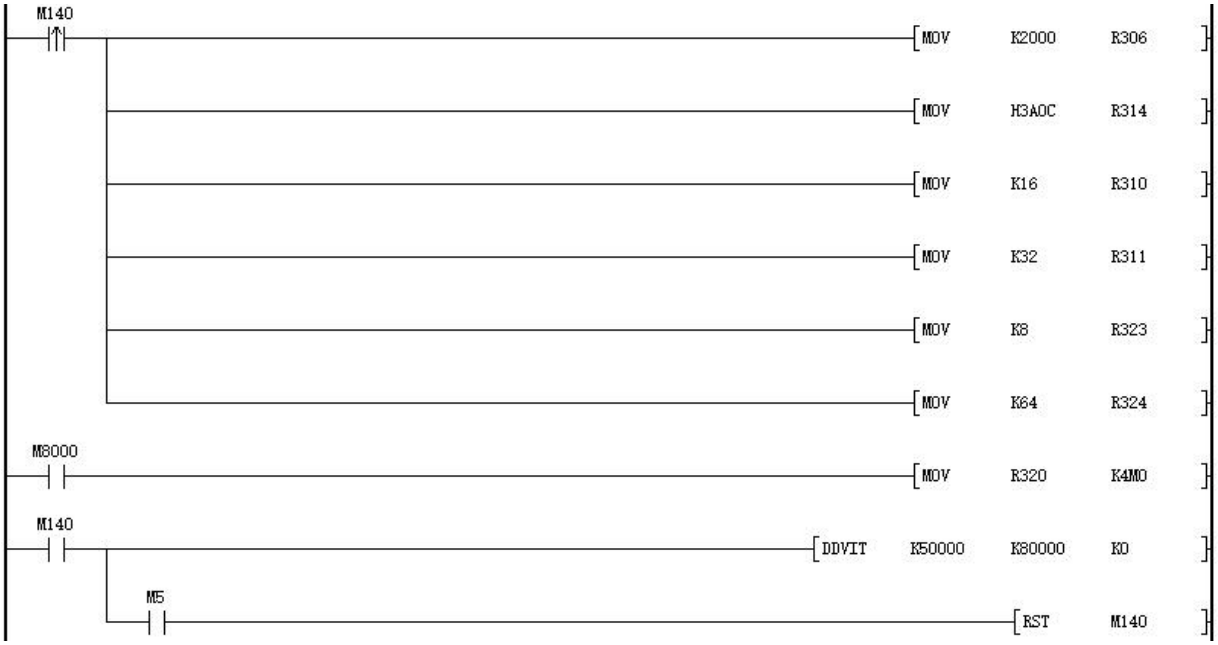
说明案例 3

使 用 DDVIT 指令设定从 20KHZ 加速至 80KHZ，在接到中断信号后使虚拟轴 0 发出 50000 个脉冲后停止。默认轴最大加速度，轴最高限速。中断信号指定 X6，急停信号 X3，左限位信号 X4，右限位信号X5， 均为常开信号。

需要配置的参数	定义	配置值
D1	脉冲数	50000
D2	脉冲频率	800000
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	2000
R307	轴最大加速度	1000（默认）
R324	轴中断信号指定	64
R314Bit8	中断信号极性	0
R314Bit9	中断信号使能	1
R314Bit13	输入信号源选择	1
R323	轴急停事件	8
R314Bit10	急停信号极性	0
R314Bit11	急停信号使能	1
R314Bit12	急停类型选择	1
R314Bit13	输入信号源选择	1
R310	左限位信号输入端口指定	16
R314Bit2	左限位信号使能	1
R314Bit6	左限位信号极性	0
R311	右限位信号输入端口指定	32
R314Bit3	右限位信号使能	1
R314Bit7	右限位信号极性	0

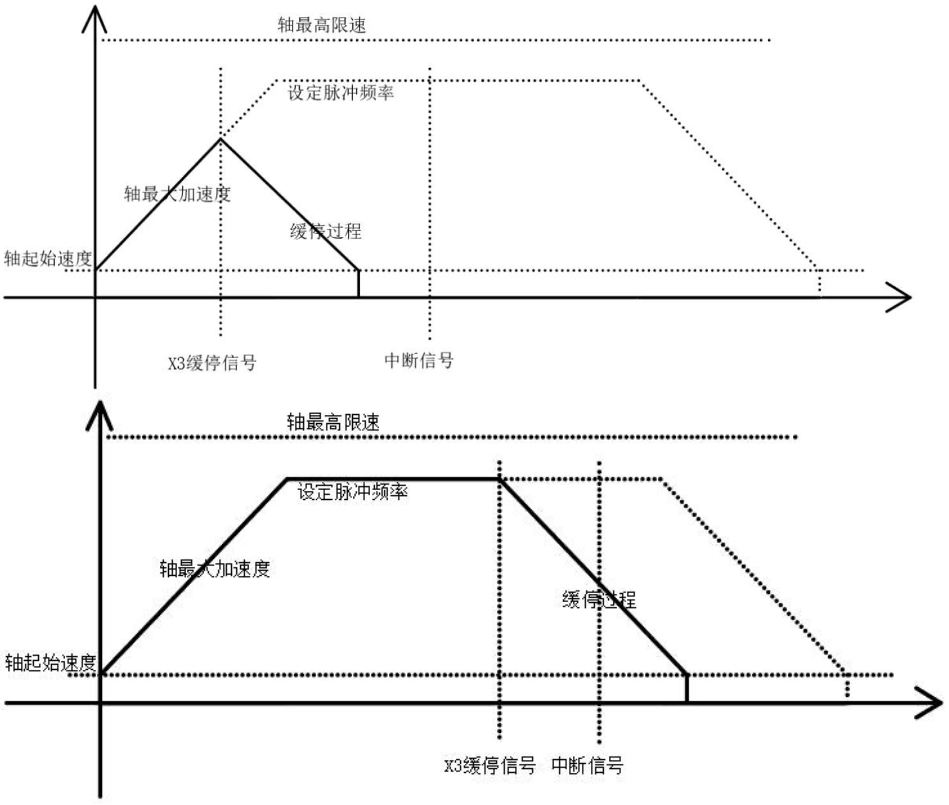
R314 为 0011 1010 0000 1100 化为16 进制，H3A0C

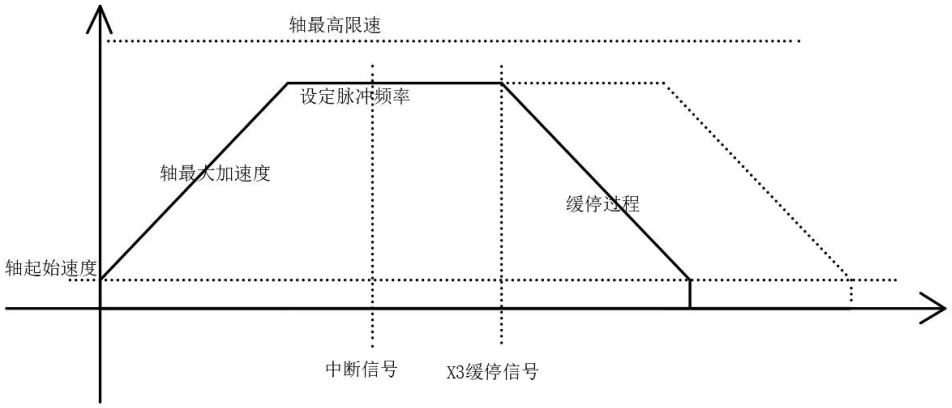
示例梯形图：（其中指令触点复位方式仅供参考。）

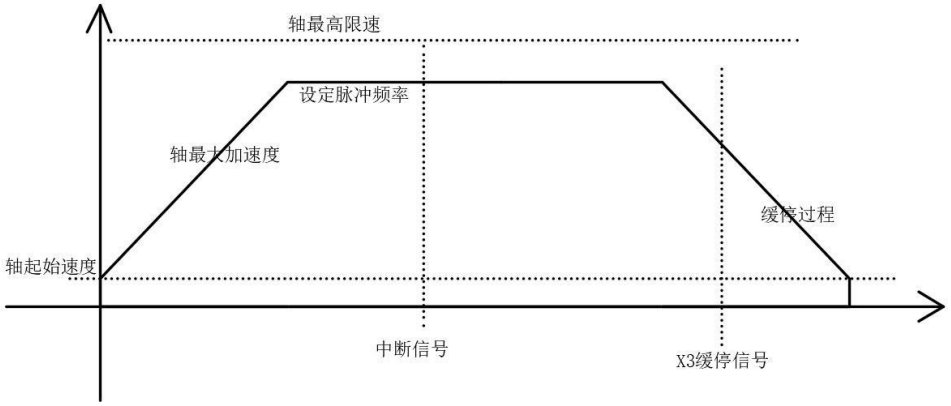


效果图解：

分别在升速，匀速中断信号前，匀速中断信号后，减速时给出急停信号：







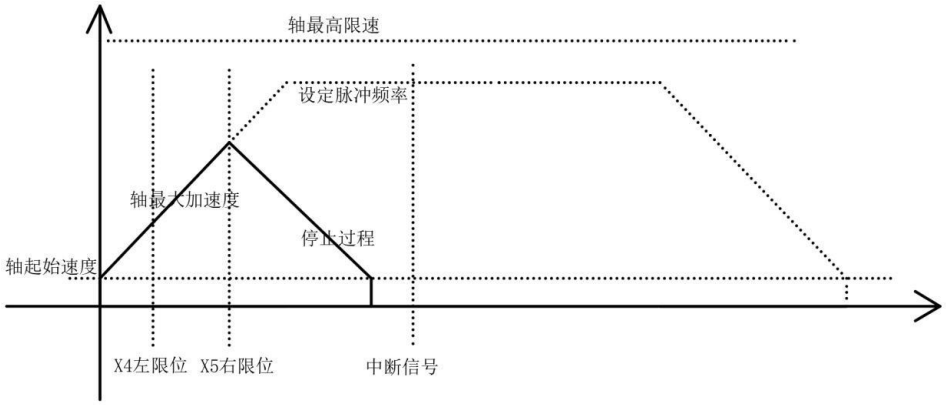
①在升速段遇到缓停信号时，直接减速，按照轴最大加速度减速至起始速度然后停止。

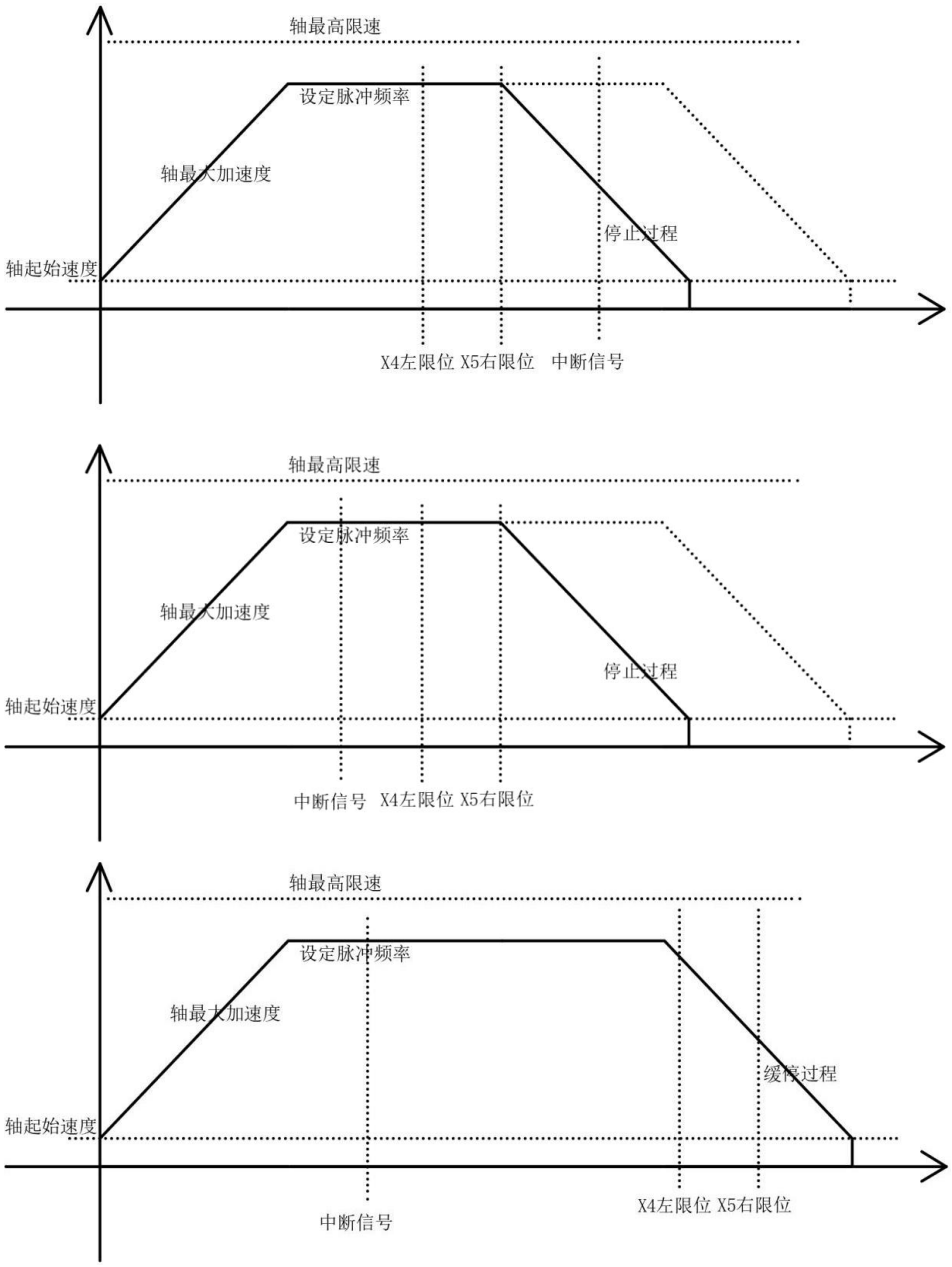
②在匀速段中断信号前，遇到缓停信号时，则立即减速，按照轴最大加速度减速至起始速度然后停止。

③在匀速段中断信号后，遇到缓停信号时，则立即减速，按照轴最大加速度减速至起始速度然后停止

④在减速段遇到缓停信号时，按照轴最大加速度减速至起始速度然后停止，此时运行状况与正常运行状况相同。如果设置为急停信号则全部立即停止。

分别在升速，匀速，减速时给出左限位和右限位信号：

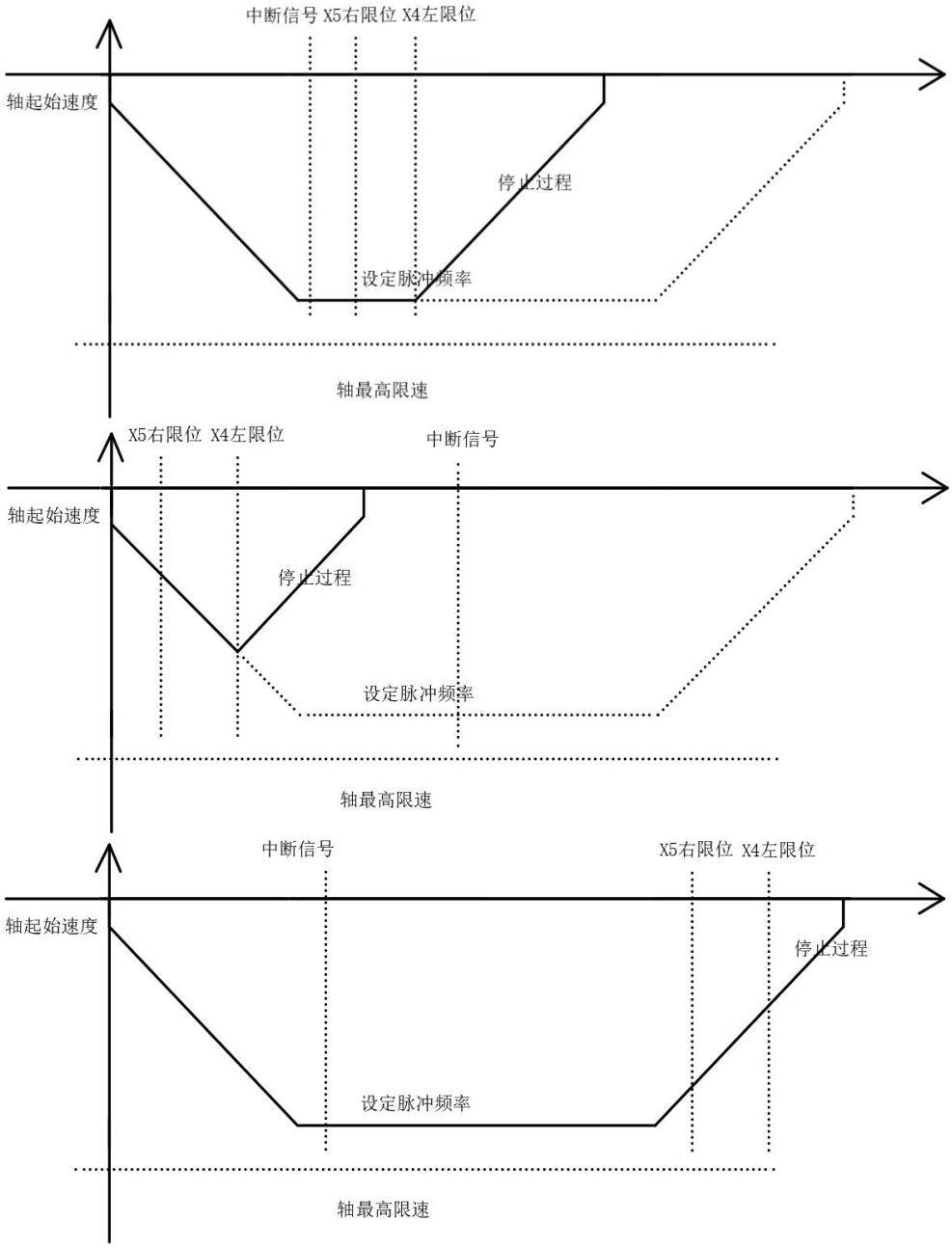




在升速，匀速，减速时给出左限位信号无效，给出右限位信号进入缓停。

左限位信号也叫负限位信号，为实际物理轴脉冲负向极限位置上的信号。只对负方向脉冲发送进行限制。同理，右限位只对正方向脉冲发送进行限制。

将脉冲方向设置为负向，即 D1 改为-50000。再次运行：



负方向发脉冲是只有左限位生效，右限位不生效。

当指令未启动时给出左限位信号。指令只能发出正方向脉冲，如果设定负方向脉冲，指令触点闭合后无脉冲输出。给出右限位信号后，指令再闭合只能发负方向脉冲。

- **注意要点**

1) 速度参数配置时，最好遵守轴起始速度＜设定脉冲频率＜轴最高限速
2) 如果设定脉冲频率小于轴起始速度，则指令以轴起始速度匀速发完设定脉冲数，无加减速段。
3) 设定脉冲频率大于轴最高限速时以轴最高限速运行。
4) 加减速过程以轴最大加速度运行。
5) 只能在轴状态为 0 时开始运行。

6） 如果增加缓停或急停信号时，轴状态在指令停止发脉冲后不会变为完成状态（32），急停状态为 160， 缓停状态为 144（缓停结束后变为160）。若要运动完成后断开触点，则要判断轴状态为完成状态（32）或急停/缓停状态（160），具体梯形图见说明案例 3。否则该脉冲轴将始终处于被占用状态；

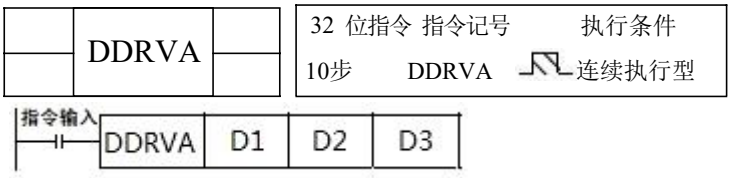
7） 指定脉冲数不得大于 16000000，否则影响定位精度。

3.4.7 绝对定位[DDRVA]

指定软件虚拟输出轴(AXV)按照指定频率和绝对位置输出脉冲数(带方向)。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	脉冲数	BIN 32 位
D2	脉冲频率	BIN 32 位
D3	软件虚拟输出轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•									

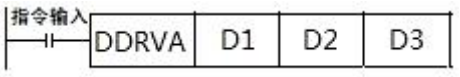
- **相关软元件**

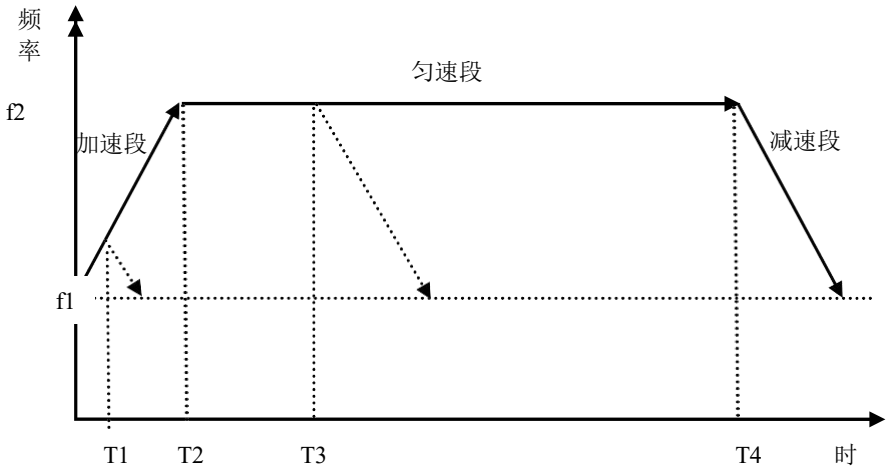
该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴				
R300	R350	R400	R450	软件虚拟轴输出绑	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R302	R352	R402	R452	时间当量	16 位	1000	默认
R305	R355	R405	R455	最大速度	16 位	20000	读写
R306	R356	R406	R456	起始速度	16 位	0	读写
R307	R357	R407	R457	最大加减速度	16 位	1000;	读写
R314	R364	R414	R464	轴事件输入点	16 位	0	读写
R318	R368	R418	R468	相对位置脉冲数	32 位	0	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读
R322	R372	R422	R472	1ms 插补增量	16 位	0	只读
R323	R373	R423	R473	轴急停事件	16 位	0	读写

- **功能和举例**

该指令可实现指定软件虚拟输出轴(AXV)按照指定频率和绝对位置输出脉冲数(带方向)。





● 以初始的 f1 频率发送脉冲进入加速段，f1 为轴参数 轴起始速度 所指定，加速度为轴参数 轴最大加减速。

◆ 若使能急停，且 T1 处产生了该事件，则根据急停模式减速/立即停止。

◆ 若使能左右限位，且 T1 处产生了该事件，则减速停止。

● T2 处进入匀速段，f2 为指令指定的脉冲频率

◆ 当指定的脉冲频率小于轴参数 轴最高限速 或计算得出的匀速段输出频率，则匀速段输出频率为三 者间最小的数值。

◆ 若使能急停，且 T3 处产生了该事件，则根据急停模式减速/立即停止。

◆ 若使能左右限位，且 T3 处产生了该事件，则减速停止。

● T4 处进入减速段，是加速段的逆过程。

◆ 若使能急停，且在减速段处产生了该事件，急停模式为减速停止则仍然以原减速段规划方式减速停止。否则立即停止。

◆ 若使能左右限位，且在减速段处产生了该事件，仍然以原减速段规划方式减速停止。

1. 在 D1 中指定脉冲数

	设定范围
32 位运算	-16000000<=D1-“轴相对位置”<=16000000

注：配置方向端口后，D1-“轴相对位置”为负值反转，端口 0FF，为正值正转，端口 ON。例如：配置虚拟轴 0 时，D1=10000，则指令执行脉冲数=D1-R318。

2. 在D2 中指定脉冲频率

	设定范围
32 位运算	0~200000（默认1000ms时间当量）

注：频率设定值受轴最大速度限制，以及硬件端口频率限制。

3.在D3 中指定软件虚拟输出轴

	设定范围
32位运算	0~3

4.输入点掩码对应表格：

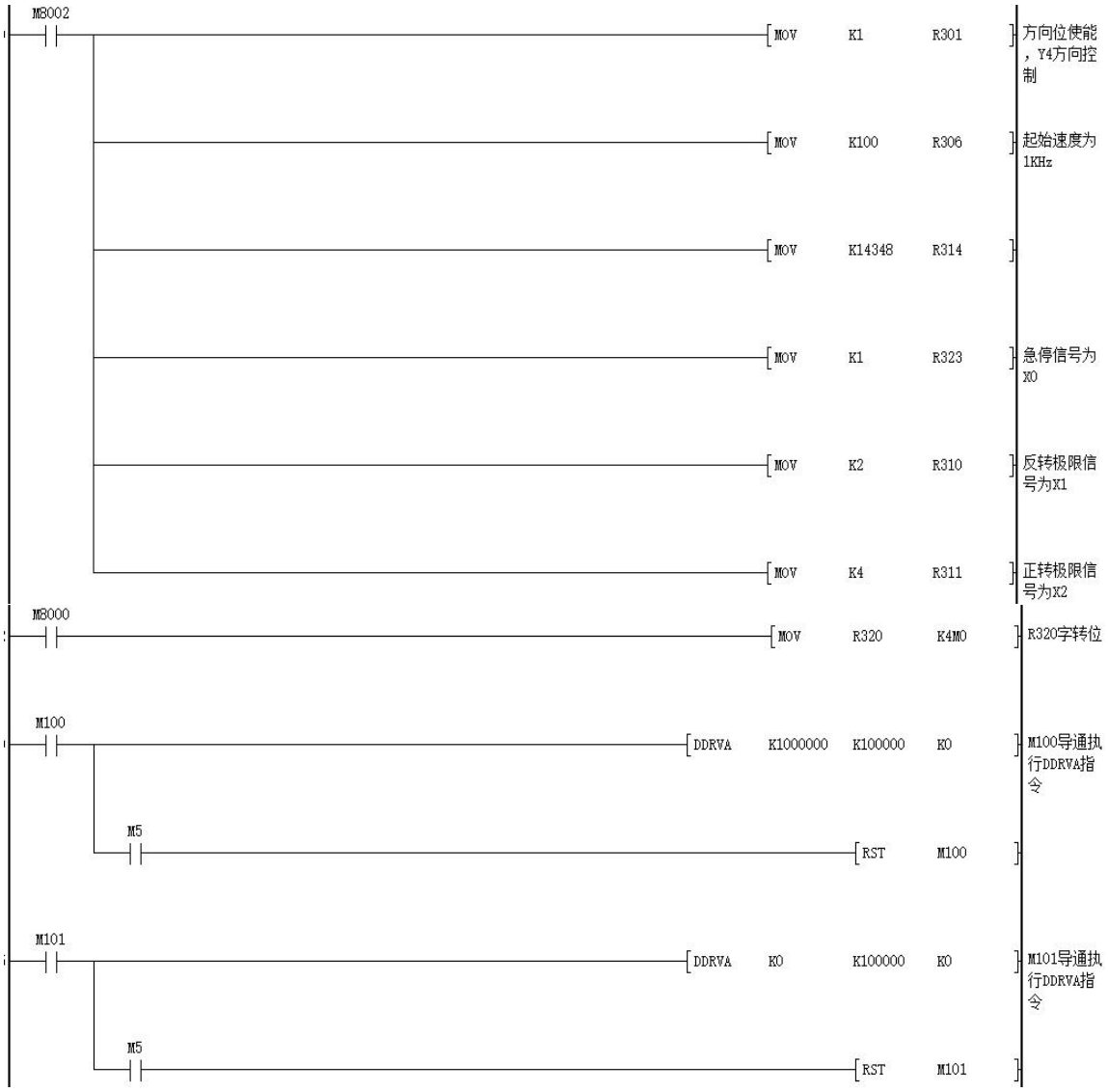
D4/D5	选定端口	D4/D5	选定端口
0	不选定	256	X10/M8104
1	X0/M8096	512	X11/M8105
2	X1/M8097	1024	X12/M8106
4	X2/M8098	2048	X13/M8107
8	X3/M8099	4096	X14/M8108
16	X4/M8100	8192	X15/M8109
32	X5/M8101	16384	X16/M8110
64	X6/M8102	-32768	X17/M8111
128	X7/M8103		

指令中左限位、右限位、急停信号的定义表格如下：

R314	Bit02	Bit03	Bit06	Bit07	Bit10	Bit11	Bit12	Bit13
急停					0：常开 1：常闭	0：关闭 1：开启	0：立即停止 1：减速停止	
左限位	0：关闭 1：开启		0：常开 1：常闭					
右限位		0：关闭 1：开启		0：常开 1：常闭				
信号源								0：指 M 1：指 X

5.当需要控制伺服电机先正转 1000000 个脉冲，后回到原位，运行频率 100KHz，起始速度 1KHz，默认最大加减速度 10p/ms²，同时 X0 端子信号急停（减速停止），X1 端子信号左限位，X2 端子信号右限位，脉冲输出端子为 Y0，方向控制端子为 Y4。

则梯形图如下：



R314赋值14348，虚拟轴0定义输入信号源为X端子，急停/反转极限/正转极限均开启，极性常开，减速急停。

监控虚拟 0 轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
0	0	0	0

置位 M0 为 ON，Y0 端口发脉冲，Y4 端口为 ON，正转发脉冲，运行过程中轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
>0	16	0	100

完成时轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
1000000	32	0	10

再置位 M1 为ON，Y0 端口发脉冲，Y4 端口为 OFF，反转发脉冲，运行过程中轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
<1000000	16	0	-100

完成时轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
0	32	0	-10

另：运行过程中导通反转限位 X1 为 ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
32	0	-10

运行过程中导通正转限位 X2 为ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
32	0	10

运行过程中导通急停信号 X0 为ON：

R320 轴状态	R321 错误代码	R322 1ms 插补增量
160	0	10/-10

- **注意要点**

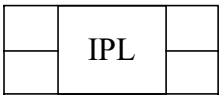
1. 必须使用常开触点导通执行指令，断开触点则轴运动立即停止；
2. 指令的运行速度受指定虚拟轴最大速度限制；
3. 配置急停功能/左限位/右限位时需设置“轴事件输入点”参数，“轴急停事件”参数，“左限位端口”参数，“右限位端口”参数；
4. 指令的第一个操作数 D1 是绝对位置，且需保证-16000000<D1-“轴相对位置”<16000000；
5. 对应虚拟轴的“轴状态”参数必须为 0 才能执行指令。

3.4.8 插补输出[IPL]

指定软件虚拟输出轴(AXV)一个扫描周期内（大于等于 1ms）以1ms 的时间输出一定个数的脉冲数(带方向)。

- **指令格式**

1） 指 令 格 式



16 位指令 指令记号 执行条件
5 步 IPL 连续执行型



2） 设 定 数 据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位
D2	插补脉冲数	BIN 16 位

3） 对 象 软 元 件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•									
D2	•	•	•							

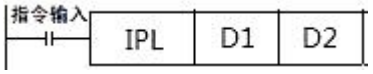
- **相关软元件**

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴				
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R318	R368	R418	R468	相对位置脉冲数	32 位	0	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读
R322	R372	R422	R472	1ms 插补增量	16 位	0	只读

- **功能和举例**

该指令使软件虚拟输出轴(AXV)在一个扫描周期内（大于等于 1ms）以1ms 的时间输出一定个数的脉冲数(带方向)。



1) 在D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

2) 在D2 中指定插补脉冲数

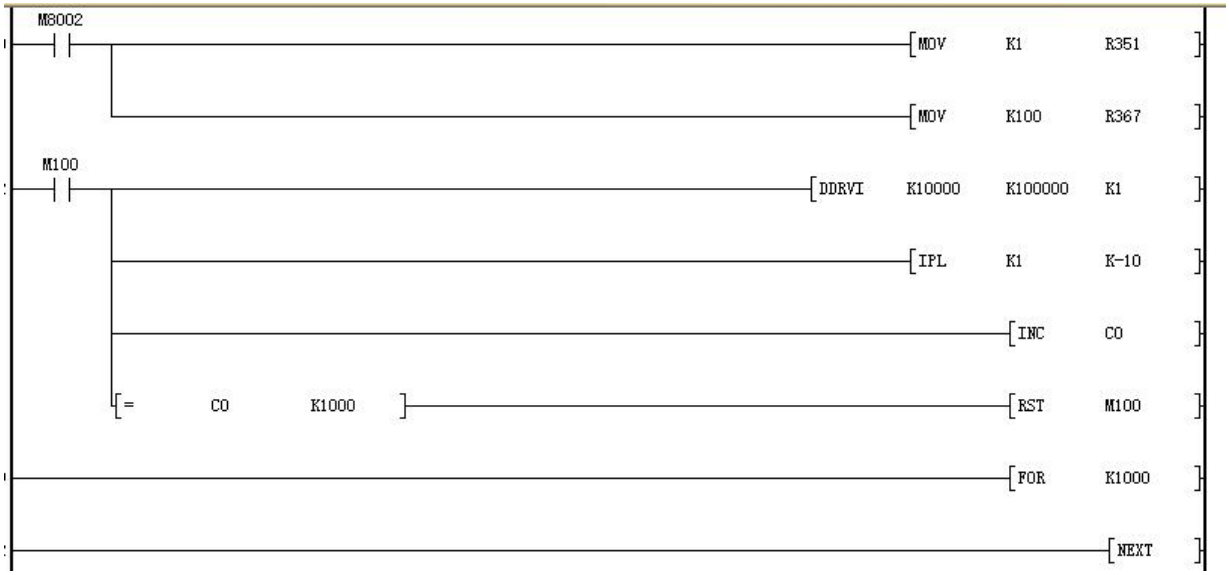
	设定范围
16 位运算	-200~200（默认10ms时间当量）

注意：为 1ms 插补脉冲数，即 1 为 1KHz，因此受端口频率限制；

配置方向端口后，负值为反转，端口 0FF，正值为正转，端口 ON 在指令运行过程中修改有效。

3) 当需要控制伺服电机正转 10000 个脉冲，同时反转插补 10 个脉冲插补 1000 次使伺服电机回到原来位置,正转以 100KHz 运行，起始频率 0KHz，最大加减速度 1p/ms^2，脉冲输出端子为 Y1，方向控制端子为 Y5，

则梯形图如下：



M0导通DDRVI指令，发送正向100KHz的10000个脉冲，M0导通IPL指令每次插补反向十个脉冲，C0计数IPL指令执行次数，C0=1000时复位M0，循环空程序1000次，使扫描周期大于1ms

置位 M0，Y0 端口发脉冲，Y4 端口为 ON，正转发 10000 个脉冲，完成时虚拟 0 轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
10000	32	0	100

由于同时执行 IPL 指令，1 次反转插补 10 个脉冲，C0=1000 时，完成时虚拟 0 轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
0	32	0	-10

可注意到 DPLSY 和 IPL 是同时执行的叠加到 Y01端口发脉冲的过程，并且 DPLSY 指令不受梯形图扫描影响，IPL 受梯形图扫描时间影响。

- **注意要点**

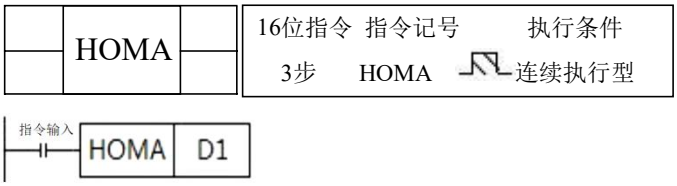
1) 该指令在梯形图扫描执行一次后在 1ms 内对虚拟轴执行一次插补，若1ms内扫描多次，则按 1次插补/ms执行；
2) 对同一软件虚拟输出轴执行多个IPL指令，则每次扫描只执行其中一个IPL指令，但具体每次执行哪个指令不确定，导致出现的输出结果不确定，所以对同一软件虚拟输出端口，同时执行的IPL指令不可超出1个；
3) 插补数量不受轴最高速度限制，但不得超过硬件端口限速；
4) 该指令属于运动叠加类型，当插补正在运行的轴时需注意叠加上的速度不得超过硬件端口限速。

3.4.9 简易回原点[HOMA]

指定对应的软件虚拟输出轴执行简易回原点功能。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟轴轴号	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	Y	M	S	V/Z
D1	•									

- **相关软元件**

该指令相关寄存器或寄存器的位如下表：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R308	R358	R408	R458	近点信号端口指定	16 位	0
R312	R362	R412	R462	原点回归速度	16 位	0
R313	R363	R413	R463	原点爬行速度	16 位	0
R314 bit0	R364 bit0	R414 bit0	R464 Bit0	近点信号使能	位	0
R314 Bit4	R364 Bit4	R414 Bit4	R464 Bit4	近点信号极性	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源类型选择	位	0
R314 Bit15	R364 Bit15	R414 Bit15	R464 Bit15	停止方式选择	位	0
R315	R365	R415	R465	过近点脉冲数	16 位	0

1) 近点信号端口指定

2) 近点信号使能

3) 近点信号极性

近点信号来源大多为机台上的光电开关或磁性开关，接在控制器的 X 点作为回原点指令开始减速和停止的判断条件。根据近点信号接入的 X 点配置 R308指定该 X 点为近点信号端口，配置方法如下：

0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14/M8108
32	X5/M8101	8192	X15/M8109
64	X6/M8102	16384	X16/M8110
128	X7/M8103	-32768	X17/M8111

信号来源有两种：X与特殊M，特殊M可应用于外部信号和内部逻辑运算并用触发近点信号事件。

以0轴为例，配置完近点信号端口后，将R314bit0 设置为 1，使能近点信号。根据接近开关实际极性配置R314bit4，常开信号设为 0，常闭信号设为 1。

4) 输入信号源类型选择

设为 0：端口信号源指向 M8096～M8111

设为 1：端口信号源指向 X0~X17

5) 停止方式选择

设为 0：停止在近点信号内，详细见简易回原点模式一

设为 1：停止在近点信号外，详细见简易回原点模式二

注意：此参数与HOME指令有所不同

6) 过近点脉冲数

设置从离开近点信号到停止的脉冲数。默认值为0。 脉冲方向由“停止方式选择”决定。

该参数为16位无符号数，超过32767将以负数形式设置/显示。设置为0时，即停在近点信号边沿上。

7) 原点回归速度

8) 原点爬行速度

使用时应遵守原点爬行速度＜原点回归速度＜轴最大速度。配置方法与轴最大速度相同。

例如：当默认轴时间当量时，R312 设为 1000 时原点回归速度对应的脉冲频率为 10KHZ。

- **功能说明**

该指令主要用于机械上电后根据外部信号寻找机械零点位置，通常仅在机械开始工作前运行一次。所以指令配置以准确为目的而不过多考虑速度。设置时保证爬行速度较小，确保系统运行平稳。

根据“停止方式选择”设置可将简易回原点分为两种模式，即最终停在近点信号内和停在近点信号外。

根据启动简易原点回归指令时刻的位置，在以上两种模式下各有两种情况，即起点位于近点信号外和起点位于近点信号内。

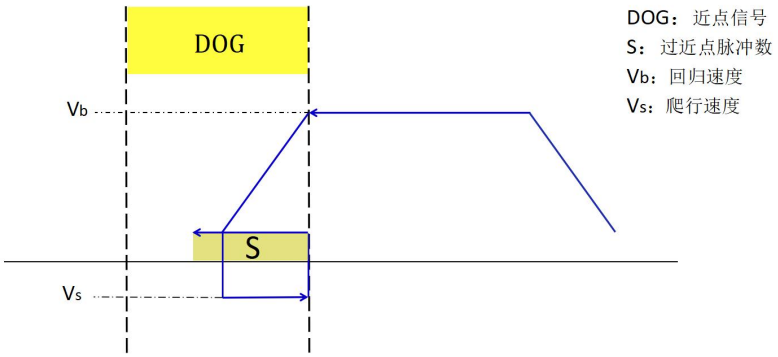
简易回原点模式一：停在近点信号（DOG）内

1. 若**起点位于DOG外**，系统运行情况如下图：

①启动HOMA指令，从Vs开始以轴最大加速度加速到Vb，朝着正向回原点方向（反转方向）运行；

②当DOG上升沿到来后，继续朝着正向回原点方向（反转方向），以轴最大加速度从Vb减速到Vs，然后立马反向，再朝着反向回原点方向（正转方向）以Vs运行；

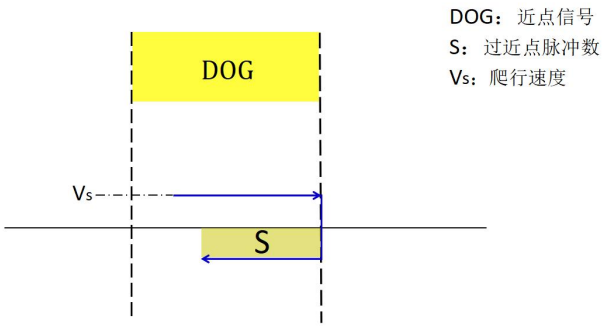
③在DOG下降沿到来后，立马反向，最后朝着正向回原点方向（反转方向）以Vs运行S个脉冲数后停止。



2. 若**起点位于DOG内**，系统运行情况如下图：

①启动HOMA指令，开始朝着反向回原点方向（正转方向）以Vs运行；

②在DOG下降沿到来后，立马反向，朝着正向回原点方向（反转方向）以Vs运行S个脉冲数后停止。



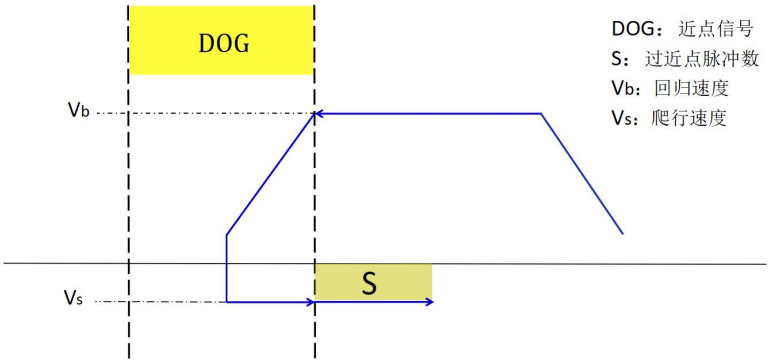
简易回原点模式二：停在近点信号（DOG）外

1. 若**起点位于DOG外**，系统运行情况如下图：

①启动HOMA指令，从Vs开始以轴最大加速度加速到Vb，朝着正向回原点方向（反转方向）运行；

②当DOG上升沿到来后，继续朝着正向回原点方向（反转方向），以轴最大加速度从Vb减速到Vs，然后立马反向，再朝着反向回原点方向（正转方向）以Vs运行；

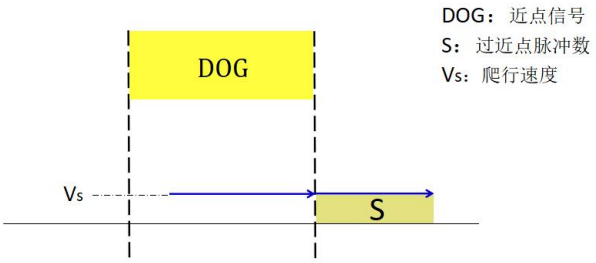
③在DOG下降沿到来后，继续朝着反向回原点方向（正转方向）以Vs运行S个脉冲数后停止。



2. 若**起点位于DOG内**，系统运行情况如下图：

①启动HOMA指令，开始朝着反向回原点方向（正转方向）以Vs运行；

②在DOG下降沿到来后，继续朝着反向回原点方向（正转方向）以Vs运行S个脉冲数后停止。



- **注意要点**

1. 速度设置中必须按照：原点爬行速度＜原点回归速度＜轴最高限速来设置。

2. 由上图可知，系统的起始速度、终止速度、换向速度都为爬行速度，此点与HOME指令有所不同，不会减小到0速，所以爬行速度尽量设置较小值。

3. 在爬行速度和回归速度之间以轴最大加减速变化。轴最大加减速的配置见“输出轴参数”章节。

4. 回原点执行方向固定为向脉冲的负方向执行。如果受到电机安装方向的影响，实际运行并非需要的方向，可以将回原点指令在数字轴上执行，然后通过电子齿轮指令反向输出到脉冲轴上。不可以通过改变近点信号的极性设置来改变运行方向，如果端口信号极性与实际不符会造成指令执行逻辑混乱。

附：左右限位信号，急停信号，中断信号具体定义与用法见说明书其他部分介绍，与HOMA指令的执行无关。

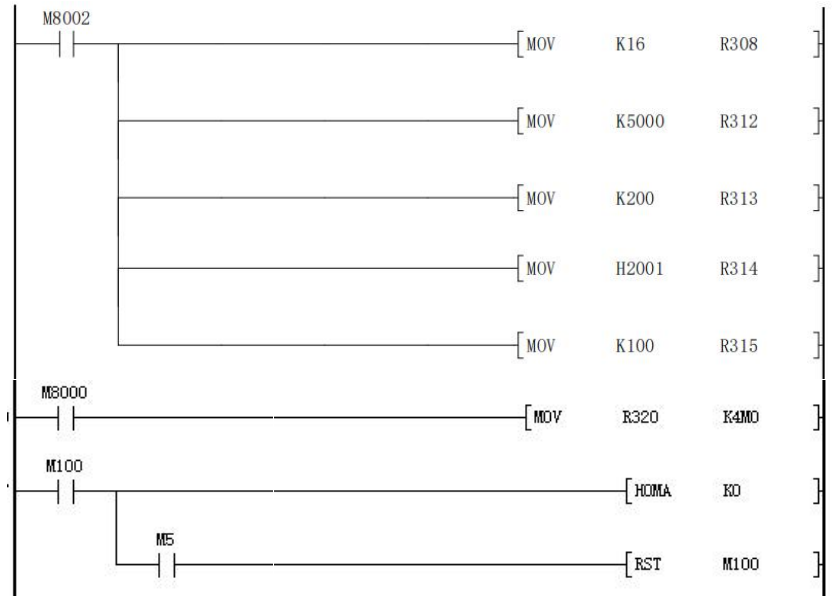
- **案例说明**

使用案例 1：

将 X4 指定为近点信号，开关为常开型。原点回归速度设为 50 KHz， 原点爬行速度设为 2KHz。过近点信号脉冲数设为100 。停在近点信号内。

R314 为 0010 0000 0000 0001，转化为 16 进制数则为 H2001。

梯形图如下：



- 近点信号指定为X4
- 原点回归速度为50kHz
- 原点爬行速度为2kHz
- 轴事件配置
- 过近点脉冲数设置为100
- 轴0执行HOMA指令
- 指令执行完或者急停完成后，复位前置触点

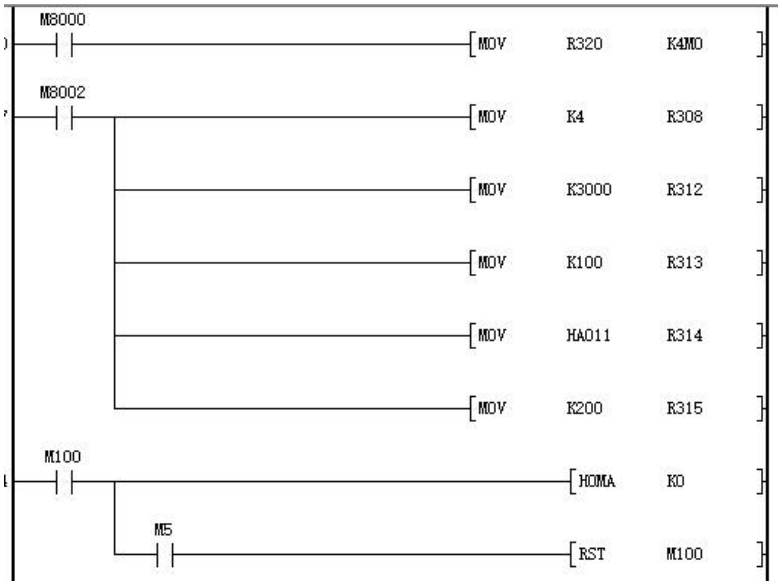
在X4=OFF和ON时，置位M100启动HOMA指令，运行情况即对应模式一的两种情况。

使用案例 2：

将 X2 指定为近点信号，开关为常闭型。原点回归速度设为 30KHz，原点爬行速度设为 1KHz。过近点信号 200 个脉冲停止。停在近点信号外。

R314 为 1010 0000 0001 0001，转化为 16 进制数则为 HA011。

梯形图如下：



- 近点信号指定为X2
- 原点回归速度为30kHz
- 原点爬行速度为1kHz
- 轴事件配置
- 过近点脉冲数设置为200
- 轴0（Y0）执行HOMA指令
- 指令执行完后，复位前置触点

在X2=ON和OFF时，置位M100启动HOMA指令，运行情况即对应模式二的两种情况。

3.4.10 多段速输出[DPTO]

用表格方式设定多段（高达255段）“速度－脉冲数”的脉冲输出定位指令。

输出时可设定成固定脉冲数定位输出，也可设定成中断定位输出。

- **指令格式**

1） 指 令 格 式



2） 设 定 数 据

操作数种类	内容	数据类型
D1	参数表起始单元	BIN 16 位
D2	软件虚拟输出轴号	BIN 16 位

3） 对 象 软 元 件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		•	•							
D2	•									

- **相关软元件**

1) D1 脉冲输出表格软元件首地址

以D1为首地址脉冲输出表格说明。

(1) 控制段：以D1 开始。

D1+地址偏移	内容
D1+0	总段数 2-255
D1+1	中断段号指定 0-254 段序号从0开始排列，如第6段的段序号为5。 发生中断的段必须是： ①脉冲数为0（无限脉冲）； ②起始速度等于终止速度且速度不为0；
D1+2	出错定位号 =0 初始状态（未报错或未工作） =1 总段数出错 =2 速度超轴最大速度 =3 加速度超轴最大加速度 =4 脉冲数小于3周期的脉冲 =5 中断段错 =6 0速段时间小于3个周期 =7 前段终点速度与本段起点速度不相等 =8 起终点速度（第一段起始速度和最后一段终止速度）大于最小速度（轴起始速度）
D1+3	保留

(2) 数据段：以D1+4 开始。

段号	起始速度（双字）	终止速度（双字）	脉冲数（双字）

第一段	D1+4	D1+6	D1+8
第二段	D1+4+6	D1+6+6	D1+8+6
...			
第N-1段	D1+4+6*(N-2)	D1+6+6*(N-2)	D1+8+6*(N-2)
第N段	D1+4+6*(N-1)	D1+6+6*(N-1)	D1+8+6*(N-1)

2) D2 指定软件虚拟输出轴号(AXV0～AXV3)，对应数值 0～3。

3) 相关轴参数列表：

软元件编号				名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴			
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3
R301	R351	R401	R451	脉冲输出方向使能	16 位	0
R302	R352	R402	R452	轴时间当量	16 位	1000
R305	R355	R405	R455	轴最高限速	16 位	20000
R306	R356	R406	R456	轴起始速度	16 位	0
R307	R357	R407	R457	轴最大加速度	16 位	1000
R323	R373	R423	R473	轴急停事件	16 位	0
R314 Bit10	R364 Bit10	R414 Bit10	R464 Bit10	急停信号极性 =0 常开 =1 常闭	位	0
R314 Bit11	R364 Bit11	R414 Bit11	R464 Bit11	急停信号使能 =0 不使能 =1 使能	位	0
R314 Bit12	R364 Bit12	R414 Bit12	R464 Bit12	急停类型选择 =0 立即停车 =1 减速停车	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源选择 =0 指向M =1 指向X	位	0
R310	R360	R410	R460	左限位信号输入 端口指定	16 位	0
R314 Bit2	R364 Bit2	R414 Bit2	R464 Bit2	左限位信号使能	位	0
R314 Bit6	R364 Bit6	R414 Bit6	R464 Bit6	左限位信号极性	位	0
R311	R361	R411	R461	右限位信号输入 端口指定	16 位	0
R314 Bit3	R364 Bit3	R414 Bit3	R464 Bit3	右限位信号使能	位	0
R314 Bit7	R364 Bit7	R414 Bit7	R464 Bit7	右限位信号极性	位	0
R324	R374	R424	R474	轴中断信号指定	位	0

R314 Bit8	R364 Bit8	R414 Bit8	R464 Bit8	中断信号极性	位	0
R314 Bit9	R364 Bit9	R414 Bit9	R464 Bit9	中断信号使能	位	0
R318	R368	R418	R468	相对位置脉冲数	32 位	0
R320	R370	R420	R470	轴状态	16 位	0
R321	R371	R421	R471	错误代码	16 位	0
R322	R372	R422	R472	1ms 插补增量	16 位	0
R326	R376	R426	R476	轴中断位置	32 位	0

① 支持左右限位减速停止。

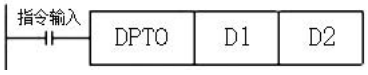
② 支持急停功能立即停止或减速停止。

③ 支持事件中断。

事件优先级
同时发生时，按①到③的优先顺序执行。① ②发生时，可中止③。

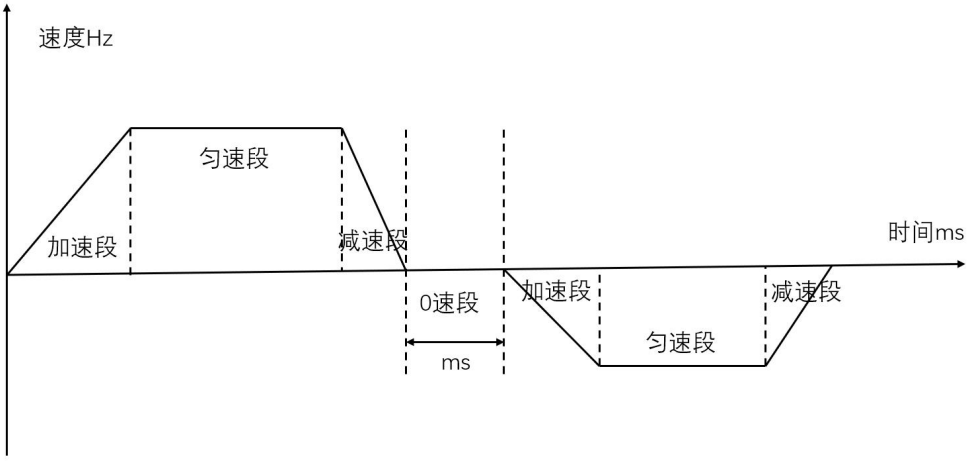
- **功能和举例**

指定的数字轴（D2）以D1指定的参数表输出多段速脉冲。



1) 参数分类

数据段共有5种类型：



① 加速段：终止速度大于起始速度。

② 减速段：终止速度小于起始速度。

③ 匀速段： 终止速度等于起始速度。

④ 中断段：终止速度等于起始速度且不为0，脉冲数为0（表示无限脉冲）。

⑤ 0 速段：终止速度和起始速度都为0，原存放脉冲数的地址，此时存放的值含义不是脉冲数而是暂停时间，单位ms。

2) 参数设置共性要求

a、常规段（加速、减速、匀速）每段脉冲数要求：

脉冲数≥3*（起始速度+终止速度）/2000，即脉冲数不小于3个插补周期的脉冲数（起始速度和终止速度单位Hz）。

中断段：脉冲数为0（表示无限脉冲），必须是匀速段且速度不为0。

0速段：终止速度和起始速度都为0，原存放脉冲数的地址表示暂停ms数，值须≥3。

b、中断定位可发生在中断段及以前段。

c、0速段的曲线用于暂停和换向。

d、脉冲数带符号。速度符号依脉冲数符号定，同一段速度不换方向。

e、速度、加速度要求。

不超过轴参数中的速度及加速度限幅。

通过修改轴限幅参数来调整。

3) 轴报错情况说明

（1）DPTO报错时，轴状态（R320/R370/R420/R470）=96，

（2）若存在参数段脉冲数、终止速度和起始速度都为0时，(D1+2)=6，报 0速段时间小于3个周期，此报错情况检测优先级较高。

（3）(D1+0) 总段数1<N≤255，设其它段数（N<2或N>255）时会报错，(D1+2)=1，报总段数出错。

（4）(D1+1) 事件中断定位段Nv（中断段号指定）从0开始计数，若第四段参数设为中断段，令Nv=4则会报中断段错，(D1+2)=5，即中断段号指定的参数段类型不为中断段（终止速度等于起始速度且不为0，脉冲数为0）时，(D1+2)=5 报中断段错。第四段参数设为中断段，令Nv=3，中断段指定正确。

（5）指令参数表中的速度超过轴最高限速时报错，(D1+2)=2，报速度超轴最大速度。

（6）指令参数表中的参数计算出的加速度超过轴最大加减速时报错，(D1+2)=3，加速度超轴最大加速度。

（7）轴起始速度大于第一段起始速度时，轴起始速度并未生效，依旧按第一段起始速度开始输出脉冲。 第一段的起始速度和最后一段的终止速度必须小于或等于轴起始速度，否则报错，(D1+2)=8，报起终点速度大于最小速度。当轴起始速度大于0时，第一段的起始速度和最后一段的终止速度可以设大于0小于或等于轴起始速度的值，且以此设定频率启停。

（8）脉冲数<3*（起始速度+终止速度）/2000，即脉冲数小于3个插补周期的脉冲数（起始速度和终止速度单位Hz）时，(D1+2)=4，报脉冲数小于3个周期的脉冲。

（9）前段终止速度与本段起始速度不相等，(D1+2)=7，报前段终止速度与本段起始速度不相等。

使用案例 1：（多段固定脉冲数定位）

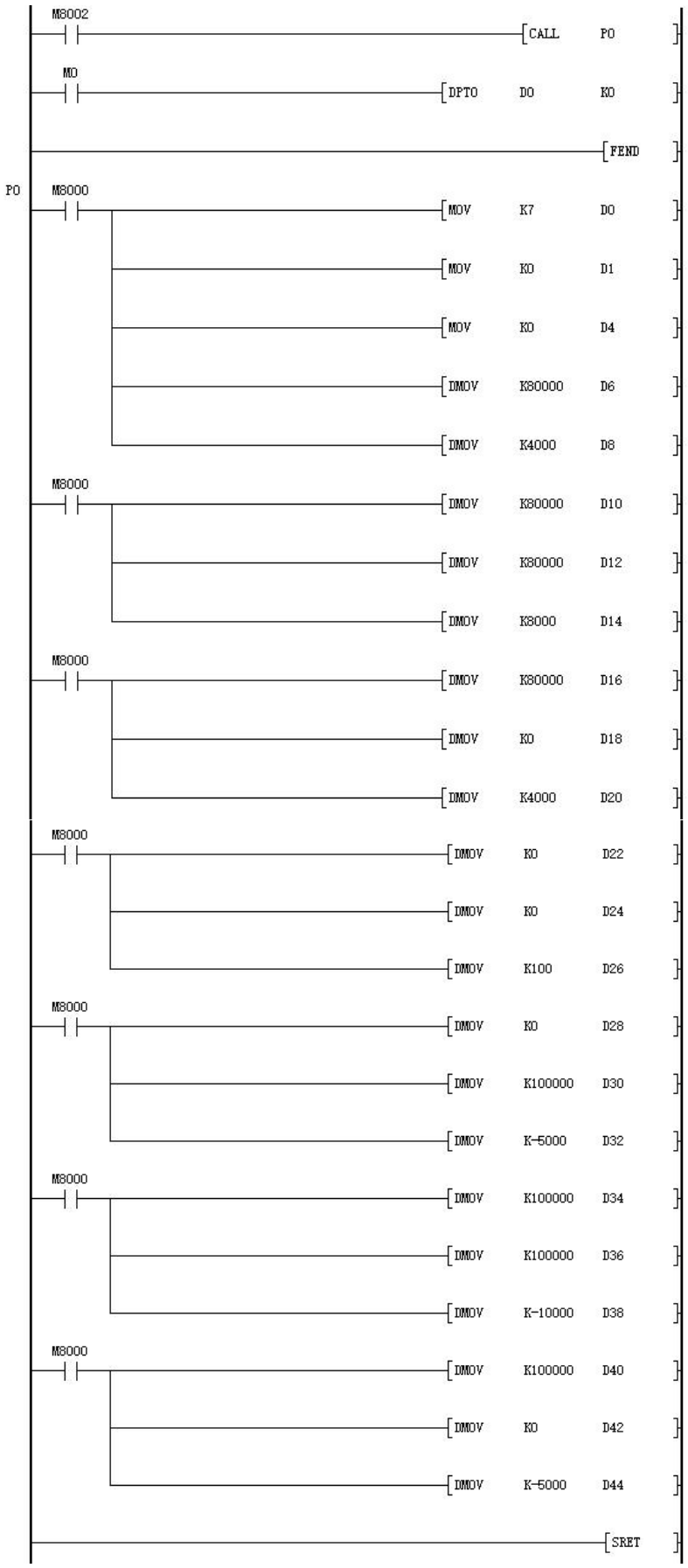
使用DPTO指令设定7段输出，第一段从0KHZ加速至80KHZ正向输出4000个脉冲，第二段以80KHZ正向输出8000个脉冲，第三段从80KHZ减速至0KHZ正向输出4000个脉冲，第四段0速暂停100ms，第五段从0KHZ加速至100KHZ反向输出5000个脉冲，第六段以100KHZ反向输出10000个脉冲，第七段从100KHZ减速至0KHZ反向输出5000个脉冲后停止。默认轴最大加速度，轴最高限速。

以D0为首地址脉冲输出表格示例：

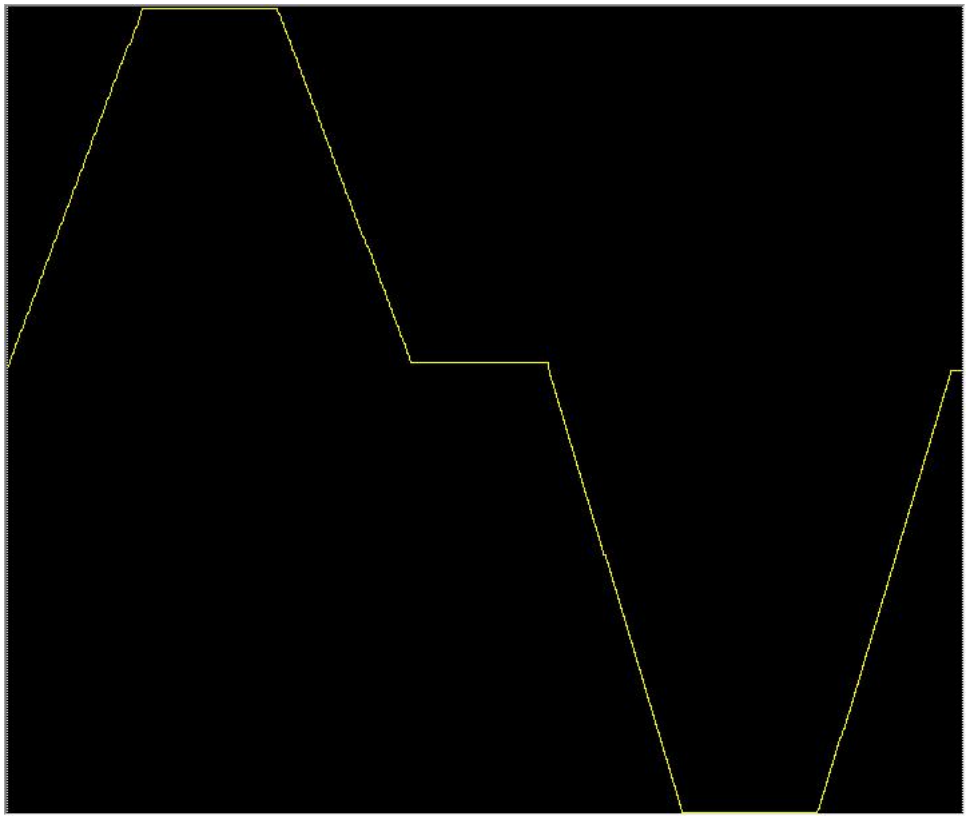
地址	含义	设定值
D0	总段数	7
D1	中断段号指定	0
D2	出错定位号	-
D3	保留	-

段号	起始速度（双字）	终止速度（双字）	脉冲数（双字）
第一段	D4=0	D6=80000	D8=4000
第二段	D10=80000	D12=80000	D14=8000
第三段	D16=80000	D18=0	D20=4000
第四段	D22=0	D24=0	D26=100
第五段	D28=0	D30=100000	D32=-5000
第六段	D34=100000	D36=100000	D38=-10000
第七段	D40=100000	D42=0	D44=-5000

梯形图示例：



1ms采集一次R322的值生成速度曲线如下：（最大80，最小-100）



使用案例 2：（含中断段多段脉冲定位）

使用DPTO指令设定7段输出，第一段从0KHZ加速至80KHZ正向输出4000个脉冲，第二段以80KHZ正向输出8000个脉冲，第三段从80KHZ减速至0KHZ正向输出4000个脉冲，第四段0速暂停100ms，第五段从0KHZ加速至100KHZ反向输出5000个脉冲，第六段为中断段以100KHZ反向输出脉冲，第七段从100KHZ减速至0KHZ反向输出5000个脉冲后停止。脉冲输出Y1，方向位Y5，默认轴最大加速度，轴最高限速，中断信号指定X1，常开信号。

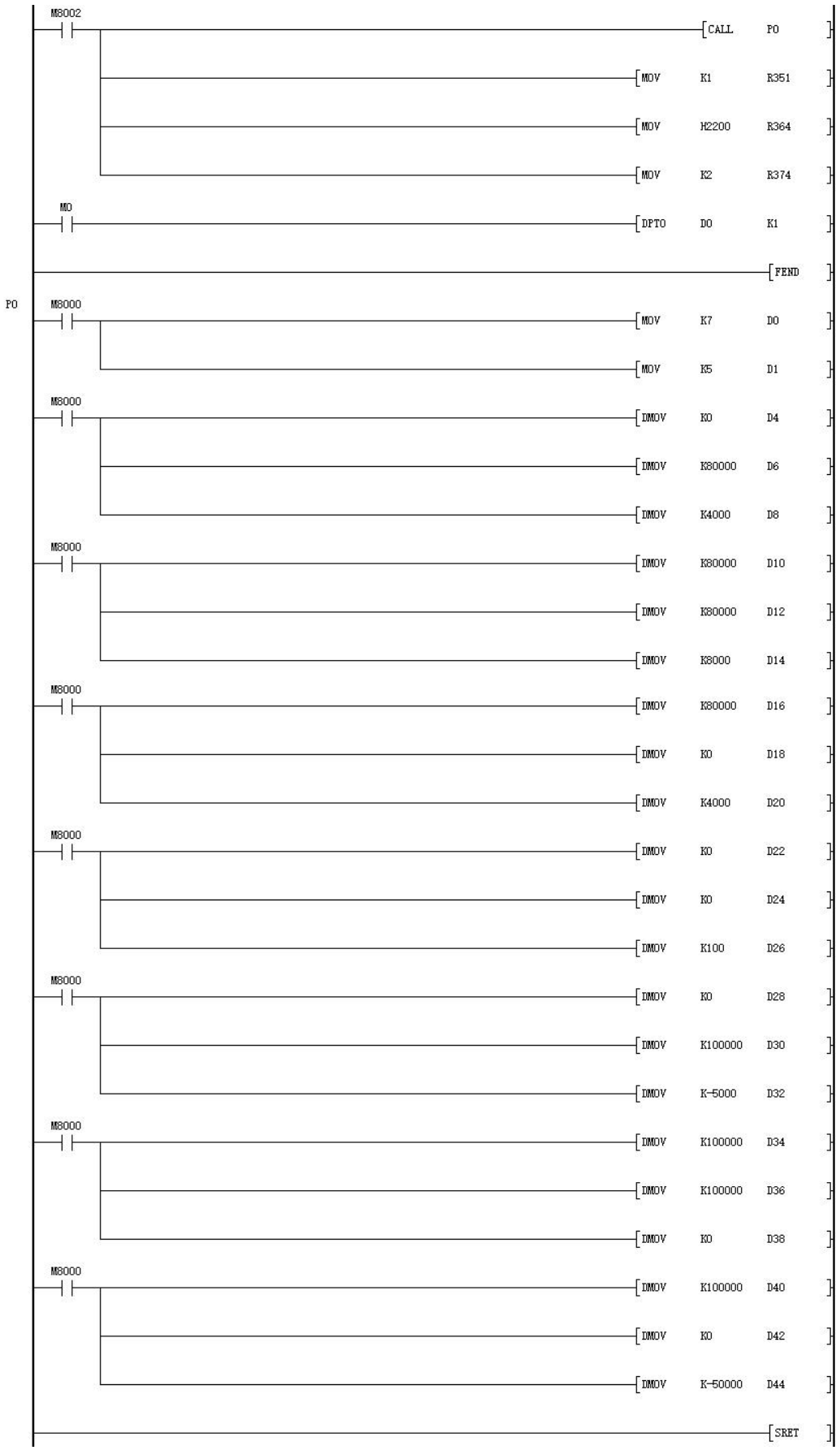
以D0为为首地址脉冲输出表格和轴参数设置示例：

地址	含义	设定值
D0	总段数	7
D1	中断段号指定	5（第六段为中断段）
D2	出错定位号	-
D3	保留	-
R305	轴最高限速	20000（默认）
R306	起始速度	0
R307	轴最大加速度	1000（默认）
R324	轴中断信号指定	2
R314Bit8	中断信号极性	0
R314Bit9	中断信号使能	1
R314Bit13	输入信号源选择	1

R364 0010 0010 0000 0000，转化为 16 进制，H2200

段号	起始速度（双字）	终止速度（双字）	脉冲数（双字）
第一段	D4=0	D6=80000	D8=4000
第二段	D10=80000	D12=80000	D14=8000
第三段	D16=80000	D18=0	D20=4000
第四段	D22=0	D24=0	D26=100
第五段	D28=0	D30=100000	D32=-5000
第六段	D34=100000	D36=100000	D38=0
第七段	D40=100000	D42=0	D44=-5000

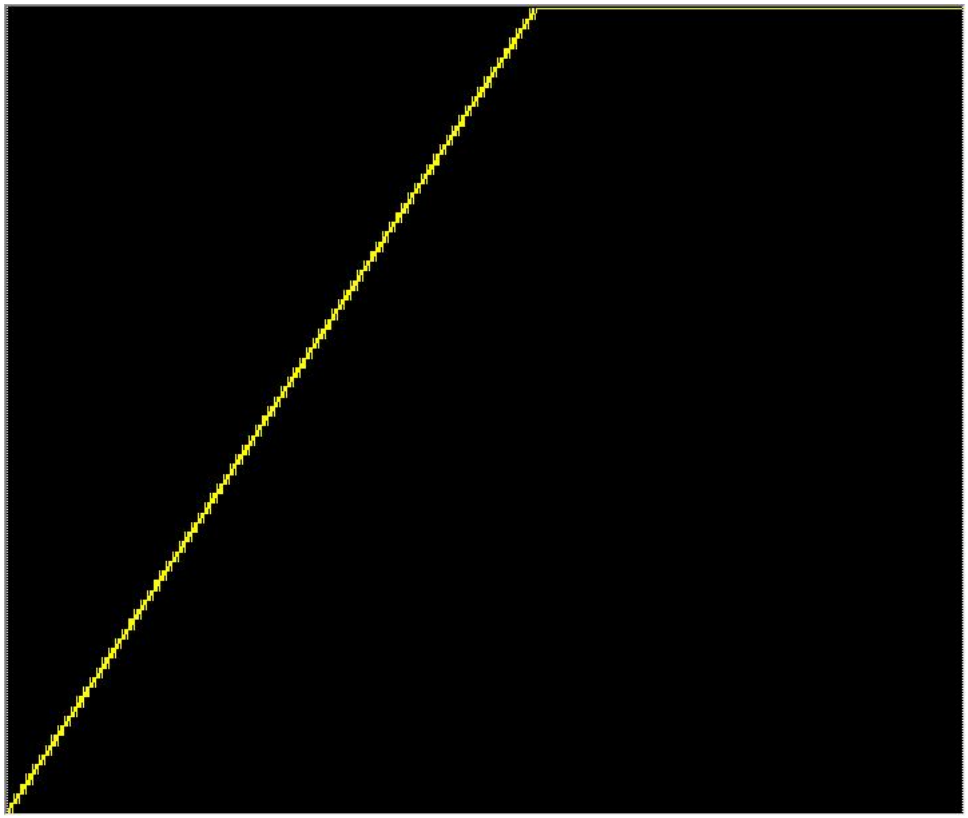
梯形图示例：



1ms采集一次R372的值生成速度曲线如下：（最大80，最小-100）



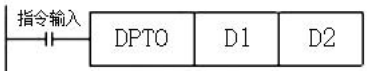
X1中断信号来后，1ms采集一次R322的值生成速度曲线如下：（最大-1，最小-100）



注：由于中断段脉冲数为0，无法区分正反向。中断段的脉冲输出方向会延续前段方向。如上例中，中断段前段为反向输出，那么中断段也为反向输出。

若中断段前段为正向输出，则中断段也为正向输出。

- **注意要点**

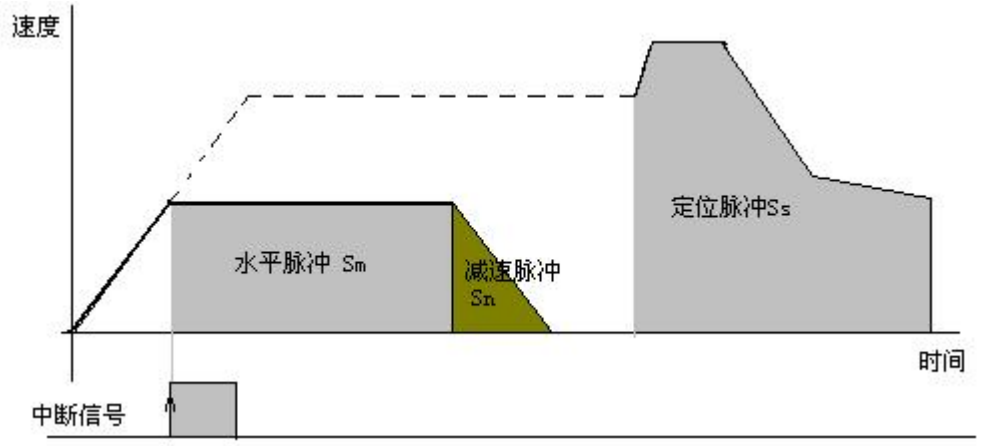


1) DPTO报错时，轴状态（R320/R370/R420/R470）=96，若存在参数段脉冲数、终止速度和起始速度都为0时，(D1+2)=6，报0速段时间小于3个周期，此报错情况检测优先级较高。
2) (D1+0) 总段数1<N≤255，设其它段数（N<2或N>255）时会报错，(D1+2)=1，报总段数出错。
3) (D1+1) 事件中断定位段Nv（中断段号指定）从0开始计数，若第四段参数设为中断段，令Nv=4则会报中断段错，(D1+2)=5，即中断段号指定的参数段类型不为中断段（终止速度等于起始速度且不为0，脉冲数为0）时，(D1+2)=5 报中断段错。第四段参数设为中断段，令Nv=3，中断段指定正确。
4) 速度参数配置时，须遵守设定脉冲频率≤轴最高限速，(D1+2)=2，否则报速度超轴最大速度。起终点速度（第一段的起始速度和最后一段的终止速度）≤轴起始速度。轴起始速度大于第一段起始速度时，轴起始速度并未生效，依旧按第一段起始速度开始输出脉冲。 起终点速度（第一段的起始速度和最后一段的终止速度）必须小于或等于轴起始速度，否则报错，(D1+2)=8，报起终点速度大于最小速度。当轴起始速度大于0时，第一段的起始速度和最后一段的终止速度可以设大于等于0小于或等于轴起始速度的值，且以此设定频率启停。
5) 加减速过程以指令参数表中的参数计算出的加速度【（终止速度^2 - 起始速度^2）/(2*脉冲数）】运行，指令参数表中的参数计算出的加速度超过轴最大加减速时报错，(D1+2)=3，加速度超轴最大加速度。
6) 脉冲数<3*（起始速度+终止速度）/2000，即脉冲数小于3个插补周期的脉冲数（起始速度和终止速度单位Hz）时，(D1+2)=4，报脉冲数小于3个周期的脉冲。
7) 前段终止速度与本段起始速度不相等，(D1+2)=7，报前段终止速度与本段起始速度不相等。
8) 只能在轴状态为 0 时开始运行。
9) 如果增加缓停或急停信号时，轴状态在指令停止发脉冲后不会变为完成状态（32），急停状态为160，缓停中为144，缓停完成为160，建议采用对轴脉冲增量做延时判断的方法来判断是否要断开触点。不断开指令触点时该脉冲轴将始终处于被占用状态。
10) 由于中断段脉冲数为0，无法区分正反向。中断段的脉冲输出方向会延续前段方向。若中断段前段为反向输出，则中断段也为反向输出。若中断段前段为正向输出，则中断段也为正向输出。
11) 事件中断发生在事件中断定位段Nv或之前，产生事件中断定位。

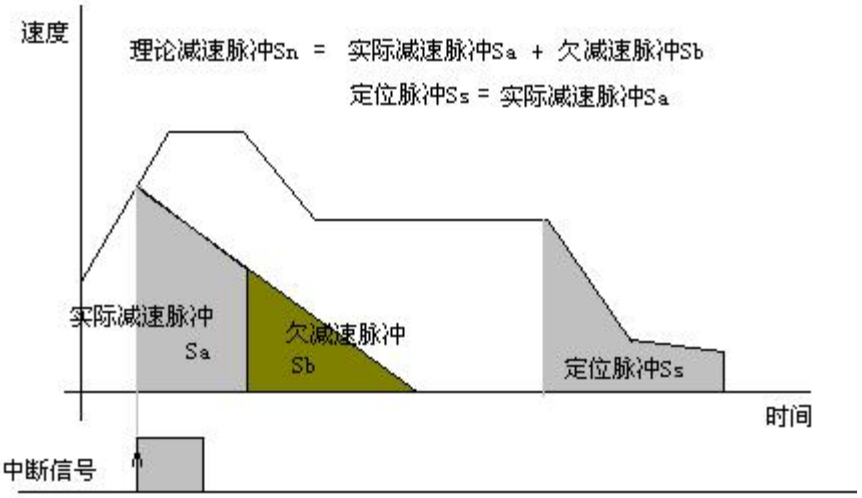
设中断定位段之后的脉冲总数为Ss，当前速度减速停止需脉冲数Sn(以轴最大加减速减速至轴起始速度停车)，发生事件中断时，分以下三种情况：

a、中断定位段Nv发生事件中断，执行中断段下一段及后续段。

b、其它段发生事件中断Ss≥Sn，以当前速度走（Sm=Ss-Sn）个脉冲，减速走Sd=Sn。(Ss=Sm+Sn)



c、其它段发生事件中断Ss=Sa<Sn，减速走Ss=Sa，立即停止。(Sn=Sa+Sb)



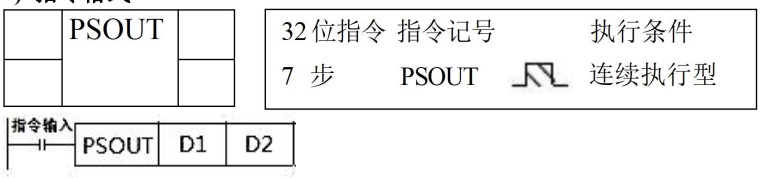
12) 若存在反向段或0速段，在中断段前给出事件中断，此时捕捉到中断信号但不执行事件中断，当DPTO指令运行到中断段时才执行事件中断，中断段运行一个ms后执行中断段下一段及后续段。

3.4.11 定位输出[PSOUT]

指定软件虚拟输出轴(AXV)输出一定个数的脉冲数。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 32 位
D2	输出脉冲数	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●									
D2	●	●								

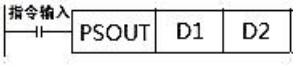
- **相关软元件**

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟0轴	虚拟1轴	虚拟2 轴	虚拟3轴				
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R302	R352	R402	4352	时间当量	16 位	1000	默认
R305	R355	R405	R455	最大速度	16 位	20000	读写
R306	R356	R406	R456	起始速度	16 位	0	读写
R307	R357	R407	R457	最大加减速度	16 位	1000	读写
R318	R368	R418	R468	相对位置脉冲数	32 位	0	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读
R322	R372	R422	R472	1ms 插补增量	16 位	0	只读

- **功能和举例**

该指令为相对运动的单速定位指令，用带正/负号的方式指定距离当前位置的移动距离，其加减速度曲线通过“四次位移曲线加减速控制算法”进行规划。



1) 在D1 中指定软件虚拟输出轴

	设定范围
32 位运算	0：脉冲输出轴，Y0为脉冲端口，Y4位方向端口。 1：脉冲输出轴，Y1为脉冲端口，Y5位方向端口。 2：脉冲输出轴，Y2为脉冲端口，Y6位方向端口。 3：脉冲输出轴，Y3为脉冲端口，Y7位方向端口。

2) 在D2 中指定相对输出脉冲数

	设定范围
32 位运算	-2147483648~2147483647

注：该参数在指令运行过程中修改无效；

负值为反转，方向端口为OFF，正值为正转，方向端口为ON。若需要反向，则需要将脉冲方向端口使能。

3) 加减速控制算法在高速高精度系统中的作用

高速数控加工的加减速直接影响加工效率、加工过程的平稳性和系统加工跟随误差的大小。不同的加减速控制策略会有不同的加工效果，因而在高速加工中需要寻找最优加减速控制方法，以保证机床高速平滑进给，减少冲击和提高加工精度。

伺服系统是数控系统的重要组成部分，是决定数控机床加工精度和生产效率高低的重要因素。高速数控加工要求伺服系统响应速度快、控制精度高、稳定性好，能够提供足够大的加速度和力矩，以保证高速加工的快速启停和加工过程平稳。而加减速控制对机床运行的平稳性、过渡时间、准确定位都有着巨大的影响，在高速加工时，加速度指令的不连续及高次不可微分将导致振动、冲击、定位精度降低等现象。

好的加减速控制算法除了应保证数控机床运动平稳，在启停和程序段间速度有变化时不产生失步、超程、冲击和振荡外；还应当具有算法简单、系统加减速处理时间短、实时性强的特点。此外，很好的柔性(通用性）也是必要的，以便适应不同配置的机床，特别是在高速加工中，这一点显得尤为重要。

4) 四次位移曲线加减速控制算法

由速度、加速度、加加速度之间的导数关系可知，如果其速度控制曲线为三次或三次以上的高阶曲线，就可保证速度，加速度的连续性，以实现高速加工过程中的柔性、平稳加减速过渡控制。

加减速方法的位移、速度、加速度、加加速度的方程如下：

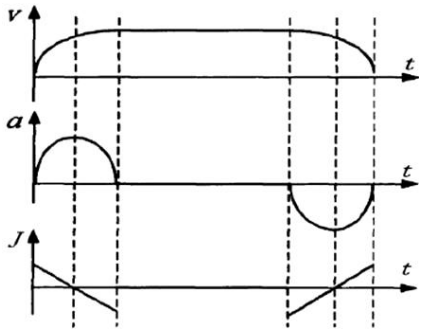
$$S(u)=a_0+a_1u+a_2u^2+a_3u^3+a_4u^4$$

$$V(u)=a_1+2a_2u+3a_3u^2+4a_4u^3$$

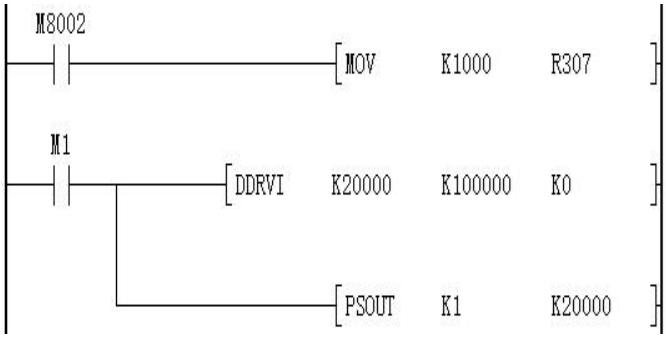
$$a(u)=2a_2+6a_3u+12a_4u^2$$

$$J(u)=6a_3+24a_4u$$

其中 V、a、J 关系如下图：



DDRVI 指令和PSOUT 指令同时发 20000 个脉冲对比速度曲线（设置相同加速度，其它参数按默认值）

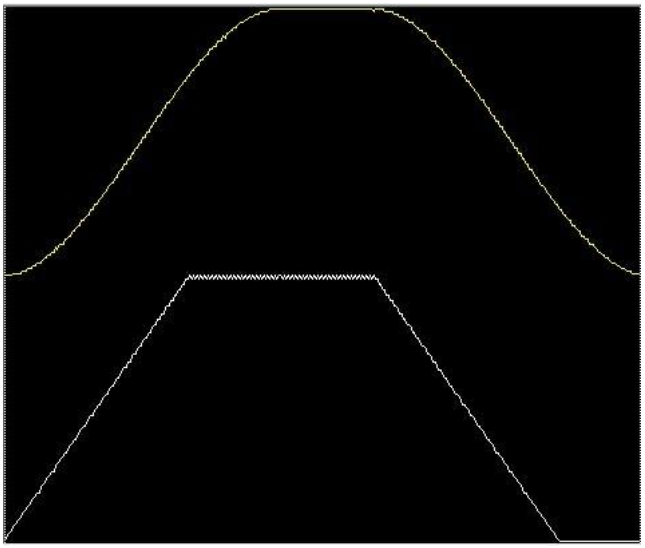


//将轴0最大加减速配置为10P/ms²，与轴1一致

//20000个脉冲，频率为100K，轴0

//20000个脉冲，频率为轴最高速（100K），轴1

置位M1为ON状态，PSOUT指令和DDRVI指令发脉冲的速度曲线对比，如下图：

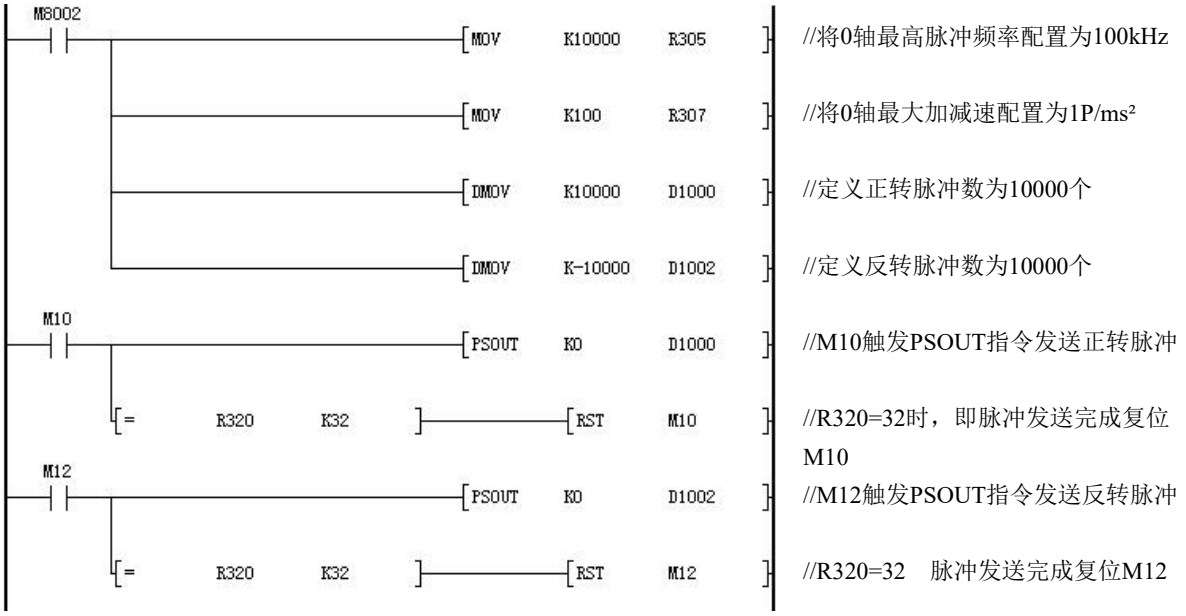


上方为PSOUT指令曲线，下方为DDRVI指令曲线。

可见PSOUT指令的速度曲线更加柔和平滑，在实际应用中可以更好的减少对电机、机床的冲击。当脉冲发送完成时，监控数据如下图：

虚拟轴号	轴参数	数值	说明
虚拟0轴	R318	20000	相对位置
	R320	32	脉冲发送完成
虚拟1轴	R368	20000	相对位置
	R370	32	脉冲发送完成

当需要控制伺服电机先正转10000个脉冲，再反转10000个脉冲，起始频率0Hz，最高频率100kHz，终止频率0Hz，最大加减速度1p/ms²，则梯形图如下：



监控虚拟0轴参数：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms插补增量
0	0	0	0

置位M10为ON，Y0端口发脉冲，Y1端口为ON，正转发10000个脉冲

指令执行完成时的轴参数如下：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
10000	32	0	1

再置位M12为ON，Y0端口发脉冲，Y1端口为OFF，反转发10000个脉冲。

指令执行完成时的轴参数如下：

R318 相对位置脉冲数	R320 轴状态	R321 错误代码	R322 1ms 插补增量
0	32	0	-1

- **注意要点**

1) 必须使用常开触点导通执行指令，断开触点则轴运动停止；

2) 该指令的运动曲线由当前轴的轴起始速度、轴最大速度、轴最大加减速等参数决定。指令的起始速度和停止速度为当前轴的轴起始速度，运动过程中的最大速度为当前轴的轴最大速度。

3) 只有当虚拟轴状态参数为0，即无运动指令执行时，才能对该虚拟轴执行PSOUT指令；

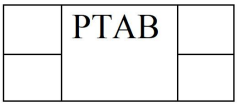
4) 设置脉冲数>0为正向运动，设置脉冲数<0为反向运动，需要在上电第一个扫描周期配置将方向位使能，否则方向无效。

3.4.12 普通制表[PTAB]

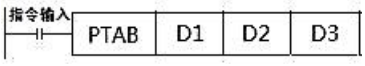
运动规划（位置均匀轨迹点表）制表，生成某个连续轨迹曲线中的一段分段曲线。

- **指令格式**

1） 指令格式



32位指令	指令记号	执行条件
10 步	PTAB	连续执行型



2） 设定数据

操作数种类	内容	数据类型
D1	段曲线参数配置缓存区起始单元	BIN 32 位
D2	轨迹表格起始单元	BIN 32 位
D3	段曲线表格间隔	BIN 32 位

3） 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		●	●							
D2		●	●							
D3	●	●	●							

- **相关软元件**

该指令主要用以生成一个位置表格，可将该表格用于PTOUT指令或CAM指令。具体使用方法见运动指令介绍，该处主要说明各类位置曲线表格的制作方法。

该指令相关于下表寄存器或寄存器的位：

1) D1 指定本段曲线的参数配置缓存区起始寄存器单元，建议为偶数地址。

指令执行时，D1指向寄存器后的连续11个寄存器均被占用即以D1为第1个，一共11个D寄存器被占用，作为一个表格，用作曲线配置的参数设置。

比如D1填D2000，则D2000到D2010被占用。该参数需要在执行PTAB指令之前完成配置。

表格具体内容如下：

制表指令配置表	定义	用法	数据类型
D1+0	曲线类别	=0 匀加速曲线 =2 四次多项式曲线	BIN 16 位
D1+1	曲线方案	=0 确定该段曲线最终位移，不确定最终速度 =1 确定该段曲线最终速度，不确定最终位移	BIN 16 位
D1+2（L）	相对位移	D1+1设为0时需要设置，即该曲线需要最终位移。D1+1设为1时无需设置。	BIN 32 位

D1+4	曲线量程	D1+4必须是D3曲线表格间隔的倍数。 D1+4÷D3等于曲线型值点个数。一般为了方便使用，D3填1，该值直接填需要配置的曲线规划的点个数。	BIN 16 位
D1+5	起始速度	若为首次制表指令的参数配置，推荐填0。 生成多段且连续的曲线轨迹，应填写上一段分段曲线的D1+6结束速度。 单位：kHz	BIN 16 位
D1+6	结束速度	当曲线方案设置为0（最终位移确定曲线）的情况下，可随意配置该值，计算完成后该地址数据由系统生成。 当曲线方案设置为1（最终速度确定曲线）的情况下，必须配置该值	BIN 16 位
D1+7	最大加减速度	分段曲线最大加减速度。生成曲线型值点后该地址数据由系统生成。该数值是否在运动轴的极限加减速范围之内，若超过则必须做调整。 该参数无需设置，为显示值。	BIN 16 位
D1+8	时间当量	分段曲线的时间当量，与脉冲当量耦合计算，为分段曲线速度/加减速度的计量单位之一。 规划包含多个分段曲线的连续轨迹曲线制表，推荐所有的分段曲线设置相同的时间当量。	BIN 16 位
D1+9	脉冲当量	分段曲线的脉冲当量，与时间当量耦合计算，为分段曲线速度/加减速度的计量单位之一。 例如时间当量=1，脉冲当量=1，则分段曲线的速度单位为1脉冲/ms，加速度为1脉冲/ms²。规划包含多个分段曲线的连续轨迹曲线制表，推荐所有的分段曲线设置相同的脉冲当量。	BIN 16 位
D1+10	制表状态	制表指令当前状态，系统同一时间只允许1条制表指令处于运行状态。 ● =0：当前无制表指令运行，可执行制表指令 ● =16：正在生成曲线数据(系统每1ms生成1个型值点数据) ● =32：曲线数据生成完毕 ● =64：曲线生成错误 当检测到制表状态等于32时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清0。该数值变0后才能执行下一个制表流程。 该参数无需设置，为显示值。	BIN 16 位

2) D2 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。

假定某个曲线轨迹规划3个分段曲线拼接而成，则需要使用3次制表指令。这些制表指令所指向D2应填写一致。

例：D2填写D5000，总曲线表格将占用D5000以后的寄存器区域。

在执行首次制表指令前，需将D2指向的寄存器数值填1；若为非接续型曲线，将D2+2指向的寄存器数值填0(均填写32位数值)。

3) D3 指定本段曲线的表格间隔，以时间或主轴位置进行划分。

表格间隔要选取合适的范围，过大则降低曲线细分精度，过小则会造成曲线生成点数过多，使得寄存器开销变大。

表格间隔是求得本段曲线中有多少个点位数据的参数之一，点位数据又称之为型值点。每个型值点均代表从轴的一个位置点信息，通常以脉冲数形式表达。

* 在规划曲线轨迹时，所有分段曲线的表格间隔必须设置为相同数值。

- 按以上设置方法设置好参数后，执行PTAB指令，会生成对应数量的型值点。这些点为位置数据，占用32位寄存器。
- 地址的起点为D2，如点数确认为200，则一共占用404个D寄存器。如D2设置为D3000，D1+4为200，D3为1，则D3000~D3403被占用为曲线轨迹表格地址。即表头（用于记录型值点个数）+起点（默认为0）+曲线的型值点。
- 制表指令执行完毕后D2会显示自动记录曲线点个数，包括型值点加表头，比如D1+4÷D3=200，则D2在制表完成后会变为201，加D2本身共占用202个双字，即404个D寄存器。
- 在制表首次运行PTAB指令前需要手动赋值D2为1，D2+2为0。

● **功能和举例**

说明案例 1

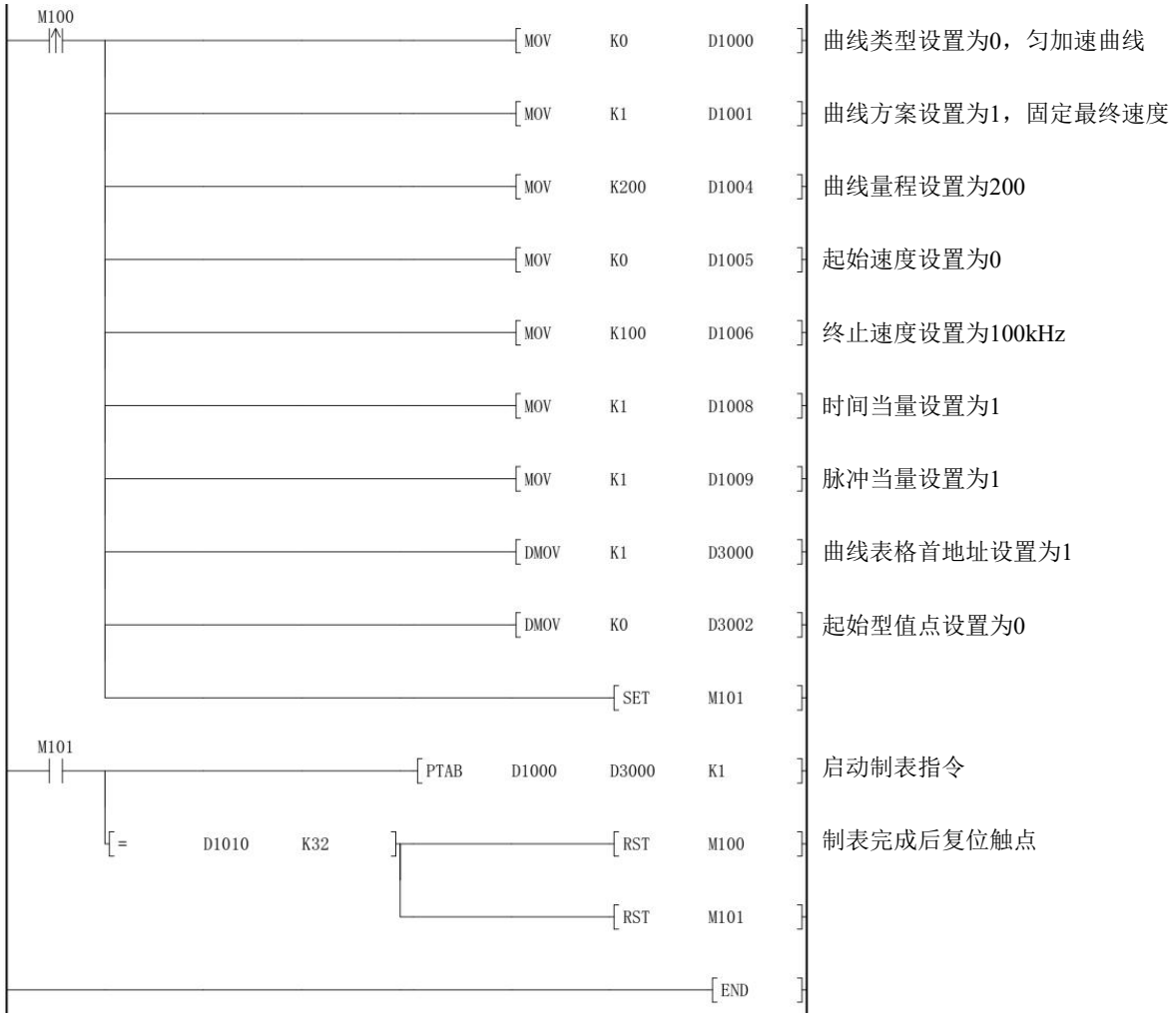
制作一段确定末速度的匀加速升速曲线。

需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	0
D1+6	结束速度	100
D1+8	时间当量	1
D1+9	脉冲当量	1

由于曲线为匀加速曲线，实际可以算出其位移。

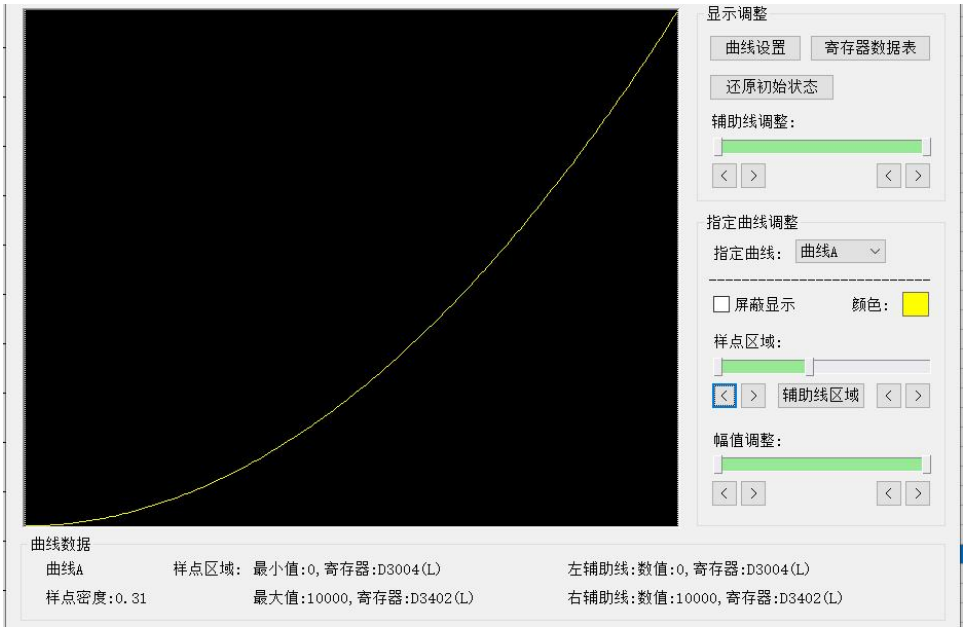
初速度为0p/ms，末速度为100p/ms，型值点个数为200个。可以理解为分200次进行加速，即1ms加速一次，共加速200次，则总位移为（0+100）÷2×200=10000个脉冲。

梯形图如下：



D1000为表格配置起始地址，D3000为曲线储存位置起点，置位M100完成曲线配置。

生成的曲线如下：



型值点的最大值为10000，即整段曲线的总位移为10000个脉冲，与计算值符合。

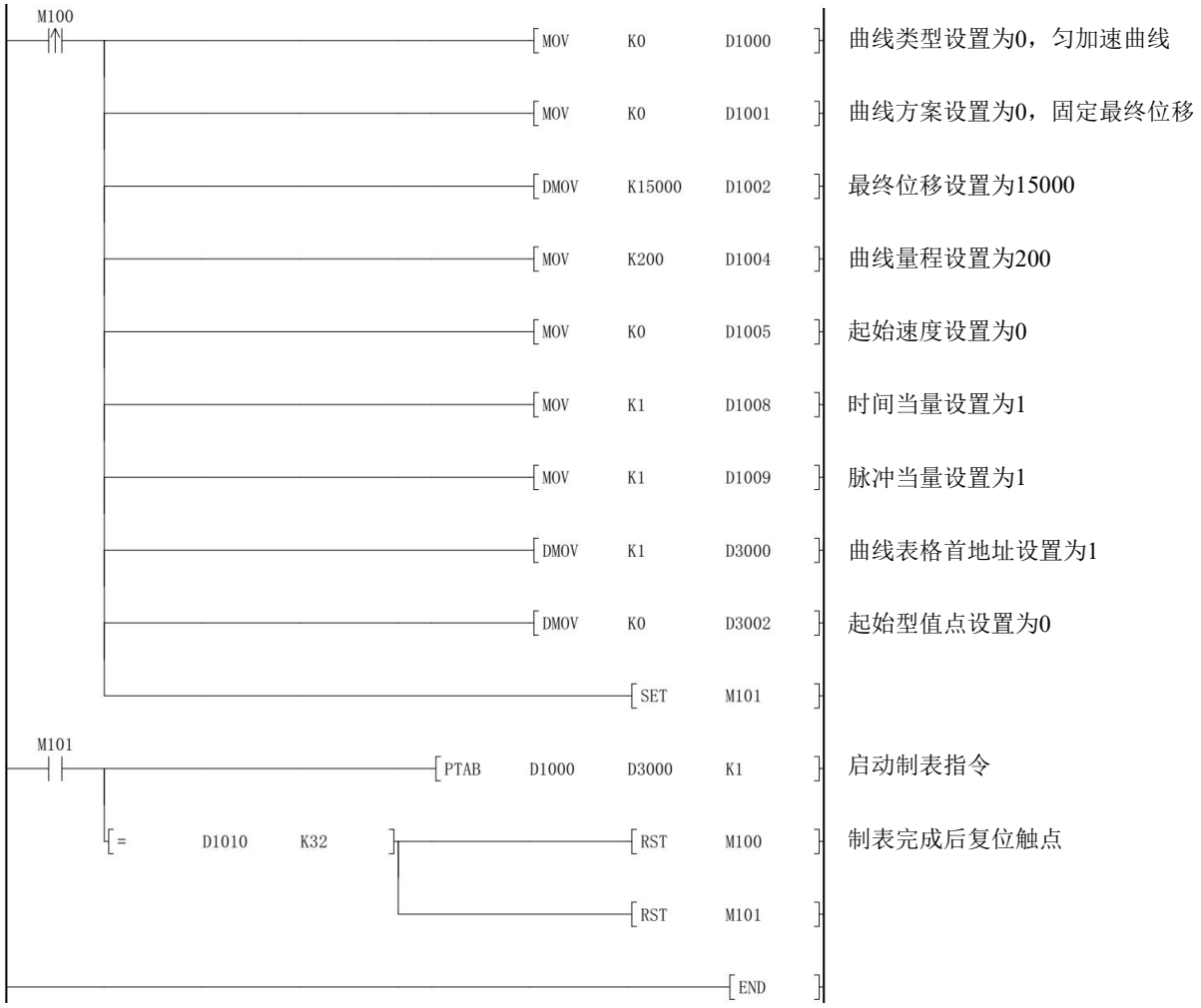
说明案例 2

制作一段确定位移的匀加速升速曲线。

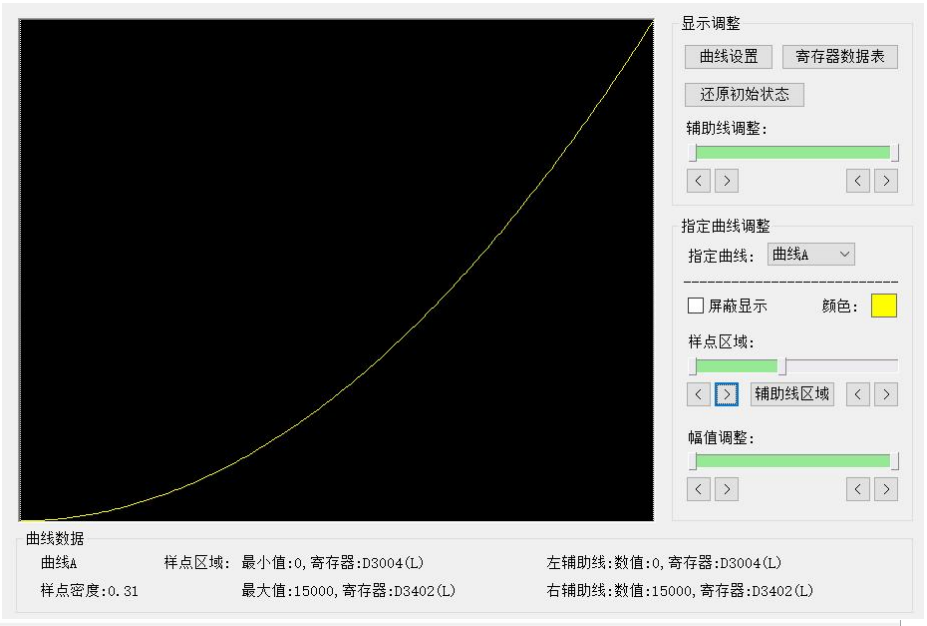
以案例1曲线为基础，如果将曲线类型改为确定最终位移型，且最终位移设为10000，生成的曲线与上述曲线一致。接着保持其他参数不变，将最终位移设为15000。

需要配置的参数	定义	配置值
D1+0	曲线类别	=0匀加速曲线
D1+1	曲线方案	=0 确定该段曲线最终位移，不确定最终速度
D1+2	相对位移	15000
D1+4	曲线量程	200
D1+5	起始速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

梯形图如下：



置位m100完成曲线生成。查看生成的曲线：





	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D3250	5766	0	5859	0	5953	0	6048	0	6144	0
D3260	6240	0	6337	0	6435	0	6534	0	6633	0
D3270	6733	0	6834	0	6936	0	7038	0	7141	0
D3280	7245	0	7350	0	7455	0	7561	0	7668	0
D3290	7776	0	7884	0	7993	0	8103	0	8214	0
D3300	8325	0	8437	0	8550	0	8664	0	8778	0
D3310	8893	0	9009	0	9126	0	9243	0	9361	0
D3320	9480	0	9600	0	9720	0	9841	0	9963	0
D3330	10086	0	10209	0	10333	0	10458	0	10584	0
D3340	10710	0	10837	0	10965	0	11094	0	11223	0
D3350	11353	0	11484	0	11616	0	11748	0	11881	0
D3360	12015	0	12150	0	12285	0	12421	0	12558	0
D3370	12696	0	12834	0	12973	0	13113	0	13254	0
D3380	13395	0	13537	0	13680	0	13824	0	13968	0
D3390	14113	0	14259	0	14406	0	14553	0	14701	0
D3400	14850	0	15000	0	0	0	0	0	0	0
D3410	0	0	0	0	0	0	0	0	0	0
D3420	0	0	0	0	0	0	0	0	0	0
D3430	0	0	0	0	0	0	0	0	0	0
D3440	0	0	0	0	0	0	0	0	0	0

最终位移为15000。

说明案例 3

制作一个两段曲线，先升速后匀速。

先制一段加速曲线。

需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	0
D1+6	结束速度	100
D1+8	时间当量	1

D1+9	脉冲当量	1

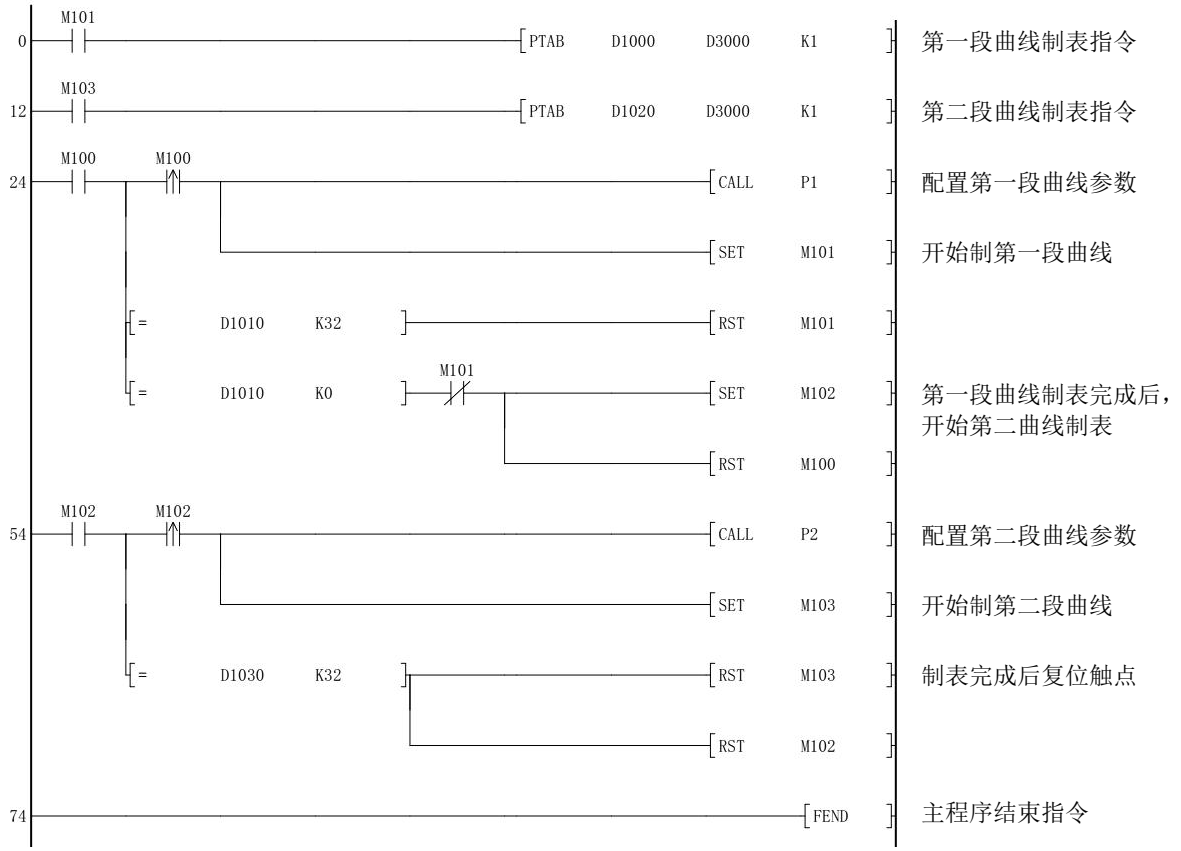
在制一段匀速曲线。即配置一个起始速度与结束速度相同的确定最终速度的匀加速曲线

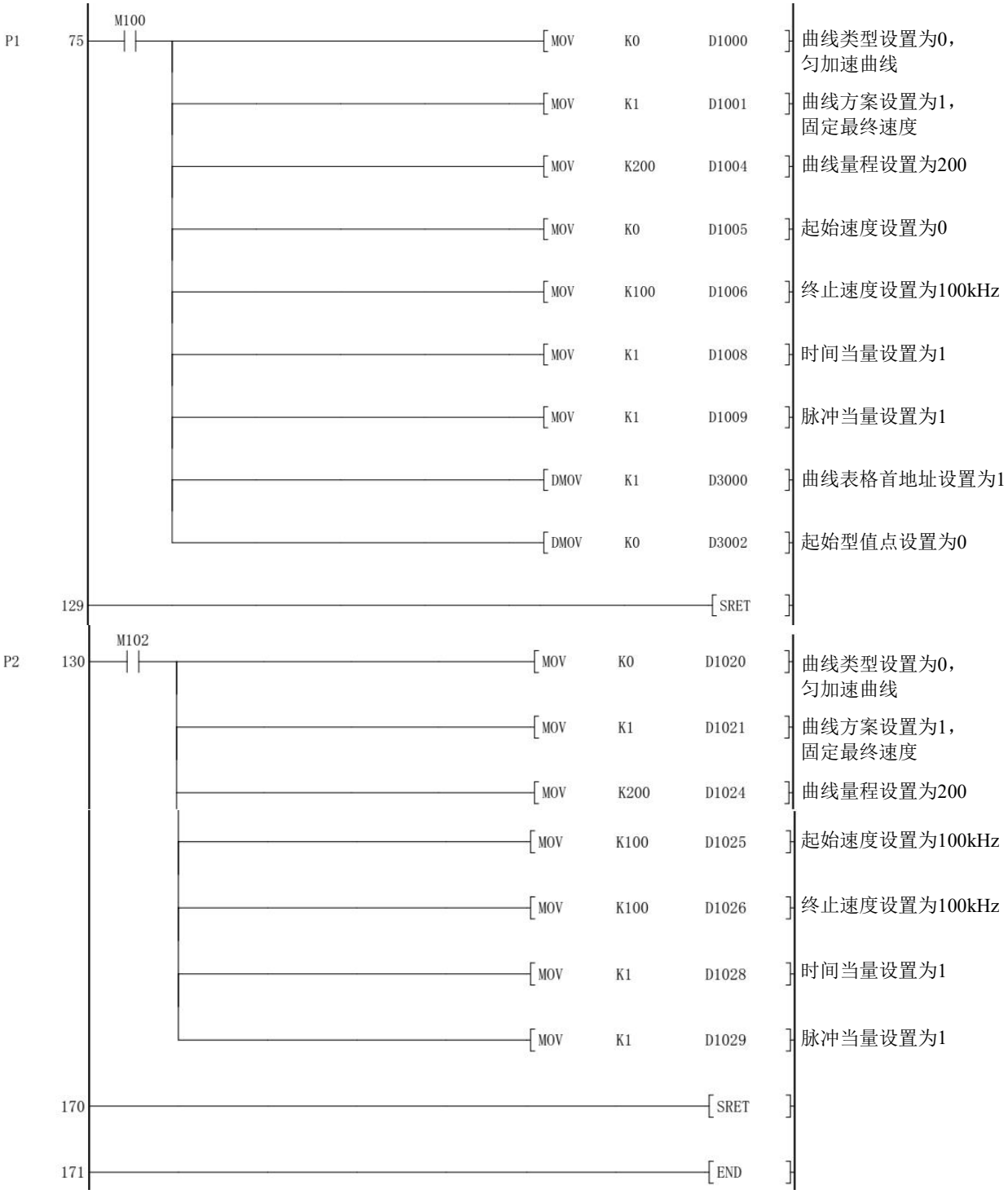
需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	100
D1+6	结束速度	100
D1+8	时间当量	1
D1+9	脉冲当量	1

即第一段曲线速度由0升至100p/ms共用200个点。可算出第一段最终位置10000

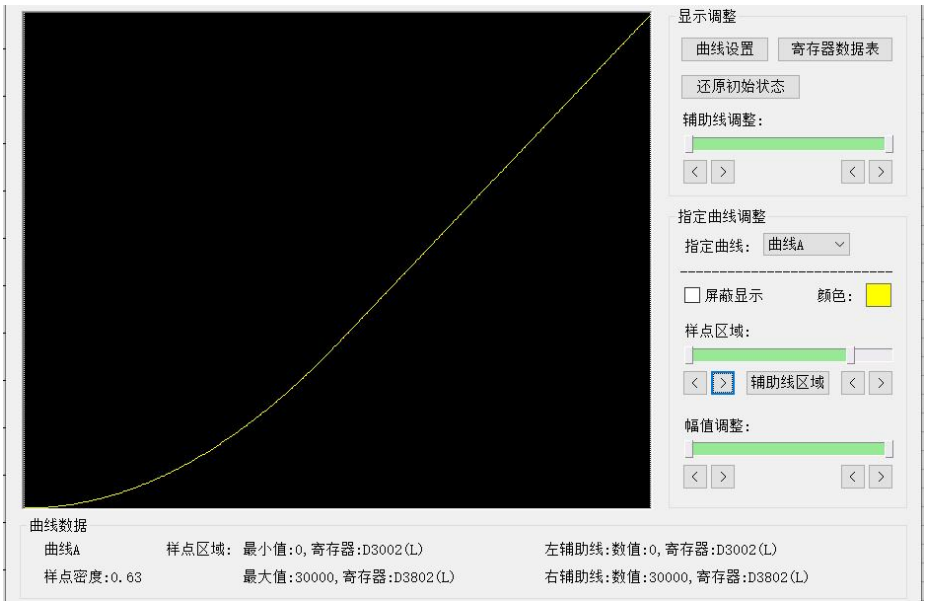
第二段曲线为速度100p/ms的匀速曲线200个点。可算出第二段最终位置为20000+10000=30000

梯形图如下：




P1
75
M100
MOV K0 D1000
曲线类型设置为0，匀加速曲线
MOV K1 D1001
曲线方案设置为1，固定最终速度
MOV K200 D1004
曲线量程设置为200
MOV K0 D1005
起始速度设置为0
MOV K100 D1006
终止速度设置为100kHz
MOV K1 D1008
时间当量设置为1
MOV K1 D1009
脉冲当量设置为1
DMOV K1 D3000
曲线表格首地址设置为1
DMOV K0 D3002
起始型值点设置为0
129
SRET
P2
130
M102
MOV K0 D1020
曲线类型设置为0，匀加速曲线
MOV K1 D1021
曲线方案设置为1，固定最终速度
MOV K200 D1024
曲线量程设置为200
MOV K100 D1025
起始速度设置为100kHz
MOV K100 D1026
终止速度设置为100kHz
MOV K1 D1028
时间当量设置为1
MOV K1 D1029
脉冲当量设置为1
170
SRET
171
END

置位M100生成曲线。



第一段曲线，为匀加速曲线；第二段曲线，为匀速曲线。数值与计算结果相同。

说明案例 4

制作一段反向运行匀加速曲线接匀减速曲线。

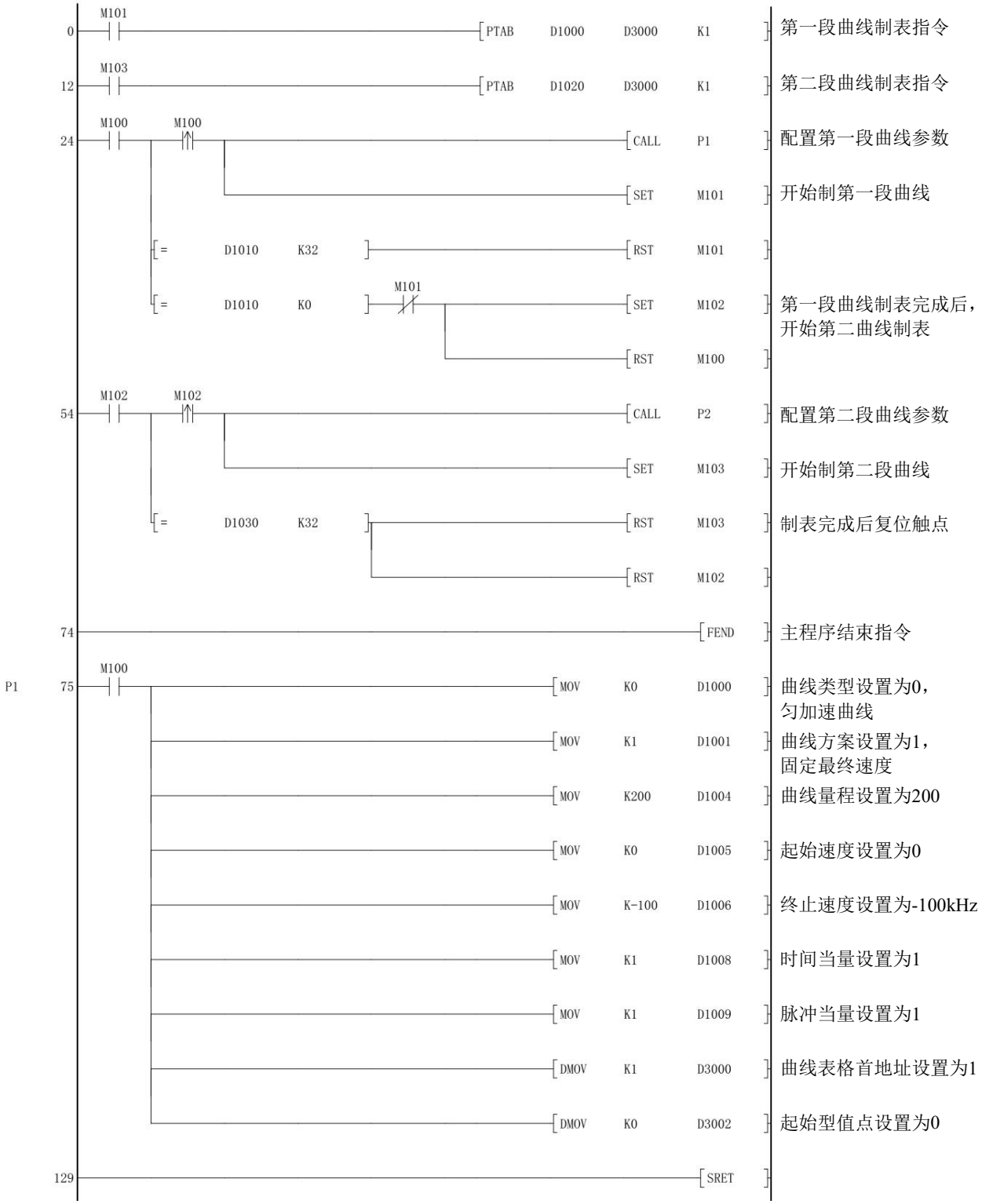
第一段曲线：

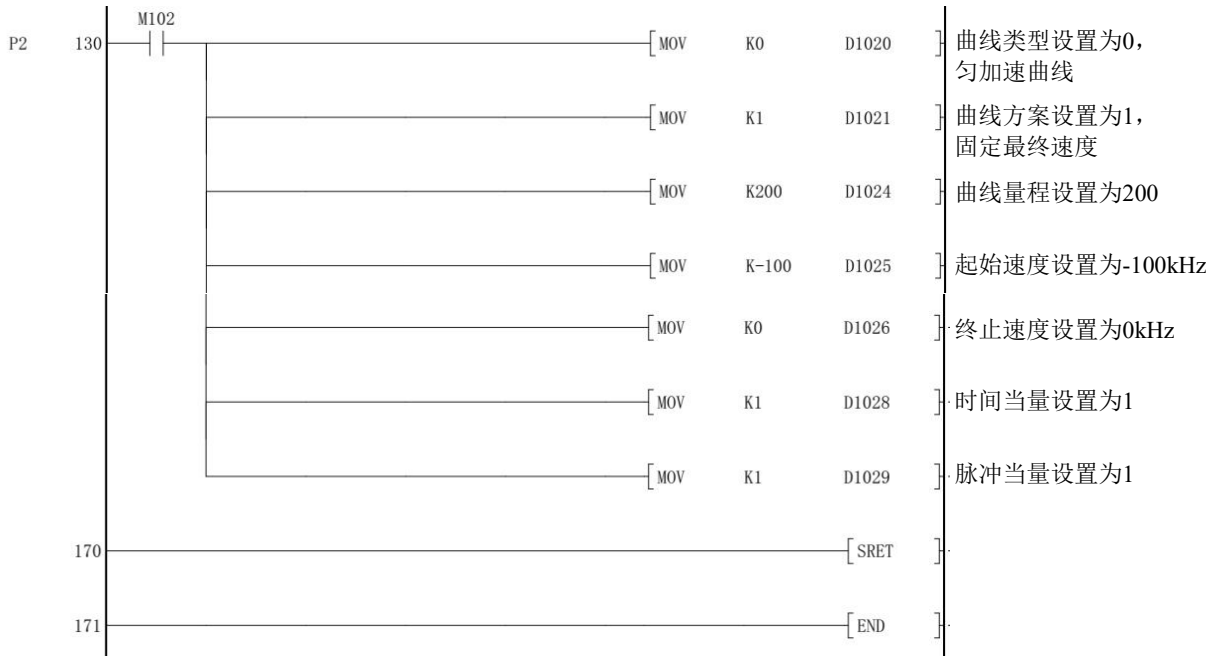
需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	0
D1+6	结束速度	-100
D1+8	时间当量	1
D1+9	脉冲当量	1

第二段曲线：

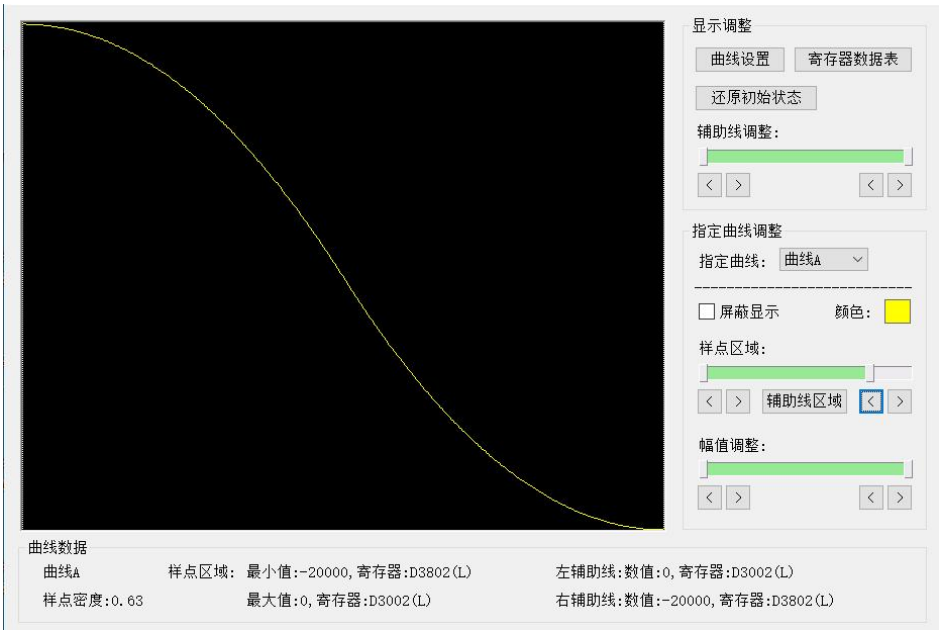
需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	-100
D1+6	结束速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

梯形图如下：


M101
0
PTAB D1000 D3000 K1
第一段曲线制表指令
M103
12
PTAB D1020 D3000 K1
第二段曲线制表指令
M100 M100
24
CALL P1
配置第一段曲线参数
SET M101
开始制第一段曲线
= D1010 K32
RST M101
M101
= D1010 K0
SET M102
第一段曲线制表完成后，开始第二曲线制表
RST M100
M102 M102
54
CALL P2
配置第二段曲线参数
SET M103
开始制第二段曲线
= D1030 K32
RST M103
制表完成后复位触点
RST M102
74
FEND
主程序结束指令
M100
P1 75
MOV K0 D1000
曲线类型设置为0，匀加速曲线
MOV K1 D1001
曲线方案设置为1，固定最终速度
MOV K200 D1004
曲线量程设置为200
MOV K0 D1005
起始速度设置为0
MOV K-100 D1006
终止速度设置为-100kHz
MOV K1 D1008
时间当量设置为1
MOV K1 D1009
脉冲当量设置为1
DMOV K1 D3000
曲线表格首地址设置为1
DMOV K0 D3002
起始型值点设置为0
129
SRET



置位M100生成曲线。查看生成的寄存器曲线：



说明案例 5

制作一段正程+返程的曲线，正程曲线由四次多项式曲线构成，先加速运行10000个脉冲接着减速运行10000脉冲减速到0速；返程曲线由匀加速曲线构成，先反向匀加速运行10000个脉冲，再反向匀减速运行10000个脉冲完成返程，即正向运行20000脉冲，反向运行20000脉冲。

该过程由4段曲线组成：

第一段曲线：

需要配置的参数	定义	配置值
D1+0	曲线类别	=2 四次多项式曲线
D1+1	曲线方案	=0 确定该段曲线最终位移，不确定最终速度
D1+2	曲线位移	10000
D1+4	曲线量程	200

D1+5	起始速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

第二段曲线

需要配置的参数	定义	配置值
D1+0	曲线类别	=2 四次多项式曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	第一段的末速度
D1+6	结束速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

第三段曲线

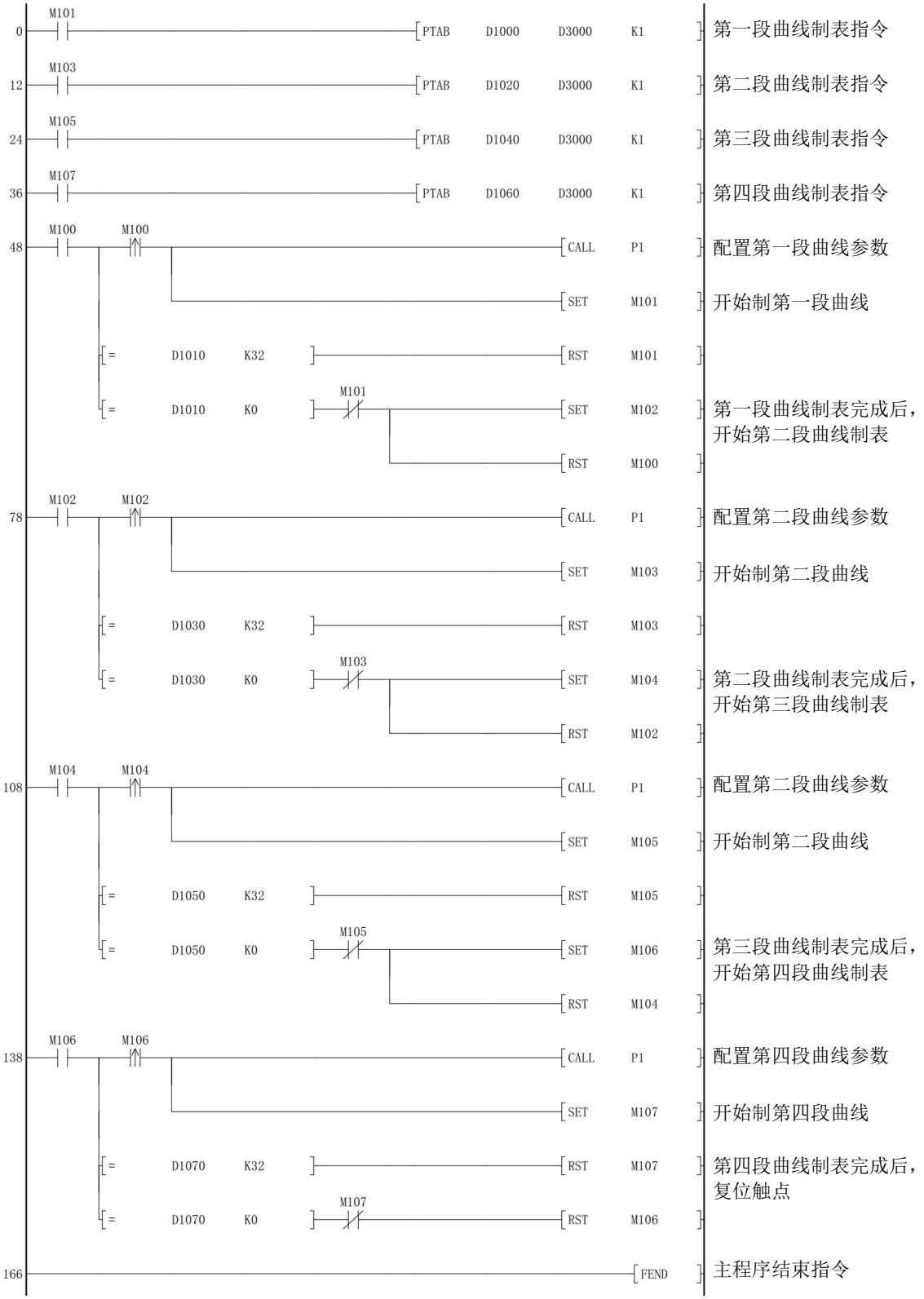
需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=0 确定该段曲线最终位移，不确定最终速度
D1+2	曲线位移	-10000
D1+4	曲线量程	200
D1+5	起始速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

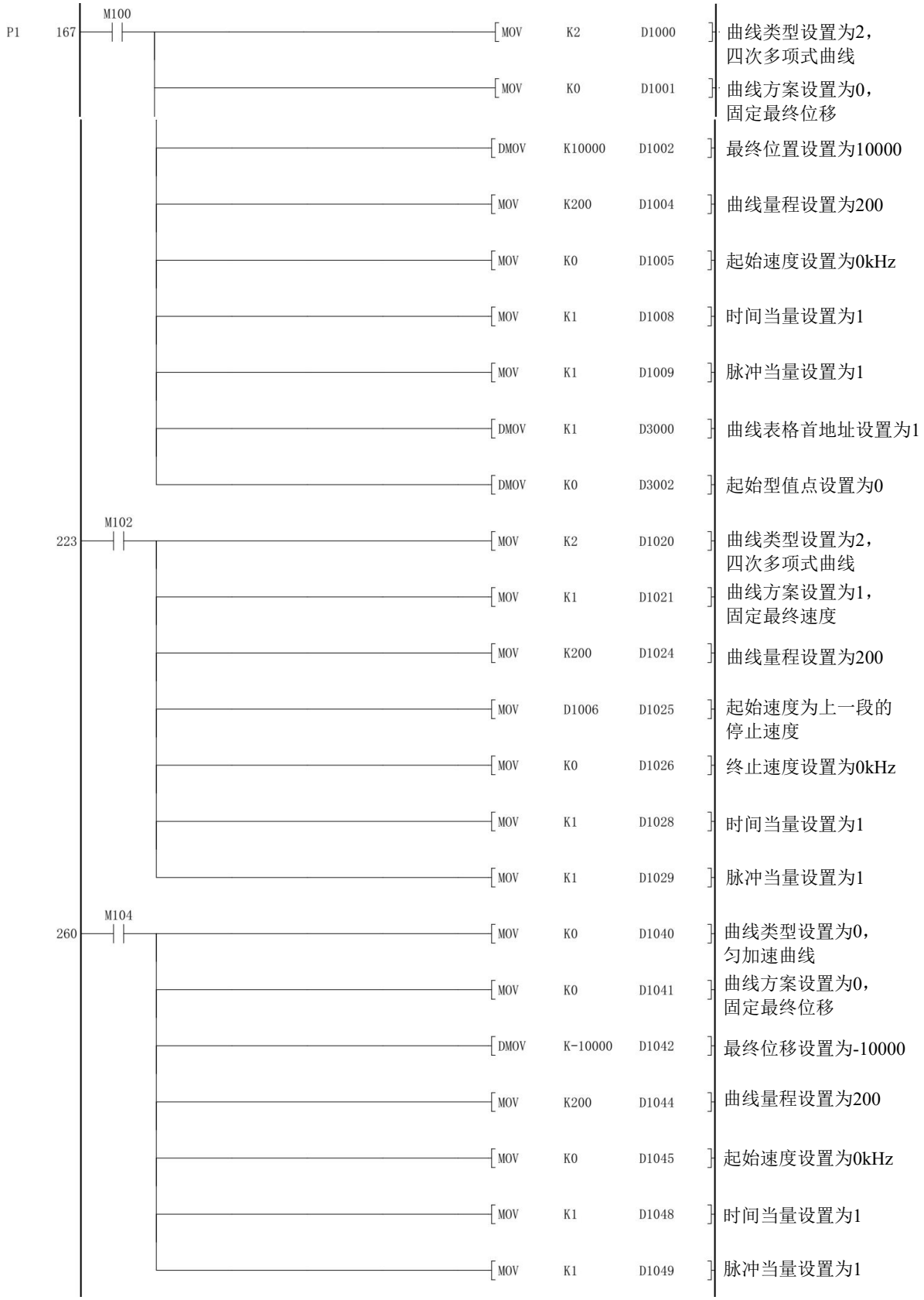
第四段曲线

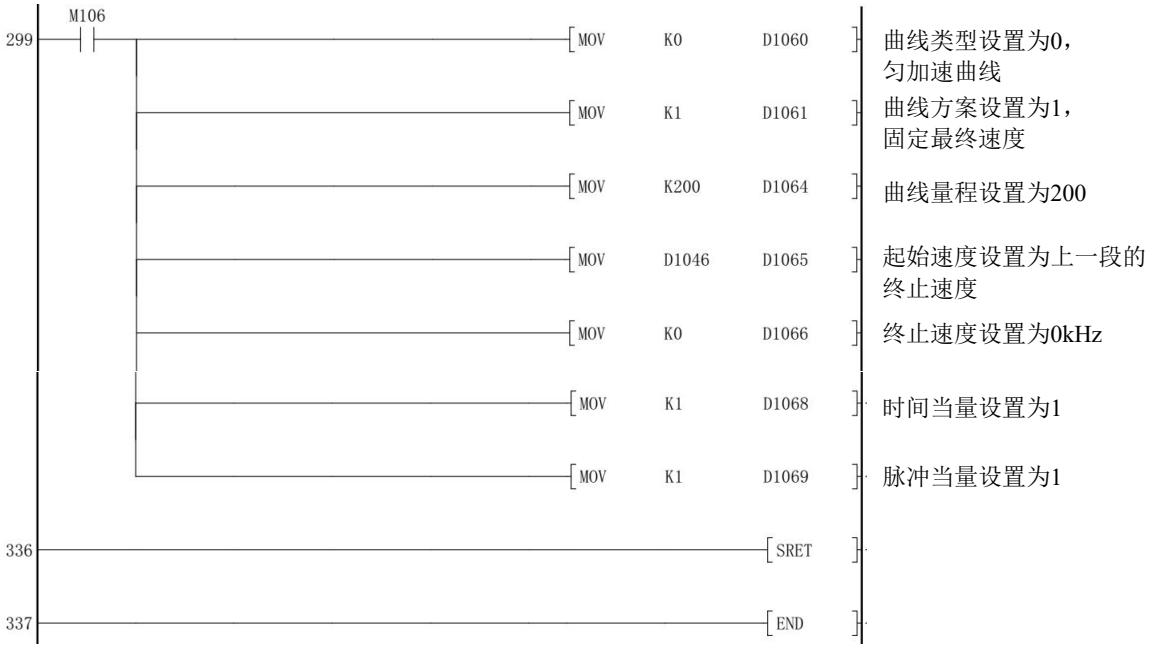
需要配置的参数	定义	配置值
D1+0	曲线类别	=0 匀加速曲线
D1+1	曲线方案	=1 确定该段曲线最终速度，不确定最终位移
D1+4	曲线量程	200
D1+5	起始速度	第三段的末速度
D1+6	结束速度	0
D1+8	时间当量	1
D1+9	脉冲当量	1

该曲线配置中涉及了四次多项式位置确定曲线，四次多项式速度确定曲线，匀加速位置确定曲线，匀加速速度确定曲线。

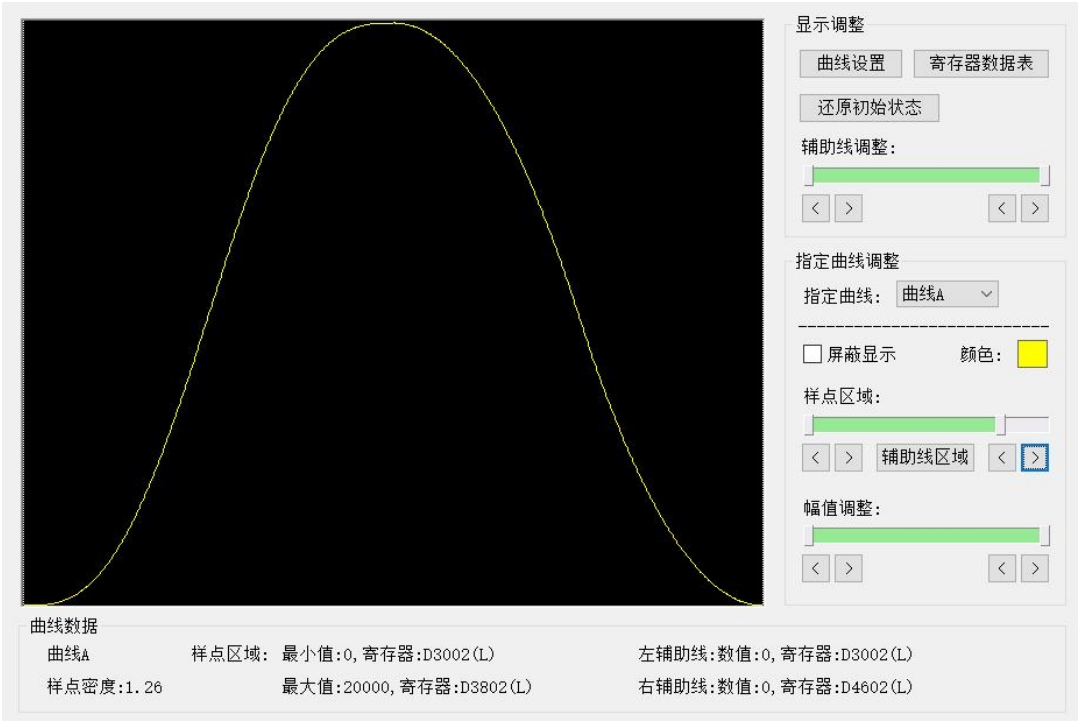
梯形图如下：


M101
0
PTAB D1000 D3000 K1
第一段曲线制表指令
M103
12
PTAB D1020 D3000 K1
第二段曲线制表指令
M105
24
PTAB D1040 D3000 K1
第三段曲线制表指令
M107
36
PTAB D1060 D3000 K1
第四段曲线制表指令
M100 M100
48
CALL P1
配置第一段曲线参数
SET M101
开始制第一段曲线
= D1010 K32
RST M101
= D1010 K0
M101
SET M102
第一段曲线制表完成后，开始第二段曲线制表
RST M100
M102 M102
78
CALL P1
配置第二段曲线参数
SET M103
开始制第二段曲线
= D1030 K32
RST M103
= D1030 K0
M103
SET M104
第二段曲线制表完成后，开始第三段曲线制表
RST M102
M104 M104
108
CALL P1
配置第二段曲线参数
SET M105
开始制第二段曲线
= D1050 K32
RST M105
= D1050 K0
M105
SET M106
第三段曲线制表完成后，开始第四段曲线制表
RST M104
M106 M106
138
CALL P1
配置第四段曲线参数
SET M107
开始制第四段曲线
= D1070 K32
RST M107
第四段曲线制表完成后，复位触点
= D1070 K0
M107
RST M106
166
FEND
主程序结束指令


P1
167
M100
MOV K2 D1000
曲线类型设置为2，四次多项式曲线
MOV K0 D1001
曲线方案设置为0，固定最终位移
DMOV K10000 D1002
最终位置设置为10000
MOV K200 D1004
曲线量程设置为200
MOV K0 D1005
起始速度设置为0kHz
MOV K1 D1008
时间当量设置为1
MOV K1 D1009
脉冲当量设置为1
DMOV K1 D3000
曲线表格首地址设置为1
DMOV K0 D3002
起始型值点设置为0
223
M102
MOV K2 D1020
曲线类型设置为2，四次多项式曲线
MOV K1 D1021
曲线方案设置为1，固定最终速度
MOV K200 D1024
曲线量程设置为200
MOV D1006 D1025
起始速度为上一段的停止速度
MOV K0 D1026
终止速度设置为0kHz
MOV K1 D1028
时间当量设置为1
MOV K1 D1029
脉冲当量设置为1
260
M104
MOV K0 D1040
曲线类型设置为0，匀加速曲线
MOV K0 D1041
曲线方案设置为0，固定最终位移
DMOV K-10000 D1042
最终位移设置为-10000
MOV K200 D1044
曲线量程设置为200
MOV K0 D1045
起始速度设置为0kHz
MOV K1 D1048
时间当量设置为1
MOV K1 D1049
脉冲当量设置为1



置位 M100 开始生成曲线。查看生成的寄存器曲线：



四次多项式正程曲线，包括升速段与减速段。匀加速返程曲线，包括升速段与减速段。

- 注意要点

1） 当制作多段曲线时，所有的曲线段需要共用一个曲线首地址；

2） 第一段曲线配置时，必须要手动对D2+0（L）赋值1和对D2+2（L）赋值0。

3） 第一段曲线的起始速度可以按需求设置，后续曲线的起始速度建议与前一段的最终速度相同。这样更加符合机械运行的状态。设定最终位移曲线的最终速度由系统生成，所以不同段曲线的PTAB指令中的配置地址建议区分开，以便将上一条曲线的最终速度写入当前曲线的起始速度而不是被覆盖。

4） 多段曲线的首地址用于记录该多段曲线的点数。只需在配置第一段曲线时将D2+0（L）赋值1和对D2+2（L）赋值0，在多段曲线配置完成后，D2+0（L）将等于每一段曲线的点数之和加1。比如第一段曲线300个点，第二段曲

线200个点，则两段曲线制表完成后，D2+0（L）为501。配置完成后建议检查该参数，如果不正确请检查梯形图，但地址占用为502个双字。

5） 该制表指令主要用来便捷生成曲线表格，指令执行的结果只是对D2+0开始型值点个数的双字进行了一次赋值。如果需要用到圆弧曲线，正余弦曲线，渐开线等曲线，可以仿照该指令生成的型值点格式对后续的D寄存器进行赋值。比如规划一条正弦曲线，需要360个点，记录在从D2000开始的寄存器中，此时赋值D2000(L)为361，将曲线各个型值点处的位置手动填写至D2002(L)~D2722(L)中，即规划完成。上述方法规划出的曲线同样可以使用PTOUT指令或者CAM指令进行调用并输出曲线。

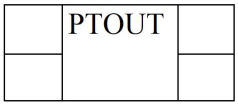
6） 伺服同时只能执行一条PTAB指令，当多段曲线生成时，需要注意各制表指令之间的时序逻辑。

3.4.13 制表输出[PTOUT]

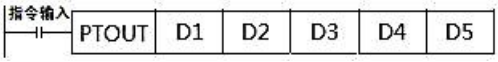
将已生成好的轨迹表进行输出(轨迹表可通过制表指令等多种方式生成)。

- **指令格式**

1) 指令格式



32位指令	指令记号	执行条件
16步	PTOUT	连续执行型



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 32 位
D2	轨迹表格起始单元	BIN 32 位
D3	主轴(横轴)实际数值	BIN 32 位
D4	主轴(横轴)实际量程	BIN 32 位
D5	输出方式	BIN 32 位

3) **对象软元件**

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●									
D2		●	●							
D3		●	●							
D4	●	●	●							
D5	●	●	●							

- **相关软元件**

1) D1 指定软件虚拟输出轴号

	设定范围
32 位运算	0：脉冲输出轴，Y0为脉冲端口，Y4位方向端口。 1：脉冲输出轴，Y1为脉冲端口，Y5位方向端口。 2：脉冲输出轴，Y2为脉冲端口，Y6位方向端口。 3：脉冲输出轴，Y3为脉冲端口，Y7位方向端口。

2) D2 指定总曲线轨迹表格的起始寄存器单元

建议为偶数地址。可以使用制表指令（PTAB、PCTAB、PBTAB）自动生成曲线轨迹表格，也可自行编程生成轨迹表格，但曲线轨迹表格的格式必须与指令生成的曲线轨迹表格保持一致。

3) D3 指定主轴当前实际值

建议为偶数地址。

●该值若对应时间轴，则表格以时间进行输出(单轴多段速轨迹表)。

推荐指定为D8008（32位数据）毫秒滴答定时器，即时间单位为ms。

●该值若对应脉冲轴，则表格关联主轴的位置进行输出(双轴关联凸轮表)。

主轴配置为外部脉冲轴时，即D3设置为外部脉冲计数值，R100（32位数据）（需要配置高速计数功能）。

主轴配置为脉冲输出轴时，即D3设置为脉冲输出轴（K0轴）相对位置，R318（32位数据）。

主轴配置为脉冲输出轴时，即D3设置为脉冲输出轴（K1轴）相对位置，R368（32位数据）。

此时PTOUT指令相当于输出一个指令啮合周期脱离的凸轮指令。但是没有表格比较事件输出与相位控制等功能，如需使用凸轮功能推荐使用CAM指令。

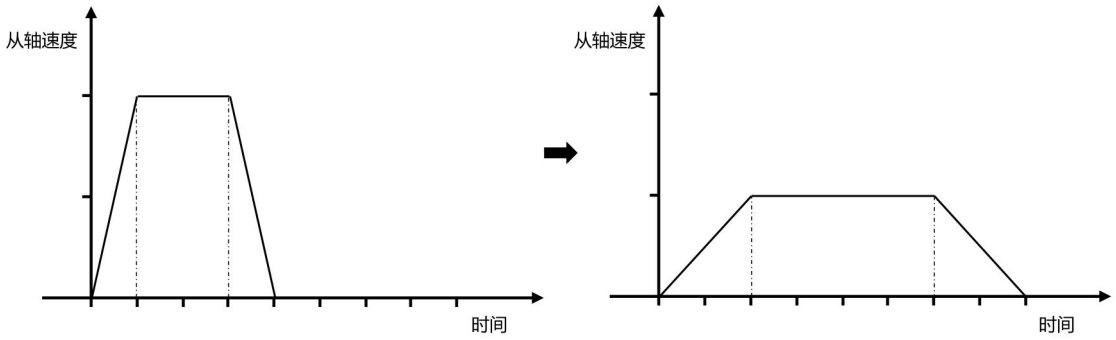
4) D4 指定主轴实际量程

该值为主轴一个运行周期的当量。例如主轴为脉冲轴，需要主轴发送100000个脉冲输出完整的表格曲线，则该值填写为100000；

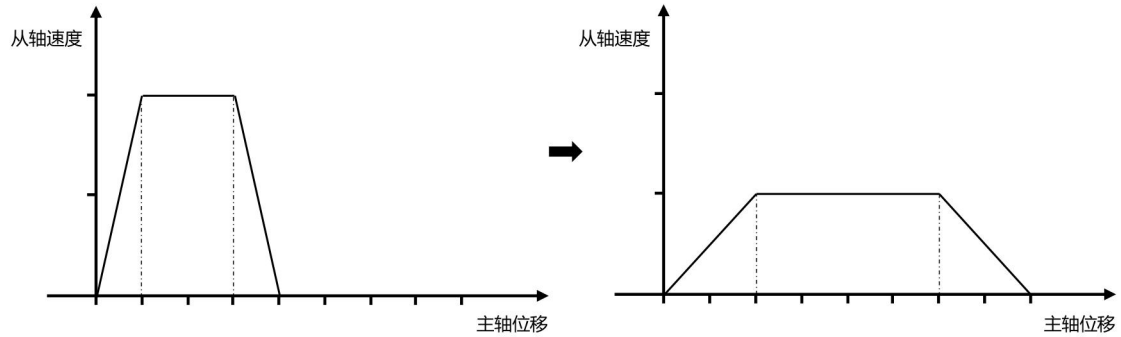
同理，如果主轴为时间轴，单位ms，当需要1000ms输出完表格曲线，则该值填写为1000。如果填写为2000，则需要2000ms输出完表格曲线，即从轴速度变为之前的一半。

当主轴实际量程的设置与实际不符时：

（1）当主轴指定为时间轴，若设置量程为实际量程的2倍时，则从轴需要两倍于之前的运行周期，方可输出完整的表格轨迹曲线，脉冲输出频率变为之前的一半。



（2）当主轴指定为脉冲轴，若设置量程为实际量程的2倍时，实际效果如下图：



由于一个运行周期内的从轴总脉冲数由曲线表格决定，所以从轴输出的总脉冲数不变，主轴的周期位移扩大一倍。所以从轴输出脉冲曲线会产生拉伸形变，整体速度变小。如果图中匀速段为同步区（具体概念见CAM凸轮指令），在拉伸之后将失去同步。当修改主轴周期之后还需要保持同步时，必须同时修改曲线表格。

引申：

当PTOUT指令输出选择位置轴作为主轴时，制得的曲线表格为从轴位置对应表格。主轴一个周期内的位移即为PTOUT指令中填写的D4（主轴实际量程），由制得曲线表格中的型值点决定从轴位置的变化过程。曲线表格的型值点数决定了从轴位置曲线的精细程度，与主轴周期位移无关。

为了方便理解，以下举一个特殊例子进行说明：

手动制作一个轨迹曲线表格，由3个型值点组成，则共占用5个双字D寄存器，以D2000为表格起点。

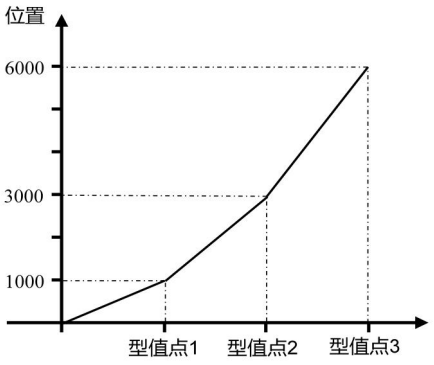
D2000(L)=4，表头（记录型值点个数，数量为实际型值点+固定的为0的位置起点）。

D2002(L)=0，位置表格起点，固定填0，若不为0则必须手动赋值。

D2004(L)=1000，第一个型值点。

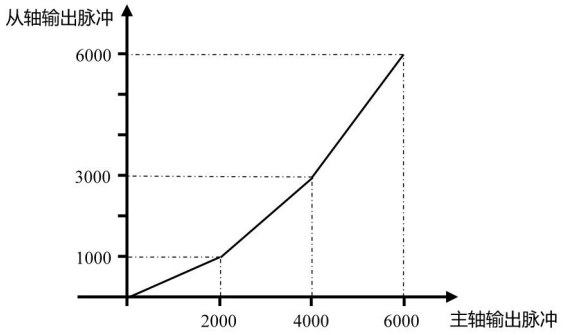
D2006(L)=3000，第二个型值点。

D2008(L)=6000，第三个型值点，制得曲线如下：



假设案例 1

如果D4主轴实际量程填写为6000，则从轴输出曲线如下图：



该曲线可分为三段：

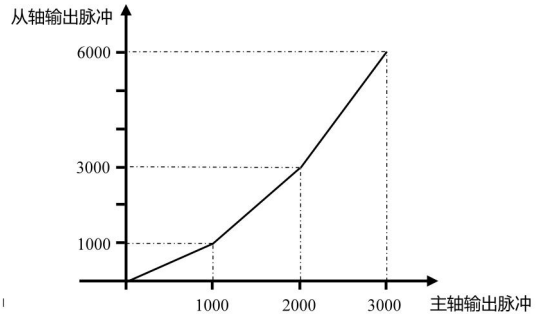
①第一段：在主轴脉冲数由0增加到2000的过程中，从轴跟随主轴以1：2的比例输出脉冲。直到主轴输出第2000个脉冲，此时从轴总共输出1000个脉冲，开始输出第二段曲线。

②第二段：在主轴脉冲数由2001增加到4000的过程中，从轴跟随主轴以1：1的比例输出脉冲。直到主轴输出第4000个脉冲，此时从轴总共输出2000个脉冲，开始输出第三段曲线。

③第三段：在主轴脉冲数由4001增加到6000的过程中，从轴跟随主轴以3：2的比例输出脉冲。直到主轴输出第6000个脉冲，此时从轴总共输出3000个脉冲，完成一个运行周期。

假设案例 2

如果D4主轴实际量程填写为3000，则从轴输出曲线如下图：



①主轴从第0个脉冲发送到第1000个脉冲，从轴从第0个脉冲发送到第1000个脉冲。速度比为1:1。

②主轴从第1000个脉冲发送到第2000个脉冲，从轴从第1000个脉冲发送到第3000个脉冲。速度比为2:1。

③主轴从第2000个脉冲发送到第3000个脉冲，从轴从第3000个脉冲发送到第6000个脉冲。速度比为3:1。

由此看出：曲线点的个数与主轴周期长度无关，只与D4主轴实际量程有关。曲线点的个数确定了主轴从轴对应关系可以细化成多少段位置对应的速度比例。增加曲线的点数会使曲线更加细化、平滑，符合实际机械运转规律。

以上主从轴关系在凸轮CAM指令中依然适用。

5）D5 指定输出模式

= 0，运行一个周期后结束，轨迹表输出完成后即停止输出。

= 1，循环输出，轨迹表输出完成后接着从第1个位置点(更新初始点)开始输出。

必须使用常开触点导通执行指令，断开触点则轴运动立即停止。

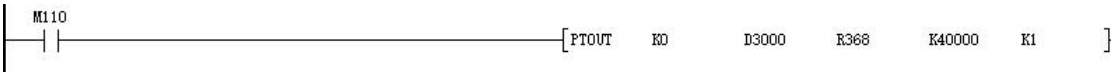
- **功能和举例**

PTOUT指令分为两种使用方法，单次输出表格内容与循环输出表格。循环输出表格功能可以用凸轮指令更好地替代，且功能更加强大，以下只做演示不做过多的用法说明。

使用表格输出指令需要首先制作表格，这里使用PTAB指令中的案例5制作的曲线作为案例进行表格输出。

说明案例1

需要配置的参数	定义	配置值
D1	软件虚拟输出轴号	0
D2	轨迹表格起始单元	D3000
D3	主轴(横轴)实际数值	R200
D4	主轴(横轴)实际量程	40000
D5	输出方式	1



- D1为K0：对虚拟轴轴0进行输出脉冲。
- D2为D3000：轨迹表格起始单元为D3000开始。实例中D3000为801，即曲线有800个点，占用D3000至D4603。
- D3为R200：跟随外部高速计数的计数值，即以外部脉冲输入为主轴。
- D4为K40000：外部每接收到40000个脉冲，即输出一个完整周期的轨迹曲线（由制表指令生成）。
- D5为1：循环输出模式。

PTAB 指令制表制得表格曲线分为4段：

① 第一段：四次多项式曲线，固定脉冲数为10000脉冲，正向加速定长段。

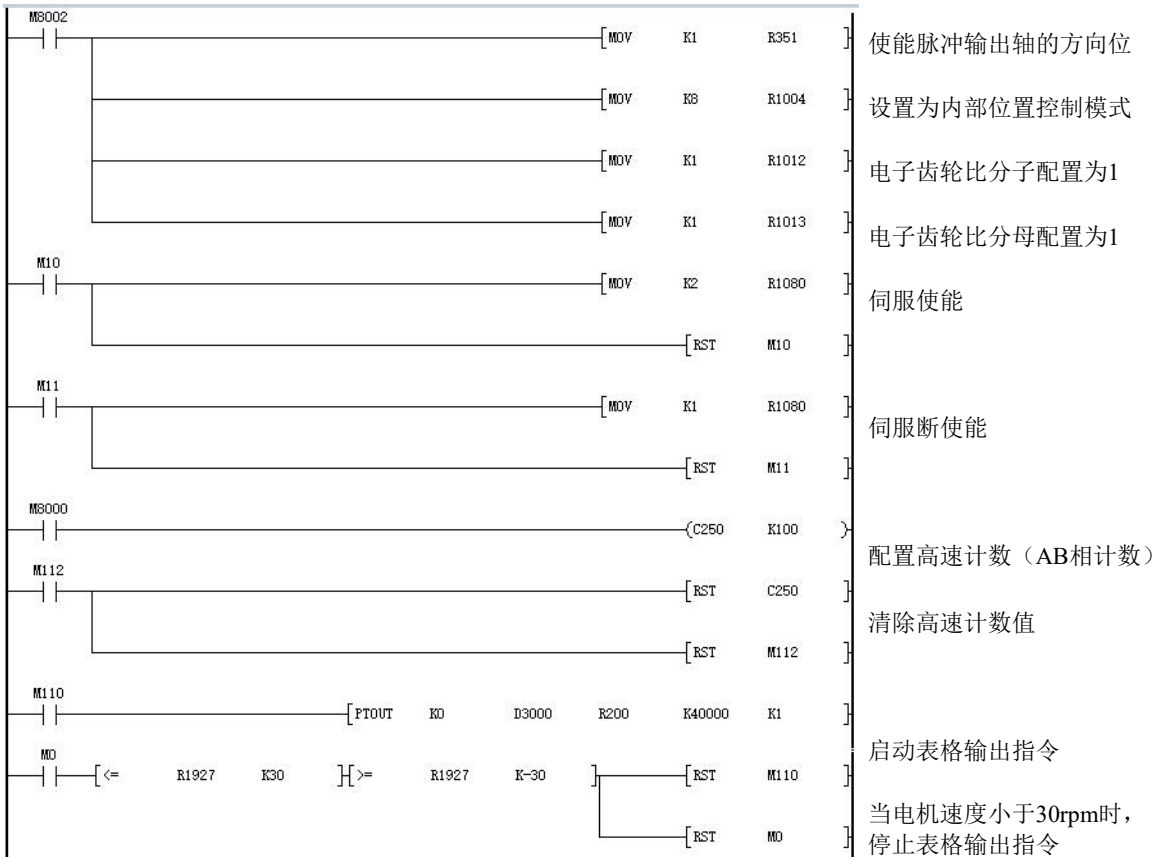
② 四次多项式曲线，固定末速度为0，正向减速段。

③ 匀加速曲线，固定脉冲数为10000脉冲，反向加速定长段。

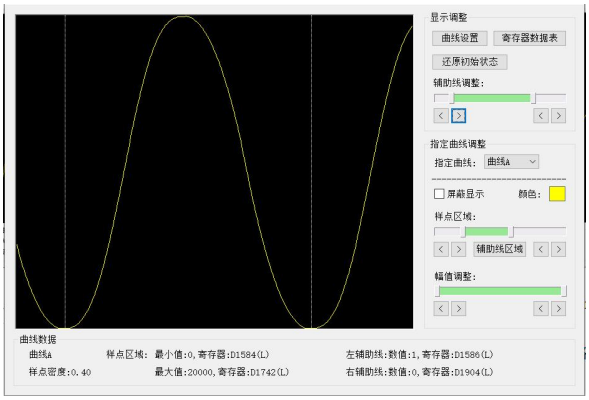
④ 匀加速曲线，固定末速度为0，反向减速段。

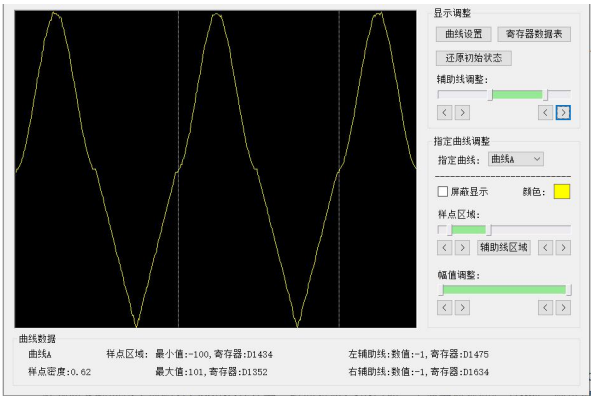
主轴配置为外部脉冲轴，从轴配置为伺服本体轴（K0），根据外部接收到的脉冲输出表格曲线。

PTAB制表指令参考本章中3.4.12 普通制表中说明案例5中示例梯形图，PTOUT指令示例如下：



表格曲线制表完成后，置位M110执行PTOUT指令。同时PLC的X0、X1端口输入12.5kHz（四倍频为50kHz）的AB相脉冲（编码器信号），PLC即控制电机以制得的表格曲线进行动作。使用示波器捕捉输出的指令脉冲的位移曲线和速度曲线如下：





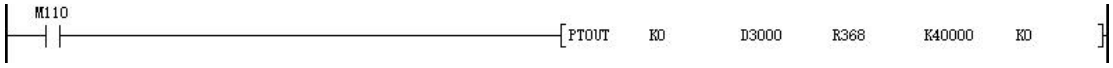
上图中两条辅助线内部的位移（速度）曲线为一个运行周期内的曲线，由速度曲线可明显看出输出脉冲的一个周期曲线由4段曲线组成，与制表指令所值表格曲线相符。

实际运行情况为：外部接收到恒定的50kHz脉冲，此时电机先正向旋转2圈，接着反向旋转2圈，即运行完成一个周期，接着循环运行。由于接收到脉冲的频率为50kHz，且电机运行一个周期所需脉冲数为40000个，所以运行一个周期用时为0.8s。

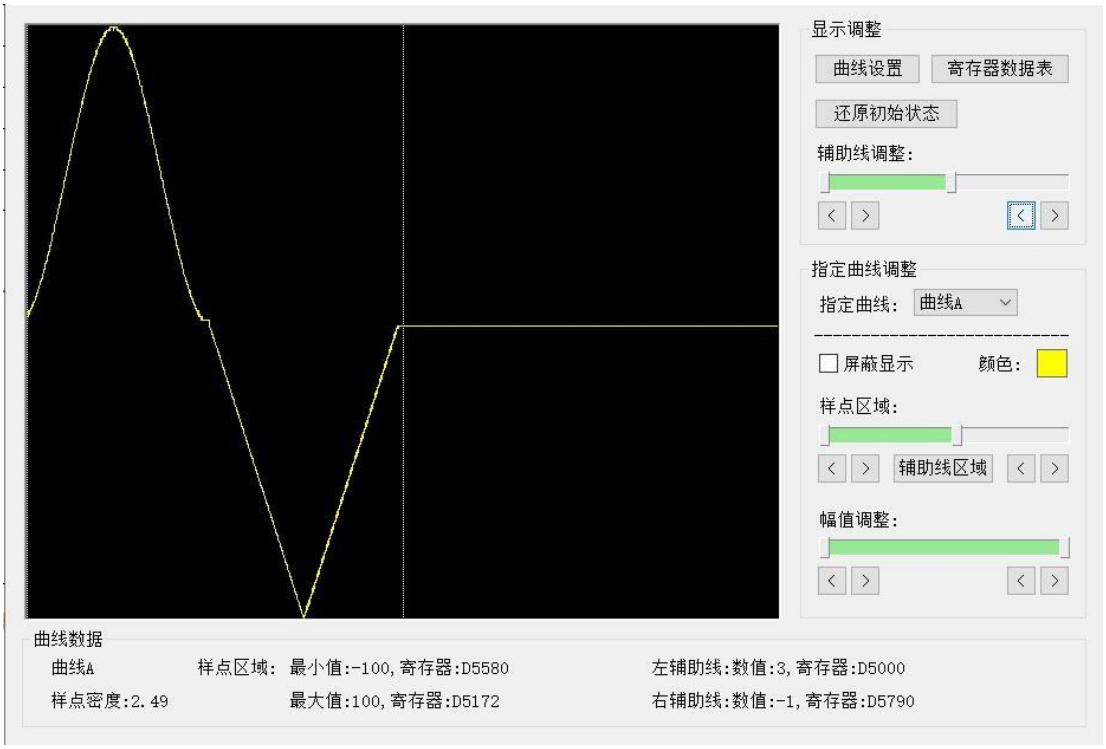
说明案例 2

其他设置与案例1相同，将输出方式改为0，即输出一个周期后停止。

需要配置的参数	定义	配置值
D1	软件虚拟输出轴号	0
D2	轨迹表格起始单元	D3000
D3	主轴(横轴)实际数值	R200
D4	主轴(横轴)实际量程	40000
D5	输出方式	0



操作方式与案例 1 相同。启动 PTOUT 之后采用示波器捕捉速度曲线。



表格周期输出完成一次后停止。

- **注意要点**

1) 该指令触点只要闭合，轴当前状态始终等于16，即运行状态。即便是单次输出模式也不可以通过轴状态来判断是否完成表格输出；

2) 该指令加速度不受轴最大加速度限制，但最大速度受轴最大速度限制。如果随意缩放主轴实际量程可能造成速度过大而轴超速报警；

3) 该指令主要功能为用于表格多段速输出，使用周期循环来运行表格建议采用凸轮指令；

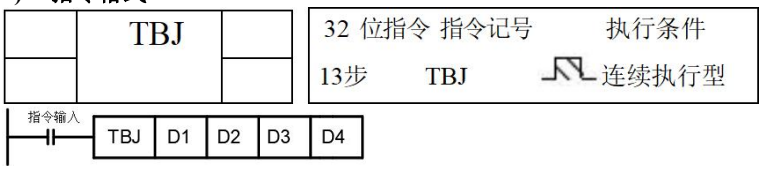
4) 主轴在指令输出过程中速度可以发生改变，该指令会根据主轴速度改变输出速度，因此尽量避免主轴速度反向。

3.4.14 贴标机[TBJ]

对指定软件虚拟输出轴(AXV)按照指定频率(带加减速)输出无限个数的脉冲数(带方向)。当遇到中断事件后，再输出一定数量的脉冲后减速停止。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	脉冲数	BIN 32 位
D2	脉冲频率	BIN 32 位
D3	输入参数配置缓存区起始寄存器单元	BIN 16 位
D4	软件虚拟输出轴号	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●	●								
D2	●	●								
D3		●								
D4	●									

- **相关软元件**

1)D1 指定输出脉冲数

	设定范围
32 位运算	-16000000~16000000

负值为反转，方向端口为OFF，正值为正转，方向端口为ON。若需要反向，则需要将脉冲方向端口使能。

2)D2 指定输出的脉冲频率

	设定范围（单位：hz）
32 位运算	0~200000

设置为10000，则脉冲输出频率为10kHz，脉冲频率必须填写正数。

D2的设置如果小于轴起始速度，那么实际的目标速度为轴起始速度。

D2的设置大于轴最高限速时，实际的目标速度不会超过轴最高限速 。无论是正向还是负向发脉冲，该值皆为正数，是速度的绝对值。

3）在D3中指定输入参数配置缓存区起始寄存器单元，建议为偶数地址

指令执行时，D3指向寄存器后的连续3个寄存器均被占用即以D3为第1个，一共3个D寄存器被占用，作为TBJ指令配置的参数设置。

比如D3填D1000，则D1000到D1002被占用。该参数可在线更改,当D1002为K100时，指令完成一个贴标动作后，

M100会置位ON，可梯形图清除标志位。

表格具体内容如下：

TBJ指令配置表	定义	备注	单位	数据类型
D3+0	物料延时	物料信号碰到后延时时间	0.2ms	BIN 16 位
D3+1	标签信号延时	标签信号碰到后延时时间	0.2ms	BIN 16 位
D3+2	指令完成标志位	标签贴完后指令完成标志置ON	/	BIN 16 位

4)D4 指定软件虚拟输出轴号

	设定范围
32 位运算	0~3

5)相关轴参数列表：

软元件编号				名称	数据类型	初始值
虚拟0 轴	虚拟1 轴	虚拟2 轴	虚拟3轴			
R302	R352	R402	R452	轴时间当量	16 位	1000
R305	R355	R405	R455	轴最高限速	16 位	20000
R306	R356	R406	R456	轴起始速度	16 位	0
R307	R357	R407	R457	轴最大加速度	16 位	1000
R323	R373	R423	R473	轴中断信号指定物料或标签（只能指定为X4-X7）	16 位	0
R324	R374	R424	R474	轴中断信号指定物料或标签（只能指定为X4-X7）	16位	0
R314 Bit10	R364 Bit10	R414 Bit10	R464 Bit10	物料或标签信号极性 =0 常开 =1 常闭	位	0
R314 Bit11	R364 Bit11	R414 Bit11	R464 Bit11	物料或标签字信号使能 =0 不使能 =1 使能	位	0
R314 Bit12	R364 Bit12	R414 Bit12	R464 Bit12	/	位	0
R314 Bit13	R364 Bit13	R414 Bit13	R464 Bit13	输入信号源选择 =0 指向M =1 指向X	位	0
R310	R360	R410	R460	/	16位	0
R314 Bit2	R364 Bit2	R414 Bit2	R464 Bit2	/	位	0
R314 Bit6	R364 Bit6	R414 Bit6	R464 Bit6	/	位	0

R311	R361	R411	R461	/	16位	0
R314 Bit3	R364 Bit3	R414 Bit3	R464 Bit3	/	位	0
R314 Bit7	R364 Bit7	R414 Bit7	R464 Bit7	/	位	0
R314 Bit8	R364 Bit8	R414 Bit8	R464 Bit8	中断信号极性	位	0
R314 Bit9	R364 Bit9	R414 Bit9	R464 Bit9	中断信号使能	位	0

- 起始速度由轴参数轴起始速度指定，加速度为“轴最大加减速”所设定的数值。
- X4-X7信号对应内部寄存器M信号，电平信号M8096-M8099,上升沿信号M8100-M8103,下降沿信号对应M8108-M8111。沿信号会自动清除，监控无法观察到。

- **功能和举例**

说明案例 1：

使用TBJ 指令设定从 0kHz 加速至 1kHz，在接到物料信号X4后延时延动1000*0.2ms,使虚拟轴 0 发出脉冲,再碰到标签信号X5后走1000个脉冲后停止运行，标志位M100置位。默认轴最大加速度，轴最高限速。中断信号指定X4，常开信号。上电第一个扫描周期必须把R999给1，否则TBJ指令不生效。

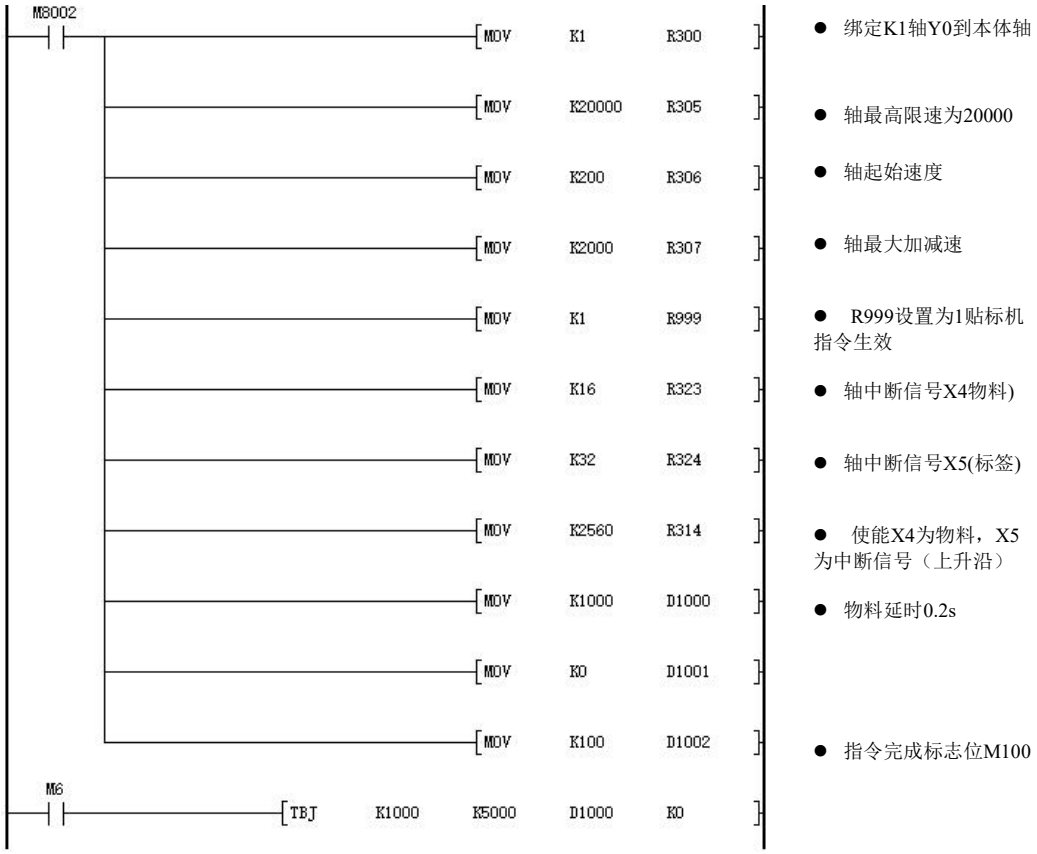
事件指定方法如下表：

0	不选定端口
16	X4/M8100
32	X5/M8101
64	X6/M8102
128	X7/M8103

需要配置的参数	定义	配置值
D1	脉冲数	1000
D2	脉冲频率	1000
D3	输入参数配置缓存区起始寄存器单元	D1000
D3+0	物料延时	1000
D3+1	标签信号延时	0
D3+2	指令完成标志位	100
D3	软件虚拟输出轴号	0
R305	轴最高限速	20000（默认）
R306	起始速度	200
R307	轴最大加速度	1000（默认）
R323	物料轴中断信号指定X10	16
R324	标签轴中断信号指定X11	32
R314Bit8	中断信号极性	0
R314Bit9	中断信号使能	1
R314Bit13	输入信号源选择	1

R314 0010 0010 0000 0000，转化为 16 进制：H2200

示例梯形图：



注意要点

1) 上电R999设置为1贴标机指令才生效，只检测第一个扫描周期。
2) 速度参数配置时，最好遵守轴起始速度＜设定脉冲频率＜轴最高限速。
3) 如果设定脉冲频率小于轴起始速度，则指令以轴起始速度匀速发完设定脉冲数，无加减速段。
4) 设定脉冲频率大于轴最高限速时以轴最高限速运行。
5) 加减速过程以轴最大加速度运行。
6) 只能在轴状态为 0 时开始运行。
7) 配置轴只能为0轴或1轴，当两个轴绑定到Y0发脉冲时，注意0轴加速度不能设置3000.
8) 指定脉冲数绝对值不得大于 16000000，否则无法进行中断定位。
9) 梯形图配置了TBJ指令后，伺服驱动器只能使用0轴或1轴。

3.4.15 停止输出[MSTOP]

指定软件虚拟输出轴(AXV)停止输出。

- **指令格式**

4） 指 令 格 式



16 位指令 指令记号 执行条件
3 步 MSTOP 连续执行型



5） 设 定 数 据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 16 位

6） 对 象 软 元 件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•									

- **相关软元件**

该指令作用于下表寄存器：

软元件编号				名称	数据类型	初始值	属性
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴				
R300	R350	R400	R450	软件虚拟轴输出绑定	16 位	0/1/2/3	上电写入
R301	R351	R401	R451	脉冲输出方向使能	16 位	0	上电写入
R318	R368	R418	R468	相对位置脉冲数	32 位	0	读写
R320	R370	R420	R470	轴状态	16 位	0	只读
R321	R371	R421	R471	错误代码	16 位	0	只读

- **功能和举例**

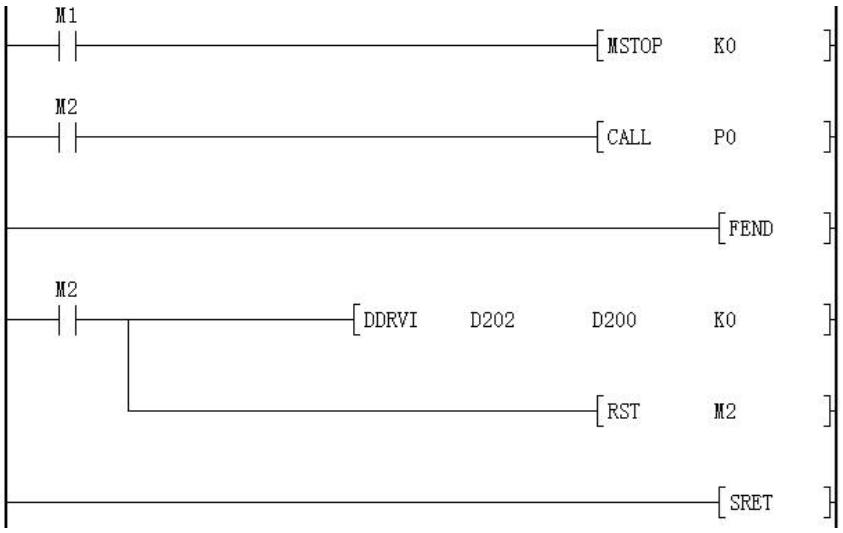
该指令使软件虚拟输出轴(AXV)停止输出。



4) 在D1 中指定软件虚拟输出轴

	设定范围
16 位运算	0~3

5) 主要针对子程序中启动了脉冲输出指令，而子程序已经退出无法进去，通过MSTOP指令可停止子程序中脉冲输出指令。当然也可停止主程序中脉冲输出指令。如下梯形图：


M1
MSTOP K0
M2
CALL P0
FEND
M2
DDRVI D202 D200 K0
RST M2
SRET

3.5 单轴关联指令

主要介绍电子齿轮、电子凸轮、追剪正程制表以及追剪返程制表指令的功能和用法

3.5.1 电子齿轮[GEAR]

对指定软件虚拟输出轴(AXV)做从轴，与时间/主轴位置成比例关系(电子齿轮)，比例换算后跟随主轴运动。

1) **指令格式**



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号(从轴)	BIN 16/32位
D2	主轴关联寄存器单元	BIN 16/32位
D3	主轴关联寄存器单元变化方式	BIN 16/32位
D4	电子齿轮分子	BIN 16/32位
D5	电子齿轮分母	BIN 16/32位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●									
D2		●	●							
D3	●	●	●							
D4	●	●	●							
D5	●	●	●							

- **功能与动作**

1) D1 指定软件虚拟输出轴号

	设定范围
32 位运算	0：脉冲输出轴，Y0为脉冲端口，Y4位方向端口。 1：脉冲输出轴，Y1为脉冲端口，Y5位方向端口。 2：脉冲输出轴，Y2为脉冲端口，Y6位方向端口。 3：脉冲输出轴，Y3为脉冲端口，Y7位方向端口。

2) D2 指定主轴关联寄存器单元

建议为偶数地址。具体设置见主轴关联参考匹配

3) D3 指定主轴关联寄存器单元的变化方式。

a. 为16位指令时：

◆＝0，累积型变化。主轴为单字16位数据，主轴单元增量×电子齿轮比产生从轴(输出轴)的插补量。适用于累积型主轴，例如软件虚拟输出轴轴参数中的轴相对位置（R318/R368）。

◆＝1，增量式变化。主轴为单字16位数据，主轴单元×电子齿轮比产生从轴(输出轴)的插补量。适用于增量型主轴，可设为任意普通寄存器用来指定每毫秒发送脉冲数。

b. 为32位指令时：

◆＝0，累积型变化。主轴为双字32位数据，主轴单元增量×电子齿轮比产生从轴(输出轴)的插补量。适用于累积型主轴，例如软件虚拟输出轴轴参数中的轴相对位置（R318/R368）。

◆＝1，增量式变化。主轴为双字32位数据，主轴单元×电子齿轮比产生从轴(输出轴)的插补量。适用于增量型主轴，可设为任意普通寄存器用来指定每毫秒发送脉冲数。

◆＝2，累积型变化。主轴为单字16位数据，主轴单元增量×电子齿轮比产生从轴(输出轴)的插补量。适用于累积型主轴，例如软件虚拟输出轴轴参数中的轴相对位置（R318/R368）。

◆＝3，增量式变化。主轴为单字16位数据，主轴单元×电子齿轮比产生从轴(输出轴)的插补量。适用于增量型主轴，可设为任意普通寄存器用来指定每毫秒发送脉冲数。

4) D4 指定电子齿轮分子

5) D5 指定电子齿轮分母

当D5=0 时，电子齿轮比视作1处理。

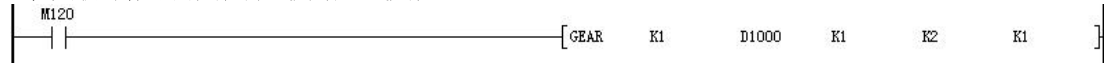
◆ 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。可在指令导通情况下修改电子齿轮分子或分母，在下一个毫秒周期生效。

◆ 电子齿轮数值若为负数，则从轴脉冲输出方向与主轴相反。

主轴关联参考匹配：

常见主轴关联	操作数 D2	操作数 D3
软件虚拟输出轴脉冲给定	轴参数 轴相对位置地址	0
软件虚拟输出轴脉冲给定增量	轴参数 轴位置插补增量地址	1
系统 1ms 滴答定时器	D8008	0

1) 主轴关联寄存器的变化方式为增量式变化



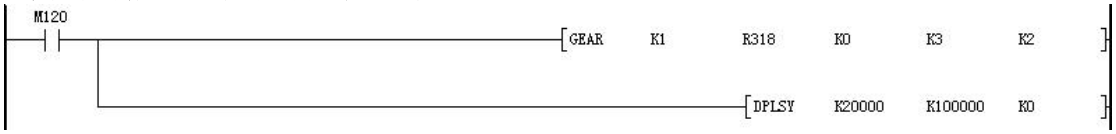
◆ 脉冲输出对象为软件虚拟输出轴 1。

◆ 以普通寄存器D1000 主轴关联单元。

◆ D1000为增量式变化单元(假定)，电子齿轮比为 2:1。

◆ 假定软件虚拟输出轴 1 绑定 Y1脉冲输出口，则 Y1 脉冲输出频率为D1000数值的 2 倍。

2) 主轴关联寄存器的变化方式为累积式变化



◆ 脉冲输出对象为软件虚拟输出轴 1。

◆ 以软件虚拟输出轴0的轴参数轴相对位置（R318）为主轴关联单元。

◆ 轴参数轴相对位置为累积式变化单元。

◆ 电子齿轮比为3:2

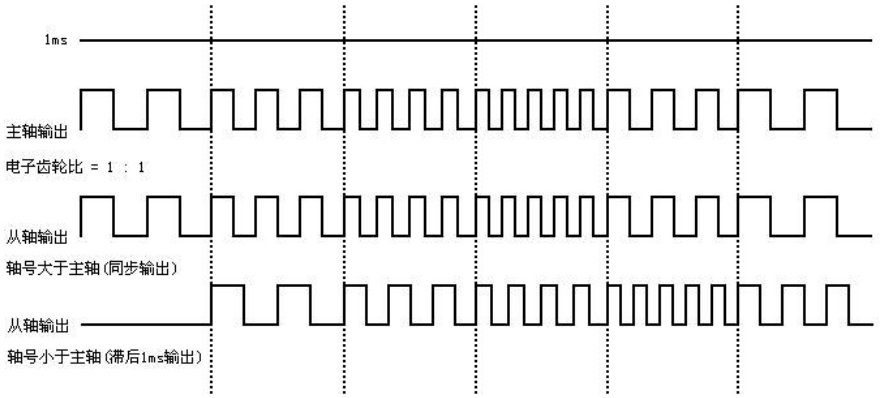
◆ DPLSY指令：虚拟输出轴0以20kHz的频率，发送100000个脉冲

◆ 假定软件虚拟输出轴1绑定Y1脉冲输出口，软件虚拟输出轴0绑定Y0脉冲输出口，则Y1输出脉冲为Y0输出脉冲数的1.5倍频(滞后1个插补周期)，即以30kHz的频率输出150000个脉冲数。

每个插补周期到来时(1ms 中断进入)，系统处理软件虚拟输出轴遵循由小到大的顺序进行脉冲插补处理和更新。假定主轴绑定对应的软件虚拟输出轴号为AxisM，从轴对应软件虚拟输出轴号(即D1对应数值)为AxisS。

a. AxisM < AxisS，系统先处理主轴插补计算，再处理从轴插补计算，能够保证同插补周期输出。

b. AxisM > AxisS，系统先处理从轴插补计算，此时主轴所有参数仍处于上一个插补周期的状态，故从轴将滞后 1 个插补周期输出。



- **注意事项：**

◆ 指令执行异常的情况有(指令不动作或动作错误)：

1） 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。

2） 设置电子齿轮的数值不合理。

3） 分子/分母设置为0

4） 电子齿轮数值太大(从轴输出变化剧烈)造成抖动。

5） 电子齿轮数值为负数，且从轴绑定Y 0脉冲端口但未绑定Y1方向端口。

6） 设置主轴单元寄存器序号为非偶数地址。

7） 设置主轴单元变化方式与主轴单元实际变化数值的关系不对应。比如主轴单元填写的是累积型变化数值，但变化方式填增量型。

8） 轴速度设置不合理导致轴参数轴错误代码报错。

9） 在对应轴状态不为0的情况下执行指令。

◆ 在线修改电子齿轮比时，如果从正电子齿轮比修改到负电子齿轮比，或由负到正，系统换向会停发1周期的脉冲，累积到下一周期补发。故容易造成超速报警，需谨慎使用。

◆ 上电PLC运行时Y0～Y7端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应Y软元件不再关联实际端口。

3.5.2 手摇轮[HAND]

对指定软件虚拟输出轴(AXV)做从轴，与主轴位置成比例关系(电子齿轮)，比例换算后跟随主轴运动。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	输入参数配置缓存区起始寄存器单元	BIN 16 位
D2	主轴关联寄存器单元	BIN 16/32 位
D3	软件虚拟输出轴号(从轴)	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		●								
D2			●	●						
D3	●									

- **相关软元件**

1) 在D1 中指定输入参数配置缓存区起始寄存器单元，建议为偶数地址

指令执行时，D1指向寄存器后的连续10个寄存器均被占用即以D1为第1个，一共10个D寄存器被占用，作为HAND指令配置的参数设置。

比如D1填D1000，则D1000到D1009被占用。该参数需要在执行HAND指令之前完成配置。

表格具体内容如下：

HAND指令配置表	定义	备注	单位	数据类型
D1+0	电子齿轮分子	主轴（手摇轮）与从轴电子齿轮分子	/	BIN 16 位
D1+1	电子齿轮分母	主轴与从轴电子齿轮分母	/	BIN 16 位
D1+2	从轴加（减）速时间	建议设200,从轴从0速到运行最高限的加速与减速时间	MS	BIN 16 位

D1+3	从轴响应时间	建议设20，主轴采样时间	MS	BIN 16 位
D1+4(L)	从轴最高频率	建议200000，主轴与从轴的电子齿轮比很大或者从轴跟随频率大于此频率时以此速度为准	HZ	BIN32位
D1+6(L)	负限位	赋值 H80000000 表示不启用此功能	脉冲数	BIN32 位
D1+8(L)	正限位	赋值 H7FFFFFFF 表示不启用此功能	脉冲数	BIN 32 位

2) 在D2 中指定主轴关联寄存器单元

具体设置见主轴关联参考匹配,如主轴为脉冲可填写虚拟轴（0-3）对应脉冲数寄存器，一般填写K1轴(R368），如主轴为编码器对应关联C250，仅支持 X0,X1 通道。

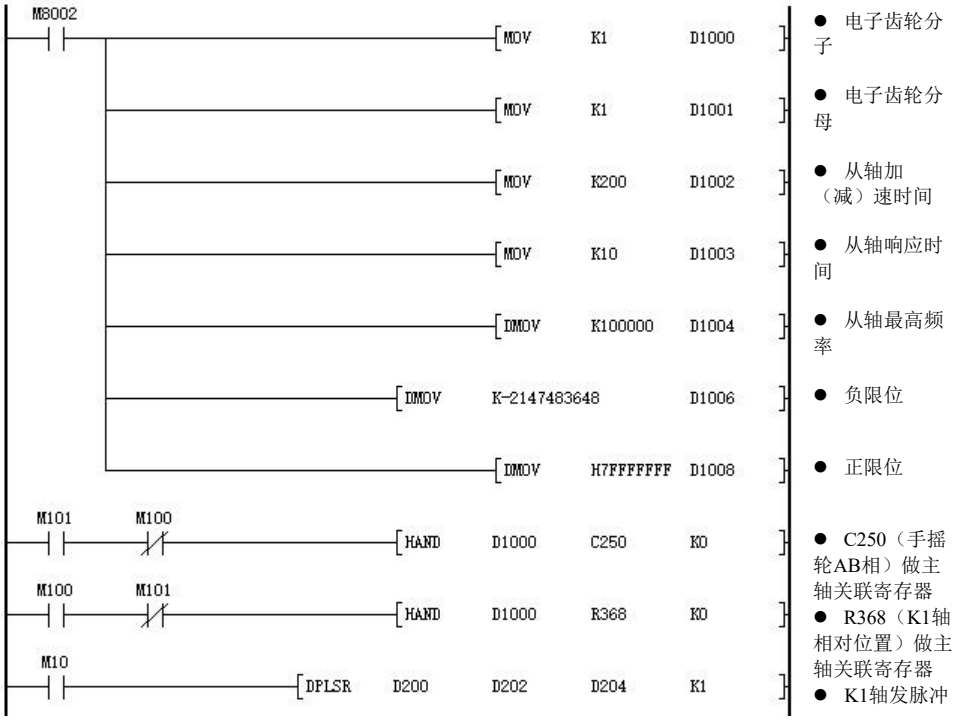
3) 在 D3 中指定软件虚拟输出轴号（从轴）

	设定范围
16/32 位运算	0：脉冲输出轴，Y0为脉冲端口，Y4位方向端口。 1：脉冲输出轴，Y1为脉冲端口，Y5位方向端口。 2：脉冲输出轴，Y2为脉冲端口，Y6位方向端口。 3：脉冲输出轴，Y3为脉冲端口，Y7位方向端口。

主轴关联参考匹配：

常见主轴关联	操作数 D2主轴关联寄存器单元
软件虚拟输出轴脉冲给定	轴参数 轴相对位置地址（R368L)
虚拟输入轴接入的脉冲计数绝对值	轴参数 轴位置高速计数地址 C250

1) 主轴关联寄存器单元



◆ 主轴关联高速计数C250(手摇轮AB相）或虚拟轴K1轴（R368)，从轴为虚拟轴K0轴，具体伺服用手摇轮（AB相计数）与虚拟轴K1发脉冲参考相关计数与脉冲部分的说明书。

◆ 根据以上梯形图，先置位M101,伺服相关参数设为AB相计数，伺服设为外部脉冲位置控制模式。

◆ 输入参数D1000-D1008相关参数可以根据现场需求设置。

◆ C250脉冲数与R318L的脉冲数发完后必须相同,或者脉冲数为电子齿轮比例关系。

● **注意事项：**

◆ 指令执行异常的情况有(指令不动作或动作错误)：

1） 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。

2） 设置电子齿轮的数值不合理。

3） 设置主轴单元寄存器序号为非偶数地址。

4） 轴速度设置不合理导致轴参数轴错误代码报错。

5） 在对应轴状态不为0的情况下执行指令。

6） 当从轴位置不在正负限位区间时，启动HAND指令，指令不运行。

7） 当负限位设置>正限位时，启动HAND指令从轴会报错。当正负限位相同时，HAND指令执行、主轴运行时，从轴不会运行也不报错。

◆ 不能在线修改电子齿轮比及相关加减速时间**[D1+0~D1+8]**，改后必须断开HAND指令后再运行才能生效。

◆ 上电PLC运行时Y0～Y7端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应Y软元件不再关联实际端口。

3.5.3 追剪正程/滚切制表[PCTAB]

制作追剪正程/滚切的轨迹曲线表。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	曲线参数配置缓存区起始单元	BIN 32 位
D2	轨迹表格起始单元	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		●	●							
D2		●	●							

- **功能与动作**

1) D1 指定追剪正程/滚切曲线的参数配置缓存区起始寄存器单元

建议为偶数地址。指令执行时，D1 指向寄存器后的连续 16 个寄存器均被占用，这些寄存器应在指令执行前配置好且只配置 1 次(配置方法后述)，指令执行过程中这些寄存器将作为计算缓存区使用，指令导通条件断开后，这些寄存器将被释放。

例：D1填写D1000，则D1000～D1015 在指令执行期间均被占用。

2) D2 指定总曲线轨迹表格的起始寄存器单元

建议为偶数地址。D2指向的寄存器单元(轨迹表点数)，与D2+2 指向的寄存器单元(轨迹表初始位置值)均在执行指令时由系统初始化，用户可以无需配置。

无论是追剪还是滚切的运动模型，都可以理解为电子凸轮的一种应用。追剪的一个工艺流程周期包括正程 (等待→加速追击→同步剪切→减速停止)和返程组成。若不考虑返程流程，则与滚切的工作原理基本一致。故追剪正程/滚切制表指令适用于此类应用。

- **追剪与滚切异同**

追剪	滚切
相同点	
● 应用于定长/色标切割，根据物料材质决定选择追剪还是滚切。 ● 主轴为送料轴(一般保持匀速)，从轴控制追剪轴或滚切轴。 ● 在切割时必须与主轴速度保持一致。	
不同点	

送料轴保持匀速单向送料，刀架伺服电机在切割台上 做往复运动。在同步区域内与送料轴速度达到同步，同时完成切割动作，然后反向回到原点，再次追踪同步切割。	滚切轴为圆周形状，始终往同一方向旋转做圆周运动， 利用轴面凸出刀口部分接触物料的时间点(同步区域) 时进行剪切，期间送料轴可不间断持续等速送料。
需要控制从轴方向，剪切完毕后回程换向。	无需控制从轴方向，始终同向运动。
切刀需要程序控制，进入同步区后控制切刀动作,执行切割动作。	切刀安装在凸轮辊轴表面凸出部分，每旋转一周刀口与物料接触一次，切刀无需程序干预。
物料切割时间长，材质较硬，如木板，管材等。	物料切割时间短，轻薄材料，如纸张，塑料袋等。

追剪正程/滚切的运动区域可划分为同步区和非同步区：

1） 同步区完成工艺操作，需与主轴速度保持一致。比如在切刀接触到物料时一定要与物料速度同步，如果接触时切刀速度大于同步速度，出现对物料一个向前牵扯的力，会照成物料切面不平；如果速度低于物料速度，会出现堵料的现象。

2） 非同步区主要保障设备运行平稳，以最小的加速度，最小的变速范围为设计条件。根据实际工艺情况，非同步区又可由加减速区和等待区组合而成。

3） 同步区的规划会影响到实际设备的运行，一个工艺周期中若同步区越大，加减速的时间就越小，表示设备需要在短时间内进行加减速，对于电机、机台、切刀的冲击都很大，而且容易导致伺服过流报警，设备无法正常运行。

4） 一个追剪正程/滚切轨迹点表通常情况下由五段曲线组成，生成的曲线型值点位置变化规律是这些曲线的走向反映。

曲线段	作用区域
起始停止段	等待区，根据配置参数决定是否存在该区域。
起始加速/减速段	加速/减速区，根据配置参数决定是否为加速或减速。
匀速段	同步区，与主轴速度一致。
终止加速/减速段	减速/加速区，根据配置参数决定是否为减速或加速。
终止停止段	等待区，根据配置参数决定是否存在该区域。

- **轨迹表定义**

同通用制表指令不同，追剪/滚切正程制表指令可以直接生成整个凸轮曲线(不包含追剪返程曲线)，而不需要执行多次制表生成。由用户指定表头寄存器地址(操作数 D2)。表内型值点数据由系统根据曲线配置参数自动计算，需留足表格数据存放的寄存器区间。

曲线轨迹表内容	内容解释	寄存器示例
表头 - (指定偶地址) 型值点总数(轨迹表量程)	无需用户初始化。调用指令后由系统填 1，计算完成后系统再次更新。	D5000(L)
初始位置点	无需用户初始化。调用指令后系统填 0。	D5002(L)
曲线型值点 1	指令计算生成。	D5004(L)
...		假定 500 个点
曲线型值点 n		D6002(L)
表格总点数和表内型值点的数据类型均为 32 位，占用 2 个寄存器。 表格占用总寄存器个数 = (表头)2 + 型值点总数 * 2		

- **曲线参数定义**

曲线配置参数缓存区的定义如下表所示。

地址后带(L)标识的参数为32位参数，占用2个D寄存器。

D1+偏移	名称	参数描述	单位
D1+0(L)	主轴周期位移	指定一个工艺周期内主轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 ＞0。	脉冲
D1+2(L)	主轴同步区位移	指定一个工艺周期内主轴同步区所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 ＞0 且≤主轴周期位移。	脉冲
D1+4(L)	从轴周期位移	指定一个工艺周期内从轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 ＞0。	脉冲
D1+6(L)	从轴同步区位移	指定一个工艺周期内从轴同步区所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 ＞0 且≤从轴周期位移。	脉冲
D1+8	周期最小时间	指定追剪正程/滚切曲线 1 个凸轮周期的最小时间。 周期最小时间 ÷ 表格时间间隔 = 曲线的型值点点数 ● 作为追剪正程曲线规划，周期最小时间为主轴最大速情况下正程的追击、同步、 脱离时间之和，返程时间在追剪返程制表中设定。 ● 作为滚切曲线规划，周期最小时间为主轴最大速情况下整个凸轮周期的时间。 ● 主轴最大速度由参数主轴最高限速确认，系统计算得出。 ● 设置该参数 ＞0。	ms
D1+9	表格时间间隔	指定追剪正程/滚切曲线的表格时间间隔。 ● 间隔要选取合适的范围，过大则降低曲线细分精度，过小则会造成曲线生成点数过多，使得寄存器开销变大。 ● 设置该参数 ＞0，且能被周期最小时间整除。	ms

D1+偏移	名称	参数描述	单位
D1+10	从轴最大速度	从轴运行最大速度。 例如设置速度为 100 kHz，即每 ms 发送 100 个脉冲，假定其计量单位 =1 脉冲/ms，则从轴最大速度设置 100。计量单位参考参数时间当量和脉冲当量描述。 ● 设置该参数 ＞0。	脉冲当量 / 时间当量
D1+11	从轴最大加减速	从轴运行最大加/减速度。 例如设置加速度为 1 脉冲/ms²,假定其计量单位 =0.1 脉冲/ms²，则从轴最大加减速设置 10。计量单位参考参数时间当量和脉冲当量描述。 ● 设置该参数 ＞0。	脉冲当量 / 时间当量 ²
D1+12	时间当量	曲线时间当量，与脉冲当量耦合计算，为速度/加减速度的计量单位之一。	ms
D1+13	脉冲当量	曲线脉冲当量，与时间当量耦合计算，为速度/加减速度的计量单位之一。 例如时间当量 =10，脉冲当量 =10，则速度单位 =10 ÷ 10 = 1 脉冲/ms， 加减速度单位 =10 ÷10² = 0.1 脉冲/ms² 。	脉冲

D1+14	制表状态	制表指令当前状态，系统同一时间只允许 1 条制表指令处于运行状态。 ● = 0：当前无制表指令运行，可执行制表指令 ● = 16：正在生成曲线数据(系统每 1ms 生成 1 个型值点数据) ● = 32：曲线数据生成完毕 ● = 64：曲线生成错误 当检测到制表状态等于 32 时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清 0。 ● 该参数无需设置，为显示值。	-
D1+15	主轴最高限速	主轴最高限速，制表完成后由系统更新。 根据曲线设置参数不同，系统求得的主轴最高速也不相同。当工艺速度小于主轴最高限速时，满足设计要求。 ● 对于追剪制表而言，正程制表生成的主轴最高限速为中间值，执行返程制表指 令后的主轴最高限速将会再次更新。 ● 该参数无需设置，为显示值。	脉冲当量 / 时间当量

- **主轴最高限速换算**

对于追剪/滚切系统，机台的速度一般而言以“米/分”为单位。其规定的最大限速需与制表完成后系统生成的主轴最高限速进行比较，若机台的速度大于系统生成的主轴最高限速，则需要重新调整曲线参数以满足设计要求。

主轴最高限速的单位与曲线参数中的脉冲当量与时间当量有关，比较时需要进行量纲转换。

示例：假定机台的最大速度为 30 米/分，脉冲当量为 10 脉冲/mm。那么机台的最大速度 = 30 * 1000 / (60 * 1000) = 0.5 mm/ms，根据脉冲当量可算出得出系统的最大速度为5pls/ms，。

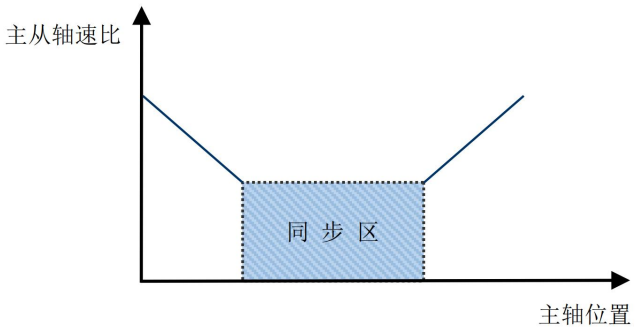
假定曲线的脉冲当量参数设置为10，曲线的时间当量参数设置为10，则生成曲线的速度单位为1pls/ms。那么系统生成的主轴最高限速数值应大于等于5，即5pls/ms。

- **凸轮曲线轨迹分析**

主从轴周期位移量和同步区位移量的设置决定了凸轮曲线的输出方式。

假定从轴同步区位移÷从轴周期位移 = $K_{从}$，主轴同步区位移÷主轴周期位移=$K_{主}$，以下是$K_{从}$与$K_{主}$在不同比例关系下的凸轮曲线轨迹：

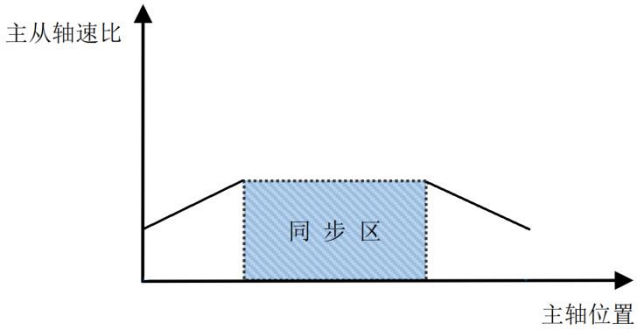
1） $K_{从} < K_{主}$：在同步区从轴速度与主轴速度同步，过了同步区后，为了赶上下一个凸轮周期，则从轴加速。



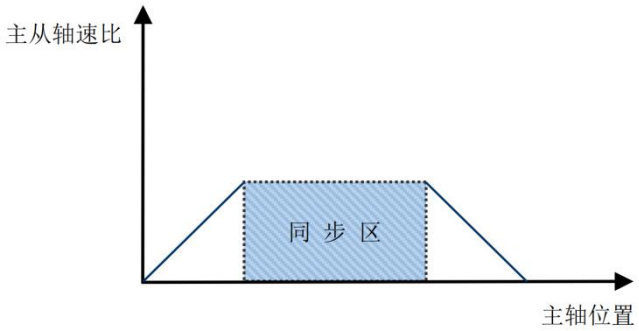
2） $K_{从} = K_{主}$：从轴与主轴同速比运动。



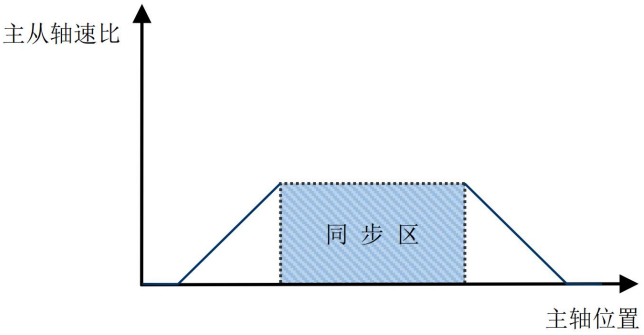
3） $K_{从} > K_{主}$ 且 $K = 2K_{主} - K_{从} > 1$；同步区结束后从轴减速，在下一个凸轮周期再加速到同步区。



4） $K_{从} > K_{主}$ 且 $K = 2K_{主} - K_{从} = 1$ ；同步区结束后从轴减速到零速，在下一个凸轮周期再加速到同步区。



5） $K_{从} > K_{主}$ 且 $K = 2K_{主} - K_{从} < 1$；最常见的一种情况。在一个凸轮周期中，同步区结束后从轴减速到停止，等待一定主轴脉冲数过去后，再加速到同步区。



* 仅当 $K_{从} > K_{主}$ 且 $K = 2K_{主} - K_{从} < 1$ 时，适用于追剪工艺的正程制表。因为接续追剪返程轨迹表时，从轴速度必须为零速(即处于等待区)。

- **追剪正程制表程序案例**

见3.5.3中案例。

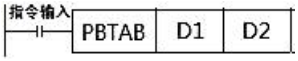
3.5.4 追剪返程制表[PBTAB]

制作追剪返程的轨迹曲线表。

1) 指令格式



32位指令	指令记号	执行条件
7 步	PBTAB	连续执行型



2) 设定数据

操作数种类	内容	数据类型
D1	曲线参数配置缓存区起始单元	BIN 32 位
D2	轨迹表格起始单元	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1		●	●							
D2		●	●							

- **功能与动作**

1) D1 指定追剪返程曲线的参数配置缓存区起始寄存器单元，建议为偶数地址。

指令执行时，D1 指向寄存器后的连续 12 个寄存器均被占用，这些寄存器应在指令执行前配置好且只配置 1 次(配置方法后述)，指令执行过程中这些寄存器将作为计算缓存区使用，指令导通条件断开这些寄存器将被释放。

例：D1 填写 D1000，则 D1000～D1011 在指令执行期间均被占用。

2) D2 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。

该起始寄存器单元接续追剪正程表的点数和位置，故不需要对 D2 指向的寄存器单元进行初始化(轨迹表点数)，也不需要初始化 D2+2 指向的寄存器单元(轨迹表初始位置值)。

对于直线追剪系统，其主轴周期长度 = 追剪正程量程 + 追剪返程量程。追剪正程轨迹表由指令PCTAB完成(本章节 3.5.2)。在执行此指令之前，需确保已完成 PCTAB 指令的制表工作。

一个追剪返程轨迹点表通常情况下由三段曲线组成，生成的曲线型值点位置变化规律是这些曲线的走向反映。

曲线段	作用区域
返程加速段	追剪返程区域的加速段回程，加速到返程最高速进入返程匀速段。
返程匀速段	追剪返程区域的匀速段回程。
返程减速段	追剪返程区域的减速段回程，从返程最高速减速到零速。

追剪返程曲线生成轨迹为四次多项式加减速计算方案。

- **轨迹表定义**

追剪返程制表指令直接接续追剪正程轨迹表进行返程轨迹的制表，执行一次制表即可生成。由用户指定表头寄存器地址(操作数D2)。表内型值点数据由系统根据曲线配置参数自动计算，需留足表格数据存放的寄存器区间。

曲线轨迹表内容	内容解释	寄存器示例
表头(指定偶地址)—型值点总数(轨迹表量程)	与追剪正程轨迹表指向同一个单元。执行制表指令前不需要初始化。 指令执行完毕由系统更新(即在原有正程制表的轨迹表点数基础上累加返程轨迹表的点数)。	D5000(L)
曲线型值点1	假定正程已占用 500 个点，则返程型值点生成地址从D6004 开始。指令计算生成。	D6004(L)
...		假定500个点
曲线型值点n		D7002(L)
表格总点数和表内型值点的数据类型均为 32 位，占用 2 个寄存器。 表格占用总寄存器个数 =表头（2个）+正程型值点总数*2+返程型值点总数*2		

- **曲线参数定义**

曲线配置参数缓存区的定义如下表所示。

地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

D1+偏移	名称	参数描述	单位
D1+0(L)	主轴回程位移	指定回程期间内主轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 ＞0。	脉冲
D1+2(L)	从轴回程位移	指定回程期间内从轴所走的脉冲数。 ● 该参数为32位数值，占用2个D寄存器单元。 ● 设置该参数为追剪正程制表中参数从轴周期位移的负值。 例如追剪正程制表中参数从轴周期位移=60000，则该参数设置-60000。	脉冲
D1+4	返程最小时间	指定追剪返程曲线完成返程的最小时间。 返程最小时间÷表格时间间隔=返程曲线的型值点点数。 ● 设置该参数>0。	ms
D1+5	表格时间间隔	指定追剪返程曲线的表格时间间隔。 ● 设置该参数与正程制表中参数表格时间间隔一致，且能被返程最小时间整除。 例如追剪正程制表中参数表格时间间隔=1，则该参数设置1。	ms
D1+6	从轴最大速度	从轴返程最大速度。 设置该参数<0。	脉冲当量/时间当量
D1+7	从轴最大加减速	从轴返程最大加/减速度。 ● 设置该参数<0。	脉冲当量/时间当量²
D1+8	时间当量	返程曲线时间当量。 ● 设置该参数为追剪正程制表中参数时间当量一致。 例如追剪正程制表中参数时间当量=10，则该参数设置10。	ms

D1+偏移	名称	参数描述	单位
D1+9	脉冲当量	返程曲线脉冲当量。 ● 设置该参数为追剪正程制表中参数脉冲当量一致。 例如追剪正程制表中参数脉冲当量 =10，则该参数设置 10。	脉冲
D1+10	制表状态	制表指令当前状态，系统同一时间只允许 1 条制表指令处于运行状态。 ● = 0：当前无制表指令运行，可执行制表指令 ● = 16：正在生成曲线数据(系统每 1ms 生成 1 个型值点数据) ● = 32：曲线数据生成完毕 ● = 64：曲线生成错误 当检测到制表状态等于 32 时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清 0。 ● 该参数无需设置，为显示值。	-
D1+11	主轴最高限速	主轴最高限速，制表指令执行前填写为追剪正程制表后生成的参数主轴最高限速的数值。制表完成后系统将再次更新该数值。 根据曲线设置参数不同，系统求得的主轴最高速也不相同。当工艺速度小于主轴最高限速时，满足设计要求。 ● 追剪正程制表生成的主轴最高限速为中间值，执行返程制表指令后生成的主轴最高限速再与主轴最高限速进行比较。 ● 该参数在指令执行前需设置初始值，指令执行后为显示值。	脉冲当量 / 时间当量

- **主轴最高限速换算**

对于追剪/滚切系统，机台的速度一般而言以“米/分”为单位。其规定的最大限速需与制表完成后系统生成的主轴最高限速进行比较，若机台的速度大于系统生成的主轴最高限速，则需要重新调整曲线参数以满足设计要求。

主轴最高限速的单位与曲线参数中的脉冲当量与时间当量有关，比较时需要进行量纲转换。

示例：假定机台的最大速度为 30 米/分，脉冲当量为 10 脉冲/mm。那么机台的最大速度 = 30 * 1000 / (60 * 1000) = 0.5 mm/ms，根据脉冲当量可算出得出系统的最大速度为5pls/ms，。

假定曲线的脉冲当量参数设置为10，曲线的时间当量参数设置为10，则生成曲线的速度单位为1pls/ms。那么系统生成的主轴最高限速数值应大于等于5，即5pls/ms。

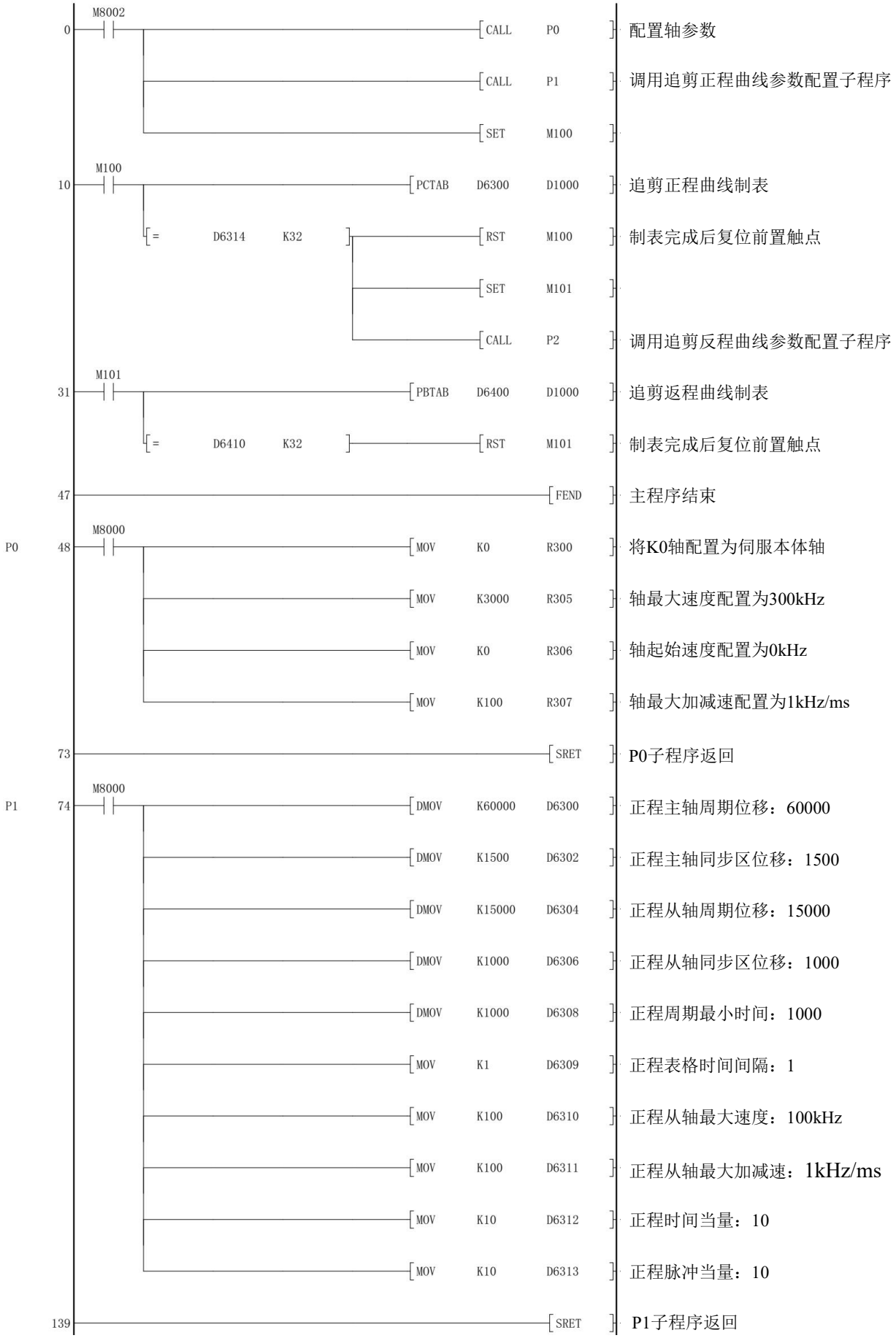
- **追剪制表应用举例**

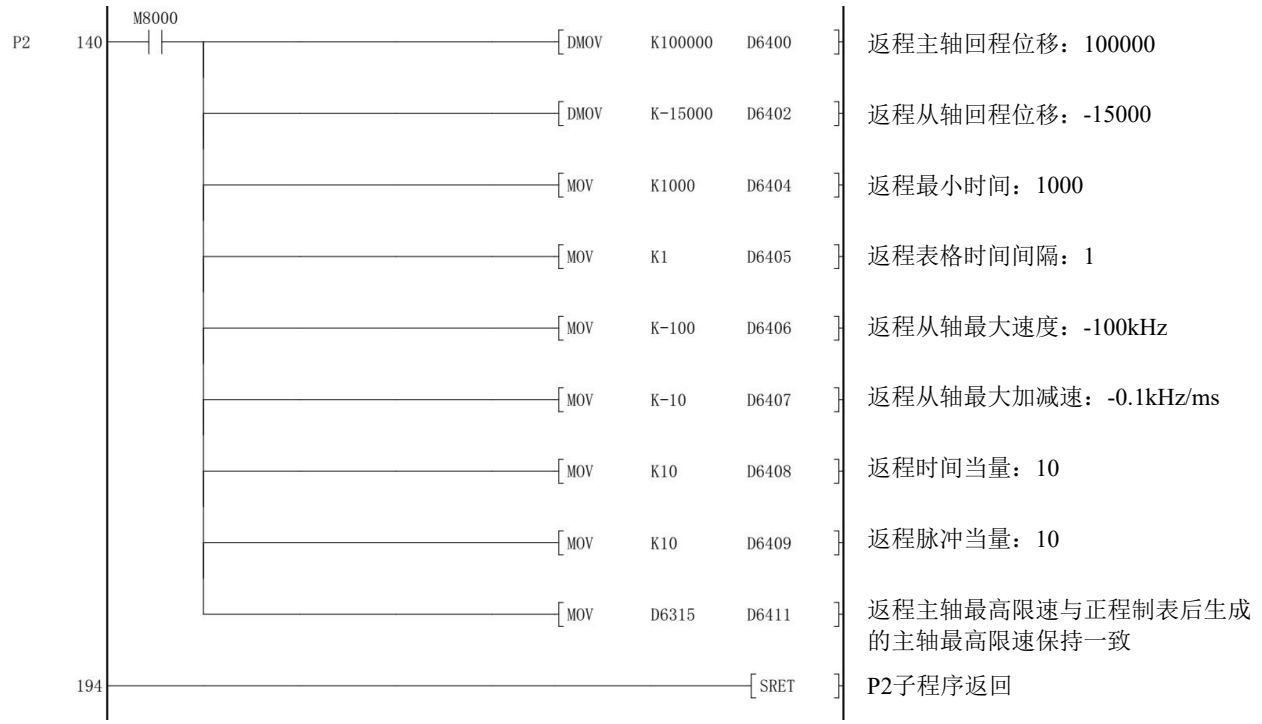
完整的追剪工艺实现梯形图示例，包含以下内容：

① 输出轴轴参数配置(输出轴为伺服本体，对应软件虚拟轴 0)。

② 追剪正程制表(正程制表梯形图可单独作为追剪正程/滚切制表的示例)。

③ 追剪返程制表。


M8002
0
CALL P0 配置轴参数
CALL P1 调用追剪正程曲线参数配置子程序
SET M100
M100
10
PCTAB D6300 D1000 追剪正程曲线制表
= D6314 K32
RST M100 制表完成后复位前置触点
SET M101
CALL P2 调用追剪反程曲线参数配置子程序
M101
31
PBTAB D6400 D1000 追剪返程曲线制表
= D6410 K32
RST M101 制表完成后复位前置触点
47
FEND 主程序结束
M8000
P0 48
MOV K0 R300 将K0轴配置为伺服本体轴
MOV K3000 R305 轴最大速度配置为300kHz
MOV K0 R306 轴起始速度配置为0kHz
MOV K100 R307 轴最大加减速配置为1kHz/ms
73
SRET P0子程序返回
M8000
P1 74
DMOV K60000 D6300 正程主轴周期位移：60000
DMOV K1500 D6302 正程主轴同步区位移：1500
DMOV K15000 D6304 正程从轴周期位移：15000
DMOV K1000 D6306 正程从轴同步区位移：1000
DMOV K1000 D6308 正程周期最小时间：1000
MOV K1 D6309 正程表格时间间隔：1
MOV K100 D6310 正程从轴最大速度：100kHz
MOV K100 D6311 正程从轴最大加减速：1kHz/ms
MOV K10 D6312 正程时间当量：10
MOV K10 D6313 正程脉冲当量：10
139
SRET P1子程序返回



- **使用方法**

下载梯形图后将PLC运行，自动生成追剪轨迹表数据。

- **追剪曲线轨迹分析**

追剪正程制表主从轴参数：

- 主轴周期位移=60000脉冲
- 主轴同步区位移=1500脉冲
- 从轴周期位移=15000脉冲
- 从轴同步区位移=1000脉冲

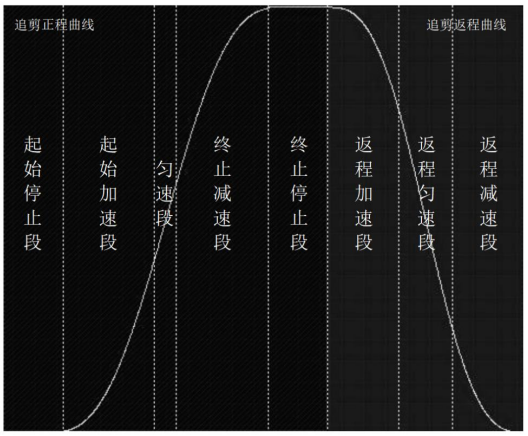
假定从轴同步区位移÷从轴周期位移 = $K_{从}$，主轴同步区位移÷主轴周期位移 = $K_{主}$， 则计算得 $K_{从}$ = 1000 ÷ 15000 = 0.067，$K_{主}$ = 1500 ÷ 60000 = 0.025。

故 $K_{从}$ > 2 倍 $K_{主}$，正程曲线应包含起始等待段、起始加速段、匀速段、终止减速段、终止停止段这 5 个阶段。

$K_{从}$ 与 $K_{主}$ 的比例关系对应不同的凸轮曲线轨迹，参考章节 3.5.2 描述。

追剪返程输出曲线为四次多项式曲线，分为返程加速段、返程匀速段和返程减速段。

以下是一个完整的从轴追剪周期位移曲线：

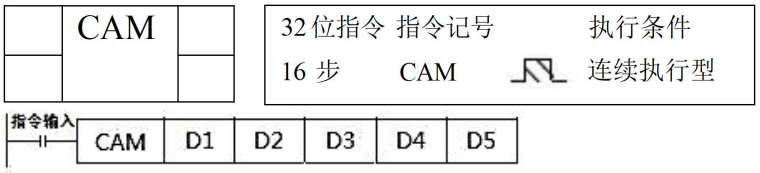

追剪正程曲线
追剪返程曲线
起始停止段
起始加速段
匀速段
终止减速段
终止停止段
返程加速段
返程匀速段
返程减速段

3.5.5 电子凸轮输出[CAM]

对指定软件虚拟输出轴(AXV)做从轴，与时间/主轴位置成函数关系(电子凸轮)。主轴变化时，从轴依电子凸轮曲线运动。采用普通制表或专用制表指令产生不同需求的凸轮表(如追剪/滚切)表，供凸轮指令使用。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	软件虚拟输出轴号	BIN 32 位
D2	轨迹表格起始单元	BIN 32 位
D3	主轴(横轴)实际数值	BIN 32 位
D4	主轴(横轴)实际量程	BIN 32 位
D5	凸轮参数缓存区起始单元	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●									
D2		●	●							
D3		●	●							
D4		●	●							
D5		●	●							

- **功能与动作**

1) D1 指定软件虚拟输出轴号(AXV0～AXV1)，对应数值 0～1。

	设定范围
16位运算	0：脉冲输出轴，Y0为脉冲端口，Y4位方向端口。 1：脉冲输出轴，Y1为脉冲端口，Y5位方向端口。 2：脉冲输出轴，Y2为脉冲端口，Y6位方向端口。 3：脉冲输出轴，Y3为脉冲端口，Y7位方向端口。

2) D2 指定凸轮曲线轨迹表格的起始寄存器单元，建议为偶数地址。

3) D3 指定主轴当前实际值，建议为偶数地址。

◆ 该值若对应时间，则表格输出以主轴时间轴进行输出，可视之为时间凸轮曲线。

◆ 该值若对应位置，则表格输出以主轴位置轴进行输出(指令推荐用法)。

4) D4 指定主轴实际量程。

当主轴规划为位置信息时，推荐设置为 1 个凸轮周期内啮合阶段主轴所走的脉冲数，也称之为主轴最大相位。

若主轴实际量程设置与主轴最大相位不相等，两者比例可视作乘以了 1 个电子齿轮的系数，但一般不推荐这样设置。

5) D5 指定凸轮中间参数及事件设置的缓存区，相关具体定义后述。

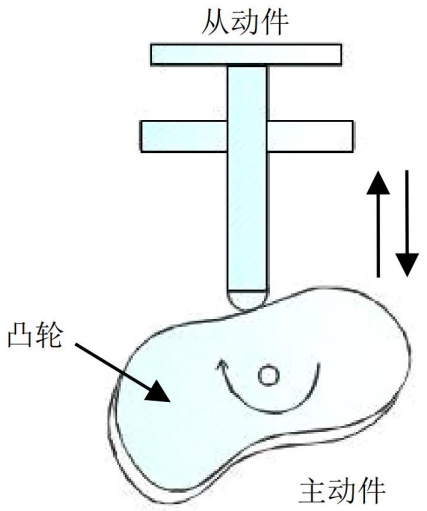
◆ 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。

- **电子凸轮功能**

传统的机械凸轮机构，是由凸轮、从动件以及机架等三部份组成。

机械凸轮是一种不规则形状的机件，一般为等转速的输入件，可经由直接接触传递运动到从动件，使从动件依设定的规律运动。

从动件为机械凸轮所驱动的被动件，一般为产生不等速、不连续、不规则运动的输出件。机架则是用来支持机械凸轮与从动件的机件。



电子凸轮又称Electronic Cam(E-CAM)，是模拟机械凸轮的一种智能控制器。它通过位置传感器将主轴当前位置反馈给系统，系统将此位置信息进行运算处理，并按设定要求对从轴进行位置输出。

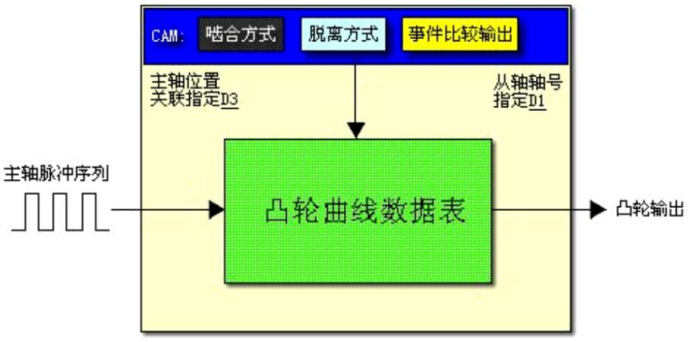
相对传统的方式，使用电子凸轮有着以下的好处：

●不同的工艺需要不同的凸轮曲线，可通过修改电子凸轮表内的数据，而不用修改机械部件。

●可以有较高的加减速性能。

●运行可以较为平稳。

电子凸轮指令功能框图：



●凸轮曲线数据表：E-CAM 数据表，由制表指令、专用制表指令(追剪/滚切制表)或上位机辅助工具生成。为一个循环凸轮周期中从轴与主轴的位置相对关系的表格。

●啮合：启动电子凸轮。以啮合点的主轴位置为相对零点，从轴根据主轴的当前位置走E-CAM 表中的凸轮曲线(输出插补值)。

●脱离：停止电子凸轮。从轴不再跟随主轴位置的变化而变化。

●事件比较输出：凸轮在某一相位产生相应的动作，需事件输出驱动(如追剪、滚切)。

- **凸轮参数缓存区 D5**

凸轮参数缓存区一共占用 28 个寄存器单元，如 D5 指定 D1200，则 D1200～D1227 寄存器区域在电子凸轮指令执行时被占用。

地址后带(L)标识的参数为 32 位参数，占用 2 个D 寄存器。

D5+偏移	名称	参数描述	单位
D5+0	相位偏差补偿	主轴虚拟相位补偿辅助轴位置值，用于消除长时间运行后由于机械或其他原因造成的主从轴相位偏差。该参数为增量式位置值。 凸轮运行时，修改该参数将会把偏差值叠加到当前周期主轴相位上(主轴相位概念解释参考下面的主轴当前相位参数)。在一个插补周期时间内，从轴按照叠加后的 主轴相位输出到轨迹表的相应位置。若补偿值较大，建议进行分时插补补偿，否则在一个插补周期内从轴位置变化剧烈可能导致轴速过大，引发轴错误。 例：运行一段时间后主轴相位偏差了10个脉冲，则相位偏差补偿填 10 消除主从轴相位偏差。若再次偏差5个脉冲，则相位偏差补偿填10+5=15。 ● 该参数填写寄存器的序号，对应占用2个D寄存器单元。例如填写K1400，则相位偏差补偿对应寄存器为D1400(L)，运行时修改该寄存器数值即设置生效。 ● 当主轴当前相位在边界值附近时(即刚进入啮合阶段或离开啮合阶段)，修改相位偏差补偿可能导致主轴当前相位越界。一般应避免该情况发生，可能导致凸轮时序运行异常。 ● 脱离阶段修改相位偏差补偿，且导致主轴相位叠加值进入凸轮啮合区。系统将会进入到啮合阶段，但此次的多余部分脉冲并不会叠加到主轴相位上。 例：主轴只需走10个脉冲即可进入啮合区，此时相位偏差补偿从0修改为20，则凸轮将进入啮合阶段，数值20作为相位偏差补偿的更新基准值。从轴输出位置为主轴相位为0时的轨迹表点位，而非主轴相位为10时的轨迹表点位。	-
D5+1	保留参数		
D5+2(L)	初始前置啮合位移	初始前置啮合位移量，仅首个凸轮周期生效。一般应用于跳过主轴空行程的需求。 ● 凸轮表格同步区位于周期正中处，可通过该参数调整主/从轴的同步区偏移量。 ● 该参数仅在啮合方式中设置后，方才生效。 ● 该参数为32位数值，占用2个D寄存器单元。 ● 不能设置该参数<0。	脉冲
D5+4(L)	周期前置啮合位移	周期前置啮合位移量，首个凸轮周期流程结束后开始生效。 ● 凸轮表格同步区位于周期正中处，可通过该参数调整主/从轴的同步区偏移量。 ● 该参数仅在啮合方式中设置后，方才生效。 ● 该参数为32位数值，占用2个D寄存器单元。 ● 不能设置该参数<0。	脉冲
D5+6(L)	相位脱离位移	主轴相位脱离对应位移量，运行到指定的主轴相位处进行脱离。此参数一般应用于单次非循环的凸轮工艺。 ● 该参数仅在脱离方式中设置后，方才生效。 ● 该参数为32位数值，占用2个D寄存器单元。 ● 不能设置设置该参数<0或>CAM指令操作数D4数值(主轴最大相位)。	脉冲
D5+8～D5+11	保留参数		

<table>
<tr><th>D5+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>D5+12(L)</td><td>主轴当前相位</td><td>主轴的位置为累积型脉冲数值，每个凸轮周期开始时将主轴的位置记录为相对零点。主轴相位是指该凸轮周期内主轴的相对位移，即主轴当前周期的实际位置减去该周期的相对零点位置之差。
一个凸轮周期主轴相位最大值即为主轴实际量程，此参数由电子凸轮指令的操作数 D4 确定，满足主轴最大相位 > 相位脱离位移 ≥0 的关系。
若凸轮脱离后又立即啮合，则当前凸轮周期的主轴最大相位位置点即为下一个凸轮周期的零点。
● 该参数为 32 位数值，占用 2 个 D 寄存器单元。
● 该参数无需设置，为显示值。</td><td>脉冲</td></tr>
<tr><td>D5+14</td><td>啮合方式</td><td>该参数决定凸轮的啮合方式。
● 0 - 指令生效啮合(指令得电状态)
● 1 - 输入触发啮合
● 2 - 前置位移量啮合
◆ 首次周期流程，主轴走初始前置啮合位移后开始啮合。
◆ 非首次周期流程，主轴走周期前置啮合位移后开始啮合。
● 3 - 输入触发 + 前置啮合
◆ 首次周期流程，输入触发啮合生效后，主轴经过初始前置啮合位移后开始 啮合。
◆ 非首次周期流程，输入触发啮合生效后，主轴经过周期前置啮合位移后开 始啮合。
必须是在脱离状态下，系统才去判断啮合条件。
凸轮轨迹表的起始点通常设计为从轴原点(即轨迹表第一个型值点的位置)，在这种情况下，需保证啮合位置点物理上也为从轴的原点位置。</td><td>-</td></tr>
<tr><td>D5+15</td><td>啮合输入触发</td><td>啮合触发输入使能，按位进行设置(掩码)。
<table>
<tr><th>数值设定 =</th><th>选定端口</th><th>数值设定 =</th><th>选定端口</th></tr>
<tr><td>0</td><td>不选定端口</td><td colspan="2"></td></tr>
<tr><td>1</td><td>X0/M8096</td><td>256</td><td>X10/M8104</td></tr>
<tr><td>2</td><td>X1/M8097</td><td>512</td><td>X11/M8105</td></tr>
<tr><td>4</td><td>X2/M8098</td><td>1024</td><td>X12/M8106</td></tr>
<tr><td>8</td><td>X3/M8099</td><td>2048</td><td>X13/M8107</td></tr>
<tr><td>16</td><td>X4/M8100</td><td>4096</td><td>X14/M8108</td></tr>
<tr><td>32</td><td>X5/M8101</td><td>8192</td><td>X15/M8109</td></tr>
<tr><td>64</td><td>X6/M8102</td><td>16384</td><td>X16/M8110</td></tr>
<tr><td>128</td><td>X7/M8103</td><td>-32768</td><td>X17/M8111</td></tr>
</table>
● 通过轴事件输入点.Bit13 可使用 M8096～M8111 代替 X00～X17 的控制。
● 该参数仅在啮合方式中设置后，方才生效。</td><td>-</td></tr>
</table>

<table>
<tr><th>D5+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>D5+16</td><td>啮合输入触发极性</td><td>啮合触发输入极性，按位设置选定端口的极性(掩码)。

<table>
<tr><th>啮合輸入触发极性.Bit</th><th>定义</th></tr>
<tr><td>Bit0</td><td>= 0：信号进入 X0 端口为 ON，离开为 OFF
= 1：信号进入 X0 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit1</td><td>= 0：信号进入 X1 端口为 ON，离开为 OFF
= 1：信号进入 X1 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit2</td><td>= 0：信号进入 X2 端口为 ON，离开为 OFF
= 1：信号进入 X2 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit3</td><td>= 0：信号进入 X3 端口为 ON，离开为 OFF
= 1：信号进入 X3 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit4</td><td>= 0：信号进入 X4 端口为 ON，离开为 OFF
= 1：信号进入 X4 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit5</td><td>= 0：信号进入 X5 端口为 ON，离开为 OFF
= 1：信号进入 X5 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit6</td><td>= 0：信号进入 X6 端口为 ON，离开为 OFF
= 1：信号进入 X6 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit7</td><td>= 0：信号进入 X7 端口为 OFF，离开为 ON
= 1：信号进入 X7 端口为 ON，离开为 OFF</td></tr>
<tr><td>Bit8</td><td>= 0：信号进入 X10 端口为 ON，离开为 OFF
= 1：信号进入 X10 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit9</td><td>= 0：信号进入 X11 端口为 ON，离开为 OFF
= 1：信号进入 X11 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit10</td><td>= 0：信号进入 X12 端口为 ON，离开为 OFF
= 1：信号进入 X12 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit11</td><td>= 0：信号进入 X13 端口为 ON，离开为 OFF
= 1：信号进入 X13 端口为 OFF，离开为 ON</td></tr>
<tr><td>Bit12</td><td>= 0：信号进入 X14 端口为 ON，离开为 OFF
= 1：信号进入 X14 端口为 OFF，离开为 ON</td></tr>
</table>
</td><td>-</td></tr>
</table>

			Bit13	= 0：信号进入 X15 端口为 ON，离开为 OFF = 1：信号进入 X15 端口为 OFF，离开为 ON
			Bit14	= 0：信号进入 X16 端口为 ON，离开为 OFF = 1：信号进入 X16 端口为 OFF，离开为 ON
			Bit15	= 0：信号进入 X17 端口为 ON，离开为 OFF = 1：信号进入 X17 端口为 OFF，离开为 ON
		● 通过轴事件输入点.Bit13 可使用 M8096～M8111 代替 X00～X17 的控制。 该参数仅在啮合输入触发中设置后，方才生效。		

<table>
<tr><th>D5+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>D5+17</td><td>脱离方式</td><td>脱离方式选择
● 0 - 周期脱离(运行完一个凸轮周期后自动脱离)
● 1 - 输入触发脱离
● 2 - 相位脱离，主轴走到相位脱离位移后脱离
● 3 - 输入触发 + 相位脱离，先满足输入触发后，再走到相位脱离位移位移量处，然后脱离。若输入触发产生时，主轴相位在相位脱离位移之后，则凸 轮将立即脱离。
例如：假定主轴最大相位为 1000，相位脱离位移为 800。若触发输入发生时，主轴当前相位为 500，则运行到本周期 800 处脱离；主轴当前相位为900，则直接脱离。
◆ 脱离方式选择 1/2/3 方式时，方式 0 仍然有效。即凸轮周期结束时，无论脱离条件是否满足，凸轮也将脱离。
◆ 凸轮指令断开时，无论凸轮处于什么状态，立即脱离。
◆ 必须在啮合状态下，系统才去判断脱离条件。脱离后若指令仍处于得电状态，系统开始判断啮合条件，满足啮合条件后会再次啮合。
◆ 必须确保主从轴在某一相对安全位置时脱离，比如追剪系统，若在同步区进行脱离，由于事件输出端口(切刀)仍处于输出状态，可能导致人身安全危险以及机件损坏！</td><td>-</td></tr>
<tr><td>D5+18</td><td>脱离输入触发</td><td>脱离触发输入使能，参考啮合输入触发参数设置。
● 通常情况下，不能和啮合输入触发指定同一个 X 端口。</td><td>-</td></tr>
<tr><td>D5+19</td><td>脱离输入触发极性</td><td>脱离触发输入极性，按位设置选定端口极性(掩码)。参考啮合输入触发参数设置。
● 该参数仅在脱离输入触发中设置后，方才生效。</td><td>-</td></tr>
<tr><td>D5+20</td><td>比较事件输出表格地址</td><td>指定比较表格的起始软元件地址。
● 该参数填写寄存器的序号，例如填写 K1000，则比较事件输出表格地址对应寄存器为 D1000，运行时修改该寄存器数值即设置生效。
● 比较表格输出表格的每个表格行占用三个数据寄存器。其中比较数据 2 个，比较输出值 1 个。具体格式参考章节 3.3.5描述。</td><td>-</td></tr>
<tr><td>D5+21</td><td>比较事件输出表格行数</td><td>指定比较表格的输出行数。
比较事件的表格输出格式填写方法参考章节 3.3.5描述。、</td><td>-</td></tr>
<tr><td>D5+22</td><td>比较事件输出元件</td><td>从起始地址开始指定连续的16 个位元件。对应的数值与位元件的关系如下表所示：
<table>
<tr><th>比较事件输出元件</th><th>定义</th></tr>
<tr><td>0～255</td><td>Y000～Y□□□</td></tr>
<tr><td>1000～1255</td><td>X000～X□□□</td></tr>
<tr><td>2000～2255</td><td>T0～T255，触点型</td></tr>
<tr><td>3000～3255</td><td>C0～C255，触点型</td></tr>
<tr><td>4000～4255</td><td>M8000～M8255</td></tr>
<tr><td>5000～5999</td><td>S0～S999</td></tr>
<tr><td>10000～13071</td><td>M0～M3071</td></tr>
</table>
当主轴当前实际值≥表格某行的比较值时，这 16 个位元件输出该行的输出值。
● 指定的位元件起始地址必须被 16 整除。</td><td>-</td></tr>
</table>

<table>
<tr><th>D5+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>D5+23</td><td>比较事件输出元件使能</td><td>指定比较输出位元件的输出使能位，按位进行设置。
● 设置为 1 时，若表格输出值对应位为 1，对应位元件输出。
● 设置为 0 时，无论表格输出值是否为 1，对应位元件始终不输出。
● 设置事件比较输出元件和对应的轴输出端口不要产生冲突。比如凸轮指令指定Y0 输出，那么必须关闭 Y0 的脉冲输出使能(配置 D8180=0)。</td><td>-</td></tr>
<tr><td>D5+24</td><td>凸轮运行状态</td><td>凸轮当前运行状态的显示，指令运行过程中进行检测。
● = 0：凸轮脱离状态。
● = 1：凸轮啮合状态。
● 该参数无需设置，为显示值。</td><td>-</td></tr>
<tr><td>D5+25</td><td>凸轮起始状态</td><td>在启动凸轮指令前，必须设置该参数。
凸轮指令启动时，凸轮运行状态默认为凸轮起始状态。
● 在首次运行凸轮指令时，该参数推荐填 0。在前次凸轮指令运行时，由于意外停止指令。此时应获取结束时的凸轮运行状态，并将其配置到凸轮起始状态，以使本次凸轮指令启动后，会沿用上次凸轮结束时的状态继续运行。</td><td>-</td></tr>
<tr><td>D5+26(L)</td><td>凸轮起始相位</td><td>在启动凸轮指令前，必须设置该参数。
该参数与凸轮其他参数的关系如下表所示：
<table>
<tr><th>凸轮起始状态</th><th>凸轮起始相位</th><th>啮合方式</th><th>后续运行状态</th></tr>
<tr><td rowspan="8">=0</td><td rowspan="4">=0</td><td>=0</td><td>立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td></tr>
<tr><td>=1</td><td>输入触发后啮合</td></tr>
<tr><td>=2</td><td>前置位移触发后啮合</td></tr>
<tr><td>=3</td><td>输入+前置位移触发后啮合</td></tr>
<tr><td rowspan="4">>0</td><td>=0/1</td><td>立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td></tr>
<tr><td>=2</td><td>凸轮起始相位为当前已发生的前置啮合位移
*凸轮起始相位不要填写超过前置啮合位移的数值</td></tr>
<tr><td>=3</td><td>输入啮合条件已触发，凸轮起始相位为当前已发生的前置啮合位移
*凸轮起始相位不要填写超过前置啮合位移的数值</td></tr>
<tr><td rowspan="2">=1</td><td>=0</td><td>=0/1/2/3</td><td rowspan="2">立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td></tr>
<tr><td>>0</td><td>=0/1/2/3</td></tr>
</table>
● 在首次运行凸轮指令时，该参数推荐填 0。在前次凸轮指令运行时，由于意外停止指令。此时应获取结束时的凸轮运行状态，并根据当前主轴相位，啮合方式等条件填写凸轮起始相位，以使本次凸轮指令启动后，会沿用上次凸轮结束时的状态继续运行。
● 当凸轮起始相位设置不为 0 时，根据当前主轴相位的数值，在比较输出表格中找到上 1 个小于该数值的输出状态，并进行比较事件输出元件输出。</td><td>脉冲</td></tr>
</table>

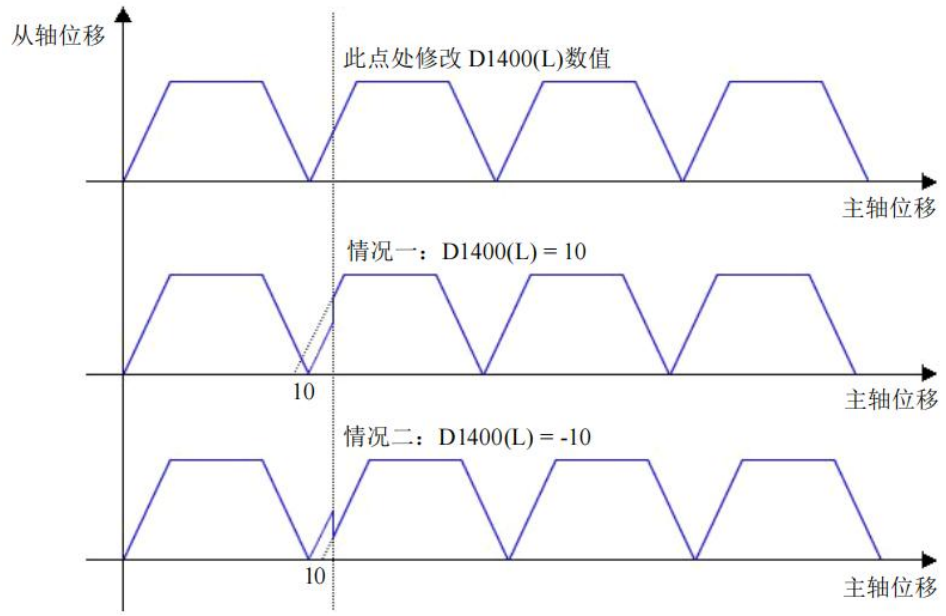
<table>
<tr><th>D5+偏移</th><th>名称</th><th>参数描述</th><th>单位</th></tr>
<tr><td>D5+28</td><td>辅助轴前置电子齿轮元件</td><td>指定辅助轴前置电子齿轮的元件地址。
<table><tr><th>辅助轴前置电子齿轮元件</th><th>定义</th></tr><tr><td>0～7999</td><td>对应 D0～D7999</td></tr><tr><td>8000～8255</td><td>对应 D8000～D8255</td></tr><tr><td>9000～9255</td><td>对应 C0～C255</td></tr><tr><td>10000～10255</td><td>对应 T0～T255</td></tr></table>● 辅助轴用于在线调整长度。</td><td>-</td></tr>
<tr><td>D5+29</td><td>辅助轴前置电子齿轮分子</td><td>用于设置辅助轴前置电子齿轮的分子。</td><td>-</td></tr>
<tr><td>D5+30</td><td>辅助轴前置电子齿轮分母</td><td>用于设置辅助轴前置电子齿轮的分母，该参数不能设置为 0。</td><td>-</td></tr>
<tr><td>D5+31</td><td>轨迹表格单元D2指向</td><td>轨迹表格单元D2指向定义，控制器版本号大于V10046。
D5+31≠5555，D2为轨迹表格起始单元地址，例如D2操作数填写D1000，表示轨迹表格起始单元地址为D1000。
D5+31＝5555，D2为轨迹表格起始单元的序号，例如D2操作数填写D1000，D1000当中数值为K2000，表示轨迹表格起始单元地址为D2000。</td><td>-</td></tr>
</table>

- **相位偏差补偿分析**

在凸轮指令启动时，系统以当前相位偏差补偿数值为基准零点，在运行过程中修改该寄存器， 则将修改过后的增量数值作为偏差值叠加到主轴相位上，同时从轴(即凸轮轴)也将输出到叠加后主轴相位对应表格的从轴输出点位置。

通常情况下对凸轮参数缓存区初始化时应将相位偏差补偿对应的 32 位寄存器数值清零。

假定相位偏差补偿值对应的寄存器为 D1400(L)，初始化数值为 0。其应用效果示意图如下所示：



- **啮合/脱离方式分析**

凸轮指令分别各有4种啮合方式和4种脱离方式，一共有16种工作模式的选择，如下表所示：

啮合方式 / 脱离方式	指令生效啮合	输入触发啮合	前置位移量啮合	输入触发+前置位移量啮合
周期脱离	A	B	C	D
输入触发脱离	E	F	G	H
主轴相位脱离	I	J	K	L
输入触发+主轴相位脱离	M	N	O	P

●无论选择哪种啮合/脱离方式，任意情况下梯形图扫描到断开凸轮指令的操作，凸轮(从轴)立即脱离。

●无论选择哪种脱离方式，周期脱离始终有效。即凸轮周期结束时，即使设定的脱离条件不满足，凸轮也将立即脱离。

●凸轮脱离后系统时刻判断啮合条件，若啮合条件满足则立即啮合，故存在脱离后马上啮合的情况(例如方式A)。由于脱离非常短暂，此种情况下可认为凸轮始终处于啮合状态。

●假定凸轮输出表格输出元件为 Y0，对应主轴相位变化位移状态变化为 2 个(即表格输出有 2 行)。

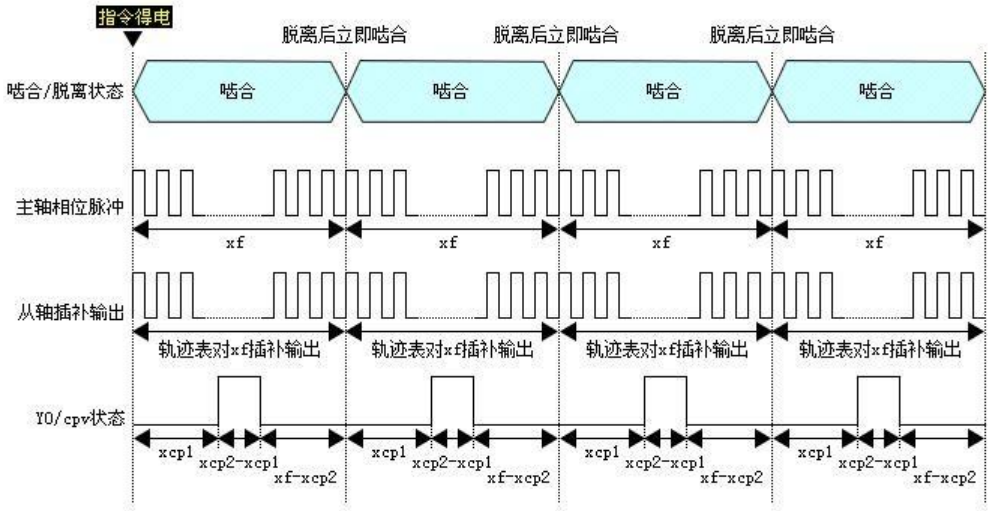
A～P 组合情况的时序分析，其他参数的设置情况以及时序图中所用到的助记符如下所示：

指令运行状态	始终得电(生效)		
输入触发啮合	X0	输入触发啮合极性	信号进入 ON，信号离开 OFF。时序图上信号来源仅为说明凸轮的工作流程，并不代表实际工况。
输入触发脱离	X1	输入触发脱离极性	信号进入 ON，信号离开 OFF。时序图上信号来源仅为说明凸轮的工作流程，并不代表实际工况。
比较事件端口	Y0	比较事件端口极性	ON 输出高电平，OFF 输出低电平。

时序图助记符		
	xf	主轴最大相位(主轴实际量程)，指令操作数 D4 指定。
	xe	相位脱离位移
	xn	主轴当前相位
	xcp1	表格输出第 1 行位移
	xcp2	表格输出第 2 行位移
	xs	初始前置啮合位移
	xsl	周期前置啮合位移
	cpv	比较事件状态

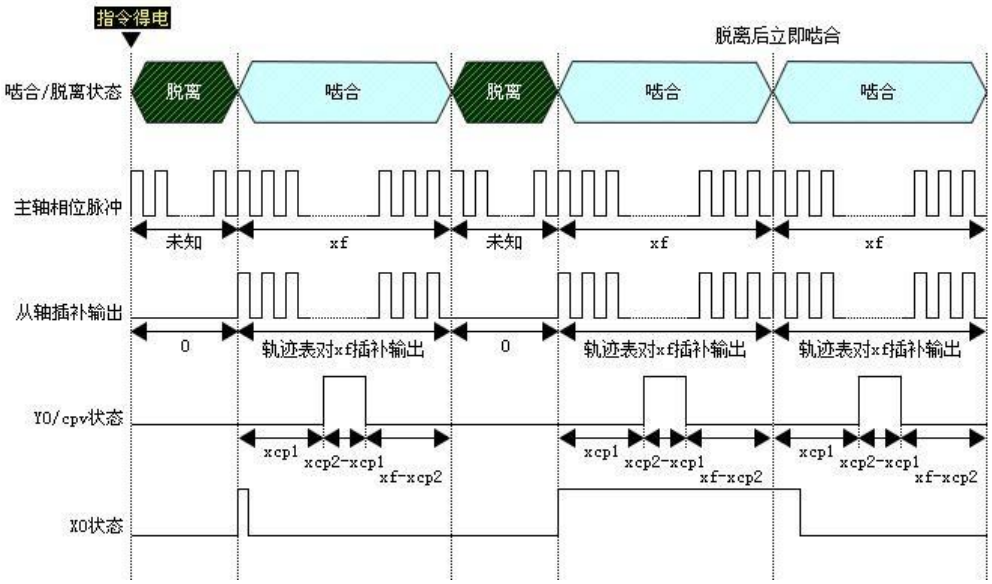
- 组合 A：

❖ 指令生效则啮合。
❖ 周期脱离。



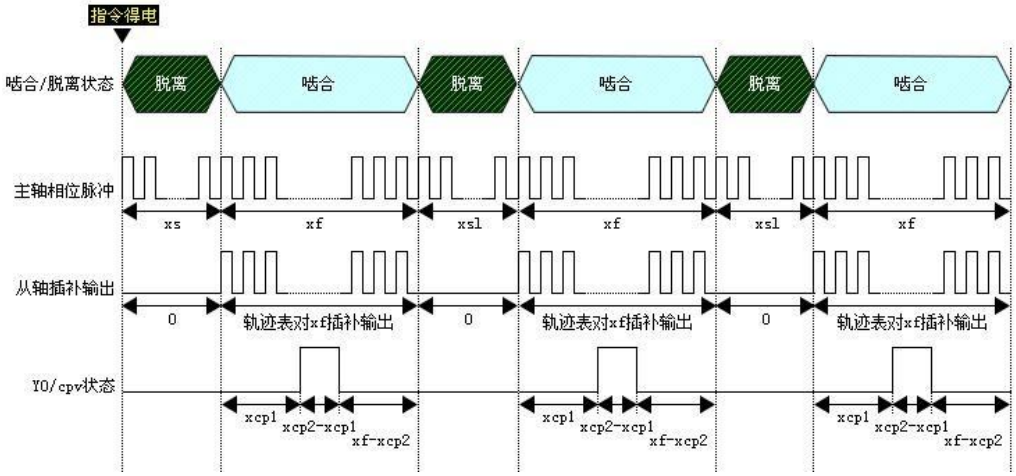
- 组合 B：

❖ 指令生效后，X0 为 ON 则啮合。
❖ 周期脱离。



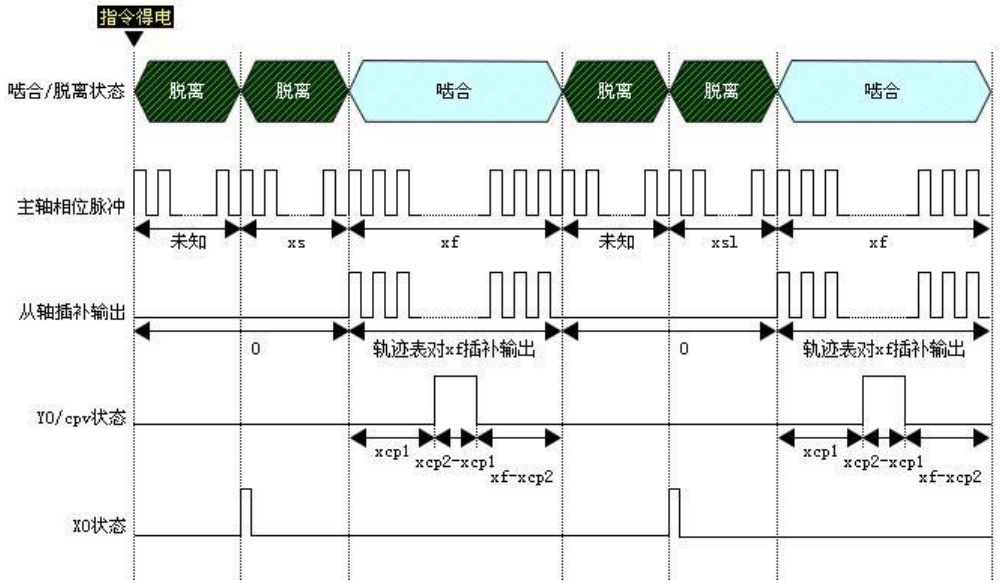
- 组合 C：

❖ 指令生效后，主轴走前置啮合位移量后则啮合。
❖ 周期脱离。



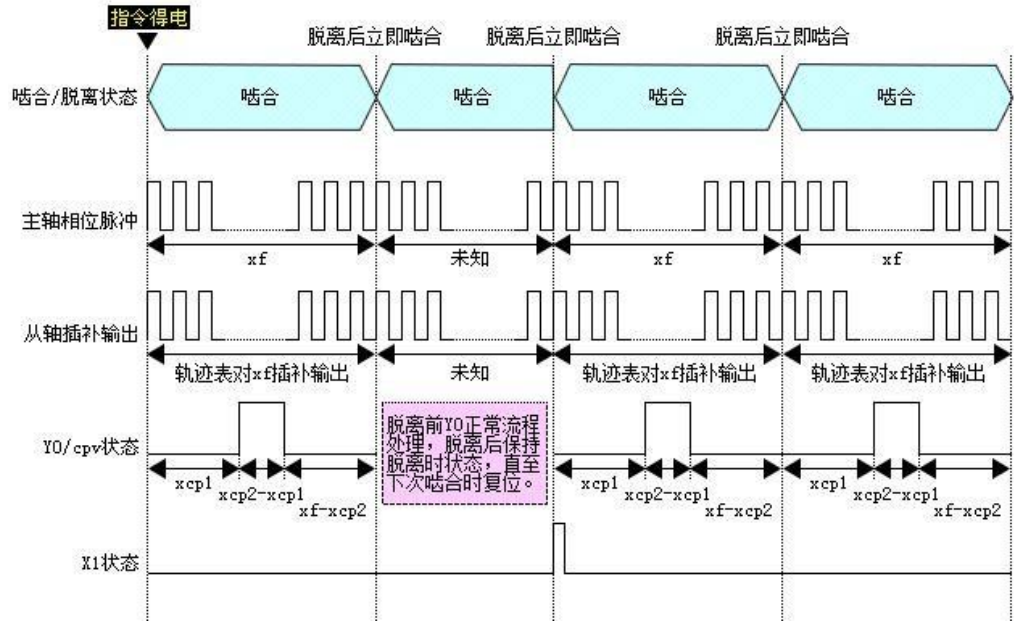
- 组合 D：

❖ 指令生效后，等待 X0 触发为ON，主轴再走前置啮合位移量后则啮合。
❖ 周期脱离。



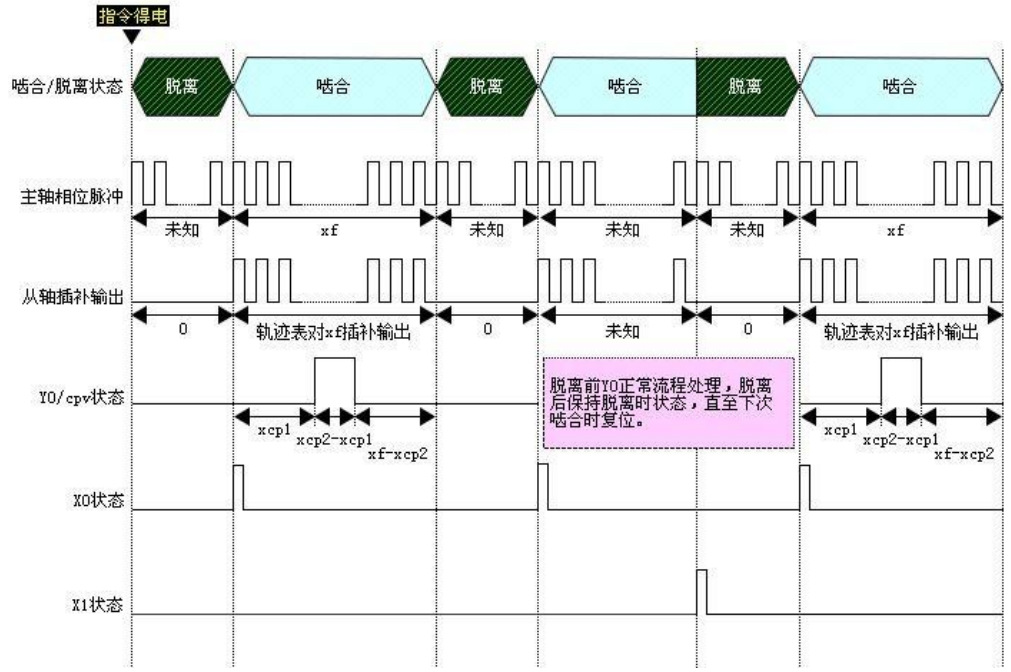
- 组合 E：

❖ 指令生效则啮合。
❖ X1 触发为 ON 或周期结束，凸轮脱离。



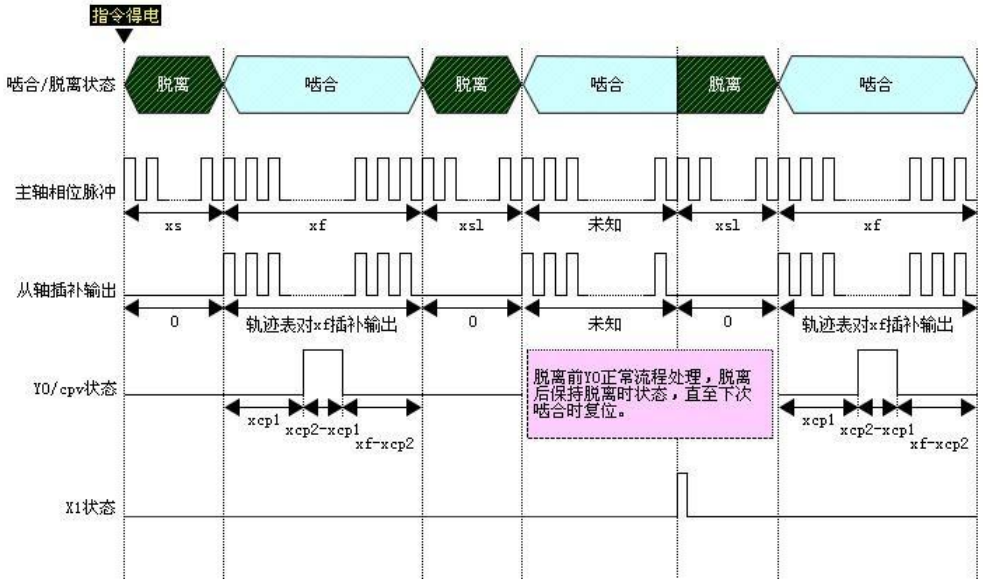
- 组合 F：

❖ 指令生效后，X0 为 ON 则啮合。
❖ X1 触发为 ON 或周期结束，凸轮脱离。



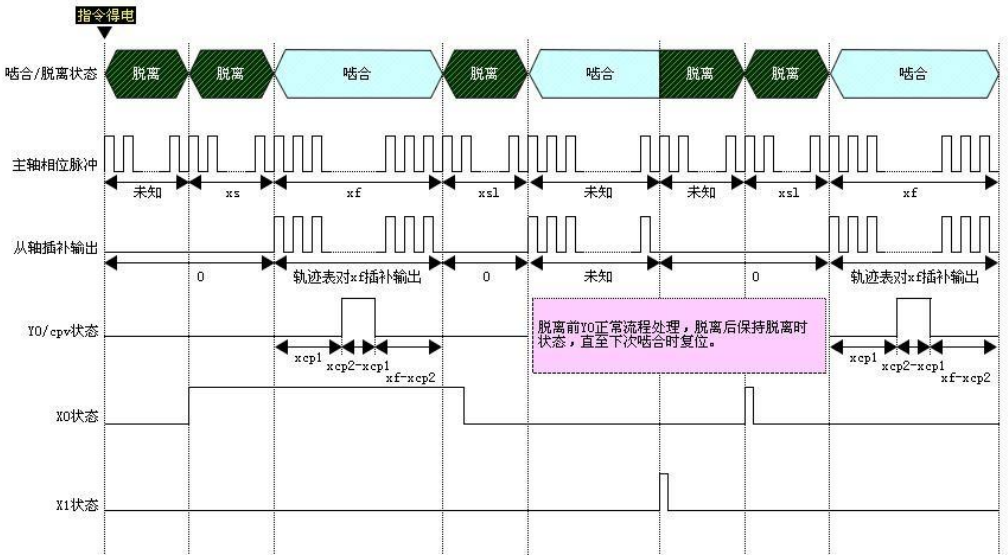
- 组合 G：

❖ 指令生效后，主轴走前置啮合位移量后则啮合。
❖ X1 触发为 ON 或周期结束，凸轮脱离。



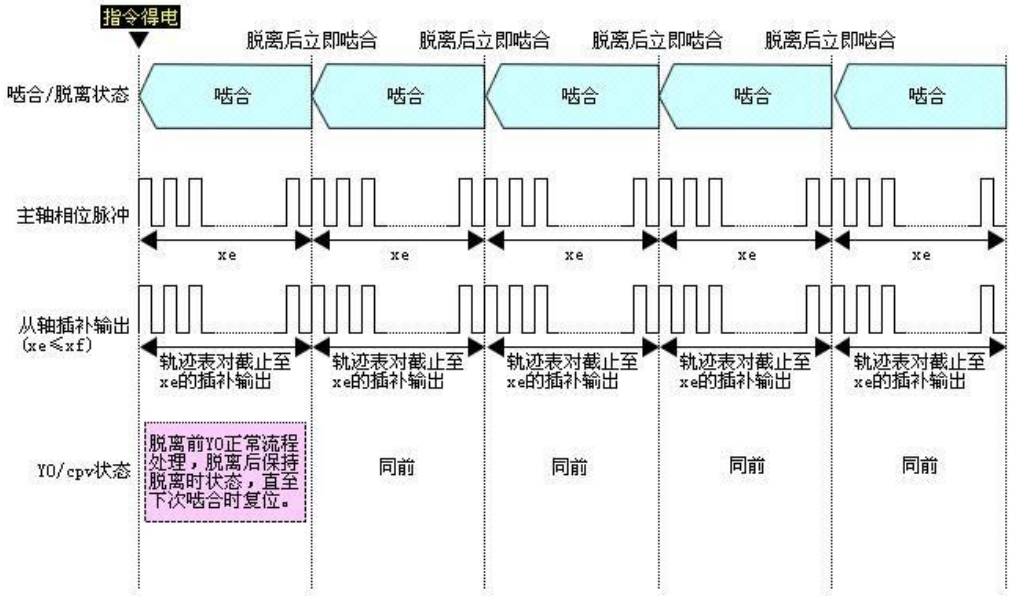
- 组合 H：

❖ 指令生效后，等待 X0 触发为ON，主轴再走前置啮合位移量后则啮合。
❖ X1 触发为 ON 或周期结束，凸轮脱离。



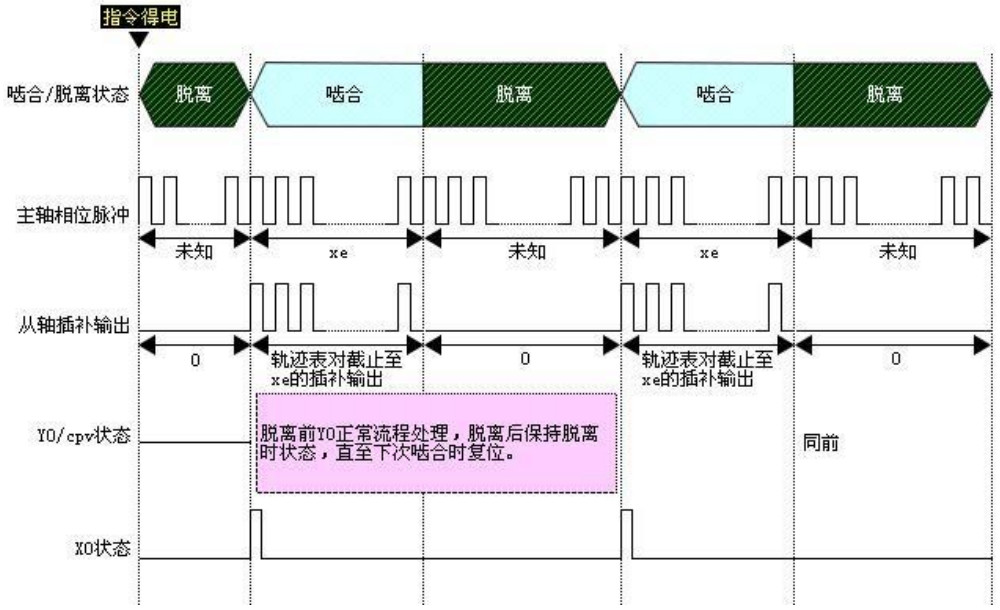
- 组合 I：

❖ 指令生效则啮合。
❖ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



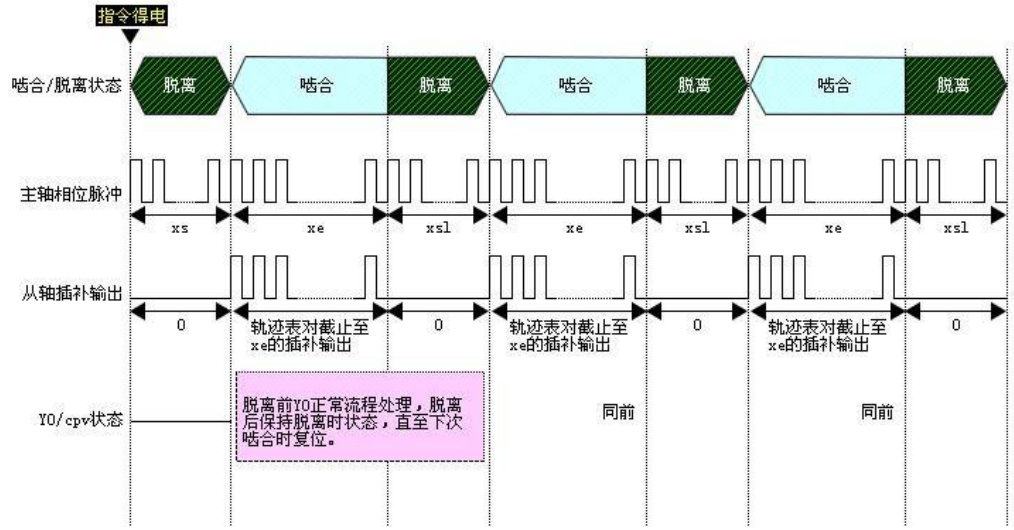
- 组合 J：

❖ 指令生效后，X0 为 ON 则啮合。
❖ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



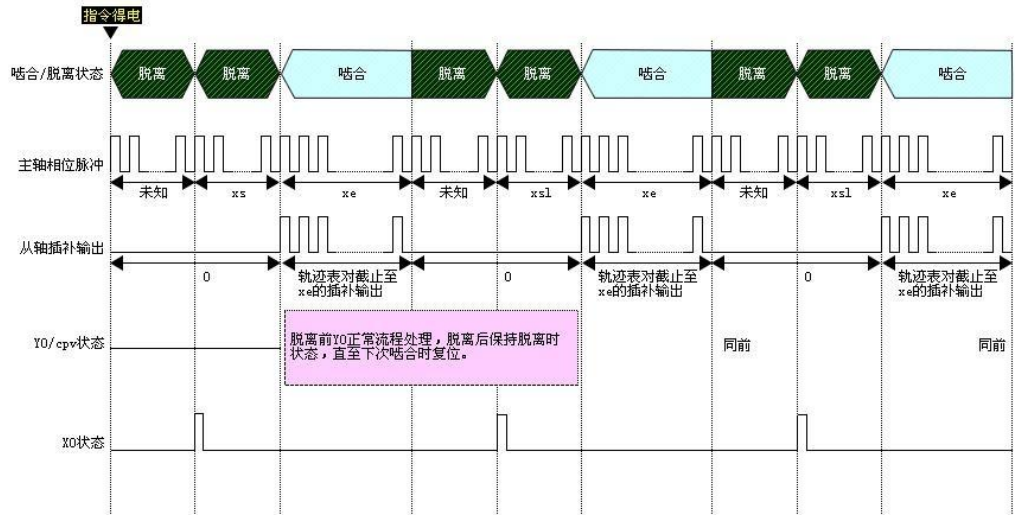
- 组合 K：

❖ 指令生效后，主轴走前置啮合位移量后则啮合。
❖ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



- 组合 L：

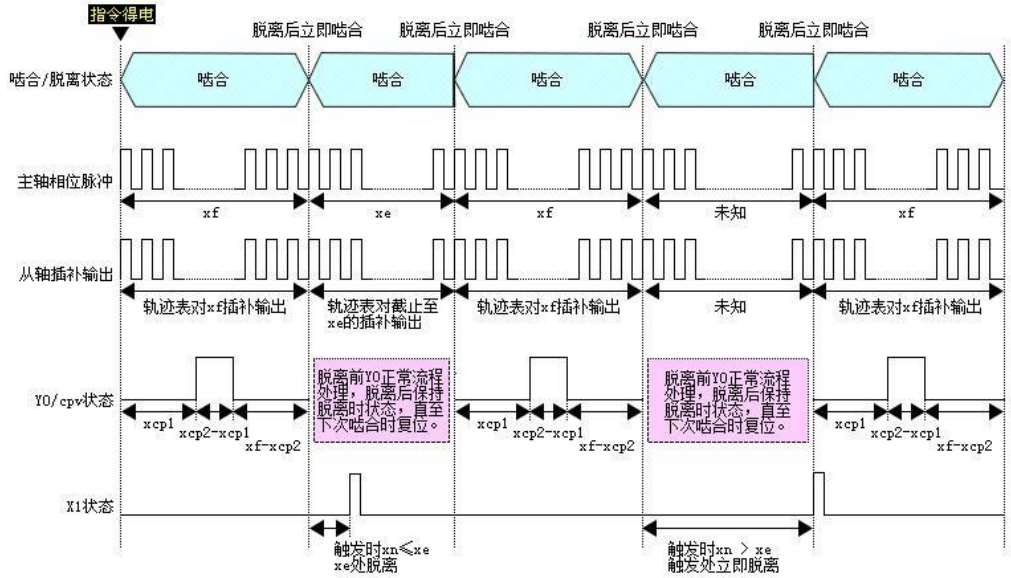
❖ 指令生效后，等待 X0 触发为ON，主轴再走前置啮合位移量后则啮合。
❖ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



- 组合 M：

❖ 指令生效则啮合。

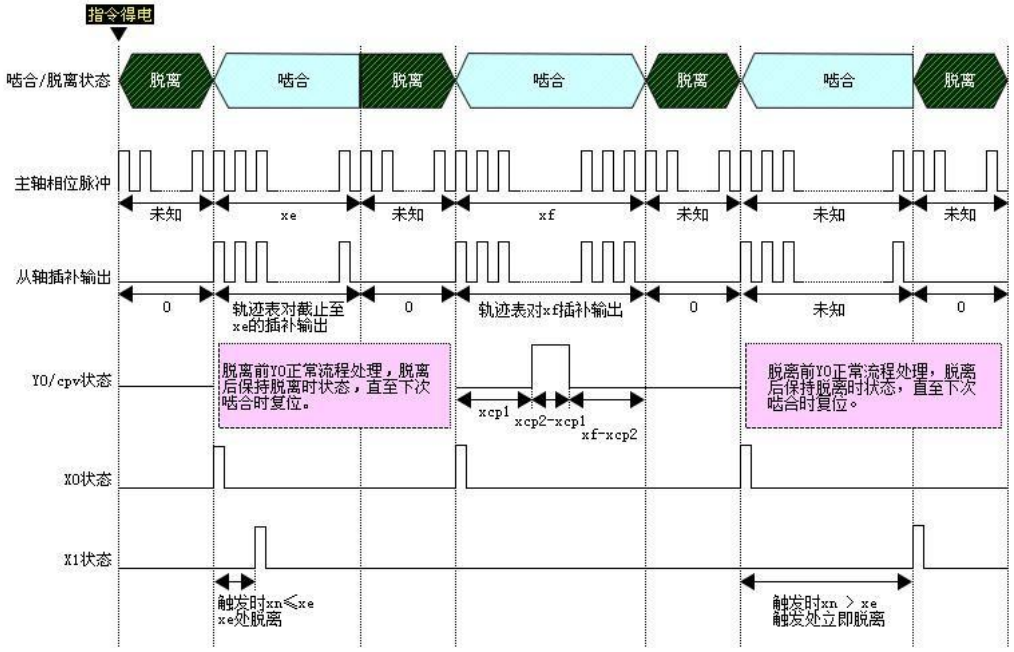
❖ X1 触发为 ON 后，到达主轴相位脱离点脱离。



- 组合 N：

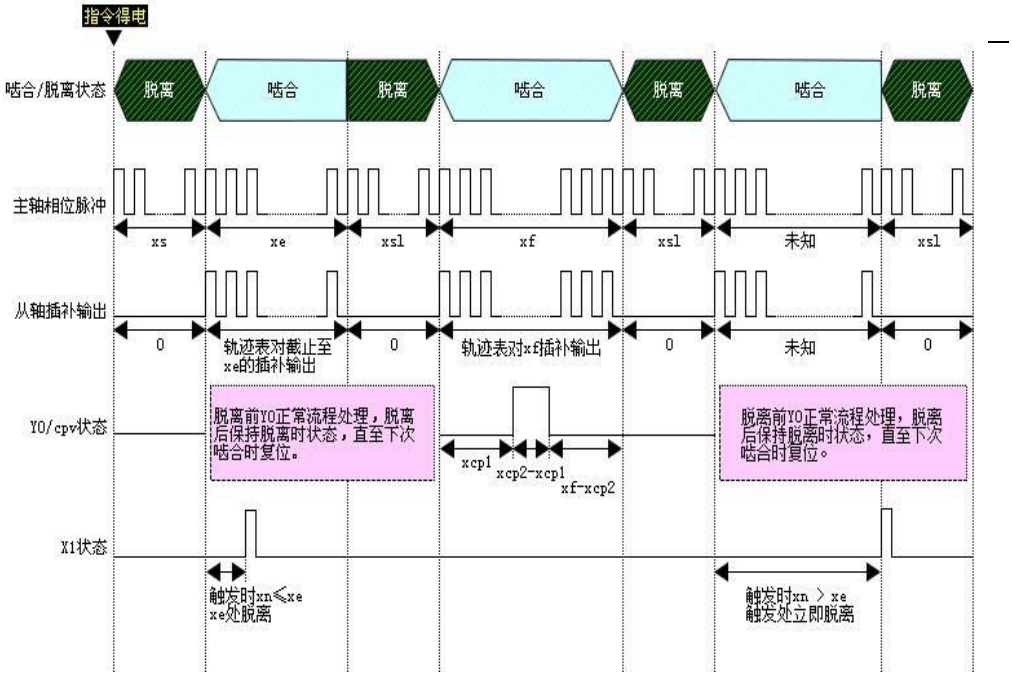
❖ 指令生效，X0 为 ON 则啮合。

❖ X1 触发为 ON 后，到达主轴相位脱离点脱离。



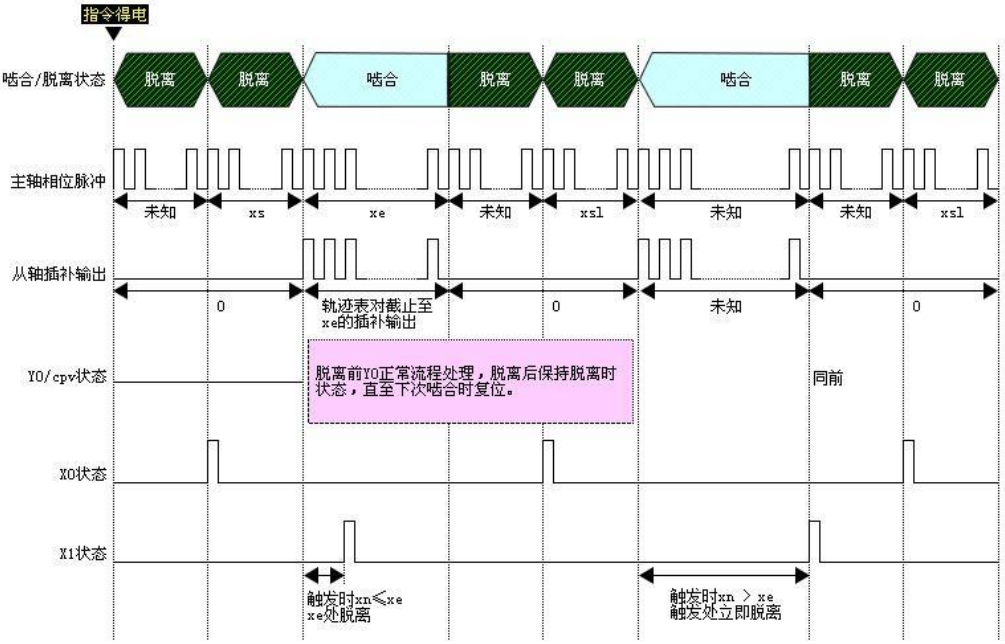
- 组合 O：

❖ 指令生效后，主轴走前置啮合位移量后则啮合。
❖ X1 触发为 ON 后，到达主轴相位脱离点脱离。



- 组合 P：

❖ 指令生效后，等待 X0 触发为ON，主轴再走前置啮合位移量后则啮合。
❖ X1 触发为 ON 后，到达主轴相位脱离点脱离。



- **追剪滚切模式分析**

常见追剪/滚切系统通常分为定长和色标两种模式，以下是模式分析。

定长	色标
啮合/脱离方式	
可选立即啮合，若需要前置偏移量，则选择前置啮合。 根据实际工况设定初始前置啮合位移和周期前置啮合位移。 根据设定的轨迹表曲线量程与工件的加工长度设计 脱离方式。若选择触发脱离，其触发点设定为从轴近 点或零点信号。 由于轨迹表设计起始点一般为零点处，故必须保证啮 合位置为从轴的原点处。	色标信号设定为触发啮合输入，若需要前置偏移量， 可选触发+前置啮合。根据实际工况设定初始前置啮合位移和周期前置啮合位移。 根据设定的轨迹表曲线量程与工件的加工长度设计脱离方式。若选择触发脱离，其触发点设定为从轴近点或零点信号。 由于轨迹表设计起始点一般为零点处，故必须保证啮 合位置为从轴的原点处。
主轴实际量程(凸轮周期)	
若运行过程中无脱离阶段(周期脱离后立即啮合，可视作主从轴始终啮合)，则应设计主轴周期长度为物料切割长度，以满足工艺周期循环的需要。 若设计脱离阶段，相位脱离位移和周期前置啮合位移应相互配合，满足两者之和等于物料的切割长度。	一般接入色标以后，从轴会有脱离等待阶段，设计主轴周期长度应≤物料切割长度。 脱离等待阶段的脉冲数等于物料切割长度对应主轴 脉冲数减去主轴周期长度脉冲数。
运行精度	
由于主轴相位仅仅通过位置的反馈确定，故长时间运 行可能导致主/从轴产生一定的相位偏差，此时应通过 调整相位偏差补偿参数消除误差。	每个凸轮周期均有触发啮合输入(色标检测)的过程， 故主从轴相位偏差在每个周期开始处将自动校正 1次。一般不需要补偿，精度高。

- **凸轮应用举例**

假定凸轮运行参数如下设置：

a. 主轴实际量程：1000

b. 初始前置啮合位移：650

c. 周期前置啮合位移：150

d. 相位脱离位移：850

e. 表格比较输出行 1 主轴相位：400

f. 表格比较输出行 2 主轴相位：600

g. 比较事件输出端口：Y0，极性为 0。

h. 输入触发啮合端口：X0，极性为 0。

i. 输入触发脱离端口：X1，极性为 0。

j. 啮合/脱离方式选择组合‘ P ’

设置凸轮输出参数的梯形图示例如下：



- **编程注意事项**

指令执行异常的情况有(指令不动作或动作错误)：

- ◆ 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- ◆ 设置轨迹表起始单元寄存器序号为非偶数地址。
- ◆ 设置主轴实际值单元寄存器序号为非偶数地址。

- 啮合/脱离相关设置与轨迹表设计相悖。比如轨迹表起始点规划为原点，但啮合点位置不在原点。
- 当脱离方式选择为触发脱离，且脱离时从轴速度较快，可能导致无法在该X端口处脱离，会略微过冲最大不超过当前速度1个ms的位移。
- 轴速度设置不合理导致轴参数轴错误代码报错。
- 在对应轴参数轴状态不为0的情况下执行指令。

上电PLC 运行时Y0～Y7端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应Y软元件不再关联实际端口。

3.6 两轴插补指令

两轴插补指令有绝对位置插补和相对位置插补，按照运动插补轨迹可分为两轴直线绝对位置插补、两轴直线相对位置插补、两轴顺圆弧绝对位置插补、两轴顺圆弧相对位置插补、两轴逆圆弧绝对位置插补、两轴逆圆弧相对位置插补。

3.6.1 设置坐标系[GCO]

执行插补指令前需要先设置坐标系。

- **指令格式**

1) 指令格式



2) 设定数据

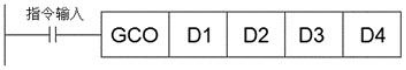
操作数种类	内容	数据类型
D1	坐标系号	BIN 32 位
D2	软件虚拟输出X轴号	BIN 32 位
D3	软件虚拟输出Y轴号	BIN 32 位
D4	软件虚拟输出Z轴号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•									
D2	•									
D3	•									
D4	•									

- **功能和举例**

该指令可在插补指令的执行前设置坐标系



1) 在D1 中指定坐标系

	设定范围
32 位运算	0~6

2) 在D2 中指定软件虚拟输出X轴

	设定范围
32 位运算	0~11

3) 在 D3 中指定软件虚拟输出Y轴

	设定范围
32 位运算	0~11

4) 在 D4 中指定软件虚拟输出Z轴

	设定范围
32 位运算	0~11

注： Z轴轴号大于30时，无Z轴。

- **注意要点**

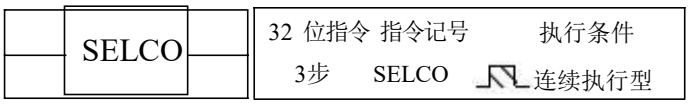
1) 设置坐标系指令要在插补指令执行前；

3.6.2 选择坐标系[SELCO]

设置坐标系后需要选择坐标系，在所选择的坐标系下执行插补指令。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	坐标系号	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•									

- **功能和举例**

该指令可在插补指令的执行前选择坐标系。



1) 在D1 中指定坐标系

	设定范围
32 位运算	0~6

- **注意要点**

1) 选择坐标系指令要在插补指令执行前；

3.6.3两轴直线绝对位置插补[G90G01]

两轴基于绝对位置以设定的直线合成脉冲输出频率，输出设定的直线插补轨迹。

- **指令格式**

4) 指令格式



5) 设定数据

操作数种类	内容	数据类型
D1	X轴目标位置的绝对值	BIN 32 位
D2	Y轴目标位置的绝对值	BIN 32 位
D3	直线插补合成脉冲频率	BIN 32 位

6) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							

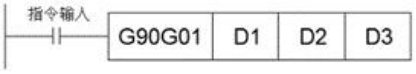
- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴	虚拟 4 轴	虚拟 5 轴	虚拟 6 轴	虚拟 7 轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的直线目标位置绝对值和直线的合成脉冲输出频率，进行两轴直线插补，从而使两轴运动到达指定的目标位置点。



4) 在D1 中指定X轴目标位置的绝对值

	设定范围
32 位运算	-16000000~16000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

5) 在D2 中指定Y轴目标位置的绝对值

	设定范围
32 位运算	-16000000~16000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

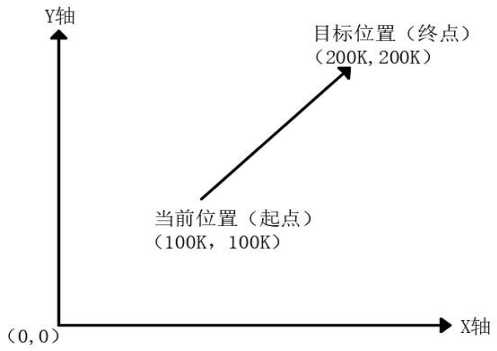
6) 在 D3 中指定直线插补合成脉冲频率

	设定范围
32 位运算	0~282800

注： D3指定的XY轴合成的输出脉冲频率，范围为0~282800Hz；合成频率分解到XY轴后，各轴输出脉冲频率范围为0~200,000Hz。频率设定值受轴最大速度限制，以及硬件端口频率限制。

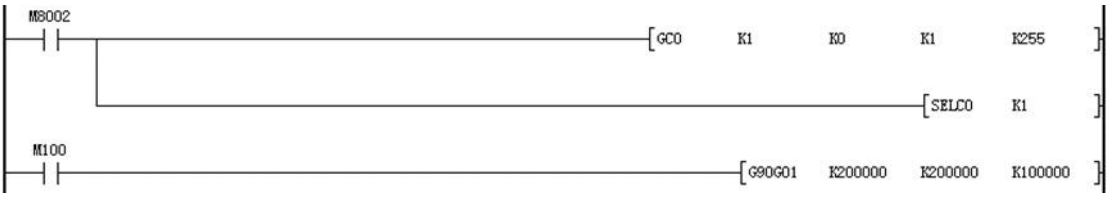
7) 案例

使用 G90G01 指令从当前位置坐标（100K,100K）直线插补到（200K，200K）的目标位置，合成速度为100khz。



需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	200000
D2	Y轴目标位置的绝对值	200000
D3	直线插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) 必须使用常开触点导通执行指令，断开触点则轴运动立即停止；

2) 指令的运行速度受指定虚拟轴最大速度限制；

3) 直线合成速度<=1.414*轴速度。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

5) 使用插补指令时，需合理设置轴加减速时间。

3.6.4两轴直线相对位置插补[G91G01]

两轴基于相对位置以设定的直线合成脉冲输出频率，输出设定的直线插补轨迹。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	目标位置在X轴相对当前位置的脉冲输出数差值	BIN 32 位
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	BIN 32 位
D3	直线插补合成脉冲频率	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							

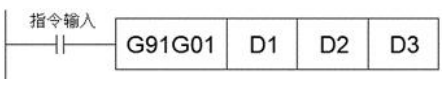
- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟 0 轴	虚拟 1 轴	虚拟 2 轴	虚拟 3 轴	虚拟 4 轴	虚拟 5 轴	虚拟 6 轴	虚拟 7 轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的直线目标位置相对当前位置的差值和直线的合成脉冲输出频率，进行两轴直线插补，从而使两轴运动到达指定的目标位置点。



1) 在D1 中指定目标位置在X轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-16000000~16000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) 在D2 中指定目标位置在Y轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-16000000~16000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

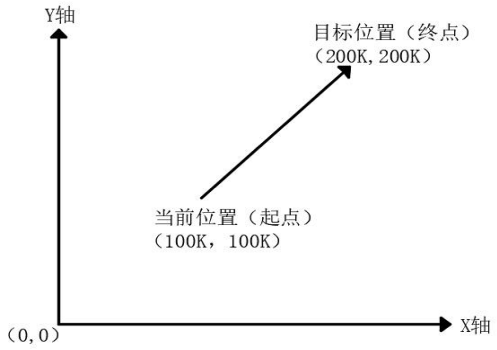
3) 在 D3 中指定直线插补合成脉冲频率

	设定范围
32 位运算	0~282800

注： D3指定的XY轴合成的输出脉冲频率，范围为0~282800Hz；合成频率分解到XY轴后，各轴输出脉冲频率范围为0~200,000Hz。频率设定值受轴最大速度限制，以及硬件端口频率限制。

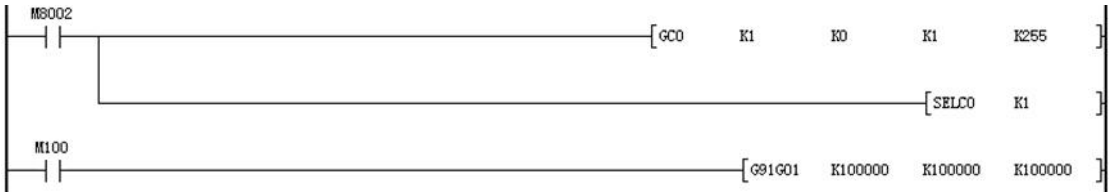
4) 案 例

使用 G91G01 指令从当前位置坐标（100K,100K）直线插补到（200K，200K）的目标位置，合成速度为100khz。



需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	100000
D3	直线插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) 必须使用常开触点导通执行指令，断开触点则轴运动立即停止；

2) 指令的运行速度受指定虚拟轴最大速度限制；

3) 直线合成速度<=1.414*轴速度。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

5) 使用插补指令时，需合理设置轴加减速时间。

3.6.5两轴顺圆弧绝对位置插补[G90G02]

两轴基于绝对位置以设定的圆弧合成脉冲输出频率，输出顺时针圆弧插补轨迹。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	X轴目标位置的绝对值	BIN 32 位
D2	Y轴目标位置的绝对值	BIN 32 位
D3	圆心在X轴上的绝对值，或半径R脉冲数	BIN 32 位
D4	圆心在Y轴上的绝对值，若D3表示半径R脉冲数，D4必须是H7FFFFFFF	BIN 32 位
D5	顺圆弧插补合成脉冲频率	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							
D5	•	•	•							

- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟0轴	虚拟1轴	虚拟2轴	虚拟3轴	虚拟4轴	虚拟5轴	虚拟6轴	虚拟7轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的顺时针圆弧目标位置绝对值、圆心位置绝对值和圆弧的合成脉冲输出频率，进行两轴圆弧插补，从而使两轴运动到达指定的目标位置点。



1) 在D1 中指定X轴目标位置的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) 在D2 中指定Y轴目标位置的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

3) 在 D3 中指定圆心在X轴上的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D3 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3可以表示半径R的脉冲数。

4) 在 D4 中指定圆心在Y轴上的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D4 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3表示半径R脉冲数时，D4必须为H7FFFFFFF。

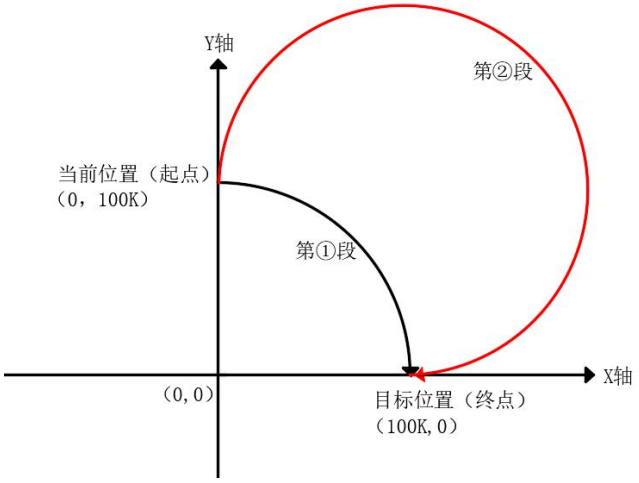
5) 在D5 中指定顺圆弧插补合成脉冲频率。

	设定范围
32 位运算	0~200000

注：D5表示顺圆弧插补合成脉冲频率<=轴速度。

6) 案例

使用 G90G02 指令从当前位置坐标（0,100K）顺时针圆弧插补到（100K，0）的目标位置，合成速度为100khz。



第①段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆心在X轴上的绝对值	0
D4	圆心在Y轴上的绝对值	0
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



第①段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆弧半径R脉冲数	100000
D4	H7FFFFFFF	H7FFFFFFF
D5	顺圆弧插补合成脉冲频率	100000

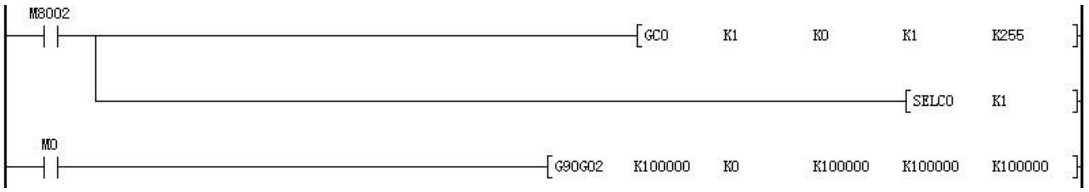
示例梯形图：



第②段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆心在X轴上的绝对值	100000
D4	圆心在Y轴上的绝对值	100000
D5	顺圆弧插补合成脉冲频率	100000

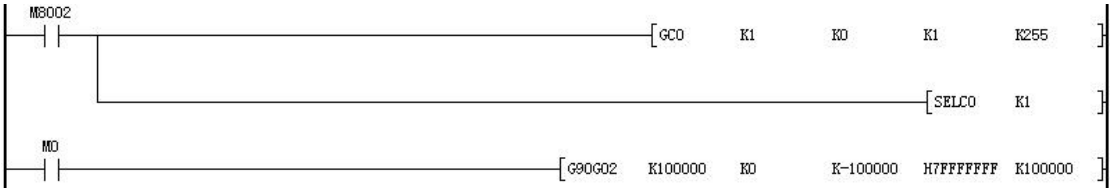
示例梯形图：



第②段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆弧半径R脉冲数	-100000
D4	H7FFFFFFF	H7FFFFFFF
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) D1，D2为圆弧的目标位置绝对值，用户需合理设置目标位置，确保能够生成正确的圆弧轨迹。当指定的XY轴目标位置与当前XY轴位置相同时，生成轨迹为一个整圆。

2) D3,D4有两种设置模式，圆心模式：D3，D4可以设置为圆心在X，Y轴上的绝对值。半径模式：D3表示半径R脉冲数,D4为H7FFFFFFF。

3) 半径模式下，R<0表示是大于180度的圆弧，R>0表示是小于180度的圆弧。在半径模式下，存在无穷多解所以不能生成整圆。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

5) 使用插补指令时，需合理设置轴加减速时间。

3.6.6两轴顺圆弧相对位置插补[G91G02]

两轴基于相对位置以设定的圆弧合成脉冲输出频率，输出顺时针圆弧插补轨迹。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	目标位置在X轴相对当前位置的脉冲输出数差值	BIN 32 位
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	BIN 32 位
D3	圆心在X轴相对当前位置的脉冲输出数差值或半径R脉冲数	BIN 32 位
D4	圆心在Y轴相对当前位置的脉冲输出数差值，若D3表示半径R脉冲数，D4必须是H7FFFFFFF	BIN 32 位
D5	顺圆弧插补合成脉冲频率	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							
D5	•	•	•							

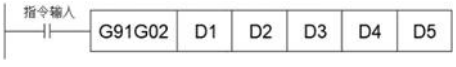
- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟0轴	虚拟1轴	虚拟2轴	虚拟3轴	虚拟4轴	虚拟5轴	虚拟6轴	虚拟7轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的顺时针圆弧目标位置的相对位置、圆心的相对位置，和圆弧的合成脉冲输出频率，进行两轴圆弧插补，从而使两轴运动到达指定的目标位置点。



1) 在D1 中指定目标位置在X轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) 在D2 中指定目标位置在Y轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

3) 在 D3 中指定圆心在X轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D3 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3可以表示半径R的脉冲数。

4) 在 D4 中指定圆心在Y轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D4 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3表示半径R脉冲数时，D4必须为H7FFFFFFF。

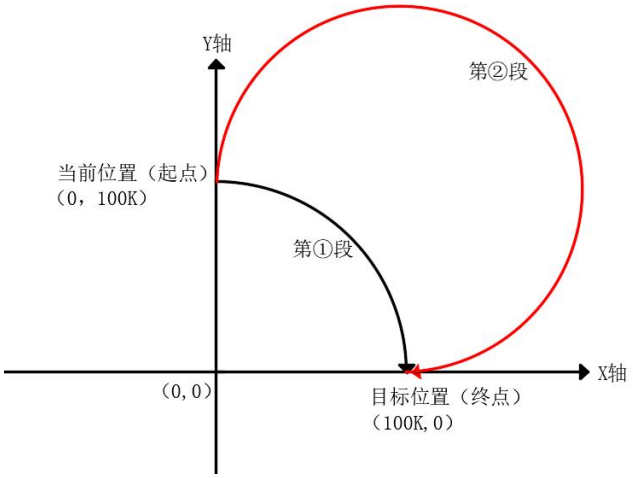
5) 在D5 中指定顺圆弧插补合成脉冲频率。

	设定范围
32 位运算	0~200000

注：D5表示顺圆弧插补合成脉冲频率<=轴速度。

6) 案例

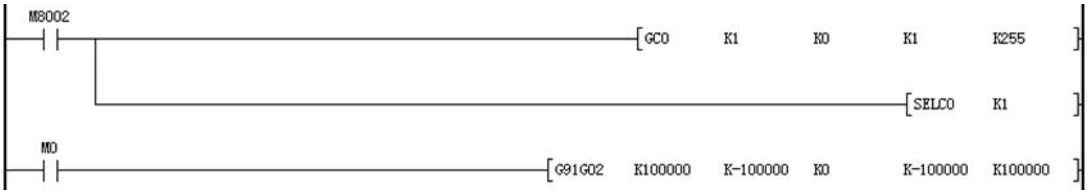
使用 G91G02 指令从当前位置坐标（0,100K）顺时针圆弧插补到（100K，0）的目标位置，合成速度为100khz。



第①段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆心在X轴相对当前位置的脉冲输出数差值	0
D4	圆心在Y轴相对当前位置的脉冲输出数差值	-100000
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



第①段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆弧半径R脉冲数	100000
D4	H7FFFFFFF	H7FFFFFFF
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



第②段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆心在X轴相对当前位置的脉冲输出数差值	100000
D4	圆心在Y轴相对当前位置的脉冲输出数差值	0
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



第②段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆弧半径R脉冲数	-100000
D4	H7FFFFFFF	H7FFFFFFF
D5	顺圆弧插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) D1，D2为圆弧的目标位置的相对位置，用户需合理设置目标位置，确保能够生成正确的圆弧轨迹。当D1=D2=0时，生成轨迹为一个整圆。

2) D3,D4有两种设置模式，圆心模式：D3，D4可以设置为圆心在X，Y轴上相对于当前位置的脉冲数差值。半径模

式：D3表示半径R脉冲数,D4为H7FFFFFFF。

3) 半径模式下，R<0表示是大于180度的圆弧，R>0表示是小于180度的圆弧。在半径模式下，存在无穷多解所以不能生成整圆。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

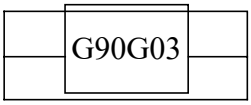
5) 使用插补指令时，需合理设置轴加减速时间。

3.6.7两轴逆圆弧绝对位置插补[G90G03]

两轴基于绝对位置以设定的圆弧合成脉冲输出频率，输出逆时针圆弧插补轨迹。

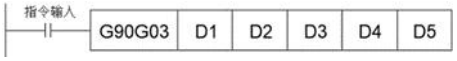
- **指令格式**

1) 指令格式



32 位指令 指令记号 执行条件

16步 G90G03 连续执行型



2) 设定数据

操作数种类	内容	数据类型
D1	X轴目标位置的绝对值	BIN 32 位
D2	Y轴目标位置的绝对值	BIN 32 位
D3	圆心在X轴上的绝对值，或半径R脉冲数	BIN 32 位
D4	圆心在Y轴上的绝对值，若D3表示半径R脉冲数，D4必须是H7FFFFFFF	BIN 32 位
D5	顺圆弧插补合成脉冲频率	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							
D5	•	•	•							

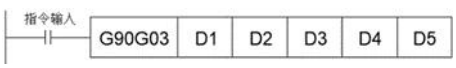
- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟0轴	虚拟1轴	虚拟2轴	虚拟3轴	虚拟4轴	虚拟5轴	虚拟6轴	虚拟7轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的逆时针圆弧目标位置绝对值、圆心位置绝对值和圆弧的合成脉冲输出频率，进行两轴圆弧插补，从而使两轴运动到达指定的目标位置点。



1) 在D1 中指定X轴目标位置的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) 在D2 中指定Y轴目标位置的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

3) 在 D3 中指定圆心在X轴上的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D3 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3可以表示半径R的脉冲数。

4) 在 D4 中指定圆心在Y轴上的绝对值

	设定范围
32 位运算	-5000000~5000000

注：当 D4 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3表示半径R脉冲数时，D4必须为H7FFFFFFF。

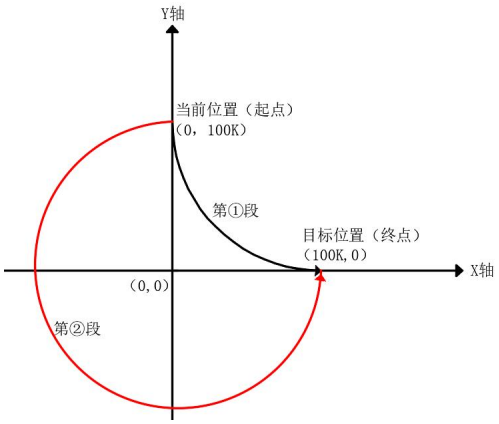
5) 在D5 中指定逆圆弧插补合成脉冲频率。

	设定范围
32 位运算	0~200000

注：D5表示逆圆弧插补合成脉冲频率<=轴速度。

6) 案例

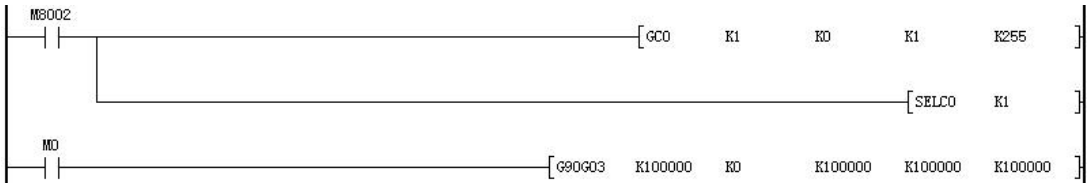
使用 G90G03 指令从当前位置坐标（0,100K）逆时针圆弧插补到（100K,0）的目标位置，合成速度为100khz。



第①段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆心在X轴上的绝对值	100000
D4	圆心在Y轴上的绝对值	100000
D5	逆圆弧插补合成脉冲频率	100000

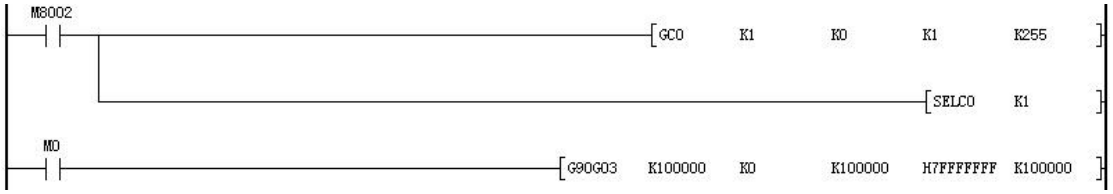
示例梯形图：



第①段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆弧半径R脉冲数	100000
D4	H7FFFFFFF	H7FFFFFFF
D5	逆圆弧插补合成脉冲频率	100000

示例梯形图：



第②段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值

D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆心在X轴上的绝对值	0
D4	圆心在Y轴上的绝对值	0
D5	逆圆弧插补合成脉冲频率	100000

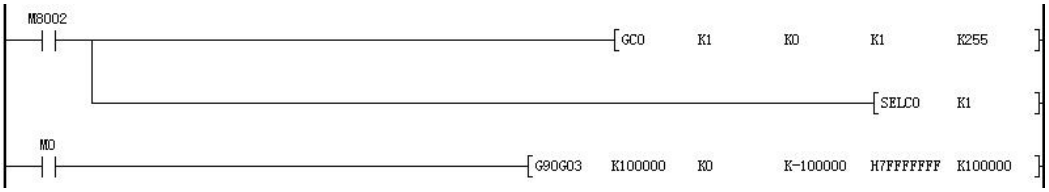
示例梯形图：



第②段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	X轴目标位置的绝对值	100000
D2	Y轴目标位置的绝对值	0
D3	圆弧半径R脉冲数	-100000
D4	H7FFFFFFF	H7FFFFFFF
D5	逆圆弧插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) D1，D2为圆弧的目标位置绝对值，用户需合理设置目标位置，确保能够生成正确的圆弧轨迹。当指定的XY轴目标位置与当前XY轴位置相同时，生成轨迹为一个整圆。

2) D3,D4有两种设置模式，圆心模式：D3，D4可以设置为圆心在X，Y轴上的绝对值。半径模式：D3表示半径R脉冲数,D4为H7FFFFFFF。

3) 半径模式下，R<0表示是大于180度的圆弧，R>0表示是小于180度的圆弧。在半径模式下，存在无穷多解所以不能生成整圆。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

5) 使用插补指令时，需合理设置轴加减速时间。

3.6.8两轴逆圆弧相对位置插补[G91G03]

两轴基于相对位置以设定的圆弧合成脉冲输出频率，输出逆时针圆弧插补轨迹。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	目标位置在X轴相对当前位置的脉冲输出数差值	BIN 32 位
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	BIN 32 位
D3	圆心在X轴相对当前位置的脉冲输出数差值或半径R脉冲数	BIN 32 位
D4	圆心在Y轴相对当前位置的脉冲输出数差值，若D3表示半径R脉冲数，D4必须是H7FFFFFFF	BIN 32 位
D5	逆圆弧插补合成脉冲频率	BIN 32 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							
D2	•	•	•							
D3	•	•	•							
D4	•	•	•							
D5	•	•	•							

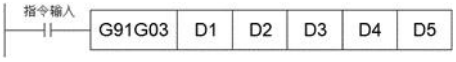
- **相关轴参数**

软元件编号								名称	数据类型	初始值
虚拟0轴	虚拟1轴	虚拟2轴	虚拟3轴	虚拟4轴	虚拟5轴	虚拟6轴	虚拟7轴			
R334	R384	R434	R484	R534	R584	R634	R684	轴加速时间	16 位	50
R335	R385	R435	R485	R535	R585	R635	R685	轴减速时间	16 位	50

注：轴加减速时间这两个参数仅可在插补指令中使用，需合理设置。

- **功能和举例**

该指令按指定的逆时针圆弧目标位置的相对位置、圆心的相对位置，和圆弧的合成脉冲输出频率，进行两轴圆弧插补，从而使两轴运动到达指定的目标位置点。



1) 在D1 中指定目标位置在X轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D1 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

2) 在D2 中指定目标位置在Y轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D2 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。

3) 在 D3 中指定圆心在X轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D3 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3可以表示半径R的脉冲数。

4) 在 D4 中指定圆心在Y轴相对当前位置的脉冲输出数差值

	设定范围
32 位运算	-5000000~5000000

注：当 D4 填写为负值，且配置了方向端口时，则为反方向脉冲输出。没有指定方向端口，则负值不起作用(按正值输出)。D3表示半径R脉冲数时，D4必须为H7FFFFFFF。

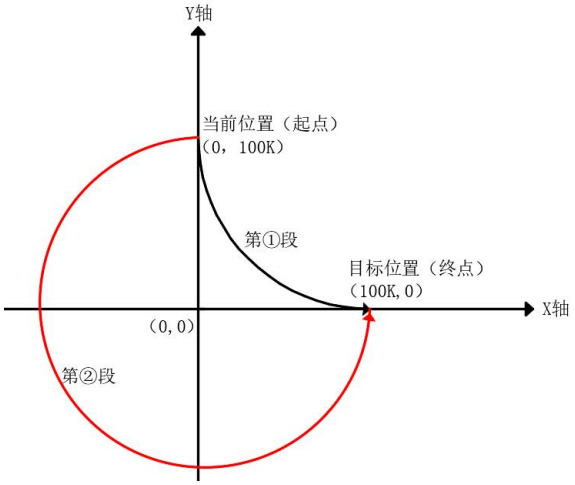
5) 在D5 中指定逆圆弧插补合成脉冲频率。

	设定范围
32 位运算	0~200000

注：D5表示逆圆弧插补合成脉冲频率<=轴速度。

6) 案例

使用 G91G03 指令从当前位置坐标（100K，0）逆时针圆弧插补到（0，100K）的目标位置，合成速度为100khz。



第①段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆心在X轴相对当前位置的脉冲输出数差值	100000
D4	圆心在Y轴相对当前位置的脉冲输出数差值	0
D5	逆圆弧插补合成脉冲频率	100000

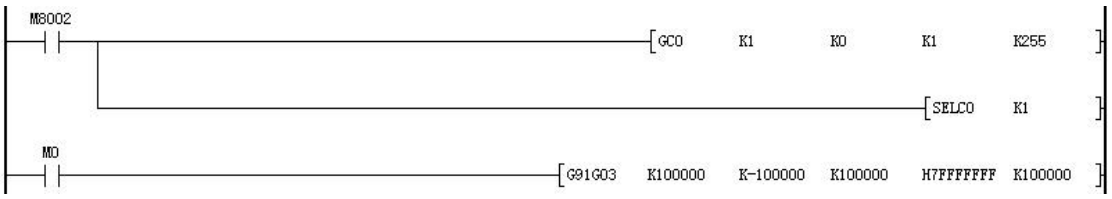
示例梯形图：



第①段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆弧半径R脉冲数	100000
D4	H7FFFFFFF	H7FFFFFFF
D5	逆圆弧插补合成脉冲频率	100000

示例梯形图：



第②段圆弧使用圆心模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆心在X轴相对当前位置的脉冲输出数差值	0
D4	圆心在Y轴相对当前位置的脉冲输出数差值	-100000
D5	逆圆弧插补合成脉冲频率	100000

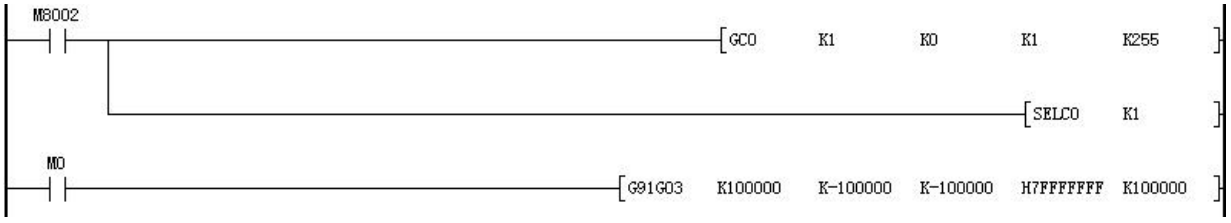
示例梯形图：



第②段圆弧使用半径模式如下：

需要配置的参数	定义	配置值
D1	目标位置在X轴相对当前位置的脉冲输出数差值	100000
D2	目标位置在Y轴相对当前位置的脉冲输出数差值	-100000
D3	圆弧半径R脉冲数	-100000
D4	H7FFFFFFF	H7FFFFFFF
D5	逆圆弧插补合成脉冲频率	100000

示例梯形图：



- **注意要点**

1) D1，D2为圆弧的目标位置的相对位置，用户需合理设置目标位置，确保能够生成正确的圆弧轨迹。当D1=D2=0时，生成轨迹为一个整圆。

2) D3,D4有两种设置模式，圆心模式：D3，D4可以设置为圆心在X，Y轴上相对于当前位置的脉冲数差值。半径模式：D3表示半径R脉冲数,D4为H7FFFFFFF。

3) 半径模式下，R<0表示是大于180度的圆弧，R>0表示是小于180度的圆弧。在半径模式下，存在无穷多解所以不能生成整圆。

4) 使用插补指令时，轴参数设置以X轴的轴参数为准。

5) 使用插补指令时，需合理设置轴加减速时间。

3.7 FLASH指令

智能伺服使用一块128MBytes的FLASH做数据扩展，可通过相关梯形图指令实现文件的查询，读写及删除等操作。系统将FLASH划分为64个扇区，每个扇区占用2MBytes字节。系统总共可创建16个文件，每个文件可占用1个或多个扇区，文件序号为0~15。

文件操作指令一览表：

指令名称	指令功能
FFMT	FLASH格式化
FDIR	FLASH状态查看
FDEL	FLASH删除文件
FOP	FLASH打开文件
FCLS	FLASH关闭文件
FRD	FLASH读文件
FWR	FLASH写文件

3.7.1 FLASH参数定义

FLASH参数定义：

<table>
<tr><th>地址</th><th>参数描述</th><th>操作属性</th></tr>
<tr><td>R2000</td><td>当前操作的FLASH操作码
<table>
<tr><th>R2000 =</th><th>R2000寄存器含义</th></tr>
<tr><td>9001</td><td>FLASH 格式化</td></tr>
<tr><td>9002</td><td>FLASH状态查看</td></tr>
<tr><td>9003</td><td>FLASH删除文件</td></tr>
<tr><td>9004</td><td>FLASH打开文件</td></tr>
<tr><td>9005</td><td>FLASH关闭文件</td></tr>
<tr><td>9006</td><td>FLASH读文件</td></tr>
<tr><td>9007</td><td>FLASH写文件</td></tr>
</table></td><td>读写</td></tr>
<tr><td>R2001</td><td>根据R2000的操作码不同，对应的含义也不一样。
<table>
<tr><th>R2000 =</th><th>R2001寄存器含义</th></tr>
<tr><td>9001</td><td>无意义</td></tr>
<tr><td>9002</td><td>无意义</td></tr>
<tr><td>9003</td><td>指定删除的文件号</td></tr>
<tr><td>9004</td><td>指定打开的文件号</td></tr>
<tr><td>9005</td><td>无意义</td></tr>
<tr><td>9006</td><td>指定读取数据的文件号</td></tr>
<tr><td>9007</td><td>指定写入数据的文件号</td></tr>
</table></td><td>读写</td></tr>
<tr><td>R2002</td><td>根据R2000的操作码不同，对应的含义也不一样。
<table>
<tr><th>R2000 =</th><th>R2002寄存器含义</th></tr>
<tr><td>9001</td><td>无意义</td></tr>
<tr><td>9002</td><td>无意义</td></tr>
<tr><td>9003</td><td>无意义</td></tr>
<tr><td>9004</td><td>文件打开方式：
=0，创建文件
=1，以读方式打开文件
=2，以写方式打开文件</td></tr>
</table></td><td>读写</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>9005</td><td>无意义</td></tr>
<tr><td>9006</td><td>指定读取的字数</td></tr>
<tr><td>9007</td><td>指定写入的字数</td></tr>
</table></td><td></td></tr>
<tr><td>R2003</td><td>根据R2000的操作码不同，对应的含义也不一样。<table>
<tr><th>R2000 =</th><th>R2003寄存器含义</th></tr>
<tr><td>9001</td><td>无意义</td></tr>
<tr><td>9002</td><td>无意义</td></tr>
<tr><td>9003</td><td>无意义</td></tr>
<tr><td>9004</td><td>无意义</td></tr>
<tr><td>9005</td><td>无意义</td></tr>
<tr><td>9006</td><td>指定读取数据存放的寄存器地址</td></tr>
<tr><td>9007</td><td>指定写入数据存放的寄存器地址</td></tr>
</table></td><td>读写</td></tr>
<tr><td>R2004~R2006</td><td>保留</td><td>-</td></tr>
<tr><td>R2007</td><td>FLASH操作错误码
错误码详情见本章第3.7.9章节</td><td>只读</td></tr>
<tr><td>R2008</td><td>指令控制状态：
=0，无操作
=4，启动操作</td><td>读写</td></tr>
<tr><td>R2009</td><td>指令执行状态：
=0，空闲状态
=16，正在执行操作
=48，操作完成
=112，操作失败
*操作失败时，可观测R2007查看出错原因。</td><td>读写</td></tr>
<tr><td>R2010~R2041</td><td>FLASH的扇区状态，一共32个R，共64个字节，对应64个扇区的状态。每个字节的数值对应扇区的状态如下：
=0xFF，为空闲扇区
=0xFE，为坏扇区
=0xFD，为管理扇区
=0xFC，为未知扇区
=0，1，...，15，为对应文件的占用扇区
● FLASH扇区存在一定的损坏几率，通常是FLASH厂商出厂自带的损坏，或长时间使用磨损造成的。文件操作时，会自动避开所检测到的坏扇区。
● 当显示为未知扇区，则是出厂第一次使用FLASH的情况。执行一次格式化操作后，未知扇区状态将被更新。</td><td>只读</td></tr>
<tr><td>R2042</td><td>当前处于打开状态的文件号。
若当前没有文件处于打开状态，该数值为0xFF</td><td>只读</td></tr>
<tr><td>R2043</td><td>当前的操作指令类型。<table>
<tr><th>R2043 =</th><th>R2043寄存器含义</th></tr>
<tr><td>9001</td><td>FLASH 格式化</td></tr>
<tr><td>9002</td><td>FLASH状态查看</td></tr>
<tr><td>9003</td><td>FLASH删除文件</td></tr>
<tr><td>9004</td><td>FLASH打开文件</td></tr>
<tr><td>9005</td><td>FLASH关闭文件</td></tr>
<tr><td>9006</td><td>FLASH读文件</td></tr>
</table></td><td>只读</td></tr>
</table>

	9007 FLASH写文件	
R2044(L)	当前打开的文件总长度 若当前没有文件处于打开状态，该数值为0	只读
R2046(L)	当前打开的文件位置 若当前没有文件处于打开状态，该数值为0	读写
R2048(L)	当前打开的文件所占的最后1个扇区的剩余空闲字节数	只读

用户可通过操作和观测上述R寄存器，来对FLASH进行文件操作。具体的流程如下所示：



当然，为了简化使用，推荐使用FLASH文件操作指令来进行操作。

3.7.2 FLASH格式化[FFMT]

格式化FLASH，清除所有文件。

- **指令格式**

1) 指令格式



- **功能和举例**

该指令用于擦除FLASH内，除管理扇区外的所有扇区数据，同时也删除所有文件信息。



- **注意要点**

1) 该指令执行时间较长，需要3秒左右的时间。
2) 该指令执行过程中，会尝试擦除已被标记为坏扇区的扇区(扇区状态为0xFE)，若能擦除成功，则该扇区状态恢复为空闲状态(扇区状态为0xFF)。
3) 该指令执行完成后，FLASH内所有的文件将被删除，系统处于无文件打开的状态。
4) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

3.7.3 FLASH状态查看[FDIR]

查看FLASH的扇区状态，以及当前打开的文件状态。

- **指令格式**

1) 指令格式



- **功能和举例**

该指令用于查看当前FLASH的扇区状态，和当前使用的文件状态。



- **注意要点**

1) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

3.7.4 FLASH删除文件[FDEL]

删除指定的文件，释放文件所占用的扇区。

- **指令格式**

1) 指令格式



2) 设定数据

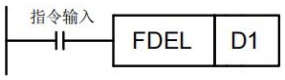
操作数种类	内容	数据类型
D1	文件号	BIN 16位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	•	•	•							

- **功能和举例**

该指令用于删除FLASH内指定的文件数据。



1) 在D1 中指定文件序号

	设定范围
16 位运算	0~15

- **注意要点**

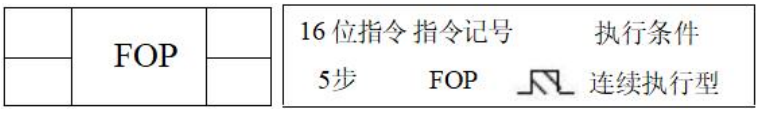
1) D1所指定的文件号，必须有相应的文件存在。
2) 执行该条语句前，确认当前没有打开的文件。
3) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

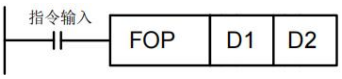
3.7.5 FLASH打开文件[FOP]

打开或者创建指定的文件。

- **指令格式**

1) 指令格式





2) 设定数据

操作数种类	内容	数据类型
D1	文件号	BIN 16 位

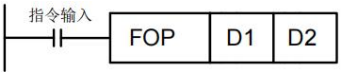
D2	打开方式	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●	●	●							
D2	●	●	●							

- **功能和举例**

该指令用于打开或创建文件，打开方式分为读模式和写模式，执行创建模式除了创建文件之外，也可以进行写文件数据。



1) 在D1 中指定文件序号

	设定范围
16 位运算	0~15

2) 在D2 中指定打开类型

	设定范围
16 位运算	0~2 0:创建文件 1:读模式打开文件 2:写模式打开文件

- **注意要点**

1) D1所指定的文件号，如果D2=0，则不能有相应的文件存在；如果D2=1或2，则必须有相应的文件存在。

2) 执行该指令前，确认当前没有打开的文件。

4) 如果是以D2=2（写模式）打开文件，指令执行完毕后，当前文件长度R2044(L)若不为512的倍数，则不允许向该文件续写数据，系统报错。

5) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

3.7.6 FLASH关闭文件[FCLS]

关闭指定的文件。

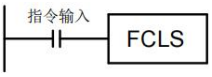
- **指令格式**

1) 指令格式



- **功能和举例**

该指令用于关闭当前正在打开的文件。



- **注意要点**

1) 执行该指令前，确认当前有1个正在打开的文件。

2) 如果之前打开的文件模式是创建或写模式，在进行相关数据录入后，在系统掉电前必须执行关闭文件的操作，否则会导致该文件损坏。建议在进行创建或写文件，梯形图里设置1个标志位，关闭文件后复位该标志位。如果在上电时，检测到该标志位没有复位，说明可能在上次系统没有关闭文件的情形下，发生了掉电，此时应对这个异常文件进行删除操作。

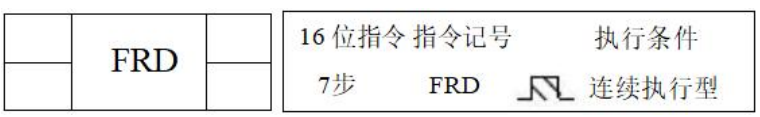
3) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

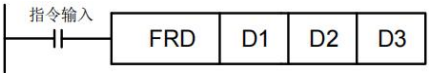
3.7.7 FLASH读文件[FRD]

对指定的文件读取数据。

- **指令格式**

1) 指令格式





2) 设定数据

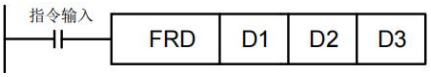
操作数种类	内容	数据类型
D1	文件号	BIN 16 位
D2	读出字数	BIN 16 位
D3	指向读取数据的D起始序号	BIN 16 位

3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●	●	●							
D2	●	●	●							
D3	●	●	●							

- **功能和举例**

该指令用于对以读模式打开的文件，读取相应位置的数据，并存储于D3所指定的D软元件区域内。



1) 在D1 中指定文件序号

	设定范围
16 位运算	0~15

2) 在D2 中指定读取字数

	设定范围
16 位运算	1~64

3) 在 D3 中指定指向读取数据的D起始序号

	设定范围
16 位运算	0~7999

- **注意要点**

1) 执行该指令前，确认当前有1个正在打开的文件，该文件序号与该指令的D1相同，且为读模式打开。

2) 指定的读取字数加上D3的数值，范围不要超过D7999。

3) 在指令执行前，可修改当前的文件位置R2046(L)，实现对任意位置的读取。

4) 指令执行时，会从当前的文件位置R2046(L),读取D2个字，然后存入D3所指定的D软元件区域，注意读取的文件位置R2046(L)不要超过文件的长度R2044(L)，否则会报错。

5) 指令执行完成后，当前的文件位置R2046(L)会自动加上本次读取的字数。

6) 当所有的读取任务完成后，记得关闭该文件。

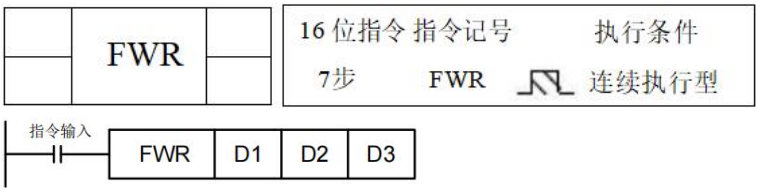
7) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

3.7.8 FLASH写文件[FWR]

对指定的文件写入数据。

- **指令格式**

1) 指令格式



2) 设定数据

操作数种类	内容	数据类型
D1	文件号	BIN 16 位
D2	写入字数	BIN 16 位
D3	指向写入数据的D起始序号	BIN 16 位

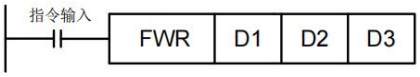
3) 对象软元件

操作数种类	位/字软元件									
	K/H	D	R	C	T	KnX	KnY	KnM	KnS	V/Z
D1	●	●	●							
D2	●	●	●							
D3	●	●	●							

- **功能和举例**

该指令用于对以创建模式或写模式打开的文件，从D3所指定的D软元件区域获取写入的数据，并从文件尾端进

行数据的写入。



1) 在D1 中指定文件序号

	设定范围
16 位运算	0~15

2) 在D2 中指定写入字数

	设定范围
16 位运算	1~64

3) 在 D3 中指定指向写入数据的D起始序号

	设定范围
16 位运算	0~7999

- **注意要点**

1) 执行该指令前，确认当前有1个正在打开的文件，该文件序号与该指令的D1相同，且该文件以创建模式或写模式打开。

2) 指定的写入字数加上的D3的数值，范围不要超过D7999。

3) 指令执行时，会从D3指定的D软元件区域，取出需要写入的数据，然后将这些数据存入到当前文件的尾端。

4) 在执行写入操作前，不要操作当前的文件位置R2046(L)，在写文件情况下，该值始终等于文件的长度R2044(L)。

5) 指令执行完成后，当前的文件位置R2046(L)会自动加上本次写入的字数，同时文件的长度R2044(L)也进行相应的增加。当前文件所占扇区的剩余容量大小R2048(L)，也会进行相应的缩减，当减至0后，系统会自动将下一个空闲扇区作为本文件占用扇区。

6) 当所有的数据写入完后，记得关闭该文件。

7) 该指令执行中，观测指令执行状态R2009，为48或112时说明指令执行完成，此时可断开前置条件。指令执行完成后，对应的R2010~R2041,R2042~R2048(L)将被更新。

3.7.9 FLASH 错误码

FLASH报警码（R2007）	类型	详情
0	无错误	-
1	ECC 校验错误	文件已损坏，删除该文件
2	擦除错误	扇区已损坏，系统记录为坏扇区
3	写 FLASH 错误	文件已损坏，删除该文件
4	读 FLASH 的 ID 错误	未检测到 FLASH，更换硬件
5	管理区错误	FLASH 已损坏，更换硬件
6	FLASH 未格式化	执行 FFMT 格式化指令
7	不能识别的 FLASH 操作码	R2000 需设置正确的操作码
8	扇区已满	删除不需要的文件，腾出空间
9	扇区匹配错误	文件已损坏，删除该文件
10	上一个 FLASH 操作尚未执行完成	等待 R2009 变为 48 或者 112，再执行本次操作
11	当前已有文件处于打开状态	先执行关闭文件，再执行本次操作
12	没有文件处于打开状态	先打开相应的文件，再执行读/写/关闭操作
13	文件号错误	指定的文件号必须是 0~15
14	文件已损坏	文件已损坏，删除该文件
15	当前打开的文件序号和指令操作的文件序号不一致	执行读/写操作时指定的文件号，必须与当前打开的文件号相同
16	文件不存在	执行删除/打开(读/写模式)/读/写文件时，指定的文件号没有被创建过
17	文件已存在	执行打开文件时，类型为创建，但该文件已经被创建过
18	文件不为读模式	文件打开方式为创建或写模式，不能进行读数据
19	文件不为写模式	文件打开方式为读模式，不能进行写数据
20	文件操作字节数错误	文件操作字节数不能超过 128 个字节，即 64 个字

21	文件操作地址错误	● 读模式 指定的文件操作地址超出了文件的长度，或操作地址加上操作字节数超出了文件的长度 ● 写模式 指定的文件操作地址不为文件的末尾，或操作地址不为 512 的倍数
22	文件读写填充地址错误	指定的软元件地址加上操作的字数超出了 D7999
23	初始化未完成	等待FLASH初始化完成后，再进行操作 * FLASH 初始化大约 3 秒左右的时间，建议 PLC 上电 5 秒后，再执行相关 FLASH 的操作
24	操作超时	无法连接到 FLASH，更换硬件

八、C语言功能说明

1.概述

MC系列PLC支持用户在科威编程软件上利用C语言函数编写功能块，在程序中以子程序形式调用，其中支持大部分C语言函数。对用户来说，不仅增加了程序的灵活性，提高了编程效率，还进一步增强了程序的保密性。

2.操作步骤

1、打开KWPro软件（需为V1.2.6.9以上版本），在工具栏中选择用户C程序的图标，或者直接在科威编程软件的tools文件夹下，打开用户C程序软件。软件画面如下：



2、用户可以根据上述提示，选择打开程序、保存程序、另存程序、编译程序等操作。

新建程序：程序编辑区还原成默认的模板

打开程序：打开用户之前编写的.c文件。

保存程序：打开用户之前编写的.c文件。

另存程序：另存用户当前编写的.c文件。

编译程序：编译用户当前的.c文件，编译信息将在编译提示区进行显示

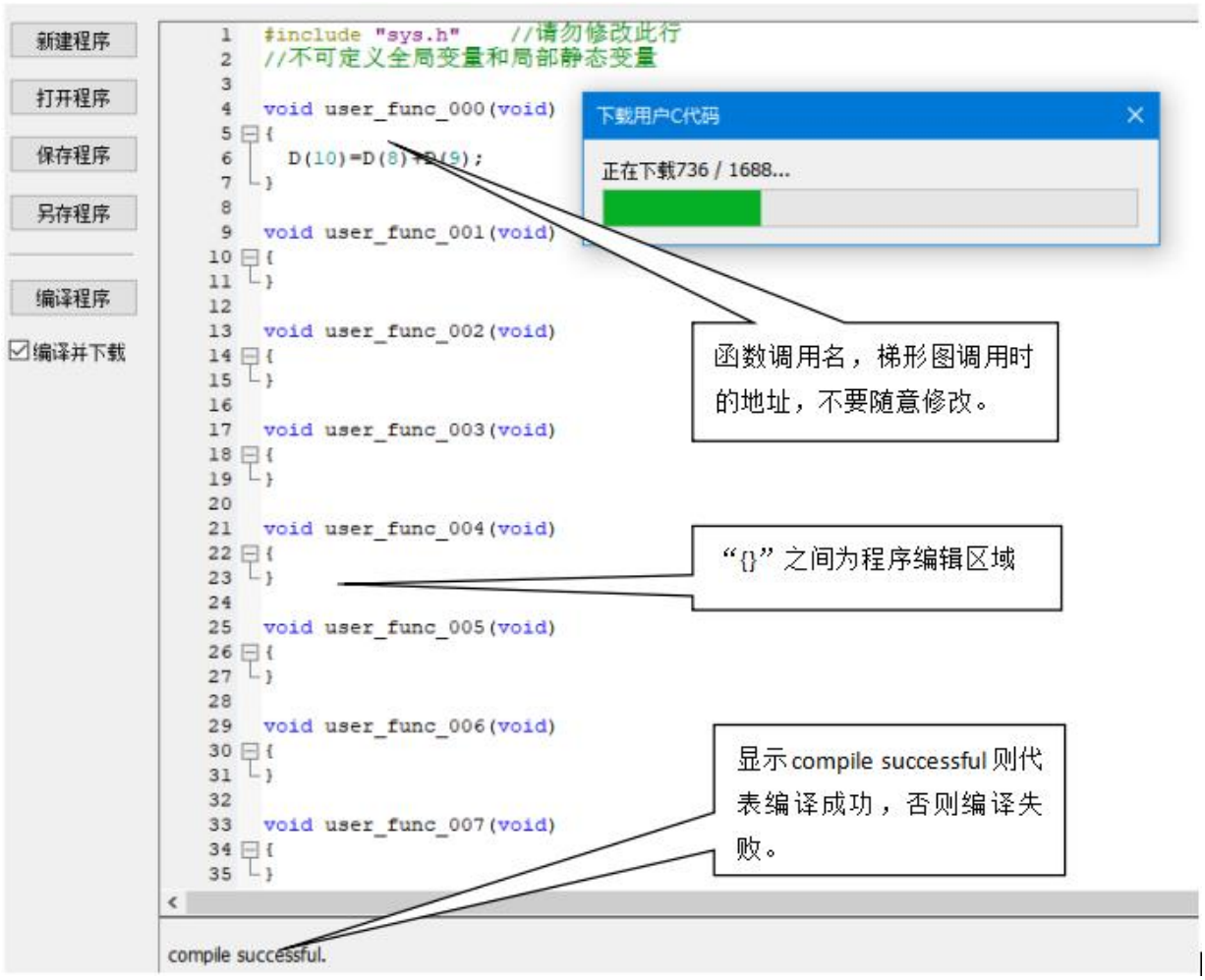
编译并下载：勾选此项后，若用户当前的.c文件没有错误，则编译后自动进行下载。

编辑区：用户在此区域进行编辑C语言程序。

3、用户编辑好程序后，就可以在梯形图写入程序进行调用，梯形图调用程序如下：



代表调用c文件里面的void user_func_000号函数如下：



当用户写完c程序后，可以点击“编译程序”，如果将下面“编译并下载”选中后，当程序编译通过时，则会自动将程序下载进入伺服里面。

注意：

- **用户也可以自定义函数名，然后在user_func_xxx里去调用。函数定义以及函数内容可以放在不同的函数之间。**
- **必须保留文档开头的#include”sys.h”文件。**
- **不可定义全局变量和局部静态变量。**

3、C语言变量

本小节主要介绍C语言常用的变量类型以及一些基本逻辑运算方法，旨在给一些无C语言编程基础的用户提供，如需进阶请参考专业C语言书籍。

3.1 基本变量类型

类型说明符合	字节	数值范围
字符型 char	1	0～255
基本整型int	4	-214783648～214783647
短整型short	2	-32768～32767
长整型long	4	-214783648～214783647
无符号型unsigned short	2	0-65536
无符号长整型unsigned long	4	0～4294967295
单精度实型float	4	3/4E-38～3/4E+38
双精度实型double	8	1/7E-308～1/7E+308

变量定义的一般形式：数据类型 变量名，如下图：

```
1  #include "sys.h"    //请勿修改此行
2  //不可定义全局变量和局部静态变量
3
4  void user_func_000(void)
5  {
6    int i,j;
7    int k;
8    short a=1;
9    char ch;
10   float m;
11 }
```

其中每一句定义结束需要以分号结尾，同一类型变量也可以放在一行定义，以上都属于合法语句。

注意：

- 当前版本不支持定义全局变量和局部静态变量；
- 变量名必须先定义再使用。
- 一个变量名只能定义一次，一般建议放在函数刚开始的地方，变量定义不允许使用中文变量，且严格区分大小写。

3.2 整型变量

1、 整型变量可以分为以下几类：

1.**基本型**，类型说明符为 **int**，在内存中占 4 个字节，其取值为基本整常数。

2.**短整型**，类型说明符为 **short**。在内存中占 2 个字节，其取值为短整常数。

3.**长整型**，类型说明符为 **long** ，在内存中占 4个字节，其取值为长整常数。

4.**无符号型**，类型说明符为 **unsigned**。

无符号型又可与上述三种类型匹配而构成：

(1) 无符号基本型类型说明符为 **unsigned int** 或 **unsigned**。

(2) 无符号短整型类型说明符为 **unsigned short。**

(3) 无符号长整型类型说明符为 **unsigned long。**

各种无符号类型量所占的内存空间字节数与相应的有符号类型量相同。但由于省去了符号位，故不能表示负数。

2、整型变量的说明

变量说明的一般形式为： 类型说明符变量名标识符，变量名标识符，...；例如：

int a,b,c； (a,b,c 为整型变量)

long x,y； (x,y 为长整型变量)

unsigned p,q； (p,q 为无符号整型变量)

在书写变量说明时，应注意以下几点：

1）允许在一个类型说明符后，说明多个相同类型的变量。各变量名之间用逗号间隔。类型说明符与变量名之间至少用一个空格间隔。

2）最后一个变量名之后必须以“;”号结尾。

3）变量说明必须放在变量使用之前。一般放在函数体的开头部分。

3.3实型变量

1、实型也称为浮点型。实型常量也称为实数或者浮点数。在C语言中，实数只采用十进制。它有二种形式：

（1）十进制数形式，由数码 0~9 和小数点组成。例如：0.0f，.25f，5.789f，0.13f，5.0f，-267.8230f 等均为合法的实数。浮点数需使用后缀。后缀为“f”或“F”即表示该数为浮点数。

（2）指数形式，由十进制数，加阶码标志“e”或“E”以及阶码(只能为整数，可以带符号)组成。其一般形式为 aE n (a 为十进制数，n 为十进制整数)其值为 a*10,n 如： 2.1E5 (等于 2.1*10,5), 3.7E-2 (等于 3.7*10,)-2*) 0.5E7 (等于 0.5*10,7), -2.8E-2 (等于-2.8*10,)-2*)以下不是合法的实数 345 (无小数点) E7 (阶码标志 E 之前无数字) -5 (无阶码标志) 53.-E3 (负号位置不对) 2.7E (无阶码)。

2、实型变量

实型变量分为两类：单精度型和双精度型，其类型说明符为 float 单精度说明符，double 双精度说明符。单精 度型占 4 个字节(32 位)内存空间，其数值范围为 3.4E-38～3.4E+38，只能提供七位有效数字。双精度型占 8 个字节(64 位)内存空间，其数值范围为 1.7E-308～1.7E+308，可提供 16 位有效数字。

实型变量说明的格式和书写规则与整型相同。

3.4字符变量

字符型量包括字符常量和字符变量。

1、字符常量是用单引号括起来的一个字符。例如'a','b','=','+','?'都是合法字符常量。在C语言中，字符常量有以下特点：

1）字符常量只能用单引号括起来，不能用双引号或其它括号。

2）字符常量只能是单个字符，不能是字符串。

3） 字符可以是字符集中任意字符。但数字被定义为字符型之后就不能参与数值运算。如'5'和 5 是不同的。'5'是字符常量，不能参与运算。

2、字符变量

字符变量的取值是字符常量，即单个字符。字符变量的类型说明符是 char。字符变量类型说明的格式和书写规则都与整型变量相同。

C语言允许对整型变量赋以字符值，也允许对字符变量赋以整型值。在输出时，允许把字符变量按整型量输出，也允许把整型量按字符量输出。整型量为二字节量，字符量为单字节量，当整型量按字符型量处理时，只有低八位字节参与处理。

3.5 符号常量宏定义

宏定义#define是C语言中常用的功能，用宏来定义一些符号常量，可以方便程序的编制。

宏定义的基本格式：

#define 宏名 宏定义字符串

define前面以#开始，表示它在编译预处理中起作用，而不是真正的C语句，行尾无需跟分号。宏定义可以按照C语言标识符规定自己定义，一般为了与变量名、函数名区别，常采用大写字母作宏名，宏名与宏定义字符串之间用空格分隔，所以宏名中间不能有空格。宏定义字符串是宏名对应的具体过程，可以是任意字符串，中间有空格，以回车符作结束。例如：

```
#define PI 3.1415926    //后续程序中用PI直接替换3.1415926
#define TRUE 1
#define FALSE 0
```

宏定义在程序中非常有用，许多C语言编写的实用程序中都会有大量的宏定义，宏的用途包括：

（1）符号常量，如PI，数组大小定义，以增加程序灵活性。

（2）简单的函数功能实现，由于宏要在一行内完成，只能实现简单函数功能。

（3）为程序书写带来一些方便。当程序中需要多次书写一些相同内容时，不妨写成宏。

3.6 数组

3.6.1 数组定义

数组是一些具有相同类型的数据的集合，数组中的数据按照一定的顺序存放。同一数组中的每个元素都具有相同的数据类型，有统一的标识符即数组名，用不同的序号即下标来区分数组中的各元素。

定义一个数组，需要明确数组变量名，数组元素的类型和数组的大小（即数组中元素的数量）。

一维数组定义的一般形式为：

类型名 数组名[数组长度]；

类型名指定数组中每个元素的类型；数组名是数组数量（以下简称数组）的名称，是一个合格的标识符；数组长度是一个整型常量的表达式，设定数组的大小。

例如：

```
int a[50];//定义一个有50个元素的整型数组a
char c[20];//定义一个有20个字符型元素的数组c
float f[5];//定义一个有5个单精度浮点型元素的数组f
```

在定义数组之后，系统根据数组中元素的类型及个数在内存中分配了一段连续的存储单元用于存放数组中的各个元素，并对这些单元进行连续编号，即下标，以区分不同的单元。每个单元所需的字节数由数组定义时给定的类型确定。

C语言规定，数组名表示该数组所分配连续内存空间中的第一个单元的地址，即首地址。由于数组空间一经分配后在运行过程中不会改变，因此数组名是一个地址常量，不允许修改。

3.6.2 数组引用

定义数组后，就可以使用它了，只能引用单个的元素，不能一次引用整个数组。

数组元素的引用要指定下标，形式为：

数组名[下标]

下标可以是整型表达式，它的取值范围是[0,数组长度-1]，前面的数组a就有50个元素，a[0],a[1],a[2],...,a[49],不能使用a[50]。

3.6.3 数组初始化

和简单变量的初始化一样，在定义数组时，也可以对数组元素进行赋值，其一般形式为：

类型名 数组名[数组长度]={初值表}

Int a[10]={1,2,3,4,5,6,7,8,9,10};

定义数组a，并对数组元素赋值，此时a[0]为1，a[1]为2，a[9]为10。

3.7 指针

3.7.1 指针定义

如果在程序中声明一个变量并使用地址作为该变量的值，那么这个变量就是指针变量。定义指针变量的一般形式为：

类型名 *指针变量名；

类型名指定指针变量所指向变量的类型，必须是有效的数据类型，如int,float,char等指针变量名是指针变量的名称，必须是一个合格的标识符。定义指针变量使用指针声明符*。例如：

Int i,*p;

声明变量i是int型，变量P是指向int型变量的指针。指针值可以是特殊的地址0，也可以是一个代表机器地址的正整数。

注意：指针变量的类型不是指指针变量本身的类型，而是指它所指向的变量的数据类型，无论何种类型的指针变量，它们都是用来存放地址的，因此指针变量自身所占的内存空间大小和它所指向的变量数据类型无关，尽管不同类型的变量所占的内存空间不同，但不同类型的指针变量所占的内存空间大小是相同的。

3.7.2指针的基本运算

如果指针的值是某个变量的地址，通过指针就能间接访问那个变量，这些操作由取地址运算符&和间接访问运算符*完成。此外，相同类型的指针还能进行赋值，比较和算术运算。

1、取地址运算和间接访问运算

单目运算符&用于给出变量的地址，例如：

Int *p,a=3;

p=&a;

将整型变量a的地址赋给整型指针p，使指针p指向变量a。也就是说，用运算符&取变量a的地址，并将这个地址值作为指针p的值，使指针p指向变量a。

3. 8 结构体

3.8.1 结构的概念与定义

在C语言中，整型，实型等基本数据类型是被系统预先定义好的，程序员可以根据其直接定义变量。而结构类型是由用户根据需要，按规定的格式自行定义的数据类型，结构类型定义的一般形式为：

```
struct 结构名{
  类型名 结构体成员名1;
  类型名 结构体成员名2;
  ...
类型名 结构成员名n;
};
```

struct是定义结构类型的关键字，在struct后，自行命名一个结构名，它必须是一个合法的C标识符，struct与结构名两者合起来共同组成结构类型名。大括号内的内容是结构所包括的结构成员，也叫结构分量，结构成员可以有多个。这样，大括号定义中的成员信息被聚合为一个整体并形成了一个新的数据类型。注意：关键字struct和它后面的结构名一起组成一个新的数据类型名，结构的定义以分号结束，这是因为C语言中把结构的定义看做一条语句。构成结构的每一个分项（即结构成员或结构分量）可以用前面章节中学习过的任意数据类型来定义，由于结构中各成员的类型可以不同，与数组相比，它提供了一种便利的手段，将不同类型的相关信息组织在一起。

3.8.2 结构变量的定义和初始化

如同C语言中的基本数据类型一样，在定义了结构类型后，还需要定义结构类型的变量，然后才能通过结构变量来操作和访问结构的数据。因为在C语言编译器只有在定义相应的结构变量后才为其分配存储单元。

struct student s1, max;

定义了在结构变量S1、max，其数据类型都是都为struct student。

在c语言中定义结构变量有如下3种方式。

1、单独定义

单独定义是指先定义一个结构类型，再定义这种结构类型的变量，例如：

```
struct student{
  int num;
  char name[10];
  int computer,english,math;
  double average;
};
  int main(void)
{
struct student s1,max;
 …
}
```

2、混合定义

混合定义是指在定义结构类型的同时定义结构变量。

这种定义方法的一般形式为：

```
  struct student{
  int num;
  char name[10];
  int computer,english,math;
  double average;
}s1,max;
```

3、无类型名定义

无类型名定义是指在定义结构变量时省略结构名。

这种方式采用如下形式定义结构类型并同时定义结构变量；

```
struct{
        类型名   结构成员名1;
        类型名   结构成员名2;
        ...
       类型名 结构成员名n;
     }结构变量名表;
```

第3种形式省略了结构名，要注意的是，由于没有给出结构名，在此定义语句后面无法再定义这个类型的其他变量，除非把定义过程再改写一遍，一般情况下除非变量不在增加，还是建议使用前两种。

结构变量初始化，即在定义时对其赋值。例如：

```
struct student s1={101,”Zhang”,78,87,85};
```

对结构变量s1进行初始化。结构变量的初始化采用初始化表的方法，大括号内各数据项之间用逗号隔开，将大括号内的数据项按顺序对应地赋值给结构变量内各个成员，且要求数据类型一致。

3.8.3 结构变量的使用

1、结构变量成员的引用

使用结构变量主要就是对其成员进行操作，在C语言中使用结构成员操作符“.”来引用结构成员，格式为：

结构变量名.成员结构名

```
S1.num=101;
```

strcpy(s1.name,”zhang”);

2、结构变量的整体赋值

如果两个结构变量具有相同的类型，则允许将一个结构变量的值直接赋给另外一个结构的变量。赋值时将赋值符号右边的结构变量的每个成员的值都赋给了左边结构变量中相应的成员，这是结构中唯一整体操作的方式。

注意：只有相同类型的变量之间才可以直接赋值。

3、结构变量作为函数参数

在一个由多个函数组成的C语言程序中，如果程序中含有结构类型的数据，就有可能需要用结构变量作为函数的参数或者返回值，以便在函数之间传递数据。

结构变量作为函数参数的特点是，可以传递多个数据且参数形式简单，但是对于成员较多的大型结构，参数传递时所进行的结构数据复制使得效率较低。

4 程序设计

4.1 基本运算符和表达式

C语言中运算符和表达式有很多种，丰富的运算符和表达式使得C语言功能多且完善，这也是C语言的主要特点之一。

C语言的运算符不仅具有不同的优先级，而且还有一个特点，就是它的结合性。在表达式中各运算量参与运算的先后顺序不仅要遵守运算符优先级别的规定，还要受运算符结合性的制约。以便确定是自左向右进行运算还是自右向左进行运算。

C语言的运算符可以分为几下几类：

1、算术运算符：用于各类数值运算。包括加（+）、减（-）、乘（*）、除（/）、求余（%）、自增（++）、自减(--)共7种。

2、关系运算符：用于比较运算。包括大于（>）、小于（<）、等于（==）、大于等于（>=）、小于等于（<=）和不等于(!=)六种。

3、逻辑运算符:用于逻辑运算。包括（&&）、或（||）、非（!）三种。

4、位操作运算符：参与运算的量，按二进制位进行运算。包括位与（&）、位或（|）、位非（~）、位异或（^）、左移（<<）、右移（>>）六种。

5、赋值运算符：用于赋值运算，分为简单赋值（=）、复合算术赋值（+=,-=,*=,/=,%=）和复合运算赋值（&=,|=,^=,>>=,<<=）三类共十一种。

6、条件运算符：这是一个三目运算符，用于条件求值（?:）。

7、逗号运算符：用于把若干表达式组成一个表达式（,）。

8、指针运算符：用于取内容（*）和取地址（&）二种运算。

9、求字节运算符：用于计算数据类型所占的字节数（sizeof）。

对基本运算符操作进行举例如下图5-1：

```
void user_func_001(void)
{
   D(0)=8;                          //D0=8
   D(1)=D(0)<<1;                    // D0左移1位，即*2
   D(2)=D(0)>>1;                    // D0右移1位，即/2
   D(6)=D(0)<<16;                   // D0左移16位
   D(7)=D(0)>>4;                    // D0右移4位
   D(3)=100;                        // D3赋值100
   D(4)=110;                        // D4赋值110
   D(5)=D(3)^D(4);                  //将D3和D4按位异或，存入D5
   D(9)=D(4)&D(3);                  //将D3和D4按位与，存入D9
   D(10)=D(0)|D(3);                 // 将D0和D3按位或，存入D10
   D(11)=D(3)/D(0);                 //将D3除以D0，商放入D11
   D(12)=D(3)%D(0);                 //将D3对D0取余，余数放入D12
   DL(13)=D(3)*D(4)*D(4);           //D3乘以D4，积放入D13
}
```

图4-1

上例中分别应用了赋值、左移、右移、异或、位与、位或、除、取余、乘等运算。

4.2 if语句

用if语句可以构成分支结构。它根据给定的条件进行判断，以决定执行某个分支程序段。C语言的if语句有三种基本形式。

1、第一种形式为基本形式

If（表达式）语句;

其语意是：如果表达式的值为真，则执行其后的语句，否则不执行该语句。

示例如下：

```
 if(D(10)==5)     //当D10的值为5时，表达式值为真，执行D11=100，否则不执行。
{
D(11)=100;       //将D11赋值为D100。
}
```

图4-2

2、第二种形式为if-else形式

If（表达式）

语句1；

else

语句2；

其语义是：如果表达式的值为真，则执行语句1，否则执行语句2。

示例如下：

```
if(D(20)==1)       //当D20值为1，执行D21=111，否则执行 D22=222
    {
     D(21)=111;
    }
    else
    {
     D(22)=222;
    }
```

图4-3

3、第三种形式为if-else-if形式

前二种形式的if语句一般都用于两个分支的情况。当有多个分支选择时，可采用if-else-if语句，其一般形式为：

If（表达式1）
语句1;
else if（表达式2）
语句2;
else if (表达式3)
语句3;
else if（表达式m）
语句m;
else
语句n;

其语意是：依次判断表达式的值，当出现某个值为真时，则执行其对应的语句。然后跳到整个if语句之外继续执行程序。如果所有的表达式均为假，则执行语句n。然后继续执行后续程序。

示例如下：

```
if(D(100)>=90)              //当D100值大于等于90时，执行D110=1
  {
   D(110)=1;
  }
  else if(D(100)>=80)   //当D100值在80-90之间时，执行D110=2
  {
   D(110)=2;
  }
  else if (D(100)>=70) //当D100值在70-80之间时，执行D110=3
  {
   D(110)=3;
  }
   else if (D(100)>=60) //当D100值在60-70之间时，执行D110=4
  {
   D(110)=4;
  }
   else
   {
    D(110)=5;          //当D100值小于60或者大于90时，执行D110=5
   }
   D(111)=6;          // 不论上述D110为何值，均执行D111=6
```

图4-4

注意：在if语句中，所有语句应为单个语句，如果要想在满足条件时执行一组（多个）语句，则必须把这一组语句用{}括起来组成一个复合语句。

4.3 Switch语句

SwItch语句可以处理多分支选择问题，根据其中break语句的使用方法，一般分为3种情况。

1、在Switch语句中的每个语句段都使用break语句，这是SwItch语句的主要使用方法，一般形式为：

switch（表达式）{
case 常量表达式1：语句段1; break;
case 常量表达式2：语句段2; break;
...
case 常量表达式n：语句段n; break;
default: 语句段n+1; break;
}

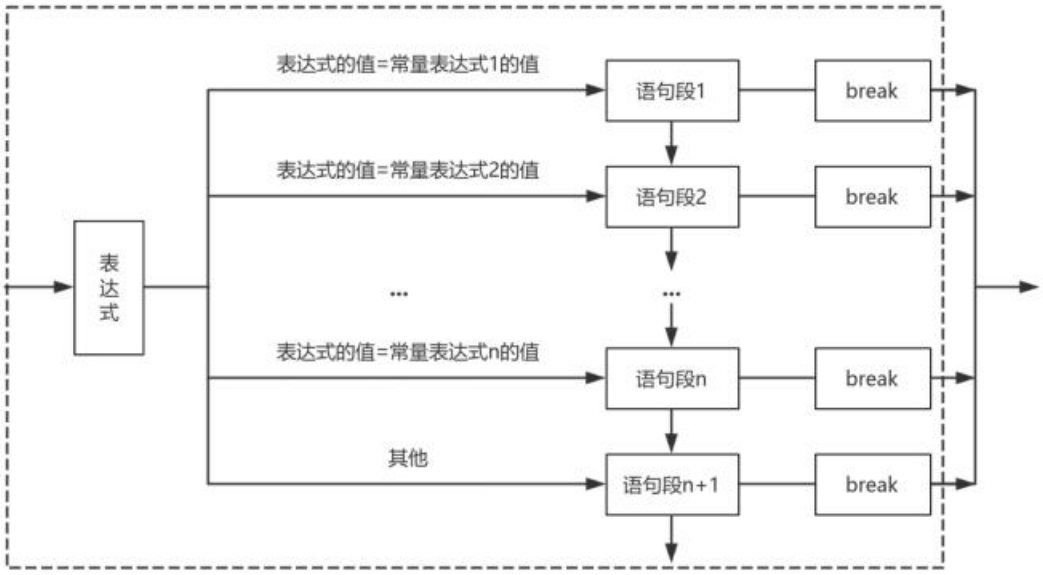


图4-5

该switch语句的执行流程如上5-5，首先求解表达式，如果表达式的值与某个常量表达式的值相等，则执行该表达式后的相应段语句段，如果表达式的值与任何一个常量表达式的值都不相等，则执行default后的语句段，最后执行break语句，跳出switch语句。

在switch语句中，表达式和常量表达式的值一般是整型或者字符型，所有的常量表达式的值都不能相等。

每个语句段可以包括一条或多条语句，也可以是空语句。

switch语句中default可以省略，如果省略了default，当表达式的值与任何一个常量表达式的值都不相等时，就什么都不执行。

示例如下：

```
switch (D(10)){  //D10寄存器数值作为比较对象
case 1:
     D(12)=2;   // 当D10=1时，执行D12=2
break;
case 2:
     D(13)=3;   // 当D10=2时，执行D13=3
break;
case 3:
     D(14)=4;   // 当D10=3时，执行D14=4
break;
default:
D(15)=5;  // 当D10值不为1、2、3时，执行D15=5
break;
}
```

图4-6

2、在switch语句中不使用break语句

break语句在switch语句中是可选的，不使用break的switch语句是：

Switch（表达式）{

case 常量表达式1：语句段1;

case 常量表达式2：语句段2;

...

case 常量表达式n：语句段n;

default: 语句段n+1;

}

求解表达式后，如果表达式的值与某个常量表达式的值相等，则执行该常量表达式后的所有语句段，如果表达式的值与任何一个常量表达式的值都不相等，则执行defalut后的所有语句段。

在上图5-6中如果去掉所有的break语句，则当D10=1时，D12=2，D13=3，D14=4，D15=5所有语句均执行，当D10=2时，则执行D13=3，D14=4，D15=5，当D10=0时，只执行D15=5。

3、在switch语句中的某些句段使用break语句

有时，在switch语句中某些句段的末尾使用break，可以实现更多的功能。

4. 4 循环结构顺序

循环结构是程序中一种很重要的结构。其特点是，在给定条件成立后，反复执行某程序段，直到条件不成立为止。给定的条件称为循环条件，反复执行的程序段为循环体。C语言中提供了多种循环体，可以组成各种不同形式的循环结构。

4.4.1 while语句

whlie语句的一般形式为：whlie（表达式）语句：其中表达式是循环条件，语句为循环体。

whlie语句的语义是：计算表达式的值，当值为非0时，执行循环语句。

使用while语句应注意以下几点：

1、whlie语句中的表达式一般是关系表达或者逻辑表达式，只要表达式的值为真（非0）即可继续循环。

2、循环体如包括有一个以上的语句，则必须用{}括起来，组成复合语句。

3、应注意循环条件的选择以避免死循环。

4.4.2 do- while语句

d0-while语句的语义是：

do 语句；

while（表达式）

先执行循环体语句一次，再判别表达式的值，若为真（非0）则继续循环，否则终止循环。

do-while语句和while语句的区别在于do-while是先执行后判断，因此do-while至少要执行一次循环体。而while是先判断后执行，如果条件不满足，则一次循环体语句也不执行。

while是先判断后执行，如果条件不满足，则一次循环体语句也不执行。

while语句和do-while一般可以相互改写。

对于do-while语句还应注意以下几点：

1、在if语句，whlie语句中，表达式后面都不能加分号，而在do-whlie语句的表达式后面则必须加分号。

2、do-whlie语句也可以组成多循环，而且也可以和whlie语句相互嵌套。

3、在do和while之间的循环体由多个语句组成，也必须用{}括起来组成一个复合语句。

4、在do-while和while语句相互替换时，要注意修改循环控制条件。

4.4.3 For语句

for语句是C语言所提供的功能更强，使用更广泛的一种循环语句。其一般形式为：

for(表达式1；表达式2； 表达3)语句；

表达式1 通常用来给循环变量赋初值，一般是赋值表达式。也允许在 for语句外给循环变量赋初值，此时可以省略该表达式。

表达式2 通常是循环条件，一般为关系表达式或逻辑表达式。

表达式3 通常可用来修改循环变量的值，一般是赋值语句。

这三个表达式都可以是逗号表达式，即每个表达式都可由多个表达式组成。三个表达式都是任选项，都可以省略。

一般形式中的“语句”即为循环体语句。for语句的语义是：

1.首先计算表达式1 的值。

2.再计算表达式2的值，若值为真(非0)则执行循环体一次，否则跳出循环。

3.然后再计算表达式3的值，转回第2步重复执行。在整个for循环过程中，表达式1只计算一次，表达式2和表达式，3则可能计算多次。循环体可能多次执行，也可能一次都不执行。

在使用for语句中要注意以下几点：

1.for语句中的各表达式都可省略，但分号间隔符不能少。如：for(；表达式；表达式)省去了表达式

<1>for(表达式；；表达式)省去了表达式

<2>for(表达式；表达式；)省去了表达式

<3>for(；；)省去了全部表达式。

2.在循环变量已赋初值时，可省去表达式1。如省去表达式2或表达式3则将造成无限循环，这时应在循环体内设法结束循环。

3.循环体可以是空语句。

本例中，省去了for语句的表达式1，表达式3也不是用来修改循环变量，而是用作输入字符的计数。这样，就把本应在循环体中完成的计数放在表达式中完成了。因此循环体是空语句。应注意的是，空语句后的分号不可少，如缺少此分号，则把后面的 printf 语句当成循环体来执行。反过来说，如循环体不为空语句时，决不能在 表达式的括号后加分号，这样又会认为循环体是空语句而不能反复执行。这些都是编程中常见的错误，要十分注意。

4.4.4 break语句

break 语句只能用在switch语句或循环语句中， 其作用是跳出 switch 语句或跳出本层循环，转去执行后面的程序。由于break语句的转移方向是明确的，所以不需要语句标号与之配合。break语句的一般形式为：

break;

上面例题中分别在 switch 语句和 for语句中使用了 break 语句作为跳转。使用 break语句可以使循环语句有多个 出口，在一些场合下使编程更加灵活、方便。

4.4.5 continue语句

continue 语句只能用在循环体中，其一般格式是：

continue;

其语义是：结束本次循环，即不再执行循环体中continue语句之后的语句，转入下一次循环条件的判断与执行。应注意的是，本语句只结束本层本次的循环，并不跳出循环。

5 PLC资源与支持函数

用户在编辑C代码过程中涉及到与PLC的资源交互，本小节主要介绍C代码和梯形图资源访问方法。

5.1 字型资源访问

- 字型方式方位位资源

类型	用法说明
DX(int)	DX(0)为X0-X17的数值，DX(1)为X20-X37的数值，依次类推。
DY(int)	DY(0)为Y0-Y17的数值，DY(1)为Y20-Y37的数值，依次类推。
DM(int)	DM(0)为M0-M15的值，DM(1)为M16-M31的数值，依次类推。
DM8(int)	DM8(0)为M8000-M8015的值，DM8(1)为M8016-M8031的数值，依次类推。
DS(int)	DS(0)为S0-S15的值，DS(1)为S16-S31的数值，依次类推。

- 字型方式方位字资源

类型	用法说明
D(int)	D(0)为D0的数值，D（1）为D1的数值，依次类推。
DL(int)	DL(0)为D0的32位数值， DL(1)为D1的32位数值，依次类推。
D8(int)	D8(0)为D8000的数值，D8(1)为D8001的数值，依次类推。
D8L(int)	D8L(0)为D8000的32位数值， D8L(8)为D8008的32位数值，依次类推。
R(int)	R(0)为D0的数值，R(1)为D1的数值，依次类推。
RL(int)	RL(0)为D0的32位数值，RL(1)为D1的32位数值，依次类推。
T(int)	T(0)为T0的数值，T(1)为T1的数值，依次类推
C(int)	C(0)为C0的数值，C(1)为C1的数值，依次类推，取值范围为0-199
C2(int)	C2(0)为C200的数值，C2(35)为C235的数值，依次类推。
V(int)	V(0)为V0的数值，V(1)为V1的数值，依次类推。
Z(int)	Z(0)为Z0的数值，Z(1)为Z1的数值，依次类推。

注意：以上资源的使用，可出现在运算符的左端，也可出现在运算符的右端，比如：

```
3
4  void user_func_000(void)
5  {
6    D(0)=100;//将D0设置为100
7    D(10) =D(0);//将D0的值送给D10
8  }
9
```

5.2 字型资源访问

位型资源的访问将采用函数调用的方式实现。

类型	用法说明
int GET_X(int)	获取X的状态，入口函数采用8进制，比如GET_X(10),获取X10的状态。
int GET_Y(int)	获取X的状态，入口函数采用8进制，比如GET_Y(10),获取Y10的状态。
int GET_M(int)	获取M的状态，比如GET_M(100),获取M100的状态。
int GET_M8(int)	获取M8000后的状态，比如GET_M8(6),获取M8006的状态。
int GET_S(int)	获取S的状态，比如GET_S(100),获取S100的状态。
void SET_Y(int)	设置Y的状态，入口函数采用8进制，比如SET_Y(10),Y10设置为ON。
void RST_Y(int)	复位Y的状态，入口函数采用8进制，比如RST_Y(10),Y10设置为OFF。
void SET_M(int)	设置M的状态，比如SET_M100(),M100设置为ON。
void RST_M(int)	复位M的状态，比如RST_M100(),M100设置为OFF。
void SET_M8(int)	设置M8000后的状态，比如说SET_M8(6)，M8006为ON。
void RST_M8(int)	复位M8000后的状态，比如说RST_M8(6)，M8006为OFF。
void SET_S(int)	设置S的状态，比如SET_S(100)，S100的状态为ON。
void RST_S(int)	复位S的状态，比如RST_S(100)，S100的状态为OFF。

示例如下,分别下载c程序和梯形图后，置位梯形图M1，调用001号C函数，可以看到D0=1，D1=1，D2=1，Y2状态为ON， M200状态为OFF。

```
void user_func_001(void)
{
  D(0)=GET_Y(4);          //将Y4的状态传入寄存器D0
  D(1)=GET_M(100);        //将M100的状态传入寄存器D1
  D(2)=GET_S(100);        //将S100的状态传入寄存器D2
  SET_Y(2);               //直接操作位，置位Y2
  RST_M(200);             //直接操作位，复位M200

}
```



5.3 支持函数

本软件支持大部分C语言字符串函数和数学函数。

1、下面列举一些常见的字符串函数：

函数	功能
void *memcpy(void *dest, void *src, unsigned int count);	由src所指内存区域复制count个字节到dest所指内存区域
int bcmp(const void *s1, const void *s2, int n);	比较字符串s1和s2的前n个字相等，相等返回0，否则返回非0，不检查空指针。
void bzero(void *s, int n);	置字节字符串s的前n个字节为零
void bcopy(const void *src, void *dest, int n);	将字符串src的前n个字节复制到dest中,bcopy不检查字符串中的空字节NULL，函数没有返回值
void *memccpy(void *dest, void *src, unsigned char ch, unsigned int count);	由src所指内存区域复制不多于count个字节到dest所指内存区域，如果遇到字符ch则停止复制,返回指向字符ch后的第一个字符的指针，如果src前n个字节中不存在ch则返回NULL
void *memset(void *buffer, int c, int count);	把buffer所指内存区域的前count个字节设置成字符

对字符串函数“memcpy”函数举例说明：

```
{
  char str[]="People";                  //定义一个字符"people"
  memcpy(&D(110), str, sizeof(str));    //将字符str[]的放入D110里面
  short *p=&D(200);                     //定义一个指针指向D200
  memcpy(&D(120), &D(200), 10);         /*将D200开始的10个字节（5个寄存器）数据复制
                                        到D120开始的5个寄存器*/
}
```

该代码运行后，从D110-D112存放字符“People”，D200-D204数据将被复制到D120-D124。

注意：该函数对PLC数据进行批量操作时，具备方便性。

2、常用基础数学函数

单精度数学函数	说明
float sqrtf(float x);	参数x的平方根
float cbrtf(floa tx);	参数x的立方根
float powf(float x,float y)；	传回参数x为底，参数y的次方值
float expf(float x)；	自然数的指数e^x

float logf(float x)；	自然对数logx
float log10f(float x)；	十为底的对数log10x
float sinf(float x)；	正弦函数
float cosf(float x)；	余弦函数
float tanf(float x)；	正切函数
float asinf(float x)；	反正弦函数
float acosf(float x)；	反余弦函数
float atanf(float x)	反正切函数
float sinhf(float x)；	双曲正弦函数
float coshf(float x)；	双曲余弦函数
floa ttanhf(float x)；	双曲正切函数
float ceilf(float x)；	向上取整，传回大于或等于参数x的最小整数
float floorf(float x)；	向下取整，传回小于或等于参数x的最大整数
float roundf(float x)；	对参数X四舍五入输出
float fabsf(float x)；	取参数X的绝对值

以平方根函数为例来说明函数表中函数的使用方法：

folat　sqrtf(float x);

folat　sqrtf中的float表示函数返回值为float型，float x中的float表示函数形参为folat型，实际使用过程中无需写float,见下例。

```
{
  float *p = (float*)&DL(300);        //定义一个指针p，p指向寄存器D300
  *(p+1)=sqrtf(*p);                   //将*p也就是D300开平方根，结果放入D302
  *(p+2)=sqrtf(36.0);                 //将36.0开平方根，结果放入D304

}
```

6 应用要点

- 必须保留文档开头的#include”sys.h”文件。
- 不可定义全局变量和局部静态变量。
- 软件不支持全部的标准C函数，常见的支持函数第5小节已经列出，当用户写入不支持的函数时，编译器并不会报错，但当函数执行时，控制器可能会卡死，出现通信超时问题，此时需重新上电，重新修改程序。
- 当调用的函数的前置条件是常开或者常闭，不是边沿信号时，C代码程序里面不要出现whlie(1)这种循环，否则会导致程序卡死。
- 当D寄存器作为浮点数使用时，推荐利用指针进行运算，示例如下：

√

```
{
  float *p = (float*)&DL(300);   //定义一个指针p，p指向寄存器D300
  *(p+1)=sqrtf(*p);              //将*p也就是D300开平方根，结果放入D302


}
```



```
{
 float *p = (float*)&DL(300);   //定义一个指针p，p指向寄存器D300
 *(p+1)=sqrtf(DL(300));         //将*p也就是D300开平方根，结果放入D302
}
```

九、以太网通信篇

MC系列PLC支持1个以太网接口，实现网络互连。

1. 以太网通讯概述

本章主要介绍以太网的几个基本概念以及TCP IP协议。

1.1 以太网的基本概念

在进行以太网通讯之前，需要先了解以太网通讯的几个基本概念，如IP地址分配、PC网络地址及设定等。

（1）分配IP地址

如果编程设备（如PC）使用网卡连接到工厂局域网（或者是互联网），则编程设备和PLC必须处于同一子网中。IP地址与子网掩码相结合即可指定设备的子网。网络ID是IP地址的第一部分，即前三个八位位组（例如IP地址为192.168.1.16，则192.168.1代表网络ID），它决定用户所在的IP网络。子网掩码的值通常为255.255.255.0；然而由于计算机处于工厂局域网中，子网掩码可能有不同的值（例如255.255.254.0）以设置唯一的子网。子网掩码通过与设备IP地址进行逻辑AND运算来定义IP子网的边界。

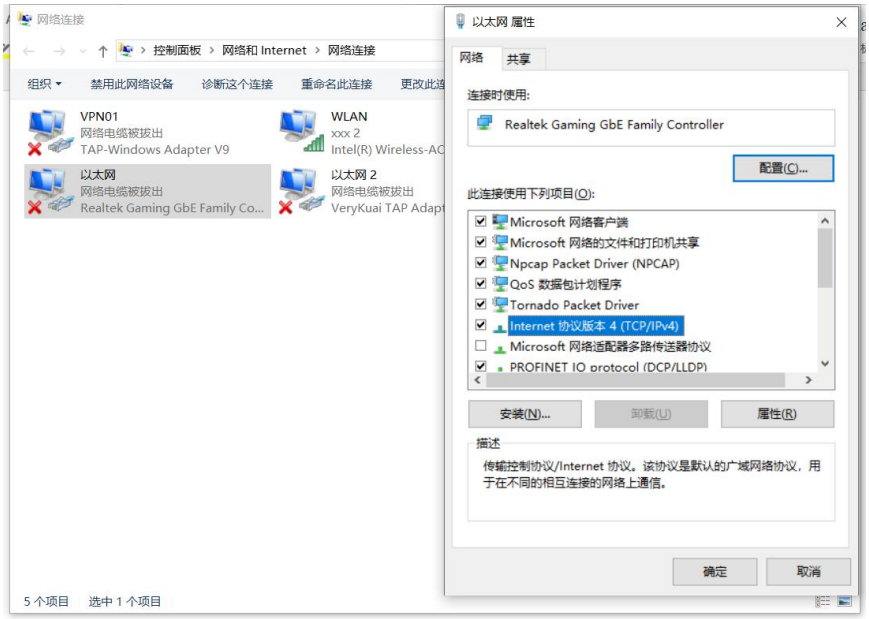
（2）设定PC网络地址信息

如果您使用的是WIN10操作系统，您可以通过以下步骤来分配或检查编程设备的IP地址：

1、鼠标右键点击电脑右下角网络图标，如下图：

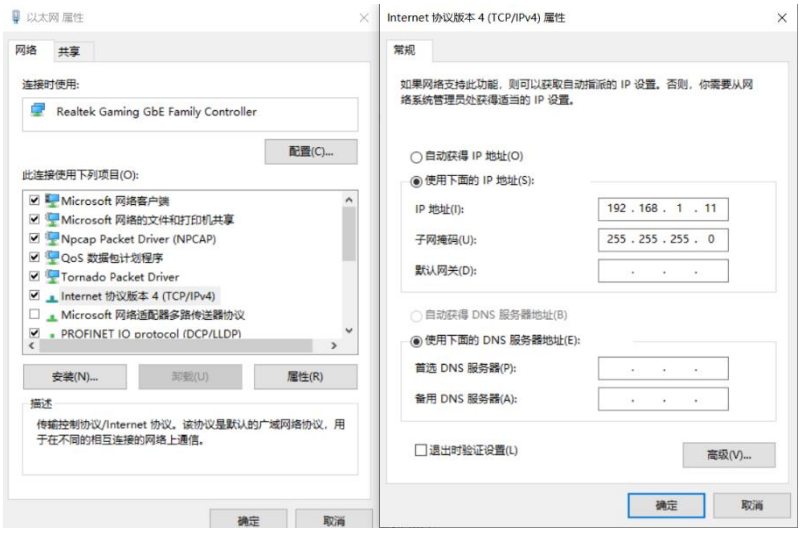


2、打开"网络和Internet设置"，选择对应的网卡，右击打开属性：



3、设定PC的IP地址，使其与PLC处于同一子网下。

PLC默认的IP地址为192.168.1.30，需将PC的IP地址设为具有相同网络ID的地址（如192.168.1.11），设定



子网掩码为255.255.255.0。默认网关可留空。这样，可使PC连接到PLC。如图所示：

1.2 TCP/IP协议

TCP/IP协议是现在比较通用的以太网通信协议，与开放互联模型ISO相比，采用了更加开放的方式。TCP/IP协议可以用在各种各样的信道和底层协议之上。确切地说，TCP/IP协议是包括TCP协议、IP协议、UDP协议、ICMP协议和其他一些协议的协议组。

（1）端口号

在以太网中，基于TCP协议或UDP协议的通信必须使用端口号才能与上层应用进行通信，端口号的范围从1到65535，有一些端口号对应有默认功能，比如用于浏览网页服务的80端口，用于FTP服务的21端口，用于MODBUS TCP通信的502端口等等。

（2）UDP协议

UDP为用户数据协议，是使用一种协议开销最小的简单无连接传输模型。UDP协议中没有握手机制，因此协议的可靠性仅等同于底层网络。无法确保对发送、回复消息提供保护。对于数据的完整性，UDP还提供了校验和，并且通常用不同的端口号来寻址不同函数。UDP组播是Internet组管理协议，简称IGMP。组播传输是在发送者和每一接收者之间实现点对多点的网络连接，用于典型的一主多从模式，有效地解决了单点发送、多点接收的问题，能够大量节约网络带宽、降低网络负载。

（3）TCP协议

1、TCP的基本原理

TCP协议为传输控制协议（Transport Control Protocol），是一种面向连接的、可靠的传输层协议。面向连接是指一次正常的TCP传输需要通过在TCP客户端和TCP服务端建立特定的虚电路连接来完成。要通过TCP传输数据，必须在两端主机之间建立连接。在通过以太网通信的主机上运行的应用程序之间，TCP提供了可靠、有序并能够进行错误校验的消息发送功能。TCP能保证接收和发送的所有字节内容和顺序完全相同。TCP协议在主动设备（即发起连接的设备）和被动设备（即接收连接的设备）之间创建连接。连接建立后，任一方均可发起数据传送。TCP协议是一种“流”协议，这意味着消息中不存在结束标志，所有接收到的消息均被认为是数据流的一部分。例如，客户端设备向服务端发送三条消息，每条均为20个字节。服务器只看到接收到一条60字节的“流”（假设服务器在收到三条消息后执行一次接收操作）。

2、套接字（Socket）的基本概念

套接字（Socket）是通信的基石，是支持TCP/IP协议的网络通信的基本操作单元。它是网络通信过程中端点的抽象表示，包含进行网络通信必须的五种信息：连接使用的协议、本地主机的IP地址、本地进程的协议端口、远端主机的IP地址、远端进程的协议口。应用层通过传输层进行数据通信时，TCP会遇到同

时为多个应用程序进程提供并发服务的问题。多 个TCP 连接或多个应用程序进程可能需要通过同一个TCP协议端口传输数据。为了区别不同的应用程序进程和连接，许多计算机操作系统为应用程序与TCP/IP协议交互提供了套接字接口。应用层可以和传输层通过套接字接口，区分来自不同应用程序进程或网络连接的通信，实现数据传输的并发服务。

3、建立套接字（Socket）连接

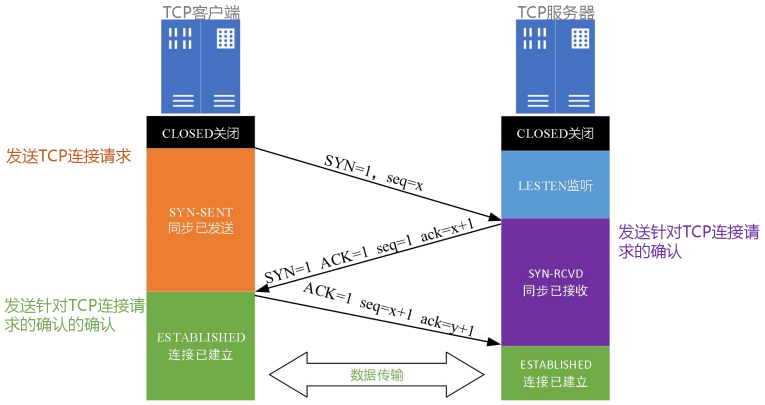
建立套接字连接至少需要一对套接字，其中一个运行于客户端（也称之为TCP客户端），称为ClientSocket，另一个运行于服务端（也称之为TCP服务器），称为ServerSocket。套接字之间的连接过程分为三个步骤：服务端监听，客户端请求，连接确认。

服务端监听：服务端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态，等待客户端的连接请求。

客户端请求：指客户端的套接字提出连接请求，要连接的目标是服务端的套接字。为此，客户端的套接字必须首先描述它要连接的服务端的套接字，指出服务端套接字的地址和端口号，然后就向服务端套接字提出连接请求。

连接确认：当服务端套接字监听到或者说接收到客户端套接字的连接请求时，就响应客户端套接字的请求，建立一个新的线程，把服务端套接字的描述发给客户端，一旦客户端确认了此描述，双方就正 式建立连接。而服务端套接字继续处于监听状态，继续接收其他客户端套接字的连接请求。创建套接字连接时，可以指定使用的传输层协议，套接字可以支持不同的传输层协议（TCP或UDP），当使用TCP协议进行连接时，该套接字连接就是一个TCP连接。

TCP通讯示意图：



上图中，服务端的的套接字处于监听状态，客户端向服务端提出连接请求，服务端接收到连接请求并发送回复确认信息给客户端，客户端收到后向服务端发送确认信息，完成资源分配后，一个TCP连接成立，此过程称为“三次握手”。

连接建立后，客户端和服务端进行数据的收发，数据收发完成后，客户端或服务端均可以发起连接关闭请求，经过“四次挥手”后，TCP连接关闭，一切数据收发中断。

2. 以太网参数配置

本章主要介绍以太网的基本参数以及参数分别在KWPro编程软件中的配置方法。

2.1 以太网参数在编程软件中的配置

打开科威PLC编程工具软件，软件上侧一栏中找到“元件监控”，如下图。



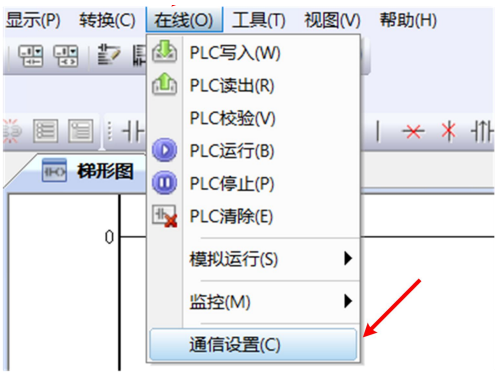
在元件监控中，监控R2300~R311的值，用户可根据IP地址设置规则自由设置PLC的IP地址，子网掩码及网关不建议改动。



有停电保护（D8020值为2222）时，使用R2300~2311内容设置

//R2300 默认值192 本地IP地址的第一个字节（192.168.1.30）
//R2300 默认值168 本地IP地址的第二个字节
//R2302 默认值1 本地IP地址的第三个字节
//R2303 默认值30 本地IP地址的第四个字节
//R2304 默认值255 本地IP掩码地址的第一个字节(255.255.255.0)
//R2305 默认值255 本地IP掩码地址的第二个字节
//R2306 默认值255 本地IP掩码地址的第三个字节
//R2307 默认值0 本地IP掩码地址的第四个字节
//R2308 默认值192 网关IP地址的第一个字节(192.168.1.1)
//R2309 默认值168 网关IP地址的第二个字节
//R2310 默认值1 网关IP地址的第三个字节
//R2311 默认值1 网关IP地址的第四个字节

设置完PLC的IP地址后，打开KWPro软件，点击菜单栏在线、通信设置

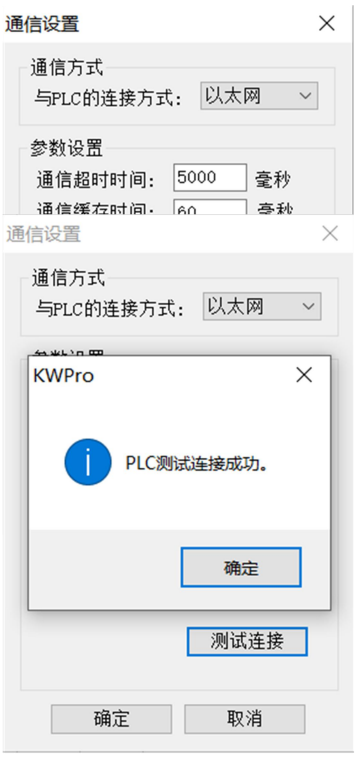


点开通信设置后，配置通信设置中的参数，与PLC的连接方式选择以太网连接，参数设置中的超时时间及缓存时间使用默认值即可，适配器选择以太网，PLC IP及端口号设置为在元件监控中自由设置的IP地址及端口号，配置完成后点击测试连接。

点击测试连接后显示如下图片，即表示以太网通信连接成功。

2.2 IP地址相关参数

以太网通讯中需设定IP地址作为每台设备的唯一标识。IP地址的设定共有三项参数，网络参数使用默认值：IP地址：192.168.1.30 子网掩码：255.255.255.0 默认网关：192.168.1.1。



3. 通讯协议

本章主要介绍MODBUS TCP通讯协议、自由格式通讯协议内容。

3.1 MODBUS TCP通讯协议

（1）MODBUS TCP通讯概述

MODBUS TCP结合了以太网物理网络和网络标准TCP/IP及以MODBUS作为应用协议标准的数据表示方法。通信报文被封装于以太网TCP/IP数据包中，MODBUS协议规范一帧数据的最大长度为256个字节。MODBUS TCP的通信系统中有两种类型的设备：MODBUS TCP客户端和服务器设备。

MODBUS客户端：客户端（TCP Client）主动向服务器（TCP Server）发起连接请求，连接建立成功。以太网机型作为MODBUS TCP客户端时，通过NOPEN指令建立TCP连接，通过NCLS指令关闭通信。

MODBUS服务器：服务器主动监听502端口，等待客户端连接请求，连接建立成功，响应符合Modbus TCP协议规范的数据通讯请求。

（2）MODBUS通讯地址

可编程控制器做ModbusTCP通信时，内部软元件编号与对应的Modbus地址编号如下所示：

类型	元件符号	元件编号	个数	Modbus地址（十六进制）
线圈、寄存器、字对象	X	X0—X256	257	0-FF
	Y	Y0—Y256	257	100-1FF
	M	M0—M7999	8000	200-DFF
	M8	M8000—M8255	256	E00-EFF
	S	S0—S1023	1024	F00-12FF
	TC	T0—T255(线圈)	256	1300-13FF
	TS	T0—T255(触点)	256	1400-14FF
	CC	C0—C255(线圈)	256	1500-15FF
	CS	C0—C255(触点)	256	1600-16FF
	D	D0000—D7999	8000	170-20AF
	D8	D8000—D8255	256	20B0-21AF
	DT	T0—T255	256	21B0-22AF
	DC	C0—C199	200	22B0-2377
	DC1	C200—C255	56	2378-23E7
	R	R0—R3999	4000	23E8-3387

（3）MODBUS通讯功能码

功能码	通信指令	功能
01H	NRDC	读线圈指令
02H	NRDI	读多个输入线圈指令
03H	NRDD	读多个输出寄存器指令
04H	NRDAI	读多个输入寄存器指令
05H	NWRC	写单个线圈指令
06H	NWRD	写单个寄存器指令
0FH	NWRCS	写多个线圈指令
10H	NWRDS	写多个寄存器指令

3.2 自由格式通讯协议

基于以太网的自由通信分为两大类：TCP和UDP，以太网机型采用TCP方式通信时可以作为TCP客户端（TCP客户端），也可以作为TCP服务端（TCP服务器）。

1、作为TCP客户端，主动与TCP服务器建立TCP连接，并绑定套接字ID。

2、作为TCP服务器，等待TCP客户端与之建立TCP连接，并绑定套接字ID。

3、使用UDP，监听指定的本机端口，并绑定套接字ID。

基于以上三种形式，可以实现以太网上的自由通信。

4. 以太网通讯指令

本章主要介绍以太网通讯指令、相关标志位和寄存器、错误一览表等内容。

4.1 以太网通讯指令概述

以太网通讯指令包括：通讯任务的开启和关闭、发送/接收数据、MODBUS TCP。使用以太网指令时，请按照以下步骤进行：

(1) 开启通讯任务：确认通信协议和通信类型，配置通信参数，创建TCP连接/UDP端口监听，并绑定套接字ID。

(2) 实现数据通信：开启成功的通讯任务，实现以太网自由通信或MODBUS TCP数据通讯。

(3) 关闭通讯任务：当与目标通讯设备通讯完成后，或TCP连接出现异常时，需要关闭通讯任务。

4.2 通信任务创建指令(NOPEN)

创建通信任务

1）指令格式

通信任务创建	
	
16位指令 NOPEN 执行条件 边沿触发	32位指令 无

2）设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	指定通讯类型	16，BIN
S3	本机各参数设置	16，BIN
S4	通信过程标志位	16，BIN

2）对象软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											

S3		●	●										
S4												●	●



创建了一个序号为K2的套接字，此套接字的功能为ModbusTCP服务器(K5)。使用D10作为保存目标设备通讯地址信息的起始寄存器。使用M100作为保存通讯状态标志的起始寄存器。

- S1：套接字ID，范围：K0~K31。注意：同时建立的套接字数量不超过32个，TCP数量不超过16个，UDP数量不超过16个。
- S2：通信类型，范围0~9
  =0 UDP 自由协议 客户端 （主站）
  =1 UDP 自由协议 服务器
  =2 TCP 自由协议 客户端 （主站）
  =3 TCP 自由协议 服务器
  =4 TCP Modbus 客户端 （主站）
  =5 TCP Modbus 服务器 （从站）
  =6 UDP 组播 服务器 （从站）
  =7 UDP 专用协议 服务器 （从站）
  =8 TCP 专用协议 服务器 （从站）
- S3：参数的起始地址
  S3参数说明：
  S3+0: 本机端口号
  S3+1~2: 目标IP 1高字 2字节
  S3+3: 目标端口号
  S3+4: 接收超时时间
  S3+5: 接收字节数
  S3+6: 错误码
  S3+7: 自由协议 格式选择 0 字低字节收发， 1 字整体收发
  S3+8: REV
  S3+9: REV
- S4:标志位的起始地址

S4	M(n+0)	M(n+1)	M(n+2)	M(n+3)	M(n+4)	M(n+5)	M(n+6)	M(n+7)	M(n+8)	M(n+9)
UDP	●	●		●	●	●	●	●		●
TCP	●	●	●	●	●	●	●	●		●
Modbus	●	●	●	●					●	●

上表中各协议标志 ● 表示标志位有效,未标 ● 表示此标志位未使用。

S4参数一览表

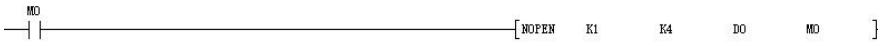
位地址	标志位	功能说明
M（n+0）	通信打开关闭标志	NOPEN，M（n+0）置ON，NCLS置OFF

M（n+1）	关闭中标志	正在关闭当前连接时，M（n+1）置ON
M（n+2）	连接中标志	连接建立过程中，M（n+2）置ON
M（n+3）	已连接标志	连接建立完成时，M（n+3）置ON
M（n+4）	发送中标志	数据发送过程中，M（n+4）置ON
M（n+5）	已发送标志	数据发送完成时，M（n+5）置ON
M（n+6）	接收中标志	数据接收过程中，M（n+6）置ON
M（n+7）	已接收标志	数据接收完成时，M（n+7）置ON
M（n+8）	ModbusTCP标志	ModbusTCP指令时，M（n+8）置ON
M（n+9）	通信异常标志	通信过程中出错，M（n+9）置ON

使用注意事项

- 0号套接字被系统上电初始化成ModbusTCP服务器。梯形图运行后，如有需要可用(NCLS K0)关闭0号套接字另作它用。
- 一般情况下不建议开启第二个ModbusTCP服务器。开启套接字时避免使用相同的端口，且不要使用默认做为ModbusTCP服务器的502端口。
- 在开启套接字前请确认已经对PLC的IP完成设置。开启套接字时，至少要设置本机端口号、目标IP高字节、目标IP低字节、目标端口信息。
- 同时建立的套接字数量不超过32个，TCP数量不超过16个，UDP数量不超过16个。
- 使用ModbusTCP通讯，但是目标端口不为502的情况下，套接字无法正常打开。

程序应用



上面的指令开启了一个ModbusTCP的客户端套接字(K4)，套接字序号为K1，从D0开始存储目标设备的地址信息，从M0开始保存套接字的状态信息。

作为客户端在创建套接字连接时，至少需要设置好D0、D1、D2、D3的值。

假设需要与地址为192.168.1.66，端口号为502的设备进行通讯。

- 首先在D0中，需要设置本机的通讯端口，可选数值范围为0 - 65535。
- 随后在D1中，需要设置目标设备IP的高字节信息，即为192.168。将192与168转换为16进制，分别为C0与A8，因此在D1中需要赋值C0A8。
- 接着在D2中，需要设置目标设备IP的低字节信息，即为1.66。将1与66转换为16进制，分别为01与42，因此在D1中需要赋值0142。
- 最后在D3中，需要设置目标设备的端口信息，即为502。

以上信息需要根据实际情况进行修改。

在本例中，标志位的首地址为M0。

在ModbusTCP通讯中：

- 如果套接字开启成功，则标志位【M0】【M2】置位。
- 如果目标通讯地址正确，则在发送完成一条ModbusTCP指令后【M2】复位，【M3】【M8】置位。

- 如果目标通讯地址不正确，则在发送完成一条ModbusTCP指令后【M2】不复位，【M8】置位。
- ModbusTCP套接字关闭后，所有标志位复位。

在TCP通讯当中，作为客户端：

- 如果套接字开启成功，则标志位【M0】【M2】置位。
- 如果目标通讯地址正确，则在完成数据发送后【M2】复位，【M3】【M5】置位。
- 如果目标通讯地址不正确，则在完成数据发送后【M2】不复位，【M5】置位。
- TCP套接字关闭后，所有标志位复位。

在TCP通讯当中，作为服务器：

- 如果套接字开启成功，则标志位【M0】【M2】置位。
- 当开启接收指令时，等待接收数据时，【M6】置位。
- 当接收到一次数据时，【M6】复位，【M7】置位。且在后续过程中【M7】不会复位。
- TCP套接字关闭后，所有标志位复位。

在UDP通讯当中，作为客户端：

- 如果套接字开启成功，则标志位【M0】【M3】置位。
- UDP通讯中，不论目标通讯地址是否正确，则在完成数据发送后【M5】置位。
- UDP套接字关闭后，所有标志位复位。

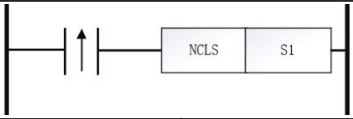
在UDP通讯当中，作为服务器：

- 如果套接字开启成功，则标志位【M0】【M3】置位。
- 当开启接收指令时，等待接收数据时，【M6】置位。
- 当接收到一次数据时，【M6】复位，【M7】置位。且在后续过程中【M7】不会复位。
- UDP套接字关闭后，所有标志位复位。

4.3 通信任务关闭指令(NCLS)

关闭通信任务

1)指令格式

<table>
<tr><td colspan="2">通信任务关闭</td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td>16位指令 NCLS
执行条件 边沿触发</td><td>32位指令
无</td></tr>
</table>

2）设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN

3）对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											



将序号为K2的套接字关闭。

- S1：套接字ID，范围：K0~K31。

使用注意事项

- 该指令不可单独使用，需和NOPEN指令配合使用。

4.4 自由通信发送指令(NSEND)

自由指令适用于客户端使用。当本机为服务器时，不建议使用自由指令。特殊情况下需要使用时，请务必掌握技术细节。

(1) 本机为TCP服务器时

TCP服务器发往套接字ID指定的客户端，由于TCP服务器对每个客户端需要独立的IP及端口需要匹配合适的客户端。当连接的只有一个客户时以此连接为通信客户端；当有多个客户端与服务器连接时，优先以用户设定的PORT选客户端，在自由指令套接字参数中设定远程IP及端口，没有对应的连接时，不通信。

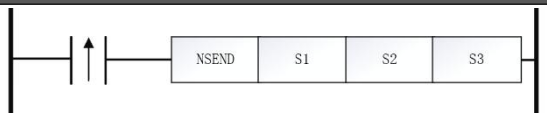
(2) 本机为UDP服务器时

自由指令套接字参数中设定了远程IP及远程端口，则该套接字ID绑定唯一客户端，可支持一对一收发通信；而不支持其它客户端。

自由指令套接字参数中未设定了远程IP及远程端口，则该套接字ID未绑定客户端，可支持多客户端接收通信，而无法发送。

NSEND，自由格式通信指令，进行一次数据的发送。

1)指令格式

自由格式通讯发送指令	
	
16位指令 NSEND 执行条件 边沿触发	32位指令 无

2）设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	本地寄存器发送首地址	16，BIN
S3	发送数据的字节数量，可指定寄存器	16，BIN

3）对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2		●	●											
S3	●	●	●											



使用自由通讯套接字K2，使用D100寄存器中保存的数据作为发送数量，从D10开始发送数据。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31。
- S2：本地寄存器发送首地址。
- S3：发送数据的字节数量，可指定寄存器或常数。

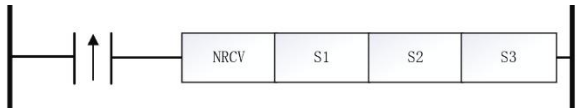
使用注意事项

- 该指令无法单独使用，需和NOPEN、NCLS指令配合使用。
- 使用时，需注意所在套接ID中NOPEN指令中的数据缓冲类型
- 当缓冲位数为8位时，只发送寄存器的低字节数据，例如：要发送D100~D107寄存器中的低字节数据时，NOPEN指令，S3指定的起始寄存器后，第5个寄存器的值应设为8。
- 当缓冲位数为16位时，寄存器的高低字节数据都将被发送，例如：要发送D100~D107中的高、低字节数据时，S3指定的起始寄存器后，第5个寄存器的值应设为16，且发送时，低字节在前高字节在后。

4.5 自由通信接收指令(NRCV)

NRCV，自由格式通信指令，进行一次数据的接收。

1)指令格式

自由格式通讯接收指令	
	
16位指令 NRCV 执行条件 边沿触发	32位指令 无

2）设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	本地寄存器接收首地址	16，BIN
S3	接收数据的字节数量，可指定寄存器	16，BIN

3）对象软元件

种类	字型软元件	位型软元件

	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2		●	●											
S3	●	●	●											



使用自由通讯套接字K2，使用D100寄存器中保存的数据作为接收数量，从D10开始存入数据。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31。
- S2：本地寄存器接收首地址。
- S3：接收数据的字节数量，可指定寄存器或常数。

使用注意事项

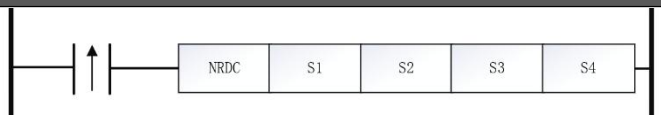
- 该指令无法单独使用，需和NOPEN、NCLS指令配合使用。
- 使用时，需注意所在套接ID中NOPEN指令中的数据缓冲类型
- 当缓冲位数为8位时，接收的数据只存放在低字节中，例如：要接收8个字节数据，依次存放在D100~D107这8个寄存器的低字节中，此时，NOPEN指令，S3指定的起始寄存器后，第5个寄存器的值应设为8。
- 当缓冲位数为16位时，寄存器的高低字节中都会存放接收的数据，例如：要接收16个字节数据，依次存放在D100~D107这8个寄存器中，此时，NOPEN指令，S3指定的起始寄存器后，第5个寄存器的值应设为16。且接收时，低字节在前高字节在后。

4.6 读多个线圈指令(NRDC)

以太网含括ModbusTCP协议，后续讲述ModbusTCP指令。

NRDC，读多个线圈指令。

1) 指令格式

读多个线圈	
	
16位指令 NRDC 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4											●	●	●	●

功能与动作

读取多个线圈数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站线圈的首地址，读取5个线圈，从主站的M0开始保存。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为线圈位地址。
- S3的通信元件个数最大为512个。
- S4的主站元件首地址需要确认为本机的线圈位地址。

4.7 读多个输入线圈指令(NRDI)

NRDI，读多个输入线圈指令。

1) 指令格式

读多个输入线圈	
	
16位指令 NRDI 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站Modbus首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4											●	●	●	●

功能与动作

读取多个输入线圈数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站输入线圈的首地址，读取5个线圈，从主站的X0开始保存。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为输入线圈位地址。
- S3的通信元件个数最大为512个。
- S4的主站元件首地址需要确认为本机的线圈位地址。

4.8 读多个输出寄存器指令(NRDD)

NRDD，读多个输出寄存器指令。

1) 指令格式

读多个输出寄存器	
	
16位指令 NRDD 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站Modbus首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4		●	●											

功能与动作

读取多个输出寄存器数据指



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站输出寄存器的首地址，读取5个寄存器，从主站的D10开始保存。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

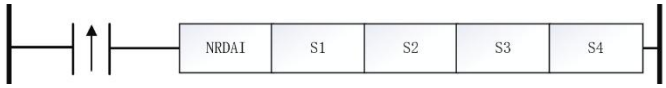
使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为输入寄存器的字地址。
- S3的通信元件个数最大为127个。
- S4的主站元件首地址需要确认为本机的寄存器字地址。

4.9 读多个输入寄存器指令(NRDAI)

NRDAI，读多个输入寄存器指令。

1) 指令格式

读多个输入寄存器	
	
16位指令 NRDAI 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4		●	●											

功能与动作

读取多个输入寄存器数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站输入寄存器的首地址，读取5个寄存器，从主站的D10开始保存。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为寄存器的字地址。
- S3的通信元件个数最大为127个。
- S4的主站元件首地址需要确认为本机的寄存器字地址。

4.10 写单个线圈指令(NWRC)

NWRC，写单个线圈指令。

1) 指令格式

写单个线圈	
	
16位指令 NWRC 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3												●	●	●

功能与动作

写单个线圈数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站线圈的地址，将主站S0线圈的状态写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为线圈的位地址。
- S3的主站元件首地址需要确认为本机的线圈位地址。

4.11 写单个寄存器指令(NWRD)

NWRD

1) 指令格式

写单个寄存器	
	
16位指令 NWRD 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	主站软元件首地址编号	16，BIN

3) 对象软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3		●	●											

功能与动作

写单个寄存器数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站寄存器的地址，将主站D100寄存器的数值写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为寄存器的字地址。
- S3的主站元件首地址需要确认为本机的寄存器的字地址。

4.12 写多个线圈指令(NWRCS)

NWRCS，写多个线圈指令。

1) 指令格式

写多个线圈	
	
16位指令 NWRCS 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 适用软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4												●	●	●

功能与动作

写多个线圈数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站线圈的首地址，将主站从M100开始的5个线圈的状态写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为线圈的位地址。
- S3的通信元件个数最大为512个。
- S4的主站元件首地址需要确认为本机的线圈位地址。

4.13 写多个寄存器指令(NWRDS)

NWRDS

1) 指令格式

写多个寄存器	
	
16位指令 NWRDS 执行条件 边沿触发	32位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	16，BIN
S2	从站软元件首地址	16，BIN
S3	通信元件个数	16，BIN
S4	主站软元件首地址编号	16，BIN

3) 适用软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4		●	●											

功能与动作

写多个寄存器数据指令。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站寄存器的首地址，将主站从D100开始的5个寄存器的值写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站软元件首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为寄存器的字地址。
- S3的通信元件个数最大为127个。
- S4的主站元件首地址需要确认为本机的寄存器字地址。

4.14 读多个寄存器指令(ERDD)

后续讲述ModbusTCP的扩展指令。

ERDD，读多个寄存器指令，地址位宽扩展到32位。

1) 指令格式

扩展读取多个寄存器	
	
32位指令 ERDD 执行条件 边沿触发	16位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	32，BIN
S2	从站(Modbus扩展)首地址	32，BIN
S3	通信元件个数	32，BIN
S4	主站软元件首地址编号	32，BIN

3) 适用软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4		●	●											

功能与动作

读取多个寄存器指令，读取地址的位宽扩展到32位。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站寄存器的32位首地址，将读取到的数据存入到主站D100开始的5个寄存器中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站(Modbus扩展)首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项：

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为寄存器的字地址，地址位宽可以为32位。
- S3的通信元件个数最大为127个。
- S4的主站元件首地址需要确认为本机的寄存器字地址。

4.15 写多个寄存器指令(EWRD)

EWRD，写多个寄存器指令，地址位宽扩展到32位。

1) 指令格式

扩展写多个寄存器	
	
32位指令 EWRD 执行条件 边沿触发	16位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	32，BIN
S2	从站(Modbus扩展)首地址	32，BIN
S3	通信元件个数	32，BIN
S4	主站软元件首地址编号	32，BIN

3) 适用软元件

种类 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4		●	●											

功能与动作

写入多个寄存器数据指令，写入地址的位宽扩展到32位。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站寄存器的32位首地址，将主站D100开始的5个寄存器中的数据写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站(Modbus扩展)首地址
- S3：通信元件个数
- S4：主站软元件首地址编号

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为寄存器的字地址，地址位宽可以为32位。
- S3的通信元件个数最大为127个。
- S4的主站元件首地址需要确认为本机的寄存器字地址。

4.16 扩展读位区数据指令(ERDC)

ERDC，扩展读位区数据指令，地址位宽扩展到32位。

1) 指令格式

扩展读位区	
	
32位指令 ERDC 执行条件 边沿触发	16位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	32，BIN
S2	从站(Modbus扩展)首地址	32，BIN
S3	通信元件个数	32，BIN
S4	主站软元件首地址编号	32，BIN

3) 适用软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4											●	●	●	●

功能与动作

扩展读位区数据指令，读取地址的位宽扩展到32位。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为读取从站线圈的32位首地址，将读取到的线圈状态存入到主站M100开始的5个线圈中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站(Modbus扩展)首地址
- S3：通信元件个数

使用注意事项

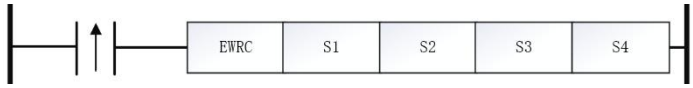
- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为线圈的位地址，地址位宽可以为32位。
- S3的通信元件个数最大为512个。
- S4的主站元件首地址需要确认为本机的线圈位地址。

- S4：主站软元件首地址编号

4.17 扩展写位区数据指令(EWRC)

EWRC，扩展写位区数据指令，地址位宽扩展到32位。

1) 指令格式

扩展写位区	
	
32位指令 EWRC 执行条件 边沿触发	16位指令 无

2) 设定数据

操作数	作用	类型
S1	指定建立通讯任务的套接字ID	32，BIN
S2	从站(Modbus扩展)首地址	32，BIN
S3	通信元件个数	32，BIN
S4	主站软元件首地址编号	32，BIN

3) 适用软元件

种类 / 操作数	字型软元件										位型软元件			
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	X	Y	M	S
S1	●	●	●											
S2	●	●	●											
S3	●	●	●											
S4												●	●	●

功能与动作

扩展写位区数据指令，读取地址的位宽扩展到32位。



使用ModbusTCP通讯套接字K2，使用D0寄存器中保存的数据作为写入从站线圈的32位首地址，将主站M100开始的5个线圈的状态写入其中。

- S1：套接字ID，可指定寄存器或常数，范围：K0~K31
- S2：从站(Modbus扩展)首地址
- S3：通信元件个数

使用注意事项

- 确认S1使用的套接字ID已经打开。
- S2从站的元件首地址需要确认为线圈的位地址，地址位宽可以为32位。
- S3的通信元件个数最大为512个。
- S4的主站元件首地址需要确认为本机的线圈的位地址。

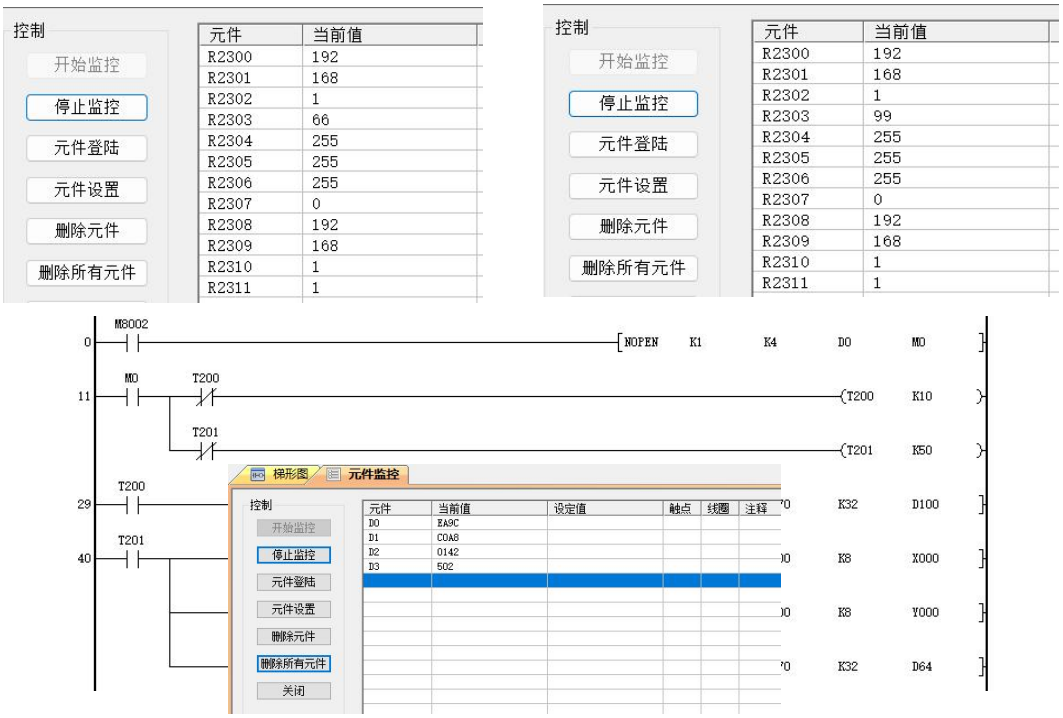
- S4：主站软元件首地址编号

5. 以太网通讯案例

5.1 使用ModbusTCP协议进行一主一从通讯

将两台PLC的IP设置为192.168.1.66与192.168.1.99。通过元件监控，对R2300至R2311寄存器进行赋值。

主站PLC程序如下：



上电使用NOPEN指令打开K1号套接字，使用K4模式(ModbusTCP 客户端)，使用D0至D3中保存的从站地址信息进行连接，将连接的状态标志保存在M0起始的线圈中。

在这个例子中，主站192.168.1.99使用60060(EA9C)作为通信端口，连接从站192.168.1.66:502 ModbusTCP服务器。

当连接成功时，M0标志位会置位，T200计时器每100ms置位一次，即程序每100ms执行以下ModbusTCP指令：

① 读取从站D0至D31，32个寄存器的数据到主站的D100至D131中；

当连接成功时，M0标志位会置位，T201计时器每500ms置位一次，即程序每500ms执行以下3条ModbusTCP指令：

② 读取从站M0至M7，8个线圈的状态到主站的X0至X7中；

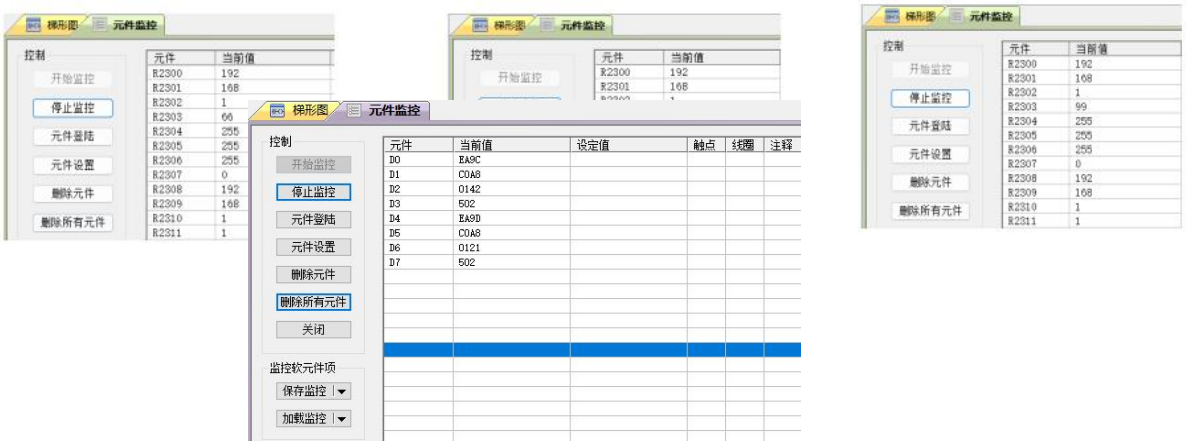
③ 将主站Y0至Y7，8个线圈的状态写入到从站的M0至M7中；

④ 将主站D64至D95，32个寄存器的值写入到从站的D0至D31中；

◆ 从站PLC的程序可以为空，但是需要完成自身的IP设置。

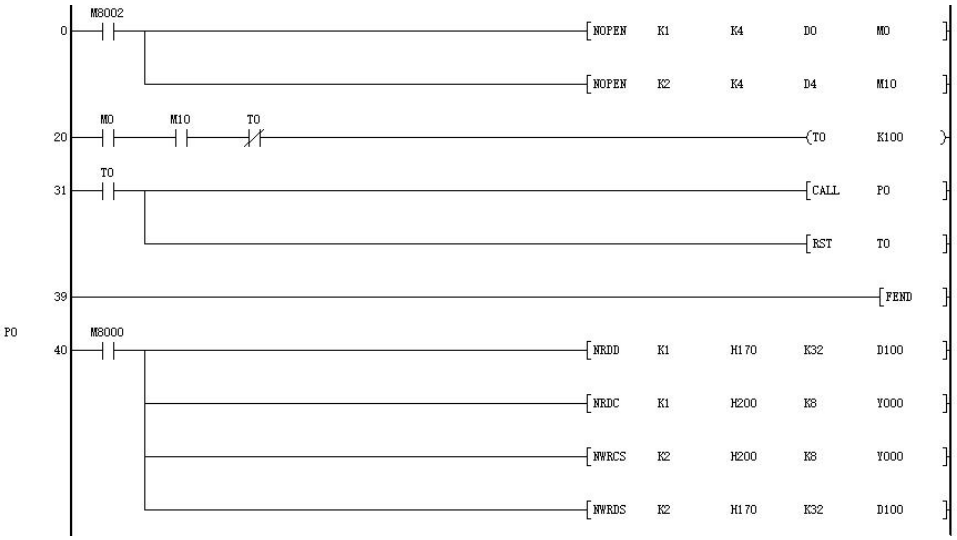
5.2 使用ModbusTCP协议进行一主二从通讯

将三台PLC的IP设置为192.168.1.33、192.168.1.66、192.168.1.99。通过元件监控，对R2300至R2311寄存器进行赋值。



主站PLC程序如下：

上电使用NOPEN指令打开K1和K2号套接字，均使用K4模式(ModbusTCP 客户端)，使用D0至D3中保存的从站地址信息连接从站一，使用D4至D7中保存的地址信息连接从站二。将一号从站的连接的状态标志保存在M0起始的线圈中，将二号从站的连接的状态标志保存在M10起始的线圈中。



在这个例子中，我们将192.168.1.99作为主站，连接192.168.1.33与192.168.1.66的ModbusTCP服务器。连接成功后，M0与M10会置位，开启T0定时器，每经过1000ms，触发中断进入P0子程序。

在子程序中，前两条指令分别对从站一进行读取线圈与读取寄存器命令：

① 将从站一中D0至D31寄存器的值保存到主站D100至131中；

② 将从站一中M0至M7线圈的状态保存到主站Y0至Y7中。

后两条指令分别对从站二进行写入线圈与写入寄存器的命令：

① 将主站Y0至Y7的值写入到从站二的M0至M7当中；

② 将主站D100至D131的值写入到从站二的D0至D31中。

- 在这个例子当中，需要注意开启两个套接字的情况下，需要确保两个套接字的ID不相同，并且存储从站地址信息的寄存器不能够相同，同时保存连接状态信息的线圈也不能够相同。
- 从站一与从站二的程序可以为空，但是需要完成自身IP设置。

5.3 使用TCP自由协议进行通讯

将两台PLC的IP设置为192.168.1.66与192.168.1.99。通过元件监控，对R2300至R2311寄存器进行赋值。在本例中，我们将192.168.1.99:123设置为服务器，192.168.1.66:234设置为客户端。





作为TCP服务器的PLC的程序如下：

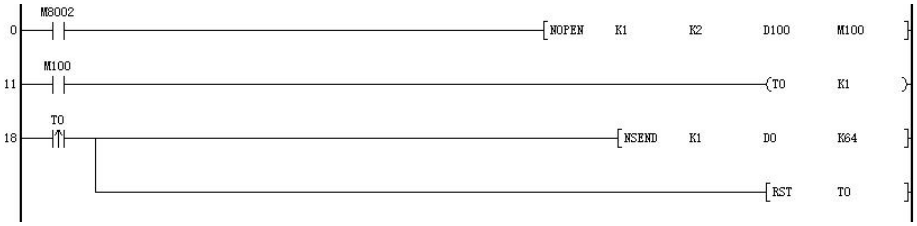


上电后使用NOPEN指令打开K1号套接字，均使用K3模式(TCP 服务端)，使用D100中保存的值作为本机服务器的监听端口，将服务器的连接状态保存在M100起始的线圈中。

首先我们设置TCP服务器，因此我们需要设置自己的TCP服务器监听端口，需要设置D100的值，范围为0-65535，但D100的值不能够设置为502。对于D100至D103中的值，作为TCP服务器时，D101、D102和D103中的需要设置为客户端的地址信息。所以，根据上述描述，D100至D103的值应该设置分别为【K123】【HC0A8】【H142】【K234】。

在服务器开启成功后，M100会置位，此时NRCV启动监听，接收64字节的数据保存到D0至D63中。如果接收数据大于64字节，则后面的数据将被丢弃。

作为TCP客户端的PLC程序如下：



上电后使用NOPEN指令打开K1号套接字，均使用K2模式(TCP 客户端)，设置D100的值作为客户端的TCP通讯端口。使用D101至D103中保存的值作为TCP服务器的地址信息，将TCP客户端的连接状态保存在M100起始的线圈中。

在客户端中，D100至D103的值需要设置为服务器的信息，根据上述描述，分别为【K234】【HC0A8】【H163】【K123】。

在客户端成功连接到服务器后，M100置位，开启T0定时器。每隔10ms向服务器发送D0至D63寄存器中的数据。

在TCP自由协议中，服务器与客户端需要一一对应，每一个套接字都需要输入对方的地址与端口信息。在进行数据收发的过程中，需要先打开服务器，再打开客户端进行连接与数据发送。

虽然TCP服务器中设置了目标客户端的IP地址，但其他客户端仍然可以连接服务器，只是在通讯时，只能接收来自目标客户端发送的数据。如果想接收来自所有TCP客户端的数据，在本例中，可以设置D101至D103为0。

5.4 使用UDP自由协议进行通讯

将两台PLC的IP设置为192.168.1.66与192.168.1.99。通过元件监控，对R2300至R2311寄存器进行赋值。在本例中，我们将192.168.1.99:123设置为服务器，192.168.1.66:234设置为客户端。





作为UDP服务器的程序如下：

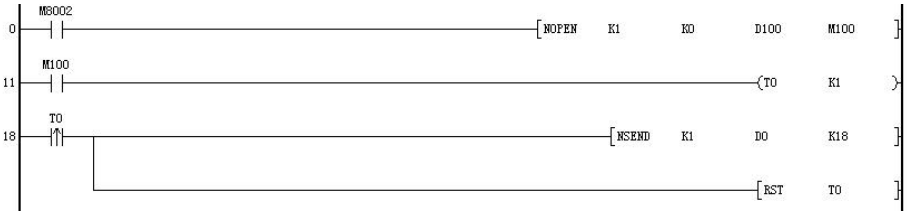


上电后使用NOPEN指令打开K1号套接字，均使用K1模式(UDP 服务端)，使用D100中保存的值作为本机服务器的监听端口，将服务器的连接状态保存在M100起始的线圈中。

首先我们设置UDP服务器，因此我们需要设置自己的UDP服务器监听端口，需要设置D100的值，范围为0-65535，但D100的值不能够设置为502。对于D100至D103中的值，由于UDP通讯的性质，作为UDP服务器时，一般不需要指定客户端的地址。所以，根据上述描述，可以将D100至D103的值应该设置分别为【K123】【K0】【K0】【K0】，以开放123端口作为UDP服务器，监听来自任意客户端的UDP连接。

在服务器开启成功后，M100会置位，同时开始接收18字节的信息，保存到D0开始的寄存当中。如果数据长度大于18字节，则后面的数据将被丢弃。

作为UDP客户端的程序如下：



上电后使用NOPEN指令打开K1号套接字，均使用K0模式(UDP 客户端)，设置D100的值作为客户端的UDP通讯端口。使用D101至D103中保存的值作为UDP服务器的地址信息，将UDP客户端的连接状态保存在M100起始的线圈中。

在客户端中，D100至D103的值需要设置为服务器的信息，根据上述描述，分别为【K234】【HC0A8】【H163】【K123】。

在客户端成功连接到服务器后，M100置位，开启T0定时器。每隔10ms向服务器发送D0至D17寄存器中的数据。

在UDP自由协议中，服务器不需要特意指定客户端IP地址，任意UDP客户端都可以向UDP服务器发送数据。在进行数据收发的过程中，客户端与服务器的开启顺序并无优先级之分。

虽然UDP服务器中不用指定客户端的IP地址。但如果想接收来自某一特定UDP客户端的数据，则在本例中，也可以将D101至D103的值设置为对应客户端的IP地址与端口信息。

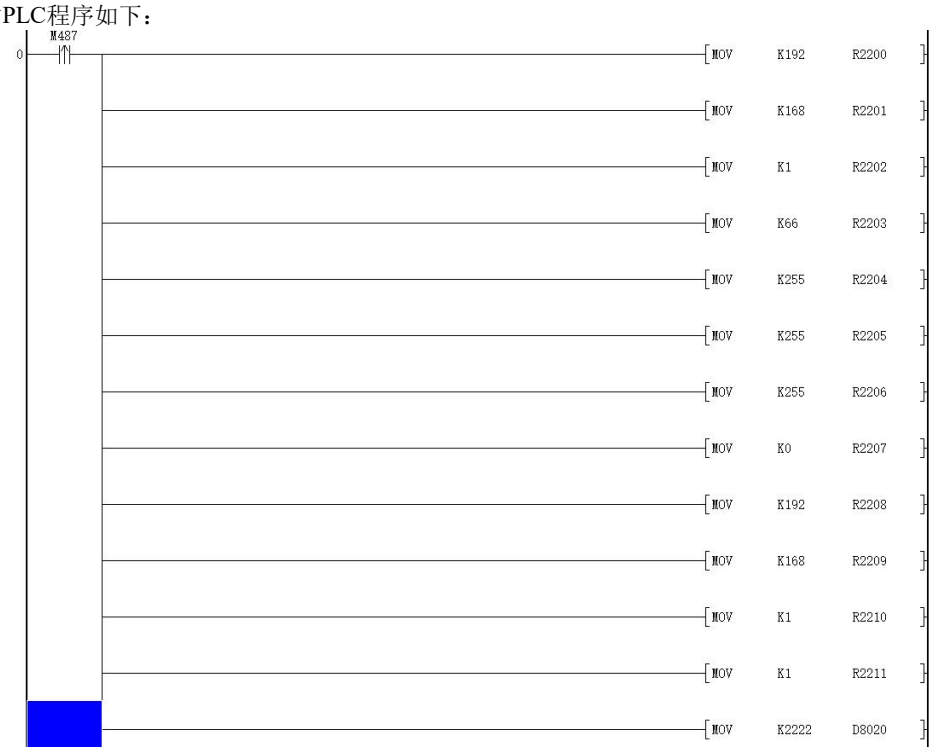
5.5 昆仑触摸屏与二台PLC使用ModbusTCP协议通信

将二台PLC的IP设置为192.168.1.66、192.168.1.99。通过下面梯形图置位M487再进入元件监控，R2300至R2311的值IP导入成功。





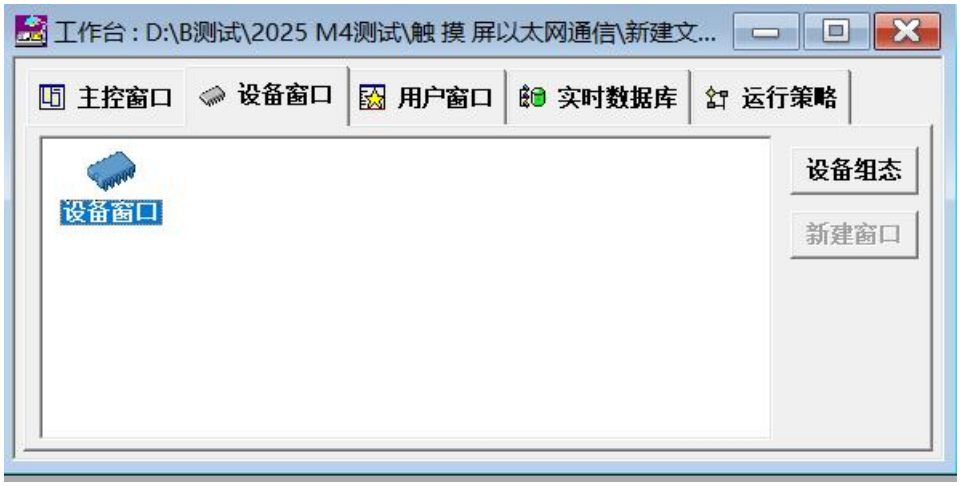
从站PLC程序如下：



在这个例子中，我们将192.168.1.99与192.168.1.66的ModbusTCP服务器作为从站，连接昆仑通态触摸屏ModbusTCP客户端。下面设置昆仑触摸屏参数，新建一个画面后如下图所示：



点设备窗口开始设置通信参数，

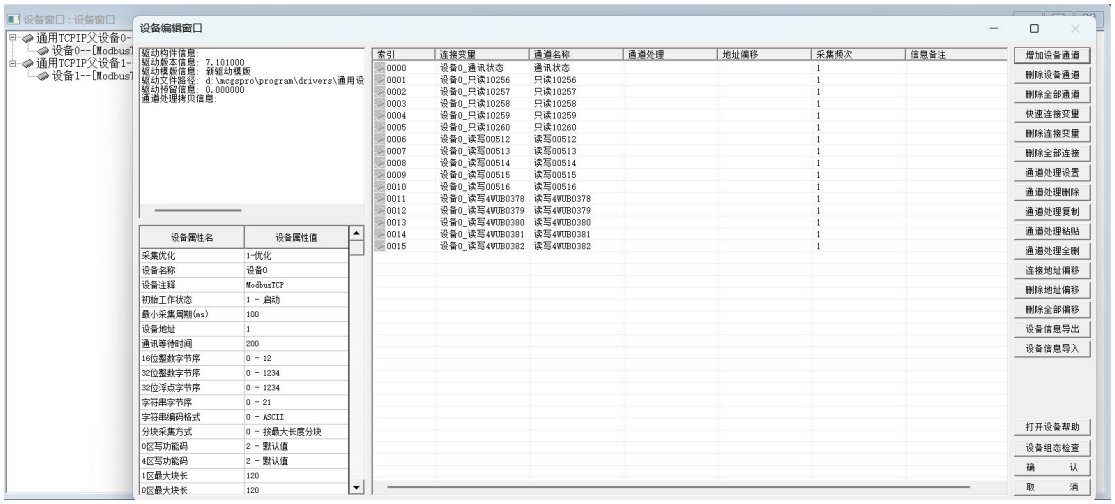


下面增加通用TCPIP父设备0，昆仑屏IP地址设置为192.168.1.21，远程IP地址设置为192.168.1.66，同样通用TCPIP父设备1，昆仑屏IP地址设置为192.168.1.21，远程IP地址设置为192.168.1.99，端口号一定要设置成502。





设置完成后，点设备0--[ModbusTCP],增加设备变量连接。根据PLC说明书D0对应地址K368，在屏上对应地址时加1，也就是K369.然后在画面里面显示出来。当改变屏上D0数据时，监控PLC的D0数据是否跟屏显示一致。如果更改有效，则通信正常。



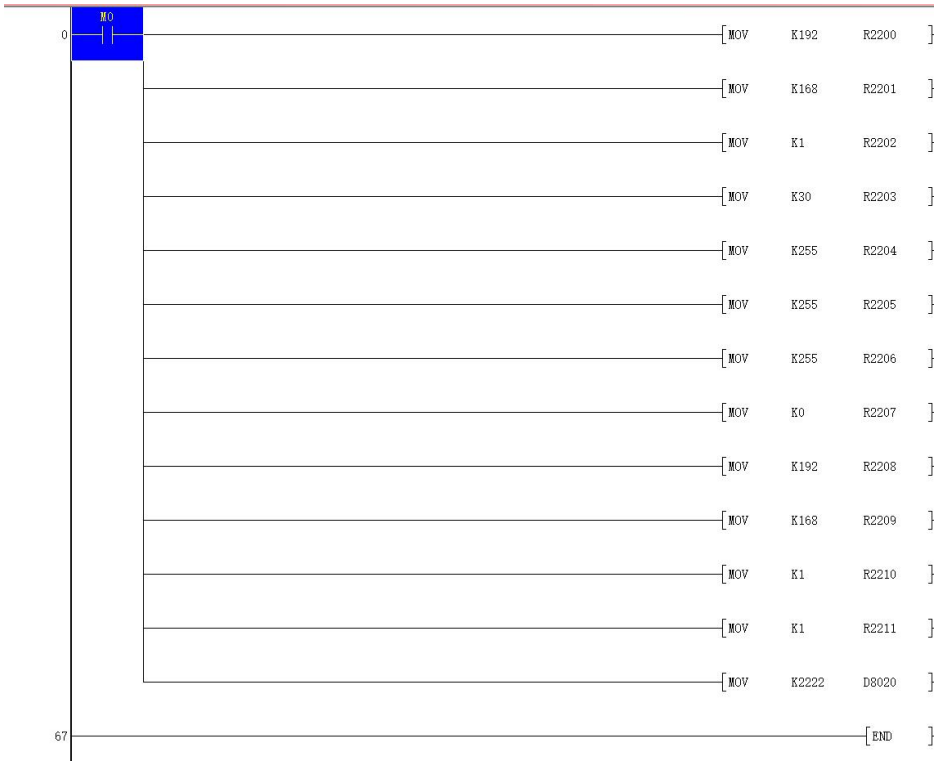
- 在这个例子当中，需要注意触摸屏的IP与PLC的IP地址不能填错或填反，否则会出现通信失败。
- PLC从站一与从站二的程序可以为空，但是需要完成自身IP设置。
- 对应变量连接时，屏上数据与实际PLC对应的MODBUS TCP地址加1，这个案例只针对昆仑通态触摸屏。

5.6 汇川H5U-A8PLC与M4PLC使用ModbusTCP协议通讯

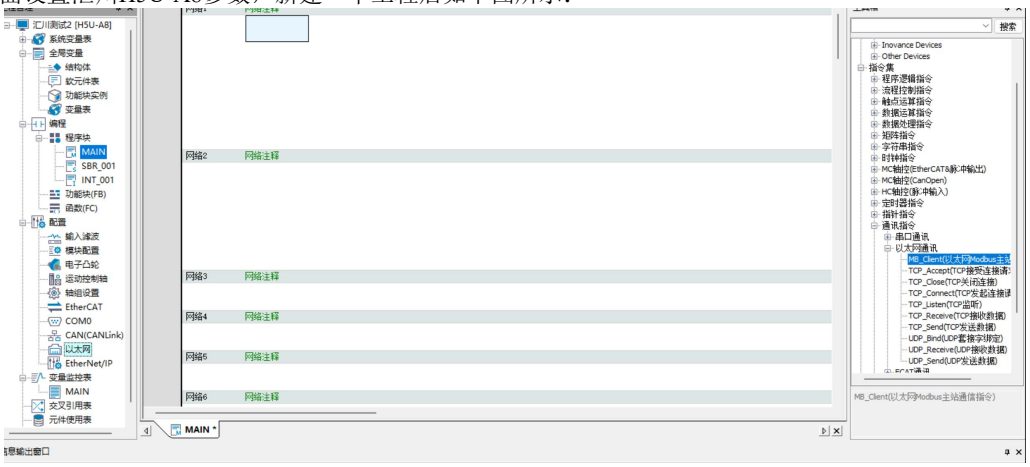
将M4PLC的IP设置为192.168.1.30。通过下面梯形图置位M0再进入元件监控，R2300至R2311的值IP导入成功。



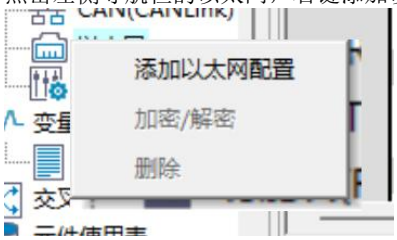
PLC程序如下：



在这个例子中，我们将192.168.1.30的ModbusTCP服务器作为从站，连接汇川H5U-A8PLC的ModbusTCP客户端。下面设置汇川H5U-A8参数，新建一个工程后如下图所示：



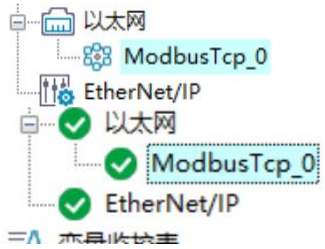
点击左侧导航栏的以太网，右键添加以太网配置



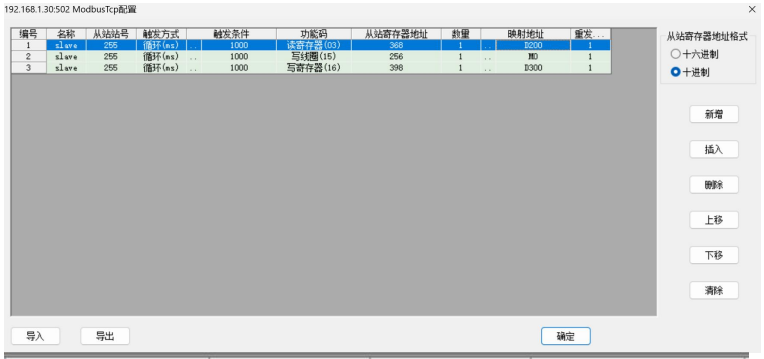
将IP地址设置为从站的IP即192.168.1.30，端口号默认502.



设置成功后以太网下出现ModbusTcp_0,即通讯成功。



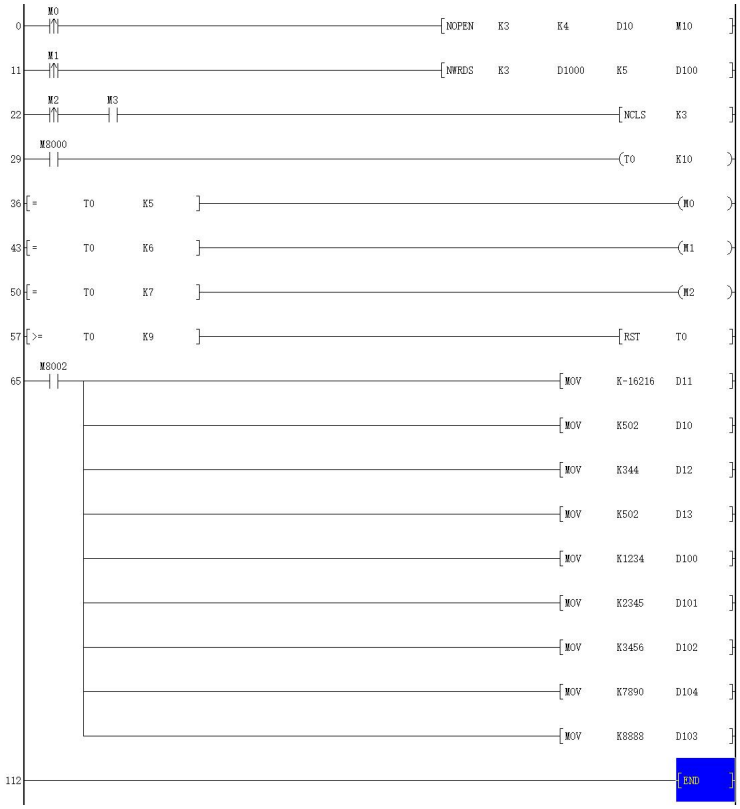
设置完成后，点设备0--[ModbusTCP],新增变量连接。根据PLC说明书在D0对应地址K368，映射汇川本地地址D200，写寄存器D30绑定汇川本地地址D300。打开汇川的变量表监控。监控汇川PLC的D200与M4PLC的D0数据是否显示一致。如果更改有效，则通信正常。



元件名称	数据类型	显示格式	当前值
D200	INT	十进制	1234
M0	BOOL	二进制	OFF

元件	当前值	设定值	触点
D0	1234		
Y000			OFF

若要将M4PLC作为服务器，将汇川的H5U-A8作为从站，需要使用NOPEN,NCLS指令。具体配置如下：



上面的指令开启了一个ModbusTCP的客户端套接字(K4)，套接字序号为K3，从D10开始存储目标设备的地址信息，从M10开始保存套接字的状态信息。

作为客户端在创建套接字连接时，至少需要设置好D10、D11、D12、D13的值。

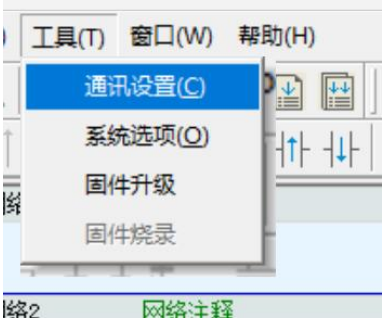
假设需要与地址为192.168.1.88，端口号为502的设备进行通讯。

- 首先在D10中，需要设置本机的通讯端口，可选数值范围为0 - 65535。
- 随后在D11中，需要设置目标设备IP的高字节信息，即为192.168。将192与168转换为16进制，分别为C0与A8，因此在D1中需要赋值C0A8。
- 接着在D12中，需要设置目标设备IP的低字节信息，即为1.88。将1与88转换为16进制，分别为01与58，因此在D1中需要赋值0158。
- 最后在D13中，需要设置目标设备的端口信息，即为502。

以上信息需要根据实际情况进行修改。

在汇川AUTOSHOP中设置PLC的IP地址

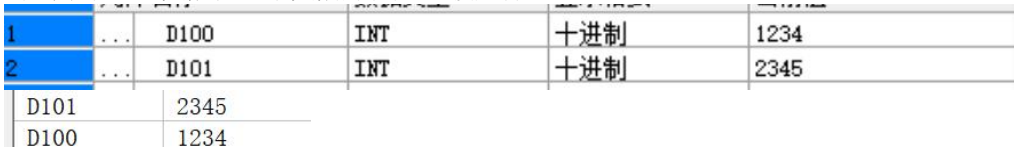
打开工具，点击通讯设置。



汇川H5U-A8PLC默认IP地址为192.168.1.88，若不清楚汇川PLC的IP可以查询汇川PLC手册复位IP地址。



当汇川的D100数据与M4的数据相同时，连接成功



使用注意事项

- 在这个例子当中，需要注意PLC的IP地址不能填错或填反，否则会出现通信失败。
- PLC从站的程序可以为空，但是需要完成自身IP设置。

5.7 信捷XDH-30A16L-E与M4PLC使用ModbusTCP协议通讯

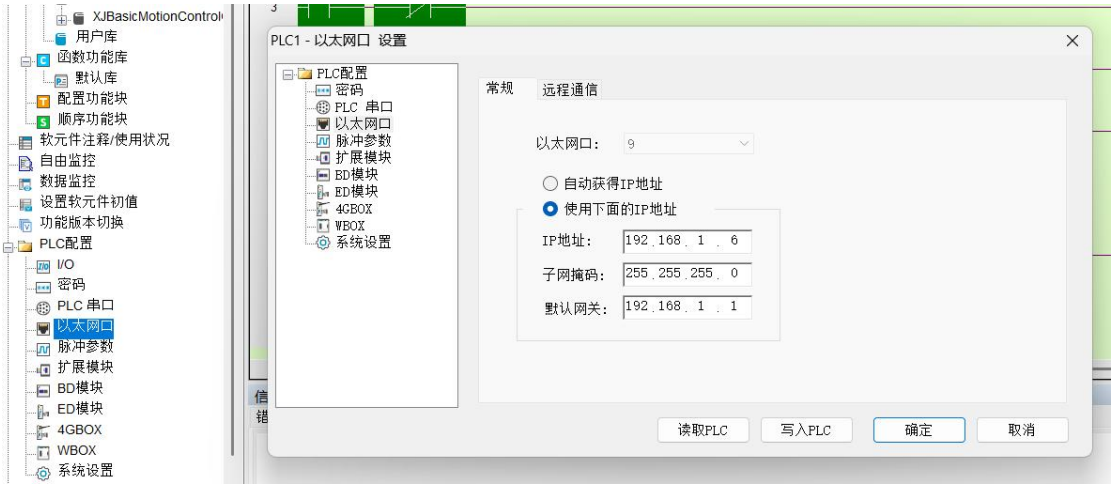
将M4PLC的IP设置为192.168.1.8。通过下面梯形图置位M42再进入元件监控，R2300至R2311的值IP导入成功。PLC程序如下：



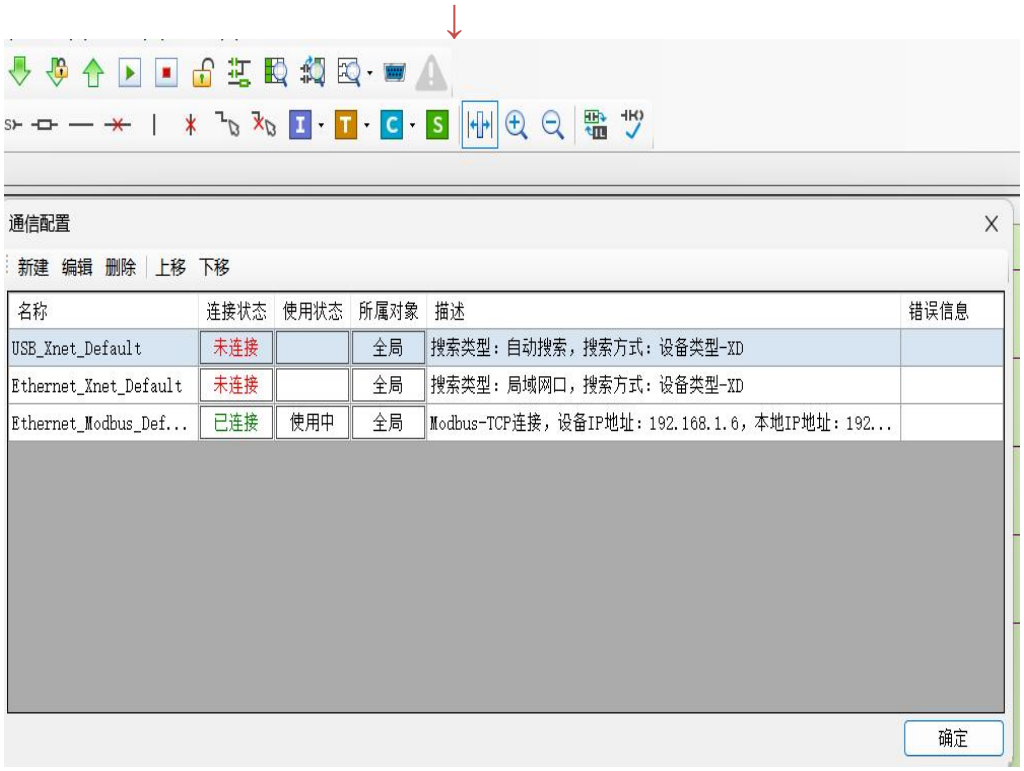
在这个例子中，我们将192.168.1.8的ModbusTCP服务器作为从站，连接信捷XDH-30A16L-E的ModbusTCP客户端。下面设置信捷PLC参数：

新建一个工程后如下图所示：点击左侧导航栏的以太网口，设置信捷PLC的IP为192.168.1.6设置IP完成后写入PLC，

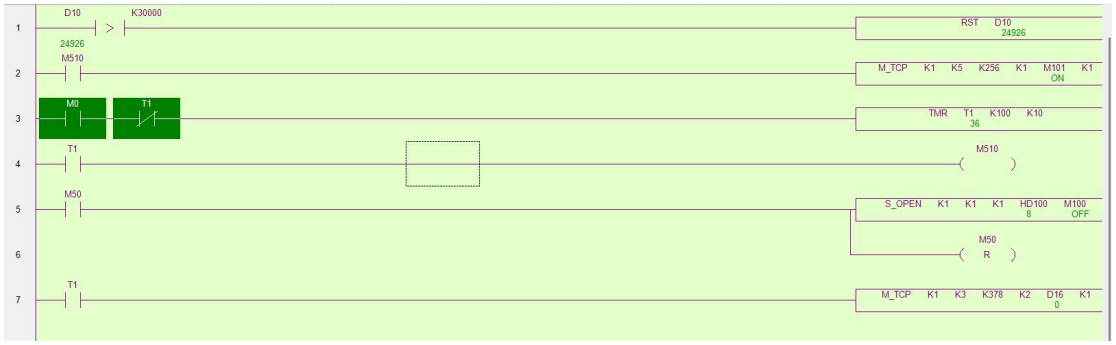
再确认。



上述完成后确定，再点下面图片红色箭头通信配置，如下图所示显示已连接，在使用中电脑与PLC以太网通信成功。



设置完成后，写入信捷以太网程序梯形图，如下图所示。



写完程序运行后，点鼠标右键S-OPEN，会出现从站IP连接设置，根据上面提示内容，设置好通信协议与从站IP

与端口号。



下一步，写出要通信的指令内容，点右键M-TCP出现下面对话框，可以任意配置相关通信内容，下面是信捷PLC的M101写入M4 PLC的Y0.当信捷PLC套接字打开后，根据上面信捷PLC梯形图M0置位ON后，就可以通过M101的控制M4 PLC的Y0。



同样点右键M-TCP出现下面对话框，出现下图所示，信捷PLC读M4 PLC的D10 D11，存放到D16 D17.



经过测试发现改变从站D10 D11 的值，信捷PLC读出来值与之一样，通信正常。

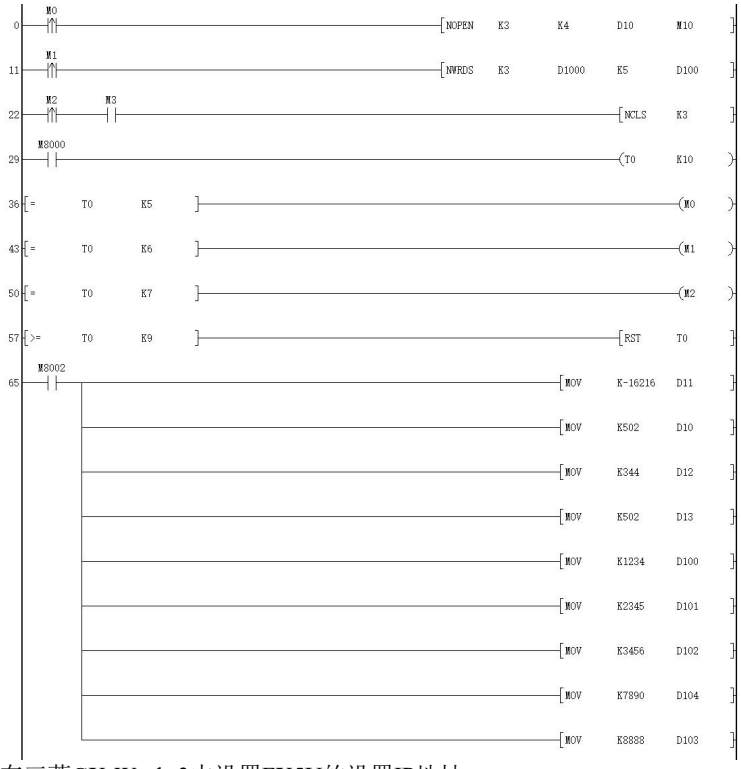
当信捷PLC当服务器时，只需要设置IP地址，M4 PLC当客户端则同二台科威PLC以太网配置相同，已不再重新举例说明。

5.8 三菱FX5U-32M与M4PLC使用ModbusTCP协议通讯

M4当主站（客户端），FX5U当从站（服务器）通讯测试，设置M4PLC的IP地址，与电脑、FX5U的在同一个网段，这里设置为192.168.1.9端口号为502



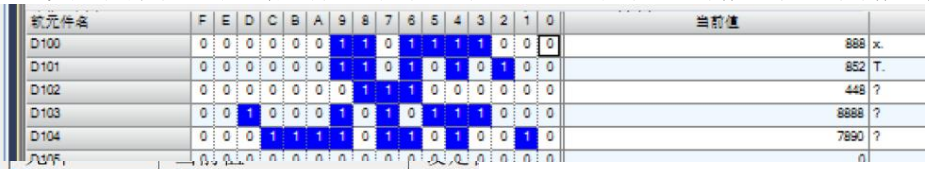
写入一段通讯程序，此程序每隔10ms会将M4PLC的D100到D104的数据
发送到客户端的D100到D104五个寄存器，按下M3通讯停止。客户端不会在接收数据。



在三菱GX-Works3中设置FX5U的设置IP地址

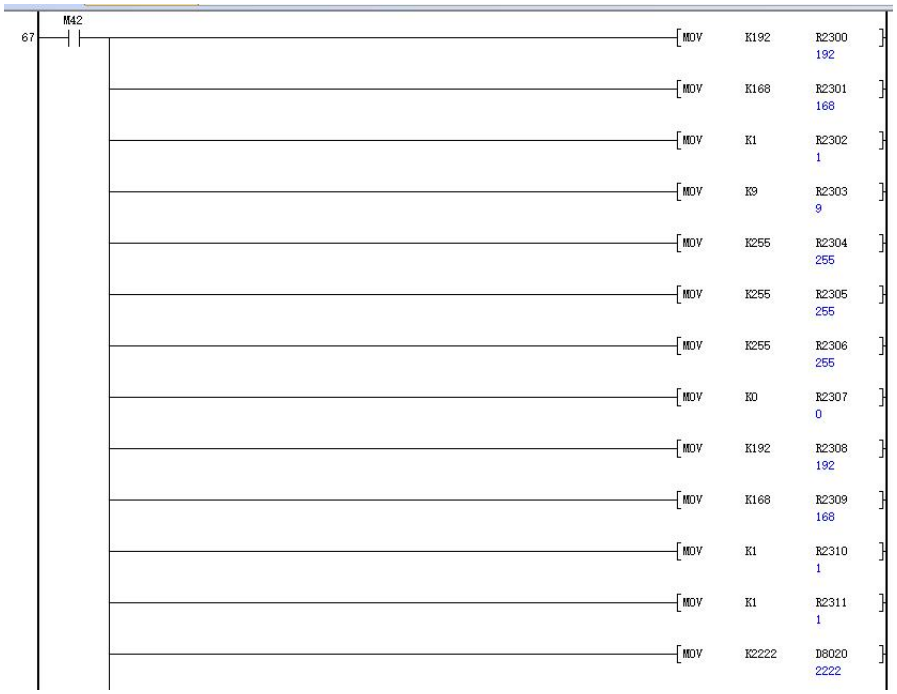


配置完成下载到PLC中，断电再上电启动，当GX Works3中的D100寄存器与M4的寄存器数据相同时，通讯成功。

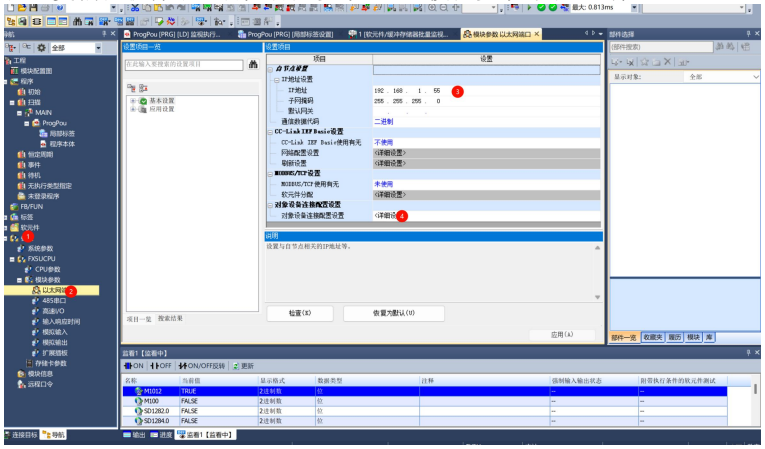


D100	888
D101	852
D102	448
D103	8888
D104	7890

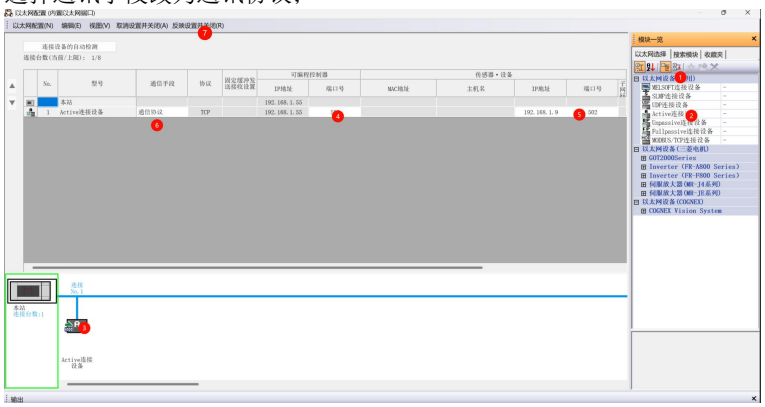
FX5U做主站，M4做从站。
在m4plc中设置本站的IP地址，无需其他操作。



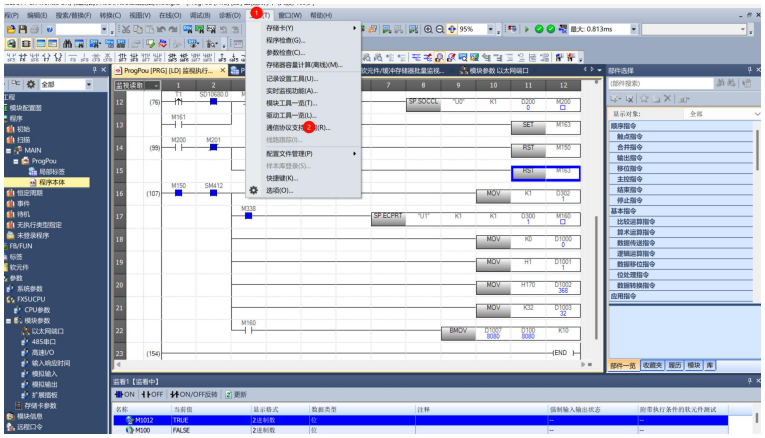
在三菱编程软件中，首先打开参数-CPU-模块参数-以太网端口；



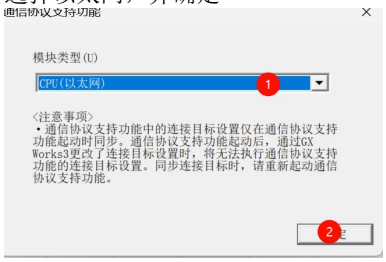
在设置完FX5U的IP地址后点击对象连接配置，因为作为主站所以我们选择active设备，并且设置客户端设备的IP地址，选择通讯手段改为通讯协议；



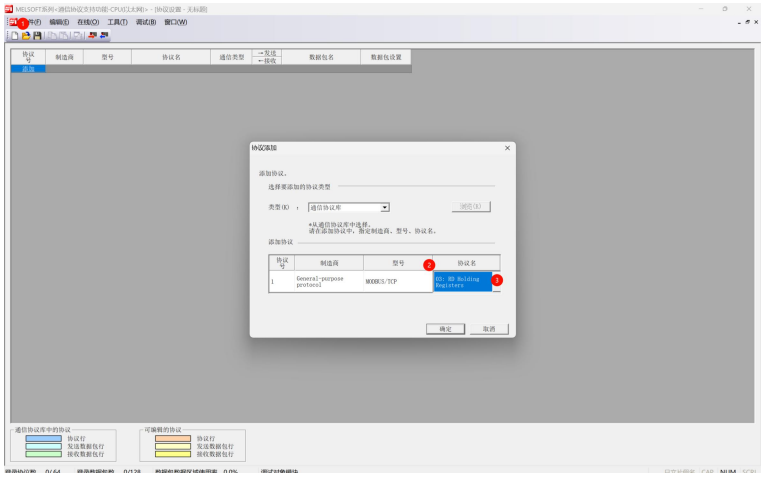
进行通讯协议设置，选择工具-通讯协议支持功能。



选择以太网，并确定



设置通信协议为Modbus TCP，功能码为03，读寄存器。



设置寄存器地址；

数据包设置

协议号	1	协议名	03: RD Holding Registers
数据包类型	发送数据包	数据包名(N)	Request

配置元素一览(L)

配置元素号	配置元素类型	配置元素名	配置元素设置
1	无转换变量	Transaction ID	[D1000-D1000](固定长度/2字节/下上字节/有更换)
2	固定数据	Protocol ID	0000(2字节)
3	长度	Length	(对象元素4-7/HEX/正/2字节)
4	无转换变量	Module ID	[D1001-D1001](固定长度/1字节/下上字节/无更换)
5	固定数据	Function Code	03(1字节)
6	无转换变量	Head holding register number	[D1002-D1002](固定长度/2字节/下上字节/有更换)
7	无转换变量	Read points	[D1003-D1003](固定长度/2字节/下上字节/有更换)

数据包设置

协议号	1	协议名	03: RD Holding Registers
数据包类型	接收数据包	数据包名(N)	Normal response
数据包号	1		

配置元素一览(L)

配置元素号	配置元素类型	配置元素名	配置元素设置
1	无转换变量	Transaction ID	[D1004-D1004](固定长度/2字节/下上字节/有更换)
2	固定数据	Protocol ID	0000(2字节)
3	长度	Length	(对象元素4-7/HEX/正/2字节)
4	无转换变量	Module ID	[D1005-D1005](固定长度/1字节/下上字节/无更换)
5	固定数据	Function Code	03(1字节)
6	长度	Number of read bytes	(对象元素7-7/HEX/1字节)
7	无转换变量	Device data	[D1006][D1007-D1131](可变长度/250字节/下上字节/有更换)

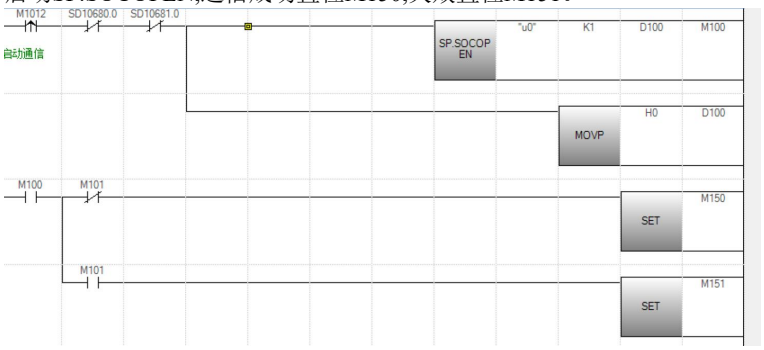


配置元素号	配置元素类型	配置元素名	配置元素设置
1	无转换变量	Transaction ID	[D1132-D1132](固定长度/2字节/下上字节/有更换)
2	固定数据	Protocol ID	0000(2字节)
3	长度	Length	(对象元素4-6/HEX/正/2字节)
4	无转换变量	Module ID	[D1133-D1133](固定长度/1字节/下上字节/无更换)
5	固定数据	Function Code	83(1字节)
6	无转换变量	Exception Code	[D1134-D1134](固定长度/1字节/下上字节/无更换)

通讯程序图如下：

软元件编号	名称	内容	R/W
SD10680	开放结束信号	各连接的开放结束信号。 [b0]～[b7]：连接No.1～连接No.8 0：关闭/开放未结束 1：开放结束	R
SD10681	开放请求信号	各连接的开放请求信号。 [b0]～[b7]：连接No.1～连接No.8 0：不可接受开放请求 1：可接受开放请求（等待开放请求状态）	R

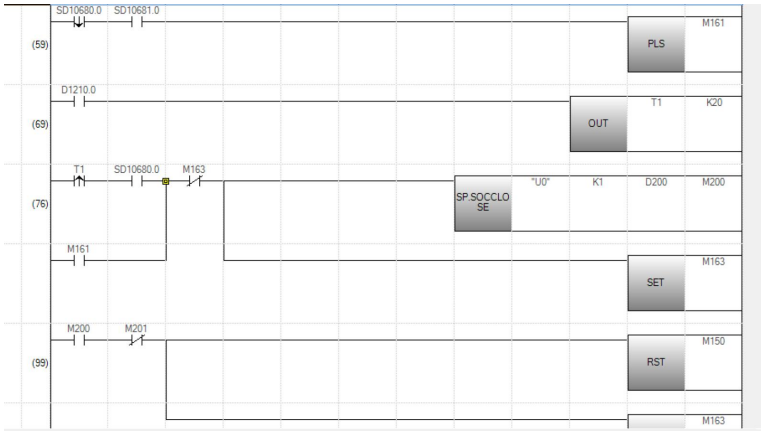
启动SP.SOCOPEN,通信成功置位M150,失败置位M151。



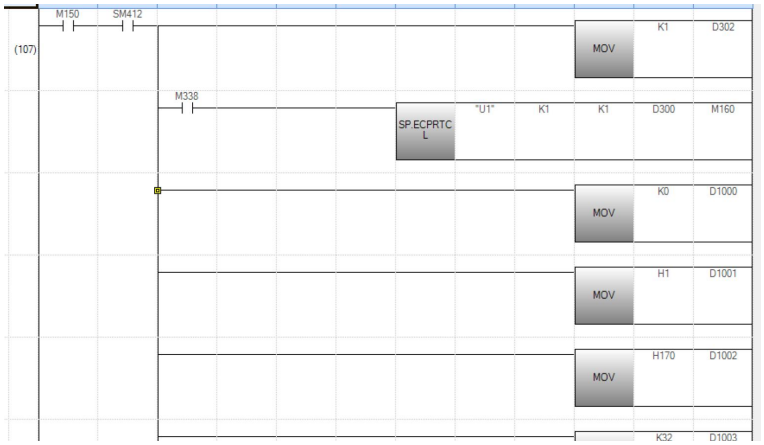
写入服务器的通信地址。



关闭通信SP.SOCCLOSE指令。



读取从站信息并且存入设定的寄存器中，D1002为寄存器起始地址，D1003为保存点数，数据会存放在D1007~D1131寄存器中。



与D0寄存器数据相同。



元件	当前值	设定值
D0(L)	8080	

6. 以太网通讯相关标志位和寄存器

6.1 以太网通讯相关寄存器

地址	功能	说明
R2300	IP网络号	IP地址的前两个字节
R2301		
R2302	IP站点号	IP地址的后两个字节
R2303		
R2304	子网掩码	子网掩码的前两个字节
R2305		
R2306		子网掩码的后两个字节
R2307		
R2308	默认网关	默认网关的前两个字节
R2309		
R2310		默认网关的后两个字节
R2311		
R2312	错误码	发生异常ID，仅在连接未建立时生效
R2313	错误码	1：套接字序号超限
		2：协议模式超限
		3：套接字已打开不能重复打开
		4：UDP内存不足
		5：UDP连接失败
		6：UDP组播加入失败
		7：UDP组播连接失败
		8：UDP组播地址超限
		13：TCP内存不足
		14：TCP客户端连接失败
		15：TCP客户端Modbus端口号≠502
		16：TCPmodbus客户端连接失败
		17：TCP服务器绑定失败
		18：TCP服务器Modbus端口号≠502
		19：TCPModbus服务器绑定失败
		24：NCLS套接字序号超限
		25：NCLS套接字未打开过
R2320	标志位	缺省ModbusTCP服务器时参数
R2321	接收超时时间，缺省值=150	缺省ModbusTCP服务器时参数
R2322	接收字节数	缺省ModbusTCP服务器时参数

6.2 以太网通讯相关标志位

位地址	标志位	功能说明
M（n+0）	通信打开关闭标志	打开通信M（n+0）置ON，关闭置OFF
M（n+1）	关闭中标志	关闭当前连接时，M（n+1）置ON
M（n+2）	连接中标志	连接建立过程中，M（n+2）置ON
M（n+3）	已连接标志	连接建立完成时，M（n+3）置ON
M（n+4）	发送中标志	数据发送过程中，M（n+4）置ON
M（n+5）	已发送标志	数据发送完成时，M（n+5）置ON
M（n+6）	接收中标志	数据接收过程中，M（n+6）置ON
M（n+7）	已接收标志	数据接收完成时，M（n+7）置ON
M（n+8）	Modbus tcp标志	ModbusTCP指令时，M（n+8）置ON
M（n+9）	通信异常标志	通信过程中出错，M（n+9）置ON

7. 以太网远程连接

本章主要讲解在公网环境中，如何通过以太网来远程控制PLC。

7.1 以太网远程连接概述

在大部分使用场景下，PLC所处的网络环境均为局域网，在没有公网IP与服务器的情况下，不处于同一局域网环境的设备无法直接连接PLC。因此对于有远程连接PLC的需求，在此提供一种基于内网穿透的实现方案，可以允许任意能够连接网络的设备远程访问PLC。

外部网络在通常情况下无法直接请求访问内网的设备，所以需要将内网设备连接到一个具有公网IP的服务器，让服务器对来自外部连接的请求进行转发。在与PLC处于同一内网的设备开启内网穿透工具的情况下，处于其他网络环境的设备可以通过访问内网穿透工具映射的公网IP与端口来访问本地的设备。

因此，在开始远程连接前，需要确保：

1、 PLC所在的局域网能够访问公共网络；

2、 准备内网穿透工具；

3、 同一局域网环境下的一台电脑（可选）；

7.2 以太网远程连接步骤

（1）准备工作

本例使用花生壳作为内网穿透工具，因此需要准备一个花生壳账号。准备好花生壳账号后，在电脑上安装花生壳工具。打开花生壳软件，登录自己的账号，在软件中首先要删掉内网穿透的全部映射，防止映射干扰。

（2）映射绑定

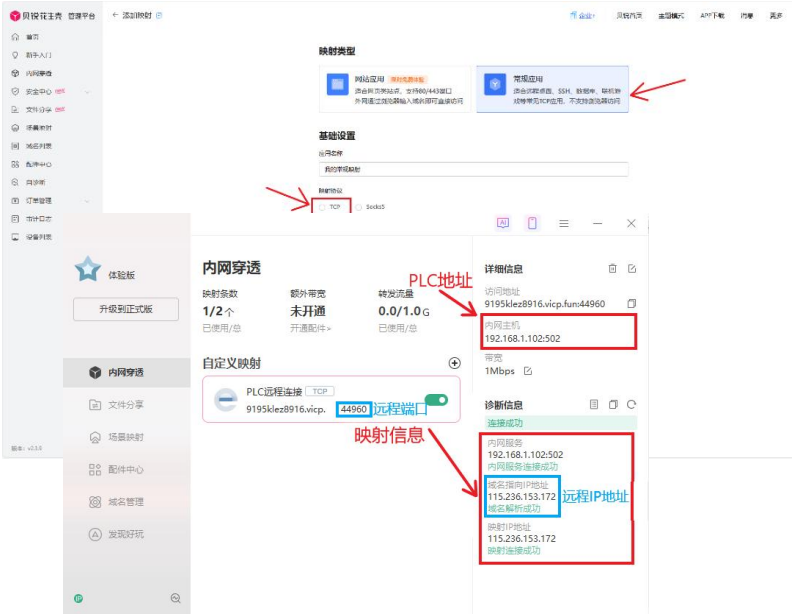
这一步中，需要将PLC的IP地址添加到花生壳的映射当中。首先确定PLC设置的IP地址，通过检查R寄存器中R2300至R2303四个寄存器中的值，通常为192.168.1.X，X为R2303当中的值。

花生壳软件中，点击加号添加自定义映射。



在弹出的网页中，映射类型选择常规应用。基础设置中的映射类型选择TCP。在内外网设置中，内网主机需要根据实际情况填写PLC的IP地址，通常为192.168.1.X，X为寄存器R2303当中的值，端口号可以设置成ModbusTCP的固定502端口，也可根据实际情况填写自由指令开启的套接字的端口号。外网域名与外网端口使用默认提供的即可。

完成后可以在花生壳软件中看到新增了一条映射，其中，内网主机显示的应该是PLC的地址与端口。下方的映射信息中会显示映射状态。映射IP地址即为可以连接PLC的公网IP。



当需要进行远程连接时，需要输入映射的远程IP地址和远程端口号。

（3）连接测试
对于已经建立好的映射，需要保持花生壳软件为开启状态，并且安装软件的电脑与PLC处于同一局域网。此时，处于任意网络环境的任意设备可以通过映射IP与映射端口连接PLC。以网络调试助手为例，使用任意网络环境的任意设备打开网络调试助手，输入映射IP与端口进行连接与ModbusTCP报文测试。



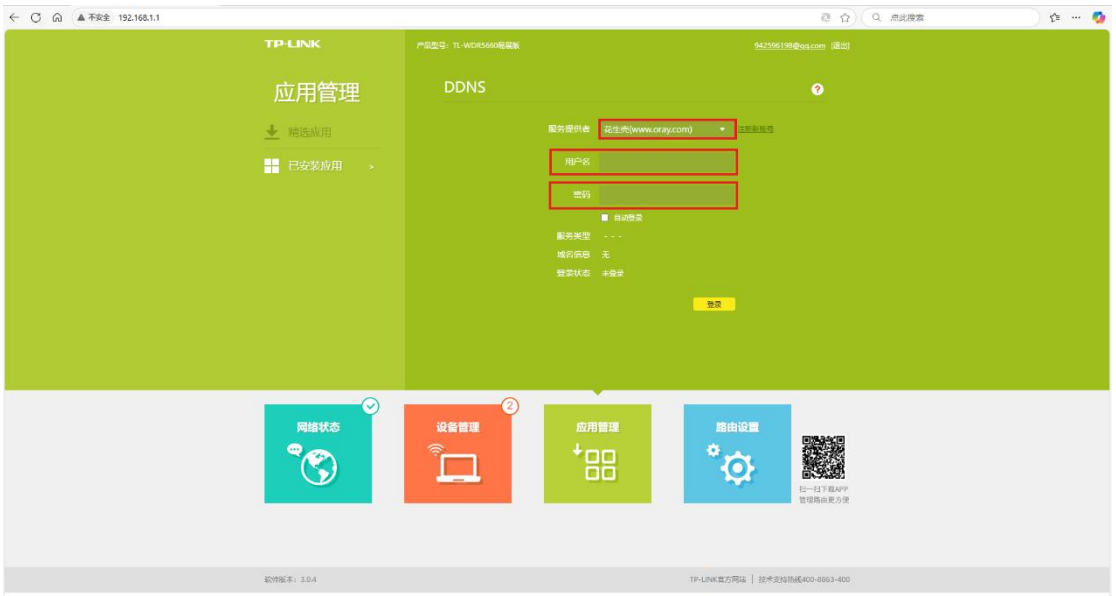
7.3 以太网远程连接注意事项

在使用花生壳内网穿透工具进行内网映射时，确保在创建对PLC的映射时，花生壳软件中不存在其他的映射，否则有可能导致创建的映射无法正常连接。
同时只能有一台设备开启内网穿透工具来进行内网映射。重复开启多个相同内网穿透工具，使用相同的IP与端口进行映射会造成冲突导致无法访问。
在路由器支持DDNS的情况下，可以使用路由器的内网穿透功能，不需要额外的一台电脑安装内网穿透工具。具体操作如下：

1、 访问路由器的设置页面。如果路由器支持DDNS，则可以在工具箱中找到DDNS工具。

2、 在DDNS界面，选择服务提供方为花生壳，并且填写好注册的花生壳账号与密码，点击登录即可完成内网映射。

以上操作仍然需要通过花生壳软件创建完成一条映射后才能够进行。
使用路由器的DDNS来进行内网映射有一定的不稳定性。



附录

附录一 常用资源列表

简称	代号及点数范围	说明
输入 X	X000～X377(8 进制)	输入继电器，单机点数由MC系列PLC具体型号而定
输出 Y	Y000～Y377(8 进制)	输出继电器，单机点数由MC系列PLC具体型号而定
辅助 M	M0～M3071	辅助继电器
	M8000～M8255	特殊辅助继电器，具体定义见相关章
状态 S	S0～S1023	状态继电器
时间 T	T0～T255	T0～T199，100ms 时基定时器
		T200～T245，10ms 时基定时器
		T246～T249，1ms 时基累计定时器
		T250～T255，100ms 时基累计定时器
计数 C	C0～C255	C0～C199，16 位增计数器
		C200～C234，32 位可逆计数器，计数方向由 M8200～M8234 控制
		C235～C255，32 位高速计数器(与高速计数指令配合使用)
寄存器 D	D0～D7999	D200～D3999，掉电保持区
	D8000～D8255	系统寄存器区，具体定义见相关章节
扩展寄存器R	R0～R7999	R0～R59，内部保留区
		R60～R83，输出轴脉冲显示区
		R84～R99，内部保留区
		R100～R259，软件虚拟输入轴参数区
		R260～R299，内部保留区
		R300～R899，软件虚拟输出轴参数区
		R900～R999，内部保留区
		R1000～R1199，内部保留区
		R1200~R1899，内部保留区
		R1900~R1999，内部保留区
		R2000~R2049，FLASH参数区
		R2050~R2099，内部保留区
		R2100~R2259，KLink总线资源配置参数区
		R2300~R2323，以太网通信参数区
		R2324~R7999，内部保留区
变址 V，Z	V0～V7，Z0～Z7	与其他软元件配合使用
常数 K，H	带符号整型数范围	K 为 10 进制表达数，H 为 16 进制表达数，如 H10=K16
程序指针 P	P0～P127	做程序跳转或子程序调用使用
中断指针 I	I000～I999	适用于各类型中断的指示标签

注意：设置 D8020=K2222，开启MC系列PLC的停电保持功能。停电保持功能设置后，停电情况下，系统会保存停电保持用的软元件状态等，再次上电时，恢复上次掉电时的状态(即上次掉电时的软元件数值)。详细说明见本手册第二章第2.5节。

附录二 特殊软元件功能一览表

特殊辅助继电器[M]功能一览表

软元件编号	软元件功能
M8000	PLC运行ON,其他状态OFF
M8001	PLC运行OFF,其他状态ON
M8002	PLC运行首次梯形图扫描周期ON,其余扫描周期OFF
M8003	PLC运行首次梯形图扫描周期OFF,其余扫描周期ON
M8004	设置PLC运行状态(ON-PLC运行，OFF-PLC停止)，操控该位必须满足以下前提条件： ● D8004 = HFAC0 ● M8005 = OFF
M8005	远程控制设置位(ON-物理控制，OFF-远程控制)
M8006	梯形图运行错误ON
M8007	外部RAM检测(ON-外部RAM运行正常，OFF-未检测到外部RAM)
M8008	掉电检测状态(ON-上电后发生掉电，OFF-上电后未发生掉电)
M8009	暂不开放使用
M8010	暂不开放使用
M8011	暂不开放使用
M8012	暂不开放使用
M8013	暂不开放使用
M8014	暂不开放使用
M8015	暂不开放使用
M8016	暂不开放使用
M8017	暂不开放使用
M8018	暂不开放使用
M8019	暂不开放使用
M8020	零点标志，加减运算结果为0时ON(保留)
M8021	借位标志，减法运算结果小于负最大值时ON(保留)
M8022	进位标志,进位循环，移位指令运算使用
M8023	暂不开放使用
M8024	暂不开放使用
M8025	暂不开放使用
M8026	暂不开放使用
M8027	暂不开放使用
M8028	暂不开放使用
M8029	暂不开放使用
M8030	暂不开放使用
M8031	暂不开放使用
M8032	输入中断I000申请
M8033	输入中断I001申请
M8034	输入中断I100申请
M8035	输入中断I101申请
M8036	输入中断I200申请
M8037	输入中断I201申请
M8038	输入中断I300申请
M8039	输入中断I301申请
M8040	定时中断I6□□申请
M8041	定时中断I7□□申请
M8042	定时中断I8□□申请
M8043	定时中断I010申请
M8044	软中断I020申请
M8045	软中断I030申请
M8046	软中断I040申请
M8047	软中断I050申请
M8048	输入中断I000禁止
M8049	输入中断I001禁止

软元件编号	软元件功能
M8050	输入中断I100禁止
M8051	输入中断I101禁止
M8052	输入中断I200禁止
M8053	输入中断I201禁止
M8054	输入中断I300禁止
M8055	输入中断I301禁止
M8056	定时中断I6□□禁止
M8057	定时中断I7□□禁止
M8058	定时中断I8□□禁止
M8059	软中断I010禁止
M8060	软中断I020禁止
M8061	软中断I030禁止
M8062	软中断I040禁止
M8063	软中断I050禁止
M8064	禁止高级中断I900(ON:DISABLE,OFF:ENABLE)
M8065	禁止高级中断I901(ON:DISABLE,OFF:ENABLE)
M8066	禁止高级中断I902(ON:DISABLE,OFF:ENABLE)
M8067	禁止高级中断I903(ON:DISABLE,OFF:ENABLE)
M8068	禁止高级中断I904(ON:DISABLE,OFF:ENABLE)
M8069	禁止高级中断I905(ON:DISABLE,OFF:ENABLE)
M8070	禁止高级中断I906(ON:DISABLE,OFF:ENABLE)
M8071	禁止高级中断I907(ON:DISABLE,OFF:ENABLE)
M8072	高级中断I900激活（ON:ACTIVE,OFF:END）
M8073	高级中断I901激活（ON:ACTIVE,OFF:END）
M8074	高级中断I902激活（ON:ACTIVE,OFF:END）
M8075	高级中断I903激活（ON:ACTIVE,OFF:END）
M8076	高级中断I904激活（ON:ACTIVE,OFF:END）
M8077	高级中断I905激活（ON:ACTIVE,OFF:END）
M8078	高级中断I906激活（ON:ACTIVE,OFF:END）
M8079	高级中断I907激活（ON:ACTIVE,OFF:END）
M8080	高级中断I900申请（ON:ASK.OFF:NO）
M8081	高级中断I901申请（ON:ASK.OFF:NO）
M8082	高级中断I902申请（ON:ASK.OFF:NO）
M8083	高级中断I903申请（ON:ASK.OFF:NO）
M8084	高级中断I904申请（ON:ASK.OFF:NO）
M8085	高级中断I905申请（ON:ASK.OFF:NO）
M8086	高级中断I906申请（ON:ASK.OFF:NO）
M8087	高级中断I907申请（ON:ASK.OFF:NO）
M8088	高级中断0进程申请字（1-ASK 0-NO）
M8089	高级中断1进程申请字（1-ASK 0-NO）
M8090	高级中断2进程申请字（1-ASK 0-NO）
M8091	高级中断3进程申请字（1-ASK 0-NO）
M8092	高级中断4进程申请字（1-ASK 0-NO）
M8093	高级中断5进程申请字（1-ASK 0-NO）
M8094	高级中断6进程申请字（1-ASK 0-NO）
M8095	高级中断7进程申请字（1-ASK 0-NO）
M8096	运动指令开关量输入（替代X000输入）
M8097	运动指令开关量输入（替代X001输入）
M8098	运动指令开关量输入（替代X002输入）
M8099	运动指令开关量输入（替代X003输入）
M8100	运动指令开关量输入（替代X004输入）
M8101	运动指令开关量输入（替代X005输入）
M8102	运动指令开关量输入（替代X006输入）
M8103	运动指令开关量输入（替代X007输入）
M8104	运动指令开关量输入（替代X010输入）
M8105	运动指令开关量输入（替代X011输入）
M8106	运动指令开关量输入（替代X012输入）
M8107	运动指令开关量输入（替代X013输入）
M8108	运动指令开关量输入（替代X014输入）
M8109	运动指令开关量输入（替代X015输入）
M8110	运动指令开关量输入（替代X016输入）
M8111	运动指令开关量输入（替代X017输入）
M8112	暂不开放使用

软元件编号	软元件功能
M8113	暂不开放使用
M8114	暂不开放使用
M8115	暂不开放使用
M8116	暂不开放使用
M8117	暂不开放使用
M8118	暂不开放使用
M8119	暂不开放使用
M8120	暂不开放使用
M8121	暂不开放使用
M8122	暂不开放使用
M8123	暂不开放使用
M8124	暂不开放使用
M8125	暂不开放使用
M8126	暂不开放使用
M8127	暂不开放使用
M8128	暂不开放使用
M8129	暂不开放使用
M8130	暂不开放使用
M8131	暂不开放使用
M8132	暂不开放使用
M8133	暂不开放使用
M8134	暂不开放使用
M8135	暂不开放使用
M8136	暂不开放使用
M8137	暂不开放使用
M8138	暂不开放使用
M8139	暂不开放使用
M8140	暂不开放使用
M8141	暂不开放使用
M8142	暂不开放使用
M8143	暂不开放使用
M8144	暂不开放使用
M8145	暂不开放使用
M8146	暂不开放使用
M8147	暂不开放使用
M8148	暂不开放使用
M8149	暂不开放使用
M8150	暂不开放使用
M8151	暂不开放使用
M8152	暂不开放使用
M8153	暂不开放使用
M8154	暂不开放使用
M8155	暂不开放使用
M8156	暂不开放使用
M8157	暂不开放使用
M8158	暂不开放使用
M8159	暂不开放使用
M8160	暂不开放使用
M8161	暂不开放使用
M8162	暂不开放使用
M8163	暂不开放使用
M8164	暂不开放使用
M8165	暂不开放使用
M8166	暂不开放使用
M8167	暂不开放使用
M8168	暂不开放使用
M8169	暂不开放使用
M8170	暂不开放使用
M8171	暂不开放使用
M8172	暂不开放使用
M8173	暂不开放使用
M8174	暂不开放使用
M8175	暂不开放使用
M8176	暂不开放使用
M8177	暂不开放使用

软元件编号	软元件功能
M8178	暂不开放使用
M8179	暂不开放使用
M8180	暂不开放使用
M8181	暂不开放使用
M8182	暂不开放使用
M8183	暂不开放使用
M8184	暂不开放使用
M8185	暂不开放使用
M8186	暂不开放使用
M8187	暂不开放使用
M8188	暂不开放使用
M8189	暂不开放使用
M8190	暂不开放使用
M8191	暂不开放使用
M8192	暂不开放使用
M8193	暂不开放使用
M8194	暂不开放使用
M8195	暂不开放使用
M8196	暂不开放使用
M8197	暂不开放使用
M8198	暂不开放使用
M8199	暂不开放使用
M8200	内部计数器C200方向控制位
M8201	内部计数器C201方向控制位
M8202	内部计数器C202方向控制位
M8203	内部计数器C203方向控制位
M8204	内部计数器C204方向控制位
M8205	内部计数器C205方向控制位
M8206	内部计数器C206方向控制位
M8207	内部计数器C207方向控制位
M8208	内部计数器C208方向控制位
M8209	内部计数器C209方向控制位
M8210	内部计数器C210方向控制位
M8211	内部计数器C211方向控制位
M8212	内部计数器C212方向控制位
M8213	内部计数器C213方向控制位
M8214	内部计数器C214方向控制位
M8215	内部计数器C215方向控制位
M8216	内部计数器C216方向控制位
M8217	内部计数器C217方向控制位
M8218	内部计数器C218方向控制位
M8219	内部计数器C219方向控制位
M8220	内部计数器C220方向控制位
M8221	内部计数器C221方向控制位
M8222	内部计数器C222方向控制位
M8223	内部计数器C223方向控制位
M8224	内部计数器C224方向控制位
M8225	内部计数器C225方向控制位
M8226	内部计数器C226方向控制位
M8227	内部计数器C227方向控制位
M8228	内部计数器C228方向控制位
M8229	内部计数器C229方向控制位
M8230	内部计数器C230方向控制位
M8231	内部计数器C231方向控制位
M8232	内部计数器C232方向控制位
M8233	内部计数器C233方向控制位
M8234	内部计数器C234方向控制位
M8235	内部计数器C235方向控制位
M8236	暂不开放使用
M8237	内部计数器C237方向控制位
M8238	暂不开放使用
M8239	内部计数器C239方向控制位
M8240	内部计数器C240方向控制位
M8241	内部计数器C241方向控制位

软元件编号	软元件功能
M8242	内部计数器C242方向控制位
M8243	暂不开放使用
M8244	暂不开放使用
M8245	暂不开放使用
M8246	暂不开放使用
M8247	暂不开放使用
M8248	暂不开放使用
M8249	暂不开放使用
M8250	暂不开放使用
M8251	暂不开放使用
M8252	暂不开放使用
M8253	暂不开放使用
M8254	暂不开放使用
M8255	暂不开放使用

特殊辅助寄存器[D]功能一览表

软元件编号	软元件功能
D8000	扫描周期定时监视器限值，单位为ms。当梯形图扫描周期超过该值时梯形图报错。
D8001	指令系列号
D8002	程序容量
D8003	驱动系列号
D8004	设置PLC远程控制(设置为HFAC0，则PLC处于远程控制模式)
D8005	文件写入操作口令(内部使用)
D8006	梯形图出错时的错误步号
D8007	100微秒定时器
D8008	毫秒滴答定时器低字
D8009	毫秒滴答定时器高字
D8010	PLC运行扫描周期实时值
D8011	PLC运行扫描周期最小值
D8012	PLC运行扫描周期最大值
D8013	时钟秒数据(保留)
D8014	时钟分数据(保留)
D8015	时钟时数据(保留)
D8016	时钟日数据(保留)
D8017	时钟月数据(保留)
D8018	时钟年数据(保留)
D8019	时钟星期数据(保留)
D8020	数据掉电保护方式，参考软元件的掉电保护章节 ● =1111 不带停电保护 ● =2222 带停电保护
D8021	掉电电压AD(实测值)
D8022	暂不开放使用
D8023	暂不开放使用
D8024	暂不开放使用
D8025	暂不开放使用
D8026	主控芯片端口A组状态
D8027	暂不开放使用
D8028	主控芯片端口B组状态
D8029	暂不开放使用
D8030	虚拟示波器触发方式
D8031	虚拟示波器触发单元
D8032	虚拟示波器触发门限
D8033	虚拟示波器采样周期
D8034	虚拟示波器采样点数
D8035	虚拟示波器记录起始寄存器序号
D8036	虚拟示波器记录区间起始寄存器序号
D8037	虚拟示波器记录区间终止寄存器序号
D8038	虚拟示波器状态设置/显示
D8039	虚拟示波器当前采样点
D8040	虚拟示波器A通道工作模式
D8041	虚拟示波器A通道采样寄存器序号
D8042	虚拟示波器B通道工作模式
D8043	虚拟示波器B通道采样寄存器序号
D8044	50us时间片内任务时间位置，数值越大则任务执行时间越短(满值3999)
D8045	定时中断最大占用时间
D8046	暂不开放使用
D8047	暂不开放使用
D8048	暂不开放使用
D8049	暂不开放使用
D8050	暂不开放使用
D8051	暂不开放使用
D8052	暂不开放使用
D8053	暂不开放使用
D8054	暂不开放使用
D8055	暂不开放使用
D8056	暂不开放使用
D8057	暂不开放使用
D8058	暂不开放使用

软元件编号	软元件功能
D8059	暂不开放使用
D8060	暂不开放使用
D8061	暂不开放使用
D8062	暂不开放使用
D8063	暂不开放使用
D8064	暂不开放使用
D8065	暂不开放使用
D8066	暂不开放使用
D8067	暂不开放使用
D8068	暂不开放使用
D8069	暂不开放使用
D8070	暂不开放使用
D8071	暂不开放使用
D8072	暂不开放使用
D8073	暂不开放使用
D8074	暂不开放使用
D8075	暂不开放使用
D8076	暂不开放使用
D8077	暂不开放使用
D8078	暂不开放使用
D8079	暂不开放使用
D8080	暂不开放使用
D8081	暂不开放使用
D8082	暂不开放使用
D8083	暂不开放使用
D8084	暂不开放使用
D8085	暂不开放使用
D8086	暂不开放使用
D8087	暂不开放使用
D8088	暂不开放使用
D8089	暂不开放使用
D8090	暂不开放使用
D8091	暂不开放使用
D8092	暂不开放使用
D8093	暂不开放使用
D8094	暂不开放使用
D8095	暂不开放使用
D8096	暂不开放使用
D8097	暂不开放使用
D8098	暂不开放使用
D8099	暂不开放使用
D8100	V0指针寄存器
D8101	V1指针寄存器
D8102	V2指针寄存器
D8103	V3指针寄存器
D8104	V4指针寄存器
D8105	V5指针寄存器
D8106	V6指针寄存器
D8107	V7指针寄存器
D8108	Z0指针寄存器
D8109	Z1指针寄存器
D8110	Z2指针寄存器
D8111	Z3指针寄存器
D8112	Z4指针寄存器
D8113	Z5指针寄存器
D8114	Z6指针寄存器
D8115	Z7指针寄存器
D8116	暂不开放使用
D8117	暂不开放使用
D8118	暂不开放使用
D8119	暂不开放使用
D8120	UART0协议控制字
D8121	UART0字节控制字
D8122	UART0占用D寄存器起始单元（默认7000）

软元件编号	软元件功能
D8123	UART1协议控制字
D8124	UART1字节控制字
D8125	UART1占用D寄存器起始单元（默认7020）
D8126	暂不开放使用
D8127	暂不开放使用
D8128	暂不开放使用
D8129	扩展占用D起始单元(默认7100)
D8130	暂不开放使用
D8131	暂不开放使用
D8132	暂不开放使用
D8133	暂不开放使用
D8134	暂不开放使用
D8135	暂不开放使用
D8136	暂不开放使用
D8137	暂不开放使用
D8138	暂不开放使用
D8139	暂不开放使用
D8140	暂不开放使用
D8141	暂不开放使用
D8142	暂不开放使用
D8143	暂不开放使用
D8144	暂不开放使用
D8145	暂不开放使用
D8146	暂不开放使用
D8147	暂不开放使用
D8148	暂不开放使用
D8149	暂不开放使用
D8150	暂不开放使用
D8151	暂不开放使用
D8152	暂不开放使用
D8153	暂不开放使用
D8154	暂不开放使用
D8155	暂不开放使用
D8156	暂不开放使用
D8157	暂不开放使用
D8158	暂不开放使用
D8159	暂不开放使用
D8160	X00上升沿计数（2KHz）
D8161	X01上升沿计数（2KHz）
D8162	X02上升沿计数（2KHz）
D8163	X03上升沿计数（2KHz）
D8164	X04上升沿计数（2KHz）
D8165	X05上升沿计数（2KHz）
D8166	X06上升沿计数（2KHz）
D8167	X07上升沿计数（2KHz）
D8168	暂不开放使用
D8169	暂不开放使用
D8170	暂不开放使用
D8171	暂不开放使用
D8172	暂不开放使用
D8173	暂不开放使用
D8174	暂不开放使用
D8175	暂不开放使用
D8176	暂不开放使用
D8177	暂不开放使用
D8178	暂不开放使用
D8179	暂不开放使用
D8180	Y0输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8181	Y1输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8182	Y2输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8183	Y3输出模式　0-PLC输出;1-保留;2-脉冲;3-保留
D8184	Y4输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8185	Y5输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8186	Y6输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位

软元件编号	软元件功能
D8187	Y7输出模式　0-PLC输出;1-高速事件输出;2-无;3-轴方向位
D8188	暂不开放使用
D8189	暂不开放使用
D8190	暂不开放使用
D8191	暂不开放使用
D8192	暂不开放使用
D8193	暂不开放使用
D8194	暂不开放使用
D8195	暂不开放使用
D8196	暂不开放使用
D8197	暂不开放使用
D8198	暂不开放使用
D8199	暂不开放使用
D8200	暂不开放使用
D8201	暂不开放使用
D8202	暂不开放使用
D8203	暂不开放使用
D8204	暂不开放使用
D8205	暂不开放使用
D8206	暂不开放使用
D8207	暂不开放使用
D8208	暂不开放使用
D8209	暂不开放使用
D8210	暂不开放使用
D8211	暂不开放使用
D8212	暂不开放使用
D8213	暂不开放使用
D8214	暂不开放使用
D8215	暂不开放使用
D8216	暂不开放使用
D8217	暂不开放使用
D8218	暂不开放使用
D8219	暂不开放使用
D8220	暂不开放使用
D8221	暂不开放使用
D8222	暂不开放使用
D8223	暂不开放使用
D8224	暂不开放使用
D8225	暂不开放使用
D8226	暂不开放使用
D8227	暂不开放使用
D8228	暂不开放使用
D8229	暂不开放使用
D8230	暂不开放使用
D8231	暂不开放使用
D8232	暂不开放使用
D8233	暂不开放使用
D8234	暂不开放使用
D8235	暂不开放使用
D8236	暂不开放使用
D8237	暂不开放使用
D8238	暂不开放使用
D8239	暂不开放使用
D8240	暂不开放使用
D8241	暂不开放使用
D8242	暂不开放使用
D8243	暂不开放使用
D8244	暂不开放使用
D8245	暂不开放使用
D8246	暂不开放使用
D8247	暂不开放使用
D8248	暂不开放使用
D8249	暂不开放使用
D8250	暂不开放使用
D8251	暂不开放使用

软元件编号	软元件功能
D8252	暂不开放使用
D8253	暂不开放使用
D8254	加密控制。PLC停止状态时写D8254为以下数值，并下载一个与MC系列PLC内相异的梯形图后，系统即进行加密操作。 ● D8254 = K12345 禁读 ● D8254 = K-11215 禁读禁写 * 加密为不可逆操作，需谨慎操作，后果自负。
D8255	内核版本号