
**Sách Hướng dẫn [Lệnh]
cho PLC dòng XD/XL**



TỰ ĐỘNG HÓA TOÀN CẦU

Sách Hướng dẫn [Lệnh]
cho PLC dòng XD/XL

Lời nói đầu

Sơ lược về lập trình

Chức năng phần tử lập trình

Lệnh lập trình cơ bản

Lệnh ứng dụng

Bộ đếm tốc độ cao

Xuất xung

Chức năng giao tiếp

Chức năng PID

Khối hàm trong ngôn ngữ C

KHỐI tuần tự

Lệnh chức năng đặc biệt

Ứng dụng

Các câu hỏi thường gặp & Trả lời

Phụ lục

- Giải thích cơ bản

Cảm ơn Quý khách đã mua PLC dòng XD/XL của chúng tôi.

Sách hướng dẫn này chủ yếu giới thiệu các lệnh PLC dòng XD/XL.

Vui lòng đọc kỹ sách hướng dẫn này trước khi sử dụng và thực hiện đấu nối sau khi hiểu rõ nội dung.

Về phần mềm và hướng dẫn lập trình, vui lòng tham khảo các sách hướng dẫn liên quan.

Vui lòng giao sách hướng dẫn này cho người sử dụng vận hành.

- Lưu ý dành cho Người sử dụng

Chỉ những người vận hành có kinh nghiệm mới có thể đấu nối dây PLC. Nếu có vấn đề gì, xin hãy vui lòng liên hệ bộ phận kỹ thuật của chúng tôi

Các ví dụ được thêm vào nhằm mục đích minh họa cho người dùng, vì vậy nó có thể không như thực tế.

Vui lòng tuân thủ các thông số và nguyên tắc của PLC khi kết nối PLC với các sản phẩm khác.

Vui lòng tuân thủ an toàn PLC và máy móc khi sử dụng PLC. Máy có thể bị hư hỏng do lỗi PLC

- Tuyên bố trách nhiệm

Nội dung trong sách hướng dẫn đã được kiểm tra cẩn thận nhưng không thể tránh khỏi sai sót. Chúng tôi sẽ thường xuyên kiểm tra nội dung của sách hướng dẫn và thực hiện chỉnh sửa trong các phiên bản tiếp theo. Chúng tôi hoan nghênh mọi đóng góp của bạn.

Vui lòng thông cảm chúng tôi sẽ không thông báo cho bạn nếu sách hướng dẫn được thay đổi.

- Thông tin liên hệ

Nếu bạn có bất kỳ thắc mắc gì về sản phẩm, vui lòng liên hệ với chúng tôi.

Số điện thoại: 0961.320.333

Địa chỉ: Lô 17-F1 khu đô thị Đại Kim, phường Đại Kim- Hoàng Mai - HN

Mã số doanh nghiệp: 0107396499

Bản quyền thuộc về **CÔNG TY CỔ PHẦN TỰ ĐỘNG HÓA TOÀN CẦU**

Không sao chép hoặc sử dụng hướng dẫn sử dụng mà không có sự chấp thuận bằng văn bản. Người vi phạm phải chịu trách nhiệm về những thiệt hại. Vui lòng giữ tất cả bản quyền của công ty chúng tôi bao gồm các mô-đun thực tế, bằng sáng chế được thiết kế và bản quyền được đề cập trong sách này.

MỤC LỤC

1 SƠ LƯỢC VỀ LẬP TRÌNH.....	10
1-1 PLC TÍNH NĂNG PLC.....	10
1-2 NGÔN NGỮ LẬP TRÌNH.....	11
1-2-1 Loại ngôn ngữ lập trình.....	11
1-2-2 Chuyển đổi ngôn ngữ lập trình.....	11
1-3 CHẾ ĐỘ LẬP TRÌNH.....	12
2 CHỨC NĂNG THÀNH PHẦN MỀM.....	13
2-1 SƠ LƯỢC VỀ CÁC THÀNH PHẦN MỀM.....	13
2-2 CẤU TRÚC CỦA THÀNH PHẦN MỀM.....	16
2-2-1 Cấu trúc bộ nhớ.....	16
2-2-2 Cấu trúc phần tử lập trình dạng bit.....	18
2-3 DANH SÁCH THÀNH PHẦN MỀM.....	19
2-3-1 Danh sách phần tử lập trình.....	19
2-4 RƠ-LE ĐẦU VÀO/ĐẦU RA (X, Y).....	36
2-5 RƠ-LE TRUNG GIAN (M, HM, SM).....	38
2-6 RƠ-LE TRẠNG THÁI (S, HS).....	39
2-7 BÔ ĐÌNH THỜI (T, HT).....	40
2-8 BÔ ĐẾM(C, HC, HSC).....	43
2-9 THANH GHI DỮ LIỆU (D, HD, SD, HSD).....	48
2-9-1 Phần tử lập trình bit.....	51
2-9-2 Ứng dụng của thanh ghi con trỏ.....	52
2-10 THANH GHI FLASH (FD, SFD, FS).....	53
2-11 HẰNG SỐ.....	54
2-12 NGUYÊN TẮC LẬP TRÌNH.....	55
3 CÁC LỆNH CHƯƠNG TRÌNH CƠ BẢN.....	59
3-1 DANH SÁCH LỆNH CƠ BẢN.....	59
3-2 LỆNH [LD] , [LDI] , [OUT].....	61
3-3 LỆNH [AND] , [ANI].....	62
3-4 LỆNH [OR] , [ORI].....	63
3-5 LỆNH [LDP] , [LDF] , [ANDP] , [ANDF] , [ORP] , [ORF].....	64
3-6 LỆNH [LDD] , [LDDI] , [ANDD] , [ANDDI] , [ORD] , [ORDI] , [OUTD].....	66
3-7 LỆNH [ORB].....	67
3-8 LỆNH[ANB].....	68
3-9 LỆNH[MCS] , [MCR].....	68
3-10 LỆNH[ALT].....	69
3-11 LỆNH[PLS] , [PLF].....	70
3-12 LỆNH [SET], [RST].....	71
3-13 LỆNH [CNT],[CNT D],[DCNT],[DCNT D],[RST].....	72
3-14 LỆNH [TMR], [TMR A].....	74
3-15 LỆNH [END].....	75

3-16 [GROUP], [GROUPE].....	76
3-17 LƯU Ý TRNG LẬP TRÌNH.....	76
4 CÁC LỆNH ỨNG DỤNG.....	78
4-1 DANH SÁCH CÁC LỆNH ỨNG DỤNG	78
4-2 PHƯƠNG PHÁP ĐỌC LỆNH ÁP DỤNG.....	82
4-3 LỆNH LƯỒNG CHƯƠNG TRÌNH.....	85
4-3-1 Lệnh [CJ]	85
4-3-2 Lệnh [CALL] và [SRET]	86
4-3-3 Lệnh [SET], [ST], [STL], [STLE]	88
4-3-4 Lệnh [FOR] và [NEXT]	93
4-3-5 Lệnh [FEND] và [END]	95
4-4 LỆNH SO SÁNH DỮ LIỆU.....	97
4-4-1 Lệnh [LD]	97
4-4-2 Lệnh [AND]	98
4-4-3 Lệnh [OR]	100
4-5 LỆNH CHUYỂN DỮ LIỆU.....	102
4-5-1 Lệnh [CMP, DCMP, QCMP].....	102
4-5-2 Lệnh [ZCP, DZCP].....	103
4-5-3 Lệnh MOV [MOV, DMOV, QMOV]	104
4-5-4 Lệnh [BMOV].....	106
4-5-5 Lệnh [PMOV]	108
4-5-6 Lệnh [FMOV, DFMOV]	109
4-5-7 Lệnh [EMOV, EDMOV]	110
4-5-8 Lệnh ghi FlashROM [FWRT, DFWRT, QFWRT].....	112
4-5-9 Lệnh [MSET]	113
4-5-10 Lệnh [ZRST].....	114
4-5-11 Lệnh [SWAP]	115
4-5-12 Lệnh [XCH, DXCH]	116
4-6 LỆNH PHÉP TOÁN DỮ LIỆU.....	118
4-6-1 Lệnh [ADD, DADD, QADD]	118
4-6-2 Lệnh [SUB]	121
4-6-3 Lệnh [MUL, DMUL, QMUL]	122
4-6-4 Lệnh [DIV, DDIV, QDIV]	124
4-6-5 Lệnh [INC, DINC, QINC] & lệnh [DEC, DDEC, QDEC]	126
4-6-6 Lệnh [MEAN, DMEAN]	128
4-6-7 Lệnh [WAND, DWAND], Lệnh [WOR, DWOR], Lệnh [WXOR, DWXOR].....	129
4-6-8 Lệnh [CML, DCML].....	131
4-6-9 Lệnh [NEG, DNEG]	132
4-7 LỆNH DỊCH CHUYỂN.....	133
4-7-1 Lệnh [SHL, DSHL], Lệnh [SHR, DSHR].....	133
4-7-2 Lệnh [LSL], Lệnh [LSR]	135
4-7-3 Lệnh [ROL, DROL], Lệnh [ROR, DROR].....	136
4-7-4 Lệnh [SFIL]	138

4-7-5 Lệnh [SFTR]	139
4-7-6 Lệnh [WSFL]	140
4-7-7 Lệnh [WSFR]	141
4-8 LỆNH CHUYỂN ĐỔI DỮ LIỆU.....	142
4-8-1 Lệnh [WTD,DWTD].....	143
4-8-2 Lệnh [BDWTD].....	144
4-8-3 Lệnh [FLT, DFLT,FLTD]	146
4-8-4 Lệnh [DFLTD,QFLTD]	147
4-8-5 Lệnh [INT, DINT]	148
4-8-6 Lệnh [DINTD,QINTD]	149
4-8-7 Lệnh [ECON].....	150
4-8-8 Lệnh [BECON].....	151
4-8-9 Lệnh [BIN]	152
4-8-10 Lệnh [BCD]	154
4-8-11 Lệnh [ASCII]	155
4-8-12 Lệnh[HEX]	157
4-8-13 Lệnh [DECO]	158
4-8-14 Lệnh [ENCO]	160
4-8-15 Lệnh [ENCOL].....	162
4-8-16 Lệnh [GRY].....	165
4-8-17 Lệnh [GBIN,DGBIN]	166
4-9. LỆNH PHÉP TOÁN SỐ THỰC DẤU PHẪY ĐỒNG.....	167
4-9-1 Lệnh [ECMP,EDCMP].....	167
4-9-2 Lệnh [EZCP]	169
4-9-3 Lệnh [EADD,EDADD]	170
4-9-4 Lệnh [ESUB,EDSUB]	172
4-9-5 Lệnh [EMUL,EDMUL].....	174
4-9-6 Lệnh [EDIV,EDDIV]	175
4-9-7 Lệnh[ESQR]	177
4-9-8 Lệnh [SIN]	178
4-9-9 Lệnh[COS].....	179
4-9-10 Lệnh [TAN]	180
4-9-11 Lệnh [ASIN]	181
4-9-12 Lệnh [ACOS]	182
4-9-13 Lệnh [ATAN]	183
4-10 LỆNH RTC.....	184
4-10-1 Lệnh [TRD]	185
4-10-2 Lệnh [TWR].....	186
4-10-3 Lệnh [MOV]	188
4-10-4 Lệnh [TO].....	189
4-10-5 Lệnh [TADD]	191
4-10-6 Lệnh [TSUB]	192
4-10-7 Lệnh [HTOS].....	194

4-10-8 Lệnh [STOH]	195
4-10-9 Lệnh [TCMP]	196
4-10-10 Lệnh [DACMP]	197
5 BỘ ĐẾM TỐC ĐỘ CAO (HSC)	200
5-1 TÓM TẮT CHỨC NĂNG	200
5-2 CHẾ ĐỘ HSC	201
5-3 PHẠM VI HSC	203
5-4 NỐI DÂY ĐẦU VÀO HSC	203
5-5 GÁN CỒNG HSC	204
5-6 CÀI ĐẶT NHẬN ĐÔI TẦN SỐ ĐẾM PHA AB	209
5-7 LỆNH HSC	210
5-7-1 Lệnh [CNT]	210
5-7-2 Lệnh [CNT_AB]	211
5-7-3 Lệnh [RST]	212
5-7-4 Lệnh [DMOV]	213
5-7-5 Lệnh [DMOV]	214
5-7-6 Sự khác biệt giữa HSC và máy đếm thông thường	215
5-8 VÍ DỤ VỀ HSC	215
5-9 NGẮT HSC	217
5-9-1 Tổng quan về chức năng và cấu hình bảng điều khiển	217
5-9-2 HSC 1 pha 100 đoạn [CNT]	219
5-9-3 HSC pha AB 100 đoạn [CNT_AB]	220
5-9-4 Cờ ngắt của HSC	222
5-9-5 Ý nghĩa giá trị cài đặt ở chế độ tuyệt đối hoặc tương đối	222
5-9-6 Chế độ chu kỳ ngắt HSC	225
5-9-7 Chức năng CAM ngắt bộ đếm tốc độ cao	226
5-9-8 Ngắt sử dụng ghi chú và địa chỉ thông số	227
5-9-9 Ứng dụng ngắt HSC	228
6 CHỨC NĂNG GIAO TIẾP	233
6-1 SƠ LƯỢC	233
6-1-1 CỒNG COM	233
6-1-2 Thông số giao tiếp	242
6-2 GIAO TIẾP MODBUS	243
6-2-1 Tổng quan chức năng	243
6-2-2 Thay đổi lệnh Modbus	244
6-2-3 Địa chỉ giao tiếp Modbus	244
6-2-4 Định dạng dữ liệu Modbus	255
6-2-5 Lệnh giao tiếp	261
6-2-6 Cấu hình cổng nối tiếp Modbus	270
6-2-7 Ứng dụng giao tiếp Modbus	274
6-2-8 Ứng dụng	276
6-3 GIAO TIẾP TỰ DO	279
6-3-1 Chế độ giao tiếp tự do	279

6-3-2 Cấu hình cổng nối tiếp.....	280
6-3-3 Các trường hợp áp dụng phù hợp.....	282
6-3-4 Lệnh giao tiếp tự do.....	282
6-3-5 Ví dụ về giao tiếp tự do.....	286
6-4 CỜ VÀ THANH GHI GIAO TIẾP.....	292
6-5 ĐỌC/GHI THAM SỐ CỔNG NỐI TIẾP.....	296
6-5-1 Lệnh đọc tham số cổng nối tiếp[CFGCR]	296
6-5-2 Lệnh ghi tham số cổng nối tiếp [CFGCW]	297
6-5-3 Tên và cài đặt tham số cổng nối tiếp.....	298
7 CHỨC NĂNG ĐIỀU KHIỂN PID.....	300
7-1 GIỚI THIỆU VỀ PID.....	300
7-2 LỆNH ĐỊNH DẠNG.....	300
7-3 CÀI ĐẶT THAM SỐ.....	302
7-3-1 Thanh ghi và các chức năng.....	303
7-3-2 Mô tả tham số.....	307
7-4 CHẾ ĐỘ ĐIỀU CHỈNH TỰ ĐỘNG.....	309
7-5 CHẾ ĐỘ NÂNG CAO.....	312
7-6 TÓM TẮT ỨNG DỤNG.....	313
7-7 ỨNG DỤNG	313
8 KHỐI HÀM TRONG NGÔN NGỮ C.....	319
8-1 SƠ LƯỢC.....	319
8-2 ĐỊNH DẠNG LỆNH.....	319
8-3 CÁC BƯỚC THAO TÁC.....	320
8-4 NHẬP VÀ XUẤT CÁC HÀM.....	323
8-5 CHỈNH SỬA KHỐI HÀM.....	324
8-6 VÍ DỤ VỀ CHƯƠNG TRÌNH.....	326
8-7 CÁC HÀM MỚI.....	329
8-8 THƯ VIỆN CHỨC NĂNG.....	331
8-8-1 Chức năng mới.....	332
8-8-2 Các chức năng cơ bản.....	332
8-8-3 Mới xây.....	332
8-8-4 Chỉnh sửa.....	336
8-8-5 Xuất.....	339
8-8-6 Nhập.....	340
8-8-7 Các chức năng khác.....	341
8-9 LƯU Ý KHI ỨNG DỤNG.....	347
8-10 HỎI VÀ ĐÁP VỀ NGÔN NGỮ C.....	350
8-11 BẢNG CHỨC NĂNG.....	352
9 KHỐI TUẦN TỰ.....	356
9-1 KHÁI NIỆM BLOCK/KHỐI.....	356
9-2 GOI BLOCK	357
9-2-1 Thêm BLOCK	357

9-2-2 Di chuyển BLOCK	360
9-2-3 Xóa BLOCK	361
9-2-4 Chỉnh sửa BLOCK	361
9-3 CHỈNH SỬA LỆNH CỦA BLOCK	362
9-3-1 Command item	362
9-3-2 Pulse Item	364
9-3-3 Wait Item	365
9-3-4 Lệnh đọc và ghi (FROM-TO) mô-đun	366
9-4 PHƯƠNG THỨC CHẠY BLOCK.....	367
9-5 QUY TẮC CHỈNH SỬA LỆNH BLOCK.....	369
9-6 CÁC LỆNH LIÊN QUAN ĐẾN BLOCK.....	371
9-6-1 Giải thích lệnh.....	371
9-6-2 Thứ tự thời gian của các lệnh.....	373
9-7 BIT CỜ VÀ THANH GHI CỦA BLOCK.....	377
10 CÁC LỆNH CHỨC NĂNG ĐẶC BIỆT.....	378
10-1 LỆNH [PWM]	378
10-2 LỆNH [FROM]	382
10-3 LỆNH [STR]	385
10-4 LỆNH [EI], [DI], [IRET]	390
10-4-1 Ngắt ngo ài	390
10-4-2 Ngắt bộ định thời.....	396
10-5 ĐỌC/GHI THẺ SD.....	398
10-5-1 Nội dung và định dạng tài liệu.....	398
10-5-2 Tên file và vị trí lưu trữ.....	399
10-5-3 Đọc/ghi thẻ SD.....	399
10-5-4 Lưu ý.....	400
10-5-5 Các công cụ sử dụng để chuyển đổi định dạng.....	402
10-6 LỆNH [MSC]	404
11 CÁC CÂU HỎI THƯỜNG GẶP VÀ TRẢ LỜI.....	411
CÂU HỎI 1:.....	411
CÂU HỎI 2:.....	414
CÂU HỎI 3:.....	416
CÂU HỎI 4:.....	421
CÂU HỎI 5:.....	422
CÂU HỎI 6:.....	422
CÂU HỎI 7:.....	422
CÂU HỎI 8:.....	423
CÂU HỎI 9:.....	423
CÂU HỎI 10:.....	423
CÂU HỎI 11:.....	423
CÂU HỎI 12:.....	424
CÂU HỎI 13:	424
CÂU HỎI 14:.....	425

<u>Câu hỏi 15:</u>	425
<u>Câu hỏi 16:</u>	425
<u>Câu hỏi 17:</u>	425
<u>Câu hỏi 18:</u>	426
<u>Câu hỏi 19:</u>	426
<u>Câu hỏi 20:</u>	426
<u>Câu hỏi 21:</u>	426
<u>Câu hỏi 22:</u>	427
<u>Câu hỏi 23:</u>	427
<u>Câu hỏi 24:</u>	428
<u>Câu hỏi 25:</u>	429
<u>Câu hỏi 26:</u>	429
<u>Câu hỏi 27:</u>	429
<u>Câu hỏi 28:</u>	431
<u>PHU LUC BIT VÀ THANH GHI ĐẶC BIỆT.....</u>	432
<u>PHU LUC 1 Rơ-le TRUNG GIAN ĐẶC BIỆT.....</u>	432
<u>PHU LUC 2 THANH GHI DỮ LIỆU ĐẶC BIỆT.....</u>	438
<u>PHU LUC 3 THANH GHI FLASH ĐẶC BIỆT.....</u>	447
<u>PHU LUC 4 BẢNG XUNG ĐỘT TÀI NGUYÊN PLC.....</u>	451
<u>PHU LUC 5 DANH SÁCH CẤU HÌNH CHỨC NĂNG PLC.....</u>	452

Chương 1: Sơ lược về lập trình

PLC dòng XD/XL nhận tín hiệu và thực hiện chương trình trong bộ điều khiển để đáp ứng yêu cầu của người dùng. Chương này giới thiệu các tính năng của PLC, hai loại ngôn ngữ lập trình, v.v.

1-1 Các tính năng của PLC

Ngôn ngữ lập trình

PLC dòng XD/XL hỗ trợ hai loại ngôn ngữ lập trình, Instruction (Lệnh) và Ladder diagram (Sơ đồ bậc thang), hai loại ngôn ngữ có thể chuyển đổi lẫn nhau.

Bảo mật chương trình

Để tránh việc chương trình người dùng bị đánh cắp hoặc thay đổi không chính xác, chúng tôi mã hóa chương trình. Khi tải lên chương trình được mã hóa, nó sẽ kiểm tra dưới dạng mật khẩu. Điều này có thể bảo vệ bản quyền của người dùng; đồng thời hạn chế việc tải xuống, tránh trường hợp thay đổi chương trình do nhầm lẫn. Dòng XD/XL đã thêm thanh ghi FS mới. (Đối với các mẫu XD/XL khác nhau, vui lòng chọn Trình giám sát dữ liệu/ Data monitor trong phần mềm XDPpro để biết phạm vi thanh ghi FS, phạm vi phổ biến là FS0~FS47). Giá trị FS có thể được sửa đổi nhưng không thể đọc được thông qua lệnh Modbus. Thanh ghi FS không thể dùng với lệnh so sánh giá trị với các thanh ghi khác, chỉ chứa dữ liệu hằng số dùng khi lập trình trong phần mềm XDPpro. Giá trị không thể được đọc. FS được sử dụng để bảo vệ bản quyền của người dùng. Thanh ghi D, HD... có thể thay thế bằng FS.

Các nhận xét chương trình

Khi chương trình người dùng quá dài thì cần có những nhận xét về chương trình và các phần mềm để dễ dàng thay đổi chương trình sau này.

Chức năng con trỏ địa chỉ

Thêm phần con trỏ bù (thanh ghi giá trị) (như X3[D100], M10[D100], D0[D100]) sau cuộn dây, thanh ghi dữ liệu có thể tạo địa chỉ gián tiếp. Ví dụ: khi D100=9,
 $X3[D100]=X[3+9]=X14$; $M10[D100]=M19$, $D0[D100]=D9$

Chức năng đa dạng

PLC dòng XD/XL có đủ các lệnh cơ bản bao gồm điều khiển tuần tự cơ bản, di chuyển và so sánh dữ liệu, phép tính số học, điều khiển logic, vòng lặp dữ liệu và truyền dữ liệu, v.v.

PLC dòng XD/XL cũng hỗ trợ ngắt, bộ đếm xung tốc độ cao, kiểm tra tần số, thời gian chính xác, điều khiển PID, v.v.

Khởi hàm trong ngôn ngữ lập trình C

PLC dòng XD/XL hỗ trợ ngôn ngữ lập trình C; người dùng có thể gọi ngôn ngữ C trong ngôn ngữ sơ đồ bậc thang. Chức năng này cải thiện hiệu quả lập trình.

Dừng PLC khi khởi động lại

PLC dòng XD/XL hỗ trợ chức năng “Dừng PLC khi khởi động lại”. Khi có sự cố nghiêm trọng trong quá trình chạy PLC, chức năng này có thể dừng tất cả đầu ra ngay lập tức. Ngoài ra, nếu thông số cổng COM bị thay đổi do nhầm lẫn, chức năng này có thể giúp PLC kết nối với PC.

Chức năng giao tiếp

PLC dòng XD/XL có nhiều chế độ giao tiếp, chẳng hạn như Modbus-RTU, Modbus-ASCII. Khi các tham số cổng COM được thay đổi, các tham số mới sẽ có hiệu lực ngay lập tức mà không cần khởi động lại PLC.

Thời gian chờ có thể được thêm vào trước các lệnh Modbus.

1-2 Ngôn ngữ lập trình

1-2-1 Loại ngôn ngữ lập trình

PLC dòng XD/XL hỗ trợ hai loại ngôn ngữ lập trình:

Mã lệnh

Tạo chương trình với các lệnh trực tiếp, chẳng hạn như “LD”, “AND”, “OUT”, v.v. Đây là dạng đầu vào cơ bản của chương trình, nhưng rất khó đọc và khó hiểu;

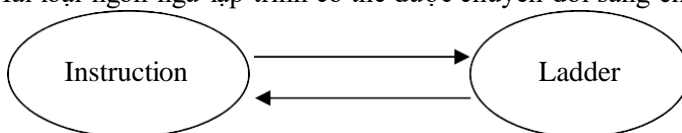
Ví dụ:	Bước	Lệnh	Toán hạng
	0	LD	X000
	1	OR	Y005
	2	ANI	X002
	3	OUT	Y005

Sơ đồ bậc thang

Tạo đồ thị điều khiển tuần tự với tín hiệu điều khiển tuần tự và các phần tử lập trình. Phương thức này được gọi là “Sơ đồ bậc thang”. Phương thức này sử dụng cuộn dây và công tắc tơ để biểu diễn mạch tuần tự. Sơ đồ bậc thang rất dễ hiểu và có thể được sử dụng để theo dõi trạng thái PLC trực tuyến.

1-2-2 Chuyển đổi ngôn ngữ

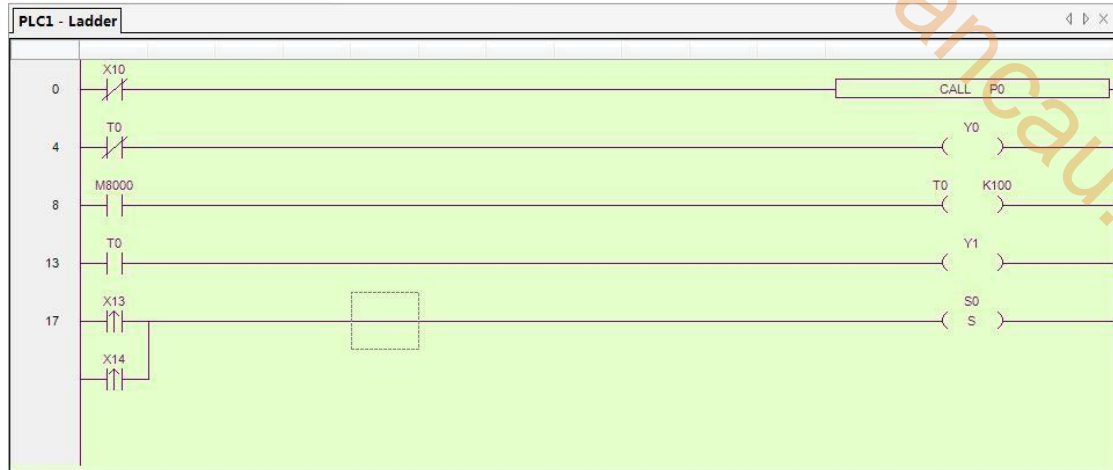
Hai loại ngôn ngữ lập trình có thể được chuyển đổi sang cho nhau.



1-3 Chế độ lập trình

Nhập lệnh trực tiếp

Hai loại ngôn ngữ lập trình có thể được nhập trực tiếp vào cửa sổ chỉnh sửa. Cửa sổ sơ đồ bậc thang có chức năng gợi ý lệnh giúp cải thiện hiệu quả lập trình đáng kể.



Instruction Configuration/ Cấu hình lệnh

Một số lệnh sử dụng phức tạp, như đầu ra xung, PID, v.v. Phần mềm XDPPro có cửa sổ cấu hình cho các lệnh đặc biệt này. Người dùng chỉ cần nhập thông số vào cửa sổ cấu hình mà không cần nhớ những lệnh phức tạp. Cửa sổ sau đây là hộp thoại cấu hình cho chân đầu ra phát xung.

The 'multi section pulse output' dialog box is shown with the following configuration:

- Data start address: D0
- user params address: D100
- System params: K1
- Output: Y0
- Mode: relative
- Start execute section count: 0

The dialog also features a table for configuring pulse sections:

frequency	pulse count	wait condition	wait register	jump register

Buttons at the bottom include: Read From PLC, Write To PLC, OK, and Cancel.

Để biết chi tiết về cấu hình lệnh, vui lòng tham khảo hướng dẫn sử dụng PLC dòng XD/XL 【 phần mềm】.

Chương 2: Chức năng phần tử lập trình

Trong chương 1, chúng tôi giới thiệu ngắn gọn về ngôn ngữ lập trình. Tuy nhiên, yếu tố quan trọng nhất trong một chương trình là các toán hạng. Những phần tử này bao gồm các rơle và các thanh ghi. Trong chương này, chúng tôi sẽ mô tả các chức năng và cách sử dụng các rơle và thanh ghi này.

2-1 Tóm tắt các phần tử lập trình

Có nhiều rơle, bộ hẹn giờ và bộ đếm bên trong PLC. Tất cả chúng đều có vô số công tắc tơ (contactor) NO (Thường Mở) và NC (Thường Đóng). Nối các contactor này với các cuộn dây sẽ tạo thành mạch điều khiển tuần tự. Tiếp theo chúng tôi sẽ giới thiệu các phần tử lập trình sau đây.

Role đầu vào/Input Relay (X)

- Chức năng của các rơle đầu vào

Role đầu vào được sử dụng để nhận tín hiệu ON/OFF từ bên ngoài, ký hiệu là **X**.

- Nguyên tắc gán địa chỉ
 - Trong mỗi thiết bị cơ bản, địa chỉ X có dạng bát phân, chẳng hạn như X0~X7, X10~X17...
 - Địa chỉ của mô-đun mở rộng: mô-đun 1 bắt đầu từ X10000, mô-đun 2 bắt đầu từ X10100...XD1/XD2/XL1 không thể hỗ trợ mô-đun mở rộng. Có thể kết nối tối đa 10 mô-đun mở rộng với thiết bị chính XD3/XL3.
 - XD5/XDM/XDC/XD5E/XDME/XDH/XL5/XL5E/XLME có thể kết nối 16 mô-đun mở rộng.
 - Bảng BD mở rộng: BD 1 bắt đầu từ X20000; PLC 24-32 điểm có thể kết nối một bảng BD mở rộng và PLC 48-60 điểm có thể kết nối hai bảng BD mở rộng. (PLC 16 điểm không hỗ trợ bảng BD mở rộng, dòng XL/XDH không hỗ trợ bảng BD mở rộng.)
 - PLC supports a left extended I/O ED module. (XDH cannot support ED module)
 - Số địa chỉ của mô-đun ED mở rộng bên trái, bắt đầu từ X30000 theo hệ bát phân, PLC dòng XD/XL hỗ trợ mô-đun I/O ED mở rộng bên trái. (XDH không thể hỗ trợ mô-đun ED)
- Lưu ý khi sử dụng

Bộ lọc kỹ thuật số được sử dụng trong bộ lọc đầu vào của rơle đầu vào. Người dùng có thể thay đổi các tham số bộ lọc bằng cách đặt thanh ghi đặc biệt SFD0, giá trị mặc định là 10ms, phạm vi sửa đổi: 0~ 1000 mili giây.

Có đủ rơle đầu vào trong PLC. Rơle đầu vào có địa chỉ lớn hơn điểm đầu vào có thể được coi là rơle trung gian.

Role đầu ra/Output Relay (Y)

- Chức năng của các rơle đầu ra

Role đầu ra là giao diện để điều khiển các tải bên ngoài, ký hiệu là **Y**;

- Nguyên tắc gán địa chỉ

Trong mỗi thiết bị cơ bản, địa chỉ Y có dạng bát phân, chẳng hạn như Y0~Y7, Y10~Y17...

Địa chỉ mô-đun mở rộng: mô-đun 1 bắt đầu từ Y10000, mô-đun 2 bắt đầu từ Y10100...

XD1/XD2/XL1 không hỗ trợ các mô-đun mở rộng, XD3/XL3 có thể kết nối 10 mô-đun mở rộng, XD5/XDM/XDC/XD5E/XDME/XDH/XL5/XL5E/XLME có thể kết nối 16 mô-đun mở rộng. Mở rộng số địa chỉ của bảng BD, bắt đầu từ X20000 theo hệ bát phân, PLC 24-32 điểm có thể mở rộng một bảng BD, 48-60 điểm PLC có thể mở rộng hai bảng BD. (PLC 16 điểm không hỗ trợ bảng BD mở rộng, dòng XL/XDH không hỗ trợ bảng BD mở rộng.)

Số địa chỉ của mô-đun ED mở rộng bên trái, bắt đầu từ Y30000 theo hệ bát phân, PLC dòng XD/XL hỗ trợ mô-đun ED đầu vào và đầu ra mở rộng bên trái. (XDH không thể hỗ trợ mô-đun ED)

Lưu ý khi sử dụng

Có đủ role đầu ra trong PLC. Role đầu ra có địa chỉ lớn hơn điểm đầu ra có thể được coi là role trung gian.

Role trung gian/Auxiliary Relays (M, HM)

- Chức năng của các Role trung gian

Role trung gian là role bên trong của PLC, ký hiệu là M và HM;

- Nguyên tắc gán địa chỉ

Trong thiết bị cơ bản, gán địa chỉ role trung gian ở dạng thập phân

- Lưu ý khi sử dụng

Loại role này khác với role đầu vào/đầu ra, chúng không thể điều khiển tải ngoài và nhận tín hiệu bên ngoài mà chỉ được sử dụng trong chương trình;

Role trung gian khả nhớ có thể giữ trạng thái BẬT/TẮT (ON/OFF) khi TẮT nguồn PLC;

Role trạng thái /Status Relays (S, HS)

- Chức năng của role trạng thái

Dùng làm role trong Sơ đồ bậc thang, ký hiệu là S, HS.

- Nguyên tắc gán địa chỉ

Trong các thiết bị cơ bản, gán địa chỉ ở dạng thập phân.

- Lưu ý khi sử dụng

Nếu nó không được sử dụng làm số bước trạng thái vận hành của chương trình, chúng có thể được sử dụng làm role phụ, lập trình như công tắc tơ/cuộn dây thông thường. Ngoài ra, chúng có thể được sử dụng làm tín hiệu cảnh báo, chẩn đoán bên ngoài.

Bộ định thời (T, HT)

- Chức năng của bộ định thời

Bộ định thời được sử dụng để tích lũy xung thời gian như 1ms, 10ms, 100ms, v.v. khi đạt đến giá trị cài đặt, các contactor đầu ra sẽ hoạt động, ký hiệu là T và HT.

- Nguyên tắc gán địa chỉ

Trong các thiết bị cơ bản, gán địa chỉ bộ định thời ở dạng thập phân. Vui lòng tham khảo chương 2-2 để biết chi tiết.

- Xung thời gian

Có ba độ phân giải của bộ định thời: 1ms, 10ms và 100ms. Ví dụ: 10ms có nghĩa là tích lũy xung 10ms.

- Tích lũy/không tích lũy

Bộ định thời có hai chế độ: bộ định thời tích lũy có nghĩa là ngay cả khi cuộn dây dẫn động của bộ định thời ở trạng thái OFF, bộ hẹn giờ vẫn giữ giá trị hiện tại; trong khi bộ định thời không tích lũy có nghĩa là khi giá trị tích lũy đạt đến giá trị cài đặt, đầu ra sẽ hoạt động, giá trị tích lũy được đặt lại về 0.

Bộ đếm (C, HC)

Tùy theo mục đích ứng dụng khác nhau, bộ đếm có nhiều loại khác nhau:

- Đếm đếm nội bộ (cho mục đích sử dụng thông thường/lưu giữ trạng thái khi tắt nguồn)

Bộ đếm 16 bit: đối mạch đếm lên, phạm vi đếm là 1 ~ 32.767

Bộ đếm 32 bit: đối với mạch đếm lên, phạm vi đếm là 1 ~ 2.147.483.647

Các bộ đếm này dành cho tín hiệu bên trong PLC. Tốc độ phản hồi là một chu kỳ quét hoặc lâu hơn.

- Đếm tốc độ cao (có lưu giữ khi tắt nguồn)

Bộ đếm 32 bit: phạm vi đếm là -2,147,483,648 ~ +2,147,483,647

(Mạch đếm lên pha đơn, mạch đếm pha AB). Đối với các thiết bị đầu cuối đầu vào đặc biệt.

Bộ đếm tốc độ cao sẽ không bị ảnh hưởng bởi thời gian quét PLC. Đối với chế độ đếm lên, nó có thể đếm xung tối đa 80KHz; đối với chế độ pha AB, nó có thể đếm xung tối đa 50KHz.

- Nguyên tắc gán địa chỉ

Trong các thiết bị cơ bản, gán địa chỉ bộ đếm ở dạng thập phân.

Thanh ghi dữ liệu (D, HD)

- Chức năng của thanh ghi dữ liệu

Thanh ghi dữ liệu dùng để lưu trữ dữ liệu, ký hiệu là D và HD.

- Nguyên tắc gán địa chỉ

Các thanh ghi dữ liệu trong PLC dòng XD/XL là 16 bit (bit cao nhất là bit dấu), kết hợp hai thanh ghi dữ liệu lại với nhau để xử lý dữ liệu 32 bit (bit cao nhất là bit dấu).

- Lưu ý khi sử dụng

Tương tự như các phần tử lập trình khác, thanh ghi dữ liệu cũng có loại thông dụng và loại lưu giữ khi tắt nguồn.

Thanh ghi FlashROM (FD)

- Chức năng của thanh ghi FlashROM

Các thanh ghi FlashROM dùng để lưu trữ dữ liệu, ký hiệu là FD.

- Nguyên tắc gán địa chỉ

Trong các thiết bị cơ bản, địa chỉ thanh ghi FlashROM ở dạng thập phân;

- Lưu ý khi sử dụng

Ngay cả khi tắt nguồn pin, vùng này vẫn có thể ghi nhớ dữ liệu. Vì vậy vùng này có thể lưu trữ các thông số quan trọng. FlashROM có thể được ghi khoảng 1.000.000 lần (times) và cần có thời gian khi ghi. Việc ghi thường xuyên có thể gây tổn hại vĩnh viễn cho FD.

Thanh ghi Bảo mật đặc biệt (FS)

- Chức năng của Secret Register

Một phần của thanh ghi FlashROM được sử dụng để lưu trữ dữ liệu trong các phần tử lập trình, được biểu thị bằng ký hiệu FS. Các giá trị trong thanh ghi FS có thể được ghi nhưng không thể đọc được, vì vậy chúng có thể được sử dụng để bảo vệ quyền sở hữu trí tuệ của người dùng.

- Nguyên tắc gán địa chỉ

Trong thiết bị cơ bản, các thanh ghi FS được gán địa chỉ bằng số thập phân.

- Vì số lượng thanh ghi FS của các loại PLC khác nhau có thể khác nhau, vui lòng tham khảo "Cài đặt gốc cho PLC/ PLC initial settings" được hiển thị trong phần mềm PLC trực tuyến, thông thường là FS0-FS47.

- Lưu ý khi sử dụng

Vùng lưu trữ có thể ghi nhớ dữ liệu ngay cả khi pin bị tắt, do đó nó có thể được sử dụng để lưu trữ các thông số quy trình quan trọng. FS có thể được ghi khoảng 1.000.000 lần và mỗi lần phải mất nhiều thời gian hơn để viết. Việc ghi thường xuyên sẽ gây ra tổn hại vĩnh viễn cho FS, vì vậy người dùng không nên ghi thường xuyên. Khi sử dụng lệnh MOV để truyền dữ liệu tới FS, sườn lên là hợp lệ.

Giá trị của phần tử lập trình có thể được đặt tùy ý trong thanh ghi FS, nhưng không thể đọc được giá trị của thanh ghi (luôn trả về 0); và nó không thể so sánh với thanh ghi trong phần mềm PLC, chỉ so sánh với hằng số nên không thể đọc được giá trị thực của thanh ghi.

Hằng số (B) (K) (H)

B có nghĩa là Hệ nhị phân, K đại diện cho Hệ thập phân, H đại diện cho Hệ thập lục phân (HEX). Chúng được sử dụng để đặt giá trị bộ định thời và bộ đếm hoặc toán hạng của các lệnh ứng dụng. Ví dụ hex FF sẽ là HFF.

2-2 Cấu trúc của Phần tử lập trình

2-2-1 Cấu trúc của Bộ nhớ

Trong PLC dòng XD/XL, có nhiều thanh ghi. Ngoài các thanh ghi D, HD, FlashROM, chúng ta còn có thể kết hợp bit vào thanh ghi.

Thanh ghi dữ liệu D, HD,FD

Cho mục đích sử dụng thông thường, 16 bit

Cho mục đích sử dụng thông thường, 32 bit (kết hợp hai thanh ghi 16 bit liên tục)

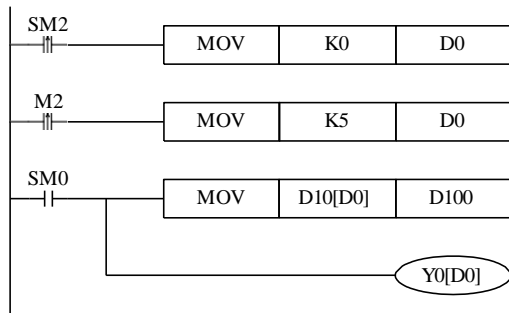
Cho mục đích sử dụng thông thường, 64 bit (kết hợp hai thanh ghi 32 bit, nhưng địa chỉ phải liên tiếp).

Cho mục đích lưu giữ khi tắt nguồn, không thể sửa đổi phạm vi lưu giữ

Cho mục đích đặc biệt, bị hệ thống chiếm dụng, không thể sử dụng cho các tham số lệnh thông thường

Cho mục đích offset (gán gián tiếp)

Hình thức: Dn[Dm], HDn[Dm], Xn[Dm], Yn[Dm], Mn[Dm], v.v.



Khi D0=0, D100=D10, Y0 ở trạng thái ON.

Khi M2 chuyển từ OFF sang ON, D0=5, thì D100=D15, Y5 ở trạng thái ON. Trong đó, D10[D0]=D[10+D0], Y0[D0]=Y[0+D0].

Địa chỉ thanh ghi từ word được kết hợp theo bit: DXn[Dm] đại diện cho DX[n+Dm].

Các phần tử lập trình địa chỉ thanh ghi từ, địa chỉ thanh ghi từ có thể biểu diễn bằng phần tử thanh ghi D, HD.

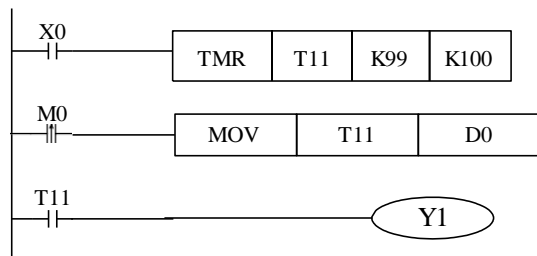
Bộ định thời T, HT/Bộ đếm C, HC

Đối với mục đích sử dụng thông thường, 16 bit, biểu thị giá trị hiện tại của bộ định thời/bộ đếm;

Đối với mục đích sử dụng phổ biến, 32 bit, (kết hợp hai thanh ghi 16 bit liên tục)

Để biểu diễn chúng, chỉ cần sử dụng phương thức chữ cái+địa chỉ, chẳng hạn như T10, C11, HT10, HC11.

Ví dụ:



Trong ví dụ trên, MOV T11 D0, T11 đại diện cho thanh ghi word;

LD T11, T11 đại diện cho thanh ghi bit.

FlashROM Register FD

Để lưu trữ khi tắt nguồn, 16 bit

Để lưu trữ khi tắt nguồn, 32 bit, (kết hợp hai thanh ghi 16 bit liên tục)

Đối với mục đích sử dụng đặc biệt, bị hệ thống chiếm dụng, không thể được sử dụng làm tham số lệnh thông thường

Thanh ghi kết hợp bởi các bit

Đối với mục đích sử dụng thông thường, 16 bit, (kết hợp 16 bit)

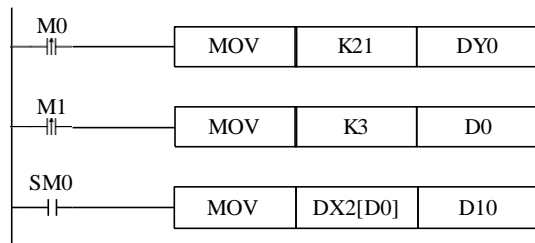
Các phần tử lập trình có thể kết hợp với các word là: X, Y, M, S, T, C, HM, HS, HT, HC.

Định dạng: thêm "D" vào trước các phần tử lập trình, như DM10, đại diện cho thanh ghi 16 bit từ M10~M25 .

Nhận 16 bit bắt đầu từ DXn, không thể vượt quá phạm vi phần tử lập trình;

Word được kết hợp bởi các bit không thể thực hiện việc gán địa chỉ bit;

Ví dụ:



Khi M0 thay đổi từ OFF sang ON, giá trị trong word được kết hợp bởi Y0~Y17 bằng 21, tức là Y0, Y2, Y4 chuyển sang ON.

Trước khi M1 kích hoạt, nếu D0=0, DX2[D0] đại diện cho một word được kết hợp bởi X2~X21. Nếu M1 thay đổi từ OFF thành ON, D0=3 thì DX2[D0] biểu thị một word được kết hợp bởi X5~X24.

2-2-2 Cấu trúc của các phần tử lập trình dạng Bit

Các phần tử lập trình bit bao gồm X, Y, M, S, T, C, HM, HS, HT, HC. Ngoài ra, bit của thanh ghi cũng có thể được sử dụng làm phần tử lập trình dạng bit nhị phân.

Relay

Role đầu vào X, dạng bát phân

Role đầu vào Y, dạng bát phân

Role trung gian M, HM, S, HS; dạng thập phân

Role phụ T, HT, C, HC dạng thập phân. Phương thức biểu diễn giống với các thanh ghi, vì vậy chúng ta cần đánh giá xem đó là thanh ghi word hay thanh ghi bit theo hướng dẫn.

Bit của thanh ghi

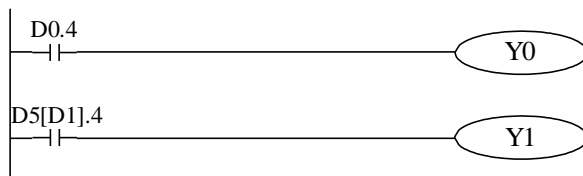
Được tạo bởi bit của thanh ghi, hỗ trợ thanh ghi D

Phương thức biểu diễn: Dn.m (0<m<15): ví dụ D10.2 nghĩa là bit thứ 2 của D10.

Phương thức biểu diễn bit có địa chỉ chứa thanh ghi con trở: Dn[Dm].x

Bit của thanh ghi không thể tạo thành phần tử lập trình dạng word;

Ví dụ:



D0.4 có nghĩa là khi bit thứ tư của D0 là 1, đặt Y0 sang trạng thái ON.

D5[D1].4 nghĩa là địa chỉ bit có offset, nếu D1=5 thì D5[D1] nghĩa là bit thứ tư của D10

2-3 Danh sách phần tử lập trình

2-3-1 Danh sách phần tử lập trình

Danh sách các phần tử lập trình PLC dòng **XD1**:

	Tên	Phạm vi				Điểm			
		10 I/O	16 I/O	24 I/O	32 I/O	10	16	24	32
X	Input points	X0~X4	X0~X7	X0~X13	X0~X17	5	8	12	16
Y	Output points	Y0~Y4	Y0~Y7	Y0~Y13	Y0~Y17	5	8	12	16
M	Internal relay	M0~M7999				8000			
HM		HM0~HM959* ¹				960			
SM		Special purpose SM0~SM2047* ²				2048			
S	Flow	S0~S1023				1024			
HS		HS0~HS127* ¹				128			
T	Timer	T0~T575				576			
HT		HT0~HT95* ¹				96			
ET		Precise timer ET0~ET31				32			
C	Counter	C0~C575				576			
HC		HC0~HC95* ¹				96			
HSC		High speed counter HSC0~HSC31				32			
D	Data register	D0~D7999				8000			
HD		HD0~HD999* ¹				1000			
SD		Special purpose SD0~SD2047				2048			
HSD		Special purpose HSD0~HSD499* ²				500			
FD	FlashROM register	FD0~FD5119				5120			
SFD		Special purpose SFD0~SFD1999* ²				2000			
FS	Special secret register	FS0~FS47				48			
ID* ⁶	Main body	ID0~ID99				100			
QD* ⁷	Main body	QD0~QD99				100			
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31				32			

Danh sách các phần tử lập trình PLC dòng **XD2**:

	Tên	Phạm vi						Điểm					
		16 I/O	24 I/O	32 I/O	42 I/O	48 I/O	60 I/O	16	24	32	42	48	60
X	Input points	X0~X7	X0~X15	X0~X21	X0~X27	X0~X33	X0~X43	8	14	18	24	28	36
Y	Output points	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y21	Y0~Y23	Y0~Y27	8	10	14	18	20	24
X	Input points* ⁴	X20000~X20077(#1 expansionBD) X20100~X20177(#2 expansionBD)						128					
Y	Output points* ⁴	Y20000~Y20077(#1 expansionBD) Y20100~Y20177(#2 expansionBD)						128					
X	Input points* ⁵	X30000~X30077(#1 expansionED)						64					

Y	Output points ^{*5}	Y30000~Y30077(#1 expansionED)	64
M	Internal relay	M0~M7999	8000
HM		HM0~HM959 ^{*1}	960
SM		Special purposeSM0~SM2047 ^{*2}	2048
S	Flow	S0~S1023	1024
HS		HS0~HS127 ^{*1}	128
T	Timer	T0~T575	576
HT		HT0~HT95 ^{*1}	96
ET		Precise timer ET0~ET31	32
C	Counter	C0~C575	576
HC		HC0~HC95 ^{*1}	96
HSC		High speed counter HSC0~HSC31	32
D	Data register	D0~D7999	8000
HD		HD0~HD999 ^{*1}	1000
SD		Special purposeSD0~SD2047	2048
HSD		Special purposeHSD0~HSD499 ^{*2}	500
FD	FlashROM register	FD0~FD5119	5120
SFD		Special purposeSFD0~SFD1999 ^{*2}	2000
FS	Special secret register	FS0~FS47	48
ID ^{*6}	Main body	ID0~ID99	100
	expansion BD	ID20000~ID20099(#1 expansionBD)	200
		ID20100~ID20199(#2 expansionBD)	
	expansion ED	ID30000~ID30099(#1 expansionED)	100
QD ^{*7}	Main body	QD0~QD99	100
	expansion BD	QD20000~QD20099(#1 expansionBD)	200
		QD20100~QD20199(#2 expansionBD)	
	expansion ED	QD30000~QD30099(#1 expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XD3:

	Tên	Phạm vi					Điểm				
		16 I/O	24 I/O	32 I/O	48 I/O	60 I/O	16	24	32	48	60
X	Input points	X0~X7	X0~X15	X0~X21	X0~X33	X0~X43	8	14	18	28	36
Y	Output points	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	8	10	14	20	24
X	Input points ^{*3}	X10000~X10077(#1 expansion module) X11100~X11177(#10 expansion module)					640				
Y	Output points ^{*3}	Y10000~Y10077(#1 expansion module) Y11100~Y11177(#10 expansion module)					640				
X	Input points ^{*4}	X20000~X20077(#1 expansionBD) X20100~X20177(#2 expansionBD)					128				
Y	Output points ^{*4}	Y20000~Y20077(#1 expansionBD) Y20100~Y20177(#2 expansionBD)					128				

X	Input points ^{※5}	X30000~X30077(#1expansionED)	64
Y	Output points ^{※5}	Y30000~Y30077(#1expansionED)	64
M	Internal relay	M0~M7999	8000
HM		HM0~HM959 ^{※1}	960
SM		special purpose SM0~SM2047 ^{※2}	2048
S	Flow	S0~S1023	1024
HS		HS0~HS127 ^{※1}	128
T	Timer	T0~T575	576
HT		HT0~HT95 ^{※1}	96
ET		precise timer ET0~ET31	32
C	Counter	C0~C575	576
HC		HC0~HC95 ^{※1}	96
HSC		High speed counter HSC0~HSC31	32
D	Data register	D0~D7999	8000
HD		HD0~HD999 ^{※1}	1000
SD		special purpose SD0~SD2047	2048
HSD		special purpose HSD0~HSD499 ^{※2}	500
FD	FlashROM register	FD0~FD5119	5120
SFD		special purpose SFD0~SFD1999 ^{※2}	2000
FS	Special secret register	FS0~FS47	48
ID ^{※6}	Main body	ID0~ID99	100
	Expansion module	ID10000~ID10099(#1 expansion module) ID10900~ID10999(#10 expansion module)	1000
		expansion BD	ID20000~ID20099(#1expansionBD) ID20100~ID20199(#2expansionBD)
	expansion ED	ID30000~ID30099(#1expansionED)	100
QD ^{※7}	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD10900~QD10999(#10 expansion module)	1000
		expansion BD	QD20000~QD20099(#1expansionBD) QD20100~QD20199(#2expansionBD)
	expansion ED	QD30000~QD30099(#1expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XD5:

	Tên	Phạm vi					Điểm				
		16 I/O	24 I/O	32 I/O	48 I/O	60 I/O	16	24	32	48	60
X	Input points	X0~X7	X0~X15	X0~X21	X0~X33	X0~X43	8	14	18	28	36
Y	Output points	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	8	10	14	20	24
X	Input points* ³	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)					1024				
Y	Output points* ³	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)					1024				
X	Input points* ⁴	X20000~X20077(#1 expansion BD) X20100~X20177(#2 expansion BD)					128				
Y	Output points* ⁴	Y20000~Y20077(#1 expansion BD) Y20100~Y20177(#2 expansion BD)					128				
X	Input points* ⁵	X30000~X30077(#1 expansion ED)					64				
Y	Output points* ⁵	Y30000~Y30077(#1 expansion ED)					64				
M	Internal relay	M0~M69999					70000				
HM		HM0~HM11999* ¹					12000				
SM		special purpose SM0~SM4999* ²					5000				
S	Flow	S0~S7999					8000				
HS		HS0~HS999* ¹					1000				
T	Timer	T0~T4999					5000				
HT		HT0~HT1999* ¹					2000				
ET		precise timer ET0~ET39					40				
C	Counter	C0~C4999					5000				
HC		HC0~HC1999* ¹					2000				
HSC		high speed counter HSC0~HSC39					40				
D	Data register	D0~D69999(firmware V3.5.3 and up)					70000				
HD		D0~D59999(firmware V3.5.2 and down)					60000				
SD		HD0~HD24999* ¹					25000				
HSD		special purpose SD0~SD4999					5000				
FD	FlashROM	FD0~FD8191					8192				
SFD	Register	special purpose SFD0~SFD5999* ²					6000				
FS	Special secret register	FS0~FS47					48				
ID* ⁶	Main body	ID0~ID99					100				
	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)					1600				
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)					200				
	expansion ED	ID30000~ID30099(#1 expansion ED)					100				
QD* ⁷	Main body	QD0~QD99					100				
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)					1600				

	expansion BD	QD20000~QD20099(#1expansionBD) QD20100~QD20199(#2expansionBD)	200
	expansion ED	QD30000~QD30099(#1expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XDM:

	Tên	Phạm vi			Điểm		
		24 I/O	32 I/O	60 I/O	24	32	60
X	Input points	X0~X15	X0~X21	X0~X43	14	18	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y27	10	14	24
X	Input points ^{*3}	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)			1024		
Y	Output points ^{*3}	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)			1024		
X	Input points ^{*4}	X20000~X20077(#1expansionBD) X20100~X20177(#2expansionBD)			128		
Y	Output points ^{*4}	Y20000~Y20077(#1expansionBD) Y20100~Y20177(#2expansionBD)			128		
X	Input points ^{*5}	X30000~X30077(#1expansionED)			64		
Y	Output points ^{*5}	Y30000~Y30077(#1expansionED)			64		
M	Internal relay	M0~M69999			70000		
HM		HM0~HM11999 ^{*1}			12000		
SM		special purpose SM0~SM4999 ^{*2}			5000		
S	Flow	S0~S7999			8000		
HS		HS0~HS999 ^{*1}			1000		
T	Timer	T0~T4999			5000		
HT		HT0~HT1999 ^{*1}			2000		
ET		precise timer ET0~ET39			40		
C	Counter	C0~C4999			5000		
HC		HC0~HC1999 ^{*1}			2000		
HSC		High speed counter HSC0~HSC39			40		
D	Data register	D0~D69999			70000		
HD		HD0~HD24999 ^{*1}			25000		
SD		special purpose SD0~SD4999			5000		
HSD		special purpose HSD0~HSD1023 ^{*2}			1024		
FD	FlashROM register	FD0~FD8191			8192		
SFD		special purpose SFD0~SFD5999 ^{*2}			6000		
FS	Special secret register	FS0~FS47			48		
ID ^{*6}	Main body	ID0~ID99			100		

	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)	1600
	expansion BD	ID20000~ID20099(#1expansionBD) ID20100~ID20199(#2expansionBD)	200
	expansion ED	ID30000~ID30099(#1expansionED)	100
QD*7	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1expansionBD) QD20100~QD20199(#2expansionBD)	200
	expansion ED	QD30000~QD30099(#1expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XDC:

	Tên	Phạm vi				Điểm			
		24 I/O	32 I/O	48 I/O	60 I/O	24	32	48	60
X	Input points	X0~X15	X0~X21	X0~X33	X0~X43	14	18	28	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	10	14	20	24
X	Input points*3	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)				1024			
Y	Output points*3	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)				1024			
X	Input points*4	X20000~X20077(#1expansionBD) X20100~X20177(#2expansionBD)				128			
Y	Output points*4	Y20000~Y20077(#1expansionBD) Y20100~Y20177(#2expansionBD)				128			
X	Input points*5	X30000~X30077(#1expansionED)				64			
Y	Output points*5	Y30000~Y30077(#1expansionED)				64			
M	Internal relay	M0~M69999				70000			
HM		HM0~HM11999*1				12000			
SM		special purpose SM0~SM4999*2				5000			
S	Flow	S0~S7999				8000			
HS		HS0~HS999*1				1000			
T	Timer	T0~T4999				5000			
HT		HT0~HT1999*1				2000			
ET		Precise timer ET0~ET39				40			
C	Counter	C0~C4999				5000			
HC		HC0~HC1999*1				2000			
HSC		High speed counter HSC0~HSC39				40			
D	Data register	D0~D69999				70000			

HD		HD0~HD24999*1	25000
SD		Special purpose SD0~SD4999	5000
HSD		Special purpose HSD0~HSD1023*2	1024
FD	FlashROM	FD0~FD8191	8192
SFD	register	Special purpose SFD0~SFD5999*2	6000
FS	Special secret register	FS0~FS47	48
ID*6	Main body	ID0~ID99	100
	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)	1600
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)	200
	expansion ED	ID30000~ID30099(#1 expansion ED)	100
QD*7	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1 expansion BD) QD20100~QD20199(#2 expansion BD)	200
	expansion ED	QD30000~QD30099(#1 expansion ED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng **XD5E**:

	Tên	Phạm vi				Điểm			
		24 I/O	30 I/O	48 I/O	60 I/O	24	30	48	60
X	Input points	X0~X15	X0~X17	X0~X33	X0~X43	14	16	28	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	10	14	20	24
X	Input points*3	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)				1024			
Y	Output points*3	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)				1024			
X	Input points*4	X20000~X20077(#1 expansion BD) X20100~X20177(#2 expansion BD)				128			
Y	Output points*4	Y20000~Y20077(#1 expansion BD) Y20100~Y20177(#2 expansion BD)				128			
X	Input points*5	X30000~X30077(#1 expansion ED)				64			

	Tên	Phạm vi				Điểm			
		24 I/O	30 I/O	48 I/O	60 I/O	24	30	48	60
Y	Output points ^{*5}	Y30000~Y30077(#1 expansion ED)				64			
M	Internal relay	M0~M69999				70000			
HM		HM0~HM11999 ^{*1}				12000			
SM		special purpose SM0~SM4999 ^{*2}				5000			
S	Flow	S0~S7999				8000			
HS		HS0~HS999 ^{*1}				1000			
T	Timer	T0~T4999				5000			
HT		HT0~HT1999 ^{*1}				2000			
ET		precise timer ET0~ET39				40			
C	Counter	C0~C4999				5000			
HC		HC0~HC1999 ^{*1}				2000			
HSC		high speed counter HSC0~HSC39				40			
D	Data register	D0~D69999				70000			
HD		HD0~HD24999 ^{*1}				25000			
SD		special purpose SD0~SD4999				5000			
HSD		special purpose HSD0~HSD1023 ^{*2}				1024			
FD	FlashROM register	FD0~FD8191				8192			
SFD		special purpose SFD0~SFD5999 ^{*2}				6000			
FS	Special secret register	FS0~FS47				48			
ID ^{*6}	Main body	ID0~ID99				100			
	Expansion module	ID10000~ID10099(#1 expansion module)				1600			
		ID11500~ID11599(#16 expansion module)							
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)				200			
expansion ED	ID30000~ID30099(#1 expansion ED)				100				
QD ^{*7}	Main body	QD0~QD99				100			
	Expansion module	QD10000~QD10099(#1 expansion module)				1600			
		QD11500~QD11599(#16 expansion module)							
	Expansion BD	QD20000~QD20099(#1 expansion BD) QD20100~QD20199(#2 expansion BD)				200			
Expansion ED		QD30000~QD30099(#1 expansion ED)				100			

	Name	Range				Points			
		24 I/O	30 I/O	48 I/O	60 I/O	24	30	48	60
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31				32			

Danh sách các phần tử lập trình PLC dòng XDME:

	Tên	Phạm vi		Điểm	
		30 I/O	60 I/O	30	60
X	Input points	X0~X17	X0~X43	16	36
Y	Output points	Y0~Y15	Y0~Y27	14	24
X	Input points ^{*3}	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)		1024	
Y	Output points ^{*3}	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)		1024	
X	Input points ^{*4}	X20000~X20077(#1 expansionBD) X20100~X20177(#2 expansionBD)		128	
Y	Output points ^{*4}	Y20000~Y20077(#1 expansionBD) Y20100~Y20177(#2 expansionBD)		128	
X	Input points ^{*5}	X30000~X30077(#1 expansionED)		64	
Y	Output points ^{*5}	Y30000~Y30077(#1 expansionED)		64	
M	Internal relay	M0~M69999		70000	
HM		HM0~HM11999 ^{*1}		12000	
SM		special purpose SM0~SM4999 ^{*2}		5000	
S	Flow	S0~S7999		8000	
HS		HS0~HS999 ^{*1}		1000	
T	Timer	T0~T4999		5000	
HT		HT0~HT1999 ^{*1}		2000	
ET		precise timer ET0~ET39		40	
C	Counter	C0~C4999		5000	
HC		HC0~HC1999 ^{*1}		2000	
HSC		high speed counter HSC0~HSC39		40	
D	Data register	D0~D69999		70000	
HD		HD0~HD24999 ^{*1}		25000	
SD		special purpose SD0~SD4999		5000	
HSD		special purpose HSD0~HSD1023 ^{*2}		1024	
FD	FlashROM register	FD0~FD8191		8192	
SFD		special purpose SFD0~SFD5999 ^{*2}		6000	
FS	Special secret register	FS0~FS47		48	
ID ^{*6}	Main body	ID0~ID99		100	

	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)	1600
	expansion BD	ID20000~ID20099(#1expansionBD) ID20100~ID20199(#2expansionBD)	200
	expansion ED	ID30000~ID30099(#1expansionED)	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1expansionBD) QD20100~QD20199(#2expansionBD)	200
	expansion ED	QD30000~QD30099(#1expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XDH:

	Tên	Phạm vi		Điểm	
		30 I/O	60 I/O	30	60
X	Input points	X0~X17	X0~X43	16	36
Y	Output points	Y0~Y15	Y0~Y27	14	24
X	Input points※3	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)		1024	
Y	Output points※3	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)		1024	
X	Input points※4	X20000~X20077(#1 expansion BD) X20100~X20177(#2 expansion BD)		128	
Y	Output points※4	Y20000~Y20077(#1 expansion BD) Y20100~Y20177(#2 expansion BD)		128	
X	Input points※5	X30000~X30077(#1 expansion ED)		64	
Y	Output points※5	Y30000~Y30077(#1 expansion ED)		64	
M	Internal relay	M0~M19999		20000	
HM		HM0~HM19999※1		20000	
SM		special purpose SM0~SM49999※2		50000	
S	Flow	S0~S19999		20000	
HS		HS0~HS1999※1		2000	
T	Timer	T0~T19999		20000	
HT		HT0~HT1999※1		2000	
ET		Precise timer ET0~ET39		40	
C	Counter	C0~C19999		20000	
HC		HC0~HC1999※1		2000	
HSC		High speed counter HSC0~HSC39		40	

D	Data register	D0~D49999	50000
HD		HD0~HD49999*1	50000
SD		special purpose SD0~SD49999	50000
HSD		special purpose HSD0~HSD49999*2	50000
FD	FlashROM register	FD0~FD65535	65536
SFD		special purpose SFD0~SFD49999*2	50000
FS	Special secret register	FS0~FS47	48
ID*6	Main body	ID0~ID99	100
	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)	1600
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)	200
	expansion ED	ID30000~ID30099(#1 expansion ED)	100
QD*7	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1 expansion BD) QD20100~QD20199(#2 expansion BD)	200
	expansion ED	QD30000~QD30099(#1 expansion ED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng XD1:

	Tên	Phạm vi	Điểm
		16 I/O	16
X	Input points	X0~X7	8
Y	Output points	Y0~Y7	8
X	Input points*3	X10000~X10077(#1 expansion module) X11100~X11177(#10 expansion module)	640
Y	Output points*3	Y10000~Y10077(#1 expansion module) Y11100~Y11177(#10 expansion module)	640
X	Input points*4	X20000~X20077(#1expansionBD) X20100~X20177(#2expansionBD)	128
Y	Output points*4	Y20000~Y20077(#1expansionBD) Y20100~Y20177(#2expansionBD)	128
X	Input points*5	X30000~X30077(#1expansionED)	64
Y	Output points*5	Y30000~Y30077(#1expansionED)	64
M	Internal relay	M0~M7999	8000
HM		HM0~HM959*1	960
SM		special purpose SM0~SM2047*2	2048

S	Flow	S0~S1023	1024
HS		HS0~HS127* ¹	128
T	Timer	T0~T575	576
HT		HT0~HT95* ¹	96
ET		precise timer ET0~ET31	32
C	Counter	C0~C575	576
HC		HC0~HC95* ¹	96
HSC		high speed counter HSC0~HSC31	32
D	Data register	D0~D7999	8000
HD		HD0~HD999* ¹	1000
SD		special purpose SD0~SD2047	2048
HSD		special purpose HSD0~HSD499* ²	500
FD	FlashROM register	FD0~FD5119	5120
SFD		special purpose SFD0~SFD1999* ²	2000
FS	Special secret register	FS0~FS47	48
ID* ⁶	Main body	ID0~ID99	100
	Expansion module	ID10000~ID10099(#1 expansion module) ID10900~ID10999(#10 expansion module)	1000
		expansion BD	ID20000~ID20099(#1expansionBD) ID20100~ID20199(#2expansionBD)
	expansion ED	ID30000~ID30099(#1expansionED)	100
QD* ⁷	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD10900~QD10999(#10 expansion module)	1000
		expansion BD	QD20000~QD20099(#1expansionBD) QD20100~QD20199(#2expansionBD)
	expansion ED	QD30000~QD30099(#1expansionED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng **XL3**:

	Tên	Phạm vi		Điểm	
		16 I/O	32 I/O	16	32
X	Input points	X0~X7	X0~X17	8	16
Y	Output points	Y0~Y7	Y0~Y17	8	16
X	Input points* ³	X10000~X10077(#1 expansion module) X11100~X11177(#10 expansion module)		640	

	Tên	Phạm vi		Điểm	
		16 I/O	32 I/O	16	32
Y	Output points ^{*3}	Y10000~Y10077(#1 expansion module) Y11100~Y11177(#10 expansion module)		640	
X	Input points ^{*4}	X20000~X20077(#1 expansion BD) X20100~X20177(#2 expansion BD)		128	
Y	Output points ^{*4}	Y20000~Y20077(#1 expansion BD) Y20100~Y20177(#2 expansion BD)		128	
X	Input points ^{*5}	X30000~X30077(#1 expansion ED)		64	
Y	Output points ^{*5}	Y30000~Y30077(#1 expansion ED)		64	
M	Internal relay	M0~M7999		8000	
HM		HM0~HM959 ^{*1}		960	
SM		special purpose SM0~SM2047 ^{*2}		2048	
S	Flow	S0~S1023		1024	
HS		HS0~HS127 ^{*1}		128	
T	Timer	T0~T575		576	
HT		HT0~HT95 ^{*1}		96	
ET		precise timer ET0~ET31		32	
C	Counter	C0~C575		576	
HC		HC0~HC95 ^{*1}		96	
HSC		high speed counter HSC0~HSC31		32	
D	Data register	D0~D7999		8000	
HD		HD0~HD999 ^{*1}		1000	
SD		special purpose SD0~SD2047		2048	
HSD		special purpose HSD0~HSD499 ^{*2}		500	
FD	FlashROM register	FD0~FD5119		5120	
SFD		special purpose SFD0~SFD1999 ^{*2}		2000	
FS	Special secret register	FS0~FS47		48	
ID ^{*6}	Main body	ID0~ID99		100	
	Expansion module	ID10000~ID10099(#1 expansion module) ID10900~ID10999(#10 expansion module)		1000	
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)		200	
	expansion ED	ID30000~ID30099(#1 expansion ED)		100	
QD ^{*7}	Main body	QD0~QD99		100	
	Expansion module	QD10000~QD10099(#1 expansion module) QD10900~QD10999(#10 expansion module)		1000	
	expansion BD	QD20000~QD20099(#1 expansion BD) QD20100~QD20199(#2 expansion BD)		200	

	Name	Range		Points	
		16 I/O	32 I/O	16	32
	expansion ED	QD30000~QD30099(#1 expansion ED)		100	
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31		32	

Danh sách các phần tử lập trình PLC dòng **XL5, XL5E, XLME**:

	Tên	Phạm vi			Điểm		
		16 I/O	32 I/O	64 I/O	16	32	64
X	Input points	X0~X7	X0~X17	X0~X37	8	16	32
Y	Output points	Y0~Y7	Y0~Y17	Y0~Y37	8	16	32
X	Input points ^{※3}	X10000~X10077(#1 expansion module) X11700~X11777(#16 expansion module)			1024		
Y	Output points ^{※3}	Y10000~Y10077(#1 expansion module) Y11700~Y11777(#16 expansion module)			1024		
X	Input points ^{※4}	X20000~X20077(#1expansio nBD) X20100~X20177(#2expansio nBD)			128		
Y	Output points ^{※4}	Y20000~Y20077(#1expansio nBD) Y20100~Y20177(#2expansio nBD)			128		
X	Input points ^{※5}	X30000~X30077(#1expansionED)			64		
Y	Output points ^{※5}	Y30000~Y30077(#1expansionED)			64		
M	Internal relay	M0~M69999			70000		
HM		HM0~HM11999 ^{※1}			12000		
SM		special purpose SM0~SM4999 ^{※2}			5000		
S	Flow	S0~S7999			8000		
HS		HS0~HS999 ^{※1}			1000		
T	Timer	T0~T4999			5000		
HT		HT0~HT1999 ^{※1}			2000		
ET		precise timer ET0~ET39			40		
C	Counter	C0~C4999			5000		
HC		HC0~HC1999 ^{※1}			2000		
HSC		high speed counter HSC0~HSC39			40		
D	Data register	D0~D69999			70000		
HD		HD0~HD24999 ^{※1}			25000		
SD		special purpose SD0~SD4999			5000		
HSD		special purpose HSD0~HSD1023 ^{※2}			1024		
FD	FlashRO	FD0~FD8191			8192		
SFD	M register	special purpose SFD0~SFD5999 ^{※2}			6000		
FS	Special secret register	FS0~FS47			48		
ID ^{※6}	Main body	ID0~ID99			100		
	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)			1600		

	expansion BD	ID20000~ID20099(#1expansion BD) ID20100~ID20199(#2expansion BD)	200
	expansion ED	ID30000~ID30099(#1expansion ED)	100
QD [*] ₇	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1expansion BD) QD20100~QD20199(#2expansion BD)	200
	expansion ED	QD30000~QD30099(#1expansion ED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

Danh sách các phần tử lập trình PLC dòng **XLH**:

	Tên	Phạm vi	Điểm
		24 I/O	24
X	Input points ^{*5}	X0-X13	12
Y	Output points ^{*5}	Y0-Y13	12
X	Internal input points	X0-X77	64
Y	Internal output points	Y0-Y77	64
X	Module input points ^{*3}	X10000~X10077(#1 expansion module) X11100~X11177(#16 expansion module)	1024
Y	Module output points ^{*3}	Y10000~Y10077(#1 expansion module) Y11100~Y11177(#16 expansion module)	1024
X	BD input points ^{*4}	X20000~X20077(#1 expansion BD) X20100~X20177(#2 expansion BD)	128
Y	BD output points ^{*4}	Y20000~Y20077(#1 expansion BD) Y20100~Y20177(#2 expansion BD)	128
X	ED input points ^{*5}	X30000~X30077(#1 expansion ED)	64
Y	ED output points ^{*5}	Y30000~Y30077(#1 expansion ED)	64
M	Internal relay	M0~M199999	200000
HM		HM0~HM19999 ^{*1}	20000
SM		Special purpose SM0~SM49999 ^{*2}	50000
S	Flow	S0~S19999	20000

	Tên	Phạm vi	Điểm
		24 I/O	24
HS		HS0~HS1999* ¹	2000
T	Timer	T0~T19999	20000
HT		HT0~HT1999* ¹	2000
ET		precise timer ET0~ET39	40
C	Counter	C0~C19999	20000
HC		HC0~HC1999* ¹	2000
HSC		High speed counter HSC0~HSC39	40
D	Data register	D0~D499999	500000
HD		HD0~HD49999* ¹	50000
SD		Special purpose SD0~SD49999	50000
HSD		Special purpose HSD0~HSD49999* ²	50000
FD	FlashROM register	FD0~FD65535	65536
SFD		Special purpose SFD0~SFD65487* ²	65488
FS	Special secret register	FS0~FS47	48
ID* ⁶	Main body	ID0~ID99	100
	Expansion module	ID10000~ID10099(#1 expansion module) ID11500~ID11599(#16 expansion module)	1600
	expansion BD	ID20000~ID20099(#1 expansion BD) ID20100~ID20199(#2 expansion BD)	200
	expansion ED	ID30000~ID30099(#1 expansion ED)	100
QD* ⁷	Main body	QD0~QD99	100
	Expansion module	QD10000~QD10099(#1 expansion module) QD11500~QD11599(#16 expansion module)	1600
	expansion BD	QD20000~QD20099(#1 expansion BD) QD20100~QD20199(# expansion BD)	200
	expansion ED	QD30000~QD30099(#1 expansion ED)	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

-
- ※1: **【】** Vùng bộ nhớ là vùng lưu giữ khi mất điện mặc định (Lưu ý: Không thể sửa đổi vùng lưu giữ khi mất điện của PLC dòng XD/XL).
- ※2: Mục đích sử dụng đặc biệt (bảo trì không mất điện) đề cập đến các thanh ghi dành cho mục đích đặc biệt được sử dụng bởi hệ thống, không thể sử dụng cho các mục đích khác. Để biết chi tiết, hãy tham khảo các phần liên quan của Danh sách các Phần tử lập trình đặc biệt trong phần phụ lục của sách hướng dẫn này.
- ※3: Gán địa chỉ I/O (bát phân) của mô-đun mở rộng, có thể được sử dụng làm rơle trung gian khi mô-đun mở rộng không được kết nối. (XL1/XD1/XD2 không hỗ trợ các mô-đun mở rộng, XD3/XL3 có thể mở rộng lên tới 10 mô-đun cùng một lúc, XD5/XDM/XDC/XD5E/XDME/XDH/XL5/XL5E/XLME có thể mở rộng lên đến 16 cùng một lúc thời gian)
- ※4: Gán địa chỉ I/O BD mở rộng (bát phân), có thể được sử dụng làm rơle trung gian khi không được kết nối với BD. (PLC 24/32/30 điểm có thể mở rộng lên 1 mô-đun, PLC 48/60 điểm có thể mở rộng lên 2 mô-đun, PLC 16 điểm không hỗ trợ BD mở rộng, dòng XL/XDH không hỗ trợ BD mở rộng)
- ※5: Gán địa chỉ I/O ED mở rộng (bát phân), có thể được sử dụng làm rơle trung gian khi không được kết nối với ED. (Dòng XD/XL có thể mở rộng tối đa một mô-đun ED, XDH không thể hỗ trợ mô-đun ED)
- ※6: Địa chỉ phần tử lập trình đầu vào tương tự (Analog), có thể được sử dụng làm thanh ghi phụ trợ khi không kết nối với thiết bị mở rộng
- ※7: Địa chỉ phần tử lập trình đầu ra analog, có thể được sử dụng làm thanh ghi phụ trợ khi không kết nối với các thiết bị mở rộng.
- ※8: Phạm vi của các phần tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số rơle không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3
-

2-4 Role đầu vào/đầu ra (X, Y)

Danh sách địa chỉ

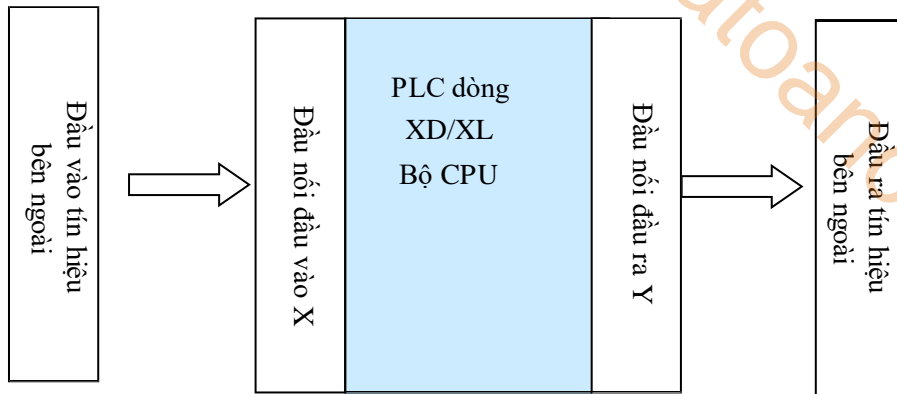
Địa chỉ đầu vào/đầu ra PLC dòng XD đều ở dạng bát phân, danh sách địa chỉ của các dòng như sau:

Dòng	Tên	Phạm vi								Điểm							
		10 I/O	16 I/O	24 I/O	30 I/O	32 I/O	42 I/O	48 I/O	60 I/O	10	16	24	30	32	42	48	60
XD1	X	X0~X4	X0~X7	X0~X13	-	X0~X17	-	-	-	5	8	12	-	16	-	-	-
	Y	Y0~Y4	Y0~Y7	X0~X13	-	Y0~Y17	-	-	-	5	8	12	-	16	-	-	-
XD2 XD3 XD5	X	-	X0~X7	X0~X15	-	X0~X21	X0~X27	X0~X33	X0~X43	-	8	14	-	18	24	28	36
	Y	-	Y0~Y7	Y0~Y11	-	Y0~Y15	Y0~Y21	Y0~Y23	Y0~Y27	-	8	10	-	14	18	20	24
XDM	X	-	-	X0~X15	-	X0~X21	-	X0~X43	-	-	14	-	18	-	-	36	
	Y	-	-	Y0~Y11	-	Y0~Y15	-	Y0~Y27	-	-	10	-	14	-	-	24	
XDC	X	-	-	X0~X15	-	X0~X21	X0~X33	X0~X43	-	-	14	-	18	-	28	36	
	Y	-	-	Y0~Y11	-	Y0~Y15	Y0~Y23	Y0~Y27	-	-	10	-	14	-	20	24	
XD5E	X	-	-	X0~X15	X0~X17	-	X0~X33	X0~X43	-	-	-	16	-	-	28	36	
	Y	-	-	Y0~Y11	Y0~Y15	-	Y0~Y23	Y0~Y27	-	-	-	14	-	-	20	24	
XDME	X	-	-	-	-	-	-	X0~X43	-	-	-	-	-	-	-	36	
	Y	-	-	-	-	-	-	Y0~Y27	-	-	-	-	-	-	-	24	
XDH	X	-	-	-	X0~X17	-	-	X0~X43	-	-	-	16	-	-	-	36	
	Y	-	-	-	Y0~Y15	-	-	Y0~Y27	-	-	-	14	-	-	-	24	

Địa chỉ đầu vào/đầu ra PLC dòng XL đều ở dạng bát phân, danh sách địa chỉ của các dòng như sau:

Dòng	Tên	Phạm vi					Điểm					
		16I/O	24I/O	30I/O	32I/O	64I/O	16	24	30	32	64	
XL1	X	X0~X7	-	-	-	-	8	-	-	-	-	
	Y	Y0~Y7	-	-	-	-	8	-	-	-	-	
XL3 XL5 XL5E	X	X0~X7	-	-	X0~X17	X0~X37	8	-	-	16	32	
	Y	Y0~Y7	-	-	Y0~Y17	Y0~Y37	8	-	-	16	32	
XLME	X	-	-	-	X0~X17	X0~X37	-	-	-	16	32	
	Y	-	-	-	Y0~Y17	Y0~Y37	-	-	-	16	32	
XLH	X	-	X0~X13	-	-	-	-	12	-	-	-	
	Y	-	Y0~Y13	-	-	-	-	12	-	-	-	

Chức năng



Role đầu vào X

Thiết bị đầu cuối đầu vào PLC được sử dụng để nhận tín hiệu bên ngoài. Role đầu vào là opto cách ly quang để kết nối PLC và các đầu nối đầu vào

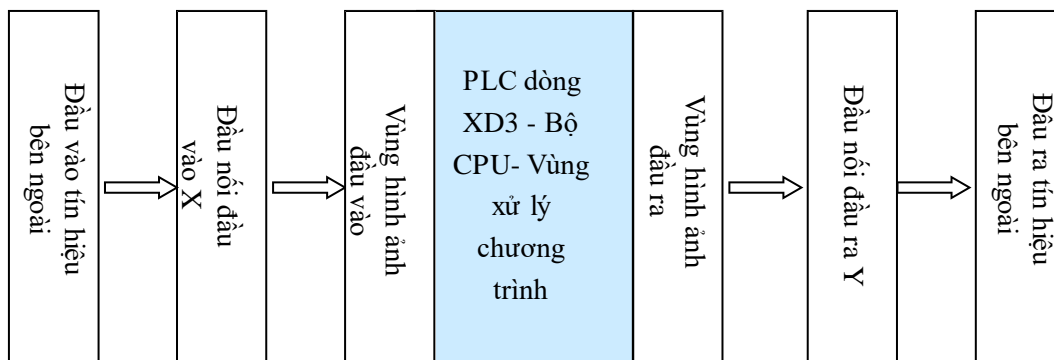
Role đầu vào không được kết nối với các thiết bị bên ngoài có thể được coi là role nội tốc độ cao.

Role đầu ra Y

Các đầu nối đầu ra của PLC có thể được sử dụng để gửi tín hiệu đến tải bên ngoài. Bên trong PLC, các công tắc tơ đầu ra bên ngoài của role đầu ra (bao gồm công tắc tơ role, công tắc tơ của linh kiện bán dẫn) kết nối với các đầu nối đầu ra.

Role đầu ra không được kết nối với các thiết bị bên ngoài có thể được coi là rowle nội tốc độ cao.

Trình tự thực hiện



Xử lý đầu vào

Trước khi PLC thực hiện chương trình, hãy đọc trạng thái ON/OFF của tất cả đầu vào vào vùng hình ảnh.

Khi chương trình đang chạy, ngay cả khi đầu vào thay đổi, nội dung trong vùng hình ảnh đầu vào sẽ không thay đổi cho đến lần quét tiếp theo.

Xử lý đầu ra

Sau khi chạy tất cả các lệnh, chuyển trạng thái ON/OFF của vùng hình ảnh đầu ra Y sang vùng bộ nhớ khóa đầu ra. Đây sẽ là đầu ra thực tế của PLC.

Các công tắc tơ đầu ra sẽ trì hoãn hoạt động tùy theo phản hồi của các phần tử lập trình đầu ra.

2-5 Role trung gian (M, HM, SM)

Danh sách địa chỉ

Các role trung gian trong PLC dòng XD/XL đều ở dạng thập phân, vui lòng tham khảo bảng sau:

Dòng	Tên	Phạm vi		
		Mục đích bình thường	Lưu giữ khi tắt nguồn	Mục đích đặc biệt
XD1/XD2/XD3	M HM SM	M0~M7999	HM0-HM959	SM0~SM2047
XD5/XDM/XDC/XD5E/XDME		M0~M69999	HM0-HM11999	SM0~SM4999
XDH		M0~M199999	HM0~HM19999	SM0~SM49999
XL1/XL3		M0~M7999	HM0-HM959	SM0~SM2047
XL5/XL5E/XLME		M0~M69999	HM0-HM11999	SM0~SM4999
XLH		M0~M199999	HM0~HM19999	SM0~SM49999

Trong PLC, role trung gian được sử dụng thường xuyên. Cuộn dây của loại role này giống với cuộn dây của role đầu ra. Chúng được điều khiển bởi các bộ phận mềm trong PLC;

Role trung gian M và HM có vô số công tắc tơ Thường hở/đóng. Chúng có thể được sử dụng tùy ý, nhưng loại công tắc tơ này không thể điều khiển tải bên ngoài.

- Đối với mục đích sử dụng thông thường

Loại role trung gian này chỉ có thể được sử dụng như role phụ thông thường. Ví dụ, nếu nguồn điện đột ngột ngắt trong khi đang chạy, role sẽ tắt.

Không thể sử dụng role sử dụng thông thường để lưu giữ khi tắt nguồn, nhưng vùng này có thể được chỉnh sửa;

- Cho mục đích lưu giữ khi tắt nguồn

Các role phụ dùng để tắt nguồn có lưu giữ, ngay cả khi PLC ở trạng thái OFF, chúng vẫn có thể giữ trạng thái ON/OFF.

Không thể chỉnh sửa vùng lưu giữ khi tắt nguồn;

Power off retentive relays are usually used to memory the status before stop the power, then when power the PLC on again, the status can run again;

Role có thể lưu giữ khi tắt nguồn thường được sử dụng để ghi nhớ trạng thái trước khi tắt nguồn, sau đó khi bật lại PLC, trạng thái có thể chạy trở lại;

- Cho mục đích sử dụng đặc biệt

Role đặc biệt là một số loại role được xác định với chức năng đặc biệt, bắt đầu từ SM0. Role đặc biệt có hai chức năng, đầu tiên là dùng để điều khiển cuộn dây, loại còn lại dùng cho mục đích chạy đặc biệt.

Ví dụ: SM2 là xung ban đầu, chỉ kích hoạt tại thời điểm bắt đầu SM34 “tắt cả đầu ra bị vô hiệu hóa”

Role trung gian đặc biệt không thể được sử dụng như role M thông thường;

Lưu ý: Phạm vi của các phần tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số role không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3

2-6 Role trạng thái (S, HS)

Danh sách địa chỉ

Địa chỉ role trạng thái của PLC dòng XD/XL có dạng thập phân, địa chỉ được hiển thị như sau:

Dòng	Tên	Phạm vi	
		Mục đích bình thường	Lưu giữ khi tắt nguồn
XD1/XD2/XD3	S HS	S0~S1023	HS0~HS127
XD5/XDM/XDC/XD5E/XDME		S0~S7999	HS0~HS999
XDH		S0~S19999	HS0~HS1999
XL1/XL3		S0~S1023	HS0~HS127
XL5/XL5E/XLME		S0~S7999	HS0~HS999
XLH		S0~S19999	HS0~HS1999

Chức năng

Role trạng thái S và HS rất quan trọng trong chương trình bậc thang; chúng được sử dụng cùng với lệnh “STL” trong luồng. Luồng này có thể giúp cho chương trình rõ ràng và dễ sửa đổi.

- Đối với mục đích sử dụng thông thường

Sau khi tắt nguồn PLC, role S sẽ TẮT

- Đối với mục đích lưu giữ khi tắt nguồn

Role HS có thể giữ trạng thái ON/OFF ngay cả khi nguồn PLC bị tắt

- Role trạng thái cũng có vô số công tác to “Thường hở/đóng”. Vì vậy người dùng có thể sử dụng chúng tùy ý trong chương trình.

Lưu ý: Phạm vi của các phần tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số role không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3.

2-7 Bộ định thời (T, HT)

Danh sách địa chỉ

Địa chỉ bộ định thời của PLC dòng XD/XL ở dạng thập phân; vui lòng xem bảng dưới đây:

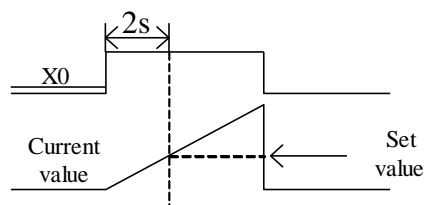
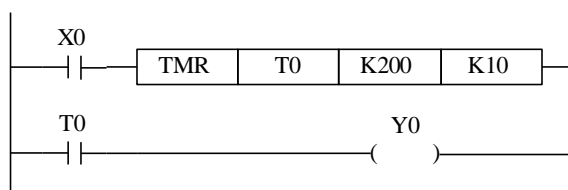
Series	Name	Range		
		Normal	Power-off holding	Precise timer
XD1/XD2/XD3	T HT ET	T0~T575	HT0~HT95	ET0~ET24
XD5/XDM/XDC/XD5E/XDME		T0~T4999	HT0~HT1999	ET0~ET24
XDH		T0~T19999	HT0~HT1999	-
XL1/XL3		T0~T575	HT0~HT95	ET0~ET24
XL5/XL5E/XLME		T0~T4999	HT0~HT1999	ET0~ET24
XLH		T0~T19999	HT0~HT1999	-

Chức năng

Bộ hẹn giờ tích lũy xung 1ms, 10ms, 100ms, công tắc tơ đầu ra kích hoạt khi tích lũy đạt đến giá trị cài đặt;

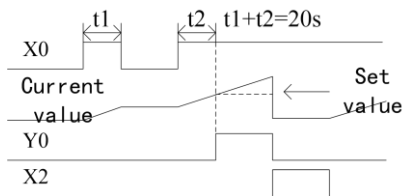
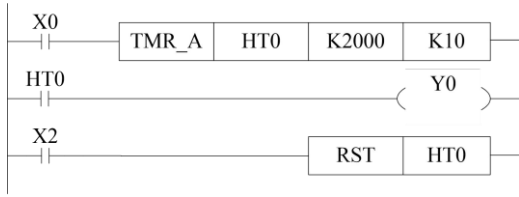
Hướng dẫn TMR dành cho bộ định thời thông thường. Giá trị cài đặt có thể là hằng số (K) hoặc thanh ghi dữ liệu (D).

Kiểu thông thường



Nếu X0 ở trạng thái ON thì T0 tích lũy xung 10ms dựa trên giá trị hiện tại; khi giá trị tích lũy đạt đến giá trị cài đặt K200, đầu ra bộ định thời sẽ kích hoạt, chẳng hạn, đầu ra kích hoạt 2 giây sau. Nếu X0 ở trạng thái OFF, bộ định thời sẽ đặt lại, đầu ra sẽ đặt lại;

Loại tích lũy



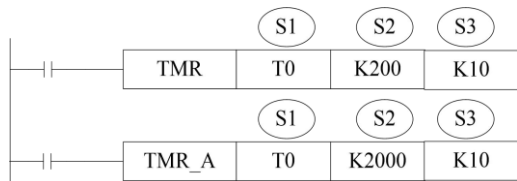
Nếu X0 ở trạng thái ON, HT0 sẽ tích lũy xung xung 10ms dựa trên giá trị hiện tại. Khi giá trị tích lũy đạt đến giá trị cài đặt K2000, đầu ra bộ định thời sẽ kích hoạt.

Nếu X0 đột ngột chuyển sang trạng thái OFF trong khi bộ định thời đang hoạt động, giá trị bộ định thời sẽ được lưu giữ. Sau đó X0 chuyển sang trạng thái ON trở lại, bộ định thời sẽ tiếp tục được chạy.

Khi X2 ở trạng thái ON, bộ định thời và đầu ra sẽ được đặt lại.

Chỉ định giá trị đã đặt

1. Định dạng Lệnh



(Không tích lũy)

(Tích lũy)

Đặt lại bộ định thời và đầu ra:



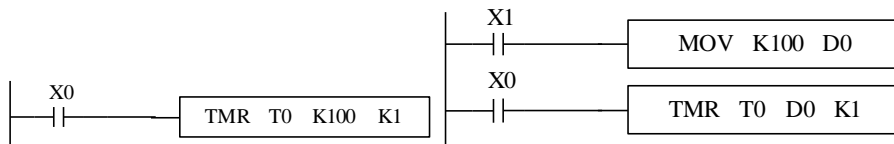
S1: timer (T0, HT10)

S2: set time (such as K100)

S3: time unit (K1—1ms, K10—10ms, K100—100ms)

Không lưu giữ khi tắt nguồn, không tích lũy

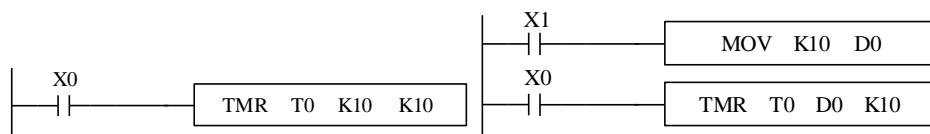
(1) Đơn vị thời gian là 1ms, thời gian cài đặt là K100, thời gian thực là $1ms * 100 = 0,1s$



Giá trị đặt là hằng số K

Giá trị đặt là thanh ghi D

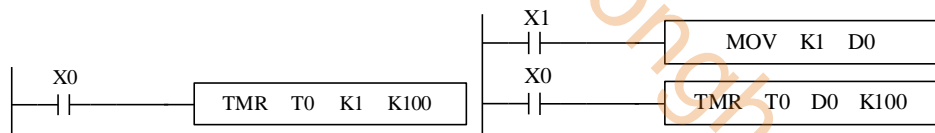
(2) Đơn vị thời gian là 10ms, thời gian cài đặt là K10, thời gian thực là $10ms * 10 = 0,1s$



Giá trị đặt là hằng số K

Giá trị đặt là thanh ghi D

(3) Đơn vị thời gian là 100ms, thời gian cài đặt là K1, thời gian thực là $100ms * 1 = 0,1s$

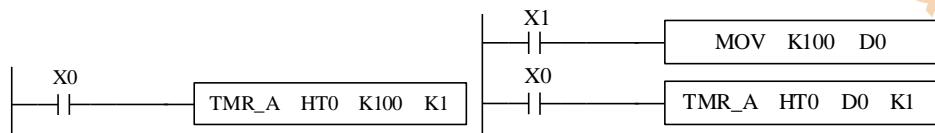


Giá trị cài đặt là hằng số K

Giá trị cài đặt là thanh ghi D

Lưu giữ khi tắt nguồn, tích lũy

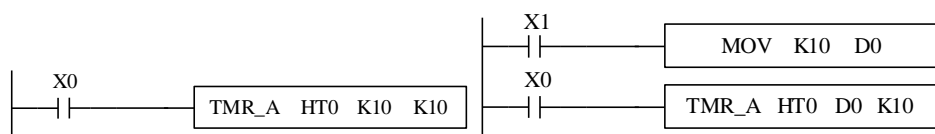
(1) Đơn vị thời gian là 1ms, thời gian cài đặt là K100, thời gian thực là $1\text{ms} * 100 = 0,1\text{s}$



Giá trị cài đặt là hằng số K

Giá trị cài đặt là thanh ghi D

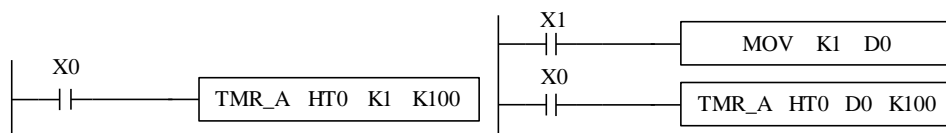
(2) Đơn vị thời gian là 10ms, thời gian cài đặt là K10, thời gian thực là $10\text{ms} * 10 = 0,1\text{s}$



Giá trị cài đặt là hằng số K

Giá trị cài đặt là thanh ghi D

(3) Đơn vị thời gian là 100ms, thời gian cài đặt là K1, thời gian thực là $100\text{ms} * 1 = 0,1\text{s}$



Giá trị cài đặt là hằng số K

Giá trị cài đặt là thanh ghi D

Lưu ý

(1) Bộ định thời có tích lũy, không tích lũy, 1ms, 10ms và 100ms nên có thể phân biệt bằng lệnh; nghĩa là, cùng một bộ định thời có thể được sử dụng dưới dạng tích lũy hoặc không tích lũy và đơn vị cơ sở thời gian của nó cũng được chỉ định bởi các lệnh là 1ms, 10ms hoặc 100ms.

(2) Mặt khác, mặc dù chương trình có thể được ghi vào phần mềm lập trình và tải xuống PLC nhưng lệnh định thời sẽ không được thực thi.

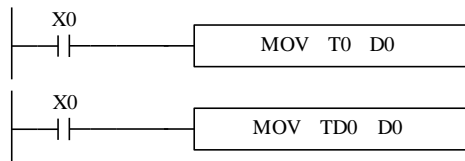
(3) Tham số thứ ba của lệnh chỉ có thể dựa trên K1, K10 và K100. Vui lòng không ghi các giá trị hoặc thanh ghi khác ngoài ba tham số này. Mặt khác, mặc dù chương trình có thể được ghi vào phần mềm lập trình và tải xuống PLC nhưng lệnh định giờ sẽ không được thực thi.

(4) Phạm vi cài đặt của hằng số K và giá trị cài đặt thực tế của bộ định thời được thể hiện trong bảng sau:

Bộ định thời	Phạm vi K	Phạm vi thực tế
Bộ định thời 1ms	1~32,767	0.001~32.767s
Bộ định thời 1ms		0.01~327.67s
Bộ định thời 1ms		0.1~3276.7s

Giá trị thời gian

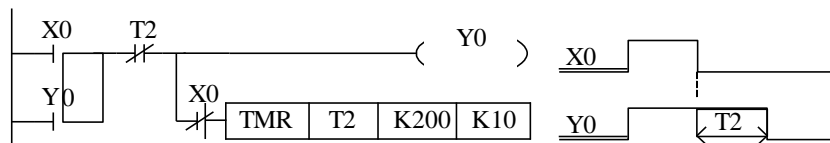
Giá trị thời gian được lưu trong thanh ghi TD. Chế độ làm việc của bộ định thời T0~T575 và HT0~HT95 là chế độ tăng tuyến tính 16 bit. Phạm vi thời gian là từ 0 đến 32767. Khi giá trị thời gian trong TD đạt 32767, bộ định thời sẽ dừng tính giờ và giữ nguyên trạng thái.



Hai lệnh đều giống nhau. Trong lệnh đầu tiên, T0 dường như là TD0.

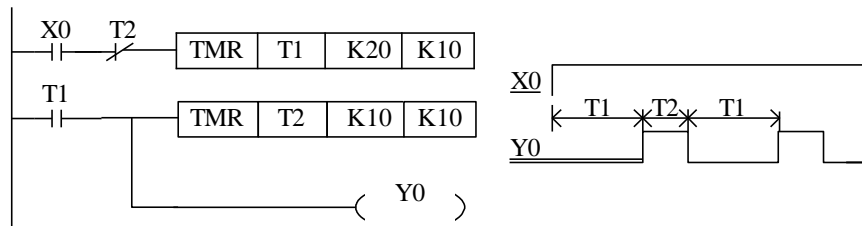
Ứng dụng

Độ trễ đầu ra



X0 ở trạng thái ON, đầu ra Y0. X0 chuyển từ ON sang OFF, trì hoãn 2s rồi cắt Y0.

Nhấp nháy



X0 ở trạng thái ON, Y0 bắt đầu nhấp nháy. T1 là thời gian Y0 ở trạng thái OFF; T2 là thời gian Y0 ở trạng thái ON.

Lưu ý: Phạm vi của các phân tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số rơle không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3.

2-8 Bộ đếm (C, HC, HSC)

Danh sách địa chỉ

Địa chỉ bộ đếm của PLC dòng XD/XL ở dạng thập phân; vui lòng xem bảng sau để biết chi tiết:

Dòng	Tên	Phạm vi		
		Mục đích bình thường	Lưu giữ khi tắt nguồn	Mục đích đặt biệt
XD1/XD2/XD3	C HC HSC	C0~C575	HC0~HC95	HSC0~HSC31
XD5/XDM/XDC/XD5E/XDME		C0~C4999	HC0~HC1999	HSC0~HSC39
XDH		C0~C19999	HC0~HC1999	HSC0~HSC39
XL1/XL3		C0~C575	HC0~HC95	HSC0~HSC31
XL5/XL5E/XLME		C0~C4999	HC0~HC1999	HSC0~HSC39
XLH		C0~C19999	HC0~HC1999	HSC0~HSC39

Phạm vi bộ đếm:

Loại bộ đếm	Mô tả
Bộ đếm lên/xuống 16/32 bit	C0~C575 HC0~HC95 (bộ đếm 32 bit chiếm hai thanh ghi, địa chỉ bộ đếm phải là số chẵn)
Bộ đếm tốc độ cao	HSC0~HSC30 (HSC0,HSC2...HSC30) (mỗi bộ đếm chiếm hai thanh ghi, địa chỉ bộ đếm phải là số chẵn)

- Vui lòng tham khảo chương 5 để biết chi tiết về bộ đếm tốc độ cao.
- Bộ đếm dòng XD/XL có thể ở chế độ đếm lên/xuống 16 hoặc 32 bit. Chế độ được chỉ định bởi lệnh. Điều đó có nghĩa là cùng một bộ đếm có thể được sử dụng dưới dạng 16 bit hoặc 32 bit. Chế độ đếm lên/xuống cũng được xác định bởi chế độ offset.

Tính năng của bộ đếm

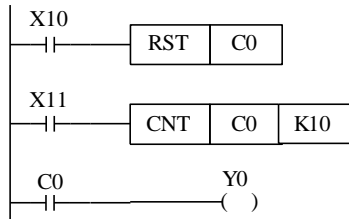
Loại	Bộ đếm 16-bit	Bộ đếm 32-bit
Chiều đếm	Đếm xuống/lên	Đếm lên/xuống
Giá trị đặt	-32,768~32,767	-2,147,483,648~+2,147,483,647
Kiểu giá trị đặt	Hằng số K hoặc thanh ghi	Hằng số K hoặc một cặp thanh ghi
Giá trị đếm	Giá trị sẽ không thay đổi khi đạt giá trị tối đa hoặc tối thiểu	Giá trị sẽ không thay đổi khi đạt giá trị tối đa hoặc tối thiểu
Đầu ra	Giữ trạng thái để đếm lên	Đặt lại để đếm xuống
Đặt lại	Chạy lệnh RST, bộ đếm và đầu ra sẽ được đặt lại	
Thanh ghi giá trị đếm hiện tại	16-bit	32-bit

Chức năng

Phần mềm sẽ chỉ định loại bộ đếm: bộ đếm cho mục đích sử dụng thông thường hoặc bộ đếm có thể lưu giữ khi tắt nguồn.

Bộ đếm 16-bit cho mục đích sử dụng thông thường và bộ đếm có thể lưu giữ khi tắt nguồn

Phạm vi giá trị cài đặt của bộ đếm đếm lên 16 bit là K1~K32.767(hệ thập phân). K0 và K1 có cùng chức năng. Chúng có nghĩa là đầu ra của bộ đếm sẽ hoạt động ở lần đếm đầu tiên. Nếu nguồn điện PLC bị cắt, giá trị bộ đếm thông thường sẽ được đặt lại. Giá trị của bộ đếm có thể lưu giữ khi tắt nguồn sẽ được duy trì.



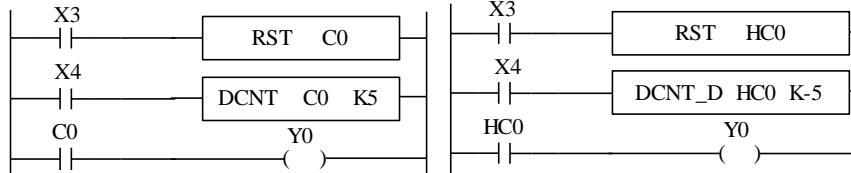
Bộ đếm C0 tăng lên 1 khi X11 chuyển trạng thái logic cao. Khi giá trị C0 đạt 10, đầu ra sẽ hoạt động. Sau đó X11 chuyển trạng thái lại, C0 sẽ tiếp tục tăng thêm 1.

Nếu X10 ở trạng thái ON, C0 và đầu ra sẽ được đặt lại.

Giá trị đặt của bộ đếm có thể là hằng số K hoặc thanh ghi. Ví dụ: nếu D10 là 123 thì giá trị cài đặt bằng K123.

Bộ đếm 32-bit cho mục đích sử dụng thông thường và bộ đếm có thể lưu giữ khi tắt nguồn

Phạm vi giá trị thiết lập của bộ đếm lên/xuống 32 bit là K+2,147,483,648~K-2,147,483,647 (thập phân). Chiều đếm được thiết lập thông qua lệnh.



Bộ đếm thông thường Bộ đếm có thể lưu giữ khi ngắt nguồn

Nếu X3 ở trạng thái ON, bộ đếm và đầu ra sẽ được đặt lại.

Đối với bộ đếm có thể lưu giữ khi tắt nguồn, giá trị bộ đếm hiện tại, trạng thái đầu ra sẽ được lưu giữ sau khi tắt nguồn.

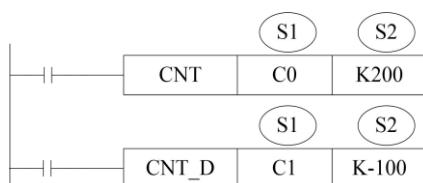
Bộ đếm 32 bit có thể được coi là thanh ghi 32 bit.

Giá trị đặt của bộ đếm

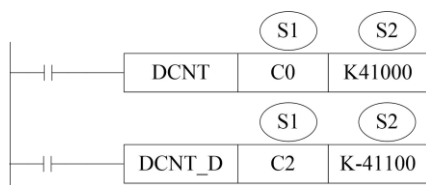
Giá trị được đặt bao gồm hai trạng thái: 16-bit và 32-bit. Các loại bộ đếm bao gồm bộ đếm thông thường (C) và bộ đếm có thể lưu giữ khi tắt nguồn (HC).

Lệnh đếm:

Bộ đếm 16-bit:



Bộ đếm 32-bit:



Lệnh reset (đặt lại):

Bộ đếm 16-bit:



Bộ đếm 32-bit:

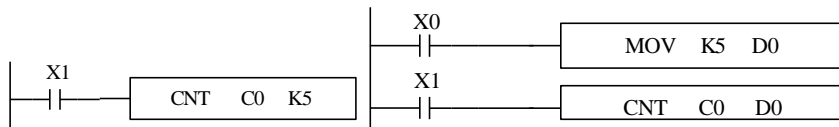


S1: bộ đếm (chẳng hạn như C0, HC10)

S2: giá trị đặt của bộ đếm (chẳng hạn như K100)

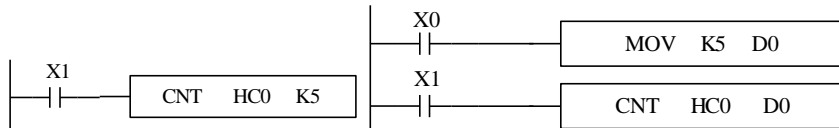
Bộ đếm 16-bit (không nhớ, đếm tiến)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



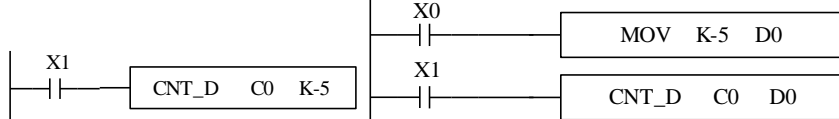
Bộ đếm 16-bit (có nhớ, đếm tiến)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



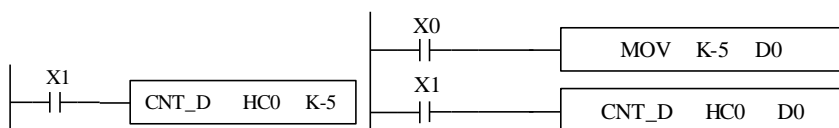
Bộ đếm 16-bit (không nhớ, đếm lùi)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



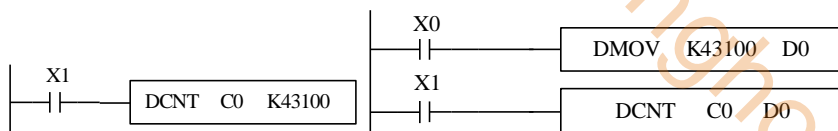
Bộ đếm 16-bit (có nhớ, đếm lùi)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



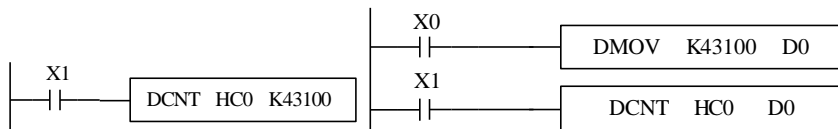
Bộ đếm 32-bit (không nhớ, đếm tiến)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



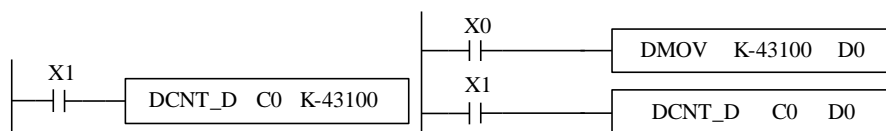
Bộ đếm 32-bit (có nhớ, đếm tiến)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



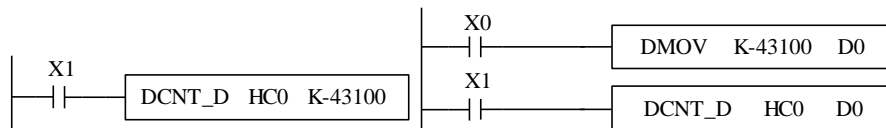
Bộ đếm 32-bit(không nhớ , đếm lùi)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



Bộ đếm 32-bit (có nhớ , đếm lùi)

《 cài đặt giá trị là hằng số 》 《 cài đặt giá trị qua thanh ghi 》



Lưu ý: Phạm vi cài đặt và giá trị cài đặt thực tế của hằng số K được thể hiện trong bảng sau:

Bộ đếm	Phạm vi cài đặt hằng số K	Phạm vi cài đặt thực tế
Bộ đếm 16-bit	1~32,767	1~32,767
Bộ đếm 32-bit	1~2,147,483,647	1~2,147,483,647

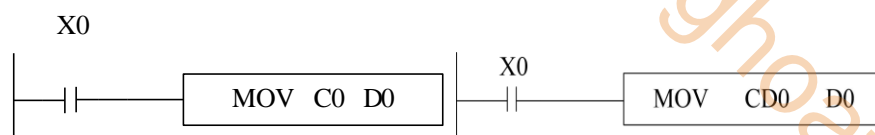
Giá trị đếm

Chế độ đếm của bộ đếm là chế độ tăng tuyến tính 16 bit (0 ~ K32,767). Khi giá trị đếm CD của bộ đếm đạt giá trị lớn nhất là K32,767 thì bộ đếm sẽ ngừng đếm và trạng thái của bộ đếm sẽ không thay đổi.

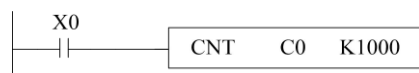
Chế độ đếm của bộ đếm là chế độ giảm tuyến tính 16 bit (-32768 ~ 0). Khi giá trị đếm của bộ đếm CD giảm xuống giá trị nhỏ nhất K-32,768 thì sẽ dừng đếm và trạng thái của bộ đếm không thay đổi.

Chế độ đếm của bộ đếm là chế độ tăng/giảm tuyến tính 32 bit (-2,147,483,648~+2,147,483,647). Khi giá trị bộ đếm tăng lên giá trị lớn nhất là K2,147,483,647 thì sẽ trở thành K 2,147,483,648. Khi giá trị đếm

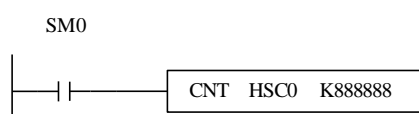
của bộ đếm giảm xuống giá trị tối thiểu K-2,147,483,648 sẽ trở thành K2,147,483,647, trạng thái ON/OFF của bộ đếm cũng sẽ thay đổi theo sự thay đổi của giá trị đếm.



Hai lệnh trên là như nhau. Trong lệnh bên trái, C0 được xử lý như một thanh ghi, trong khi ở lệnh bên phải, CD0 là thanh ghi dữ liệu tương ứng với bộ định thời C0. CD và C là sự tương ứng một-một.



Tần số cao nhất mà lệnh này có thể đếm được có liên quan đến việc lựa chọn các thông số bộ lọc và chu kỳ quét của PLC. Nên sử dụng bộ đếm tốc độ cao khi tần số đầu vào vượt quá 25Hz. Bộ đếm số tốc độ cao phải sử dụng HSC0-HSC30 và hệ thống dây phân cứng tương ứng.



Bộ đếm tốc độ cao, khi SM0 bật, HSC0 đếm tín hiệu xung của đầu vào X0. Bộ đếm tốc độ cao không bị ảnh hưởng bởi thời gian trễ phản hồi của bộ lọc đầu vào và thời gian quét chu kỳ. Do đó, các xung đầu vào tần số cao hơn có thể được xử lý. Tham khảo thông tin chi tiết ở chương 5.

Lưu ý: Phạm vi của các phân tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số role không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3.

2-9 Thanh ghi dữ liệu (D, HD, SD, HSD)

Danh sách địa chỉ

Thanh ghi dữ liệu của PLC dòng XD/XL có định dạng thập phân. Vui lòng xem bảng sau:

Dòng	Tên	Phạm vi			
		Mục đích bình thường	Lưu giữ khi tắt nguồn	Mục đích đặt biệt	Lưu giữ đặc biệt khi tắt nguồn
XD1/XD2/XD3	D HD SD HSD	D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XD5/XDM/XDC/XD5E/XDME		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XDH(except XDH-60A32)		D0~D499999	HD0~HD49999	SD0~SD49999	HSD0~HSD49999
XL1/XL3		D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XL5/XL5E/XLME		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XLH		D0~D499999	HD0~HD49999	SD0~SD49999	HSD0~HSD49999

Lưu ý: Đối với phiên bản phần mềm XD5 V3.4.6 trở lên, thanh ghi dữ liệu D nằm trong khoảng từ D0 đến D69999;

Phiên bản chương trình cơ sở XD5 dưới V3.4.6 và thanh ghi dữ liệu D nằm trong khoảng từ D0 đến D59999.

Cấu trúc

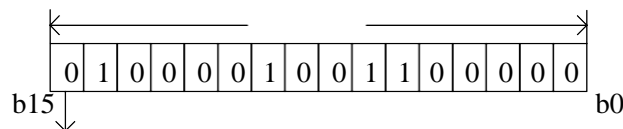
Thanh ghi dữ liệu được sử dụng để lưu trữ dữ liệu; nó bao gồm 16 bit (bit cao nhất là bit có dấu) và 32 bit. (32 bit chứa hai thanh ghi, bit cao nhất là bit dấu)

16 bits

Phạm vi thanh ghi 16 bit là -32,768 ~ +32,767

Đọc và ghi dữ liệu thanh ghi thông qua lệnh hoặc thiết bị khác như HMI.

D0 16-bits



Bit dấu

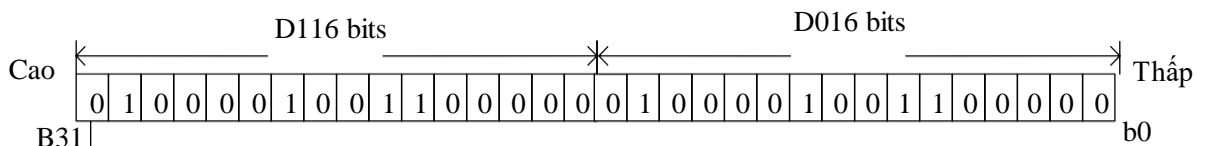
0: + 1: -

32 bits

Giá trị 32 bit bao gồm hai thanh ghi liên tục. Phạm vi là -2147483648 ~

2147483647. Ví dụ: (D1 D0) D1 là high16 bit, D0 là low16 bit.

Đối với thanh ghi 32 bit, nếu 16 bit thấp được chỉ định, chẳng hạn như D0, thì D1 sẽ tự động là 16 bit cao. Địa chỉ của thanh ghi 16 bit thấp phải là số chẵn.



Bit dấu

0: + 1: -

Chức năng

- Loại thông thường

Khi ghi một giá trị mới vào thanh ghi, giá trị cũ sẽ bị ẩn đi.

Khi PLC chuyển từ chế độ từ RUN sang STOP hoặc STOP sang RUN, giá trị trong thanh ghi sẽ bị xóa.

- Loại sử dụng cho mục đích lưu giữ khi tắt nguồn

Khi PLC chuyển từ RUN sang STOP hoặc tắt nguồn, giá trị trong thanh ghi sẽ được lưu giữ.

Phạm vi thanh ghi có thể lưu giữ không thể thay đổi.

- Loại sử dụng cho mục đích đặc biệt

Thanh ghi cho mục đích đặc biệt được sử dụng để thiết lập dữ liệu đặc biệt hoặc được hệ thống chiếm dụng.

Một số thanh ghi đặc biệt được khởi tạo khi bật nguồn PLC.

Vui lòng tham khảo phụ lục để biết địa chỉ và chức năng của thanh ghi dung cho mục đích đặc biệt.

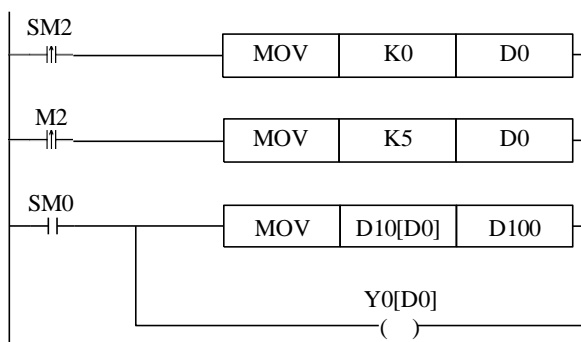
- Được sử dụng như con trỏ gián tiếp

Thanh ghi dữ liệu có thể được sử dụng làm offset của phần tử lập trình.

Định dạng : Dn[Dm], Xn[Dm], Yn[Dm], Mn[Dm].

Địa chỉ thanh ghi word: DXn[Dm] nghĩa là DX[n+Dm].

Giá trị offset chỉ có thể được đặt làm thanh ghi D.



Khi D0=0, D100=D10, Y0 ở trạng thái ON;

Khi M2 chuyển từ OFF→ON, D0=5, D100=D15, Y5 ở trạng thái ON. D10[D0]=D[10+D0], Y0[D0]=Y[0+D0].

Ví dụ

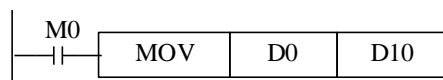
Thanh ghi dữ liệu D có thể xử lý nhiều loại dữ liệu. Lưu trữ dữ liệu



Khi M0 ở trạng thái ON, ghi 100 vào D0. (giá trị 16 bit)

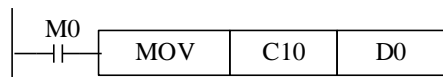
Khi M1 ở trạng thái ON, ghi 41100 vào D11,D10 (giá trị 32 bit)

Truyền dữ liệu



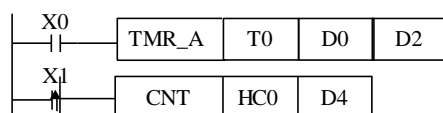
Khi M0 ở trạng thái ON, chuyển giá trị của D10 sang D0

Chuyển giá trị bộ đếm hiện tại



Khi M0 trạng thái ON, chuyển giá trị của C10 sang D0

Đặt lệnh và cài giá trị cho Timer và Counter



Khi X0 ở trạng thái ON, T10 bắt đầu hoạt động, T0 sẽ ở trạng thái ON khi giá trị D0 bằng giá trị bộ định thời, đơn vị thời gian là D2.

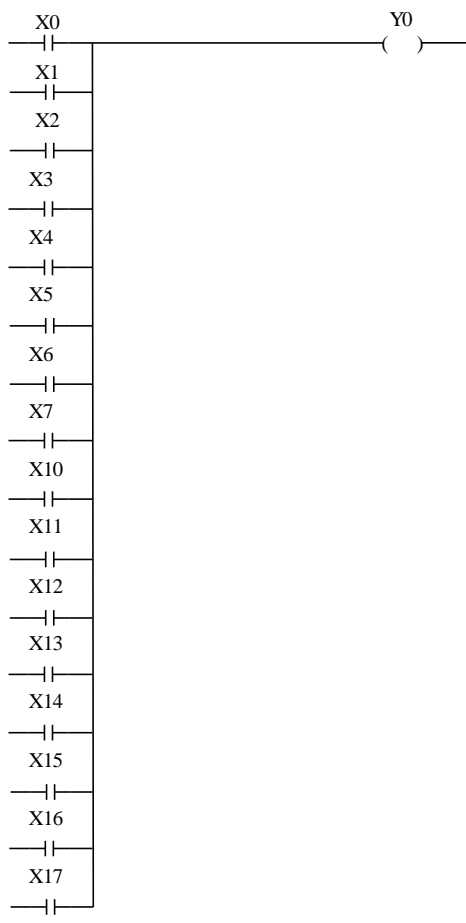
X1 ở trạng thái ON, HC0 bắt đầu hoạt động, HC0 ở trạng thái ON khi giá trị D4 bằng giá trị bộ đếm.

Lưu ý: Phạm vi của các phần tử lập trình được đề cập ở trên là phạm vi hợp lệ của PLC ở chế độ giao tiếp X-NET. Ở chế độ giao tiếp MODBUS, một số rơle không thể đọc và ghi. Phạm vi sử dụng cụ thể được trình bày trong chương 6-2-3

2-9-1 Kiểu gộp bit thành thanh ghi dạng Word

Một trong các cuộn dây từ X0 đến X17 ở trạng thái ON, Y0 sẽ ở trạng thái ON.

Phương pháp lập trình 1:



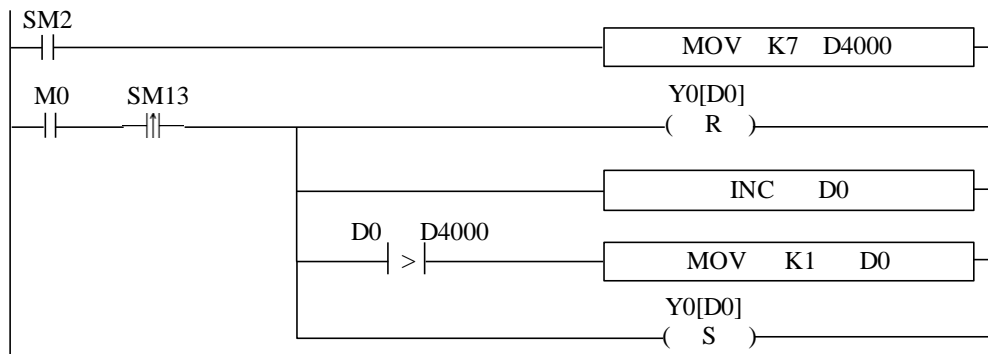
Phương pháp lập trình 2: : (áp dụng kiểu gộp bit thành word)



2-9-2 Ứng dụng của thanh ghi con trỏ

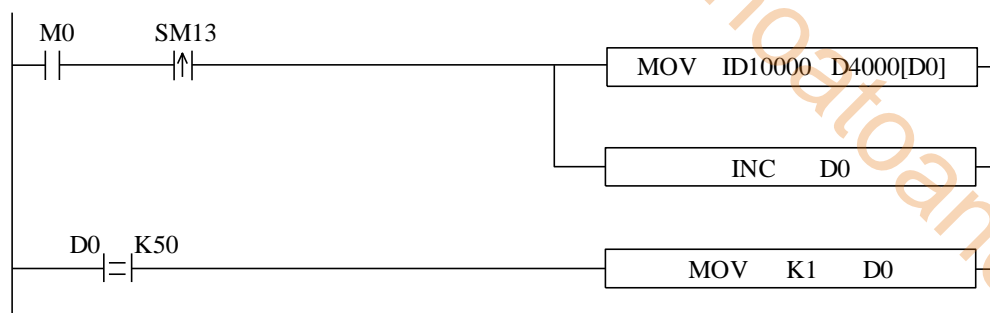
Ứng dụng 1:

Khi M0 ở trạng thái ON, đầu ra từ Y1 đến Y7 sẽ ON lần lượt. D0 là địa chỉ offset. Nếu có nhiều điểm đầu ra thì M có thể thay thế Y.



Ứng dụng 2:

Khi M0 ở trạng thái ON, đọc giá trị ID10000 mỗi giây và lưu vào thanh ghi bắt đầu từ D4000 (số lượng là 50 thanh ghi). D0 là địa chỉ offset.



2-10 Thanh ghi Flash (FD, SFD, FS)

Các thanh ghi FLASH của PLC dòng XD/XL đều được gán địa chỉ theo hệ thống thập phân. Các dòng được hiển thị trong bảng tương ứng.

Dòng	Tên	Phạm vi		
		Thanh ghi dữ liệu người dùng FLASH	Thanh ghi dữ liệu hệ thống FLASH	Thanh ghi FLASH chống đọc bằng mật khẩu
XD1/XD2/XD3	FD SFD FS	FD0~FD5119	SFD0~SFD1999	FS0~FS47
XD5/XDM/XDC/XD5E/XDME		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XDH		FD0~FD65535	SFD0~SFD4999 9	FS0~FS47
XL1/XL3		FD0~FD5119	SFD0~SFD1999	FS0~FS47
XL5/XL5E/XLME		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XLH		FD0~FD65535	SFD0~SFD6548 7	FS0~FS47

Chức năng

- Thanh ghi dữ liệu người dùng FLASH (FD)

Dùng để lưu trữ những dữ liệu quan trọng của người dùng, có thể lưu giữ khi tắt nguồn. Vùng lưu trữ này có thể ghi nhớ dữ liệu ngay cả khi tắt nguồn, do đó nó có thể được sử dụng để lưu trữ các thông số quá trình quan trọng.

- Thanh ghi dữ liệu hệ thống FLASH (SFD)

Dùng để lưu trữ các thông số hệ thống và có thể lưu giữ dữ liệu khi tắt nguồn.

Vùng lưu trữ là khối tham số hệ thống và người dùng không thể sửa đổi nó theo ý muốn.

- Thanh ghi chống đọc bằng mật khẩu FLASH (FS)

Một phần của thanh ghi FlashROM được sử dụng để lưu trữ dữ liệu của các phần tử lập trình, được biểu thị bằng ký hiệu FS. Các giá trị trong thanh ghi FS có thể được ghi nhưng không thể đọc được, vì vậy chúng có thể được sử dụng để bảo vệ quyền sở hữu trí tuệ của người dùng.

Giá trị của phần tử lập trình có thể được đặt tùy ý trong thanh ghi FS, nhưng không thể đọc được giá trị của thanh ghi (luôn trả về 0); và nó không thể so sánh với thanh ghi trong phần mềm máy tính chủ, mà chỉ so sánh với hằng số nên không thể đọc được giá trị thực của thanh ghi.

Vùng lưu trữ này có thể ghi nhớ dữ liệu ngay cả khi tắt nguồn, do đó nó có thể được sử dụng để lưu trữ các thông số quá trình quan trọng.

Lưu ý:

(1) Khi sử dụng lệnh MOV để truyền dữ liệu tới FD, SFD và FS, chỉ sườn lên là hợp lệ, ngay cả khi điều kiện điều khiển là cuộn dây thường đóng/mở thì lệnh chỉ được thực hiện một lần

(2) Các thanh ghi Flash có thể được ghi khoảng 1.000.000 lần và mỗi lần ghi sẽ bị xóa đối với toàn bộ các thanh ghi Flash, việc này rất tốn thời gian. Việc ghi thường xuyên sẽ gây tổn hại vĩnh viễn cho các thanh ghi Flash, vì vậy người dùng không nên ghi thường xuyên. Không sử dụng cuộn dây dao động (ví dụ SM11) làm điều kiện điều khiển.

(3) Khi dữ liệu được truyền đến cùng một thanh ghi Flash nhiều lần, nếu giá trị trong thanh ghi nguồn không thay đổi so với lần truyền trước đó thì lệnh truyền sẽ không được thực thi ngay cả khi điều kiện điều khiển được thiết lập lại. Ví dụ: nếu giá trị trong D0 được truyền tới FD100 thì giá trị trong D0 là 300 khi lệnh truyền được thực thi lần đầu tiên; nếu điều kiện điều khiển được thiết lập lần thứ hai, lệnh truyền sẽ không được thực thi nếu giá trị trong D0 vẫn là 300.

(4) Để tránh nhiễu tín hiệu nhiễu (burr signal) khi truyền dữ liệu đến các thanh ghi Flash, không nên sử dụng các cuộn dây như SM0 và SM2 làm điều kiện điều khiển trực tiếp. Nên thực hiện các lệnh truyền sau khi bật nguồn PLC trong một khoảng thời gian.

2-11 Hằng số

Xử lý dữ liệu

PLC dòng XD/XL có 5 hệ thống số sau.

- DEC: HỆ THẬP PHÂN

Giá trị đặt trước của bộ đếm và bộ định thời (hằng số K)

Số lượng role phụ M, HM; bộ định thời T, HT; bộ đếm C, HC; trạng thái S, HS; thanh ghi D, HD.

Đặt làm giá trị toán hạng và hành động của lệnh được ứng dụng (hằng số K)

- HEX: HỆ THẬP LỤC PHÂN

Đặt làm giá trị toán hạng và hành động của lệnh được ứng dụng (hằng số H)

- BIN: HỆ NHỊ PHÂN

Bên trong PLC, tất cả các số sẽ được xử lý ở dạng nhị phân. Nhưng khi điều khiển trên thiết bị, toàn bộ hệ nhị phân sẽ được chuyển thành HEX hoặc DEC.

- OCT: HỆ BÁT PHÂN

Role I/O PLC dòng XD/XL có dạng bát phân. Chẳng hạn như [X0-7, X10-17,....X70-77].

- BCD: MÃ BCD

BCD sử dụng số nhị phân 4 bit để biểu thị số thập phân 0-9. BCD có thể được sử dụng trong công tắc kỹ thuật số đầu ra LED và BCD 7 đoạn

- Các số khác (số thực dấu phẩy động)

PLC dòng XD/XL có thể tính toán số thực dấu phẩy động có độ chính xác cao. Nó được tính bằng số nhị phân và hiển thị bằng số thập phân.

Hiển thị

Chương trình PLC nên sử dụng K, H để xử lý các giá trị. K là số hệ thập phân, H là số hệ thập lục phân. Xin lưu ý role đầu vào/đầu ra PLC sử dụng địa chỉ bát phân.

- Hằng số K

K được sử dụng để hiển thị số thập phân. K10 có nghĩa là số thập phân 10. Nó được sử dụng để thiết lập giá trị bộ định thời và bộ đếm, giá trị toán hạng của lệnh ứng dụng.

- Hằng số H

H được sử dụng để hiển thị số thập lục phân. HA có nghĩa là số thập phân 10. Nó được sử dụng để đặt giá trị toán hạng của lệnh ứng dụng.

- Hằng số B

B được sử dụng để hiển thị số nhị phân. B10 có nghĩa là số thập phân 2. Nó được sử dụng để thiết lập giá trị toán hạng của lệnh ứng dụng.

2-12 Nguyên tắc lập trình

Ký hiệu P và I

P là ký hiệu chương trình cho lệnh nhảy có điều kiện và chương trình con.

I là ký hiệu cho ngắt chương trình (ngắt ngoài, ngắt bộ định thời, ngắt bộ đếm tốc độ cao, ngắt bộ định thời chính xác...).

Địa chỉ P và I ở dạng thập phân. Vui lòng tham khảo bảng sau:

Dòng	Ký hiệu	Địa chỉ
XD, XL	P	P0~P9999

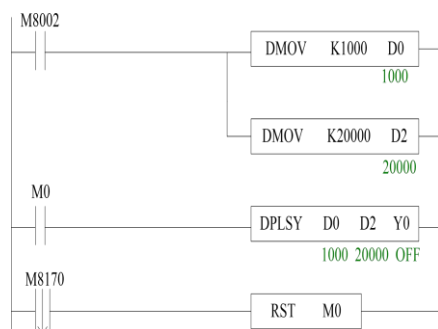
Dòng	Tên	Phạm vi			Ngắt bộ định thời
		Gián đoạn bên ngoài			
		Đầu nối đầu vào	Gián đoạn sườn lên	Gián đoạn sườn xuống	
XD/XL series 16 points	I	X2	I0000	I0001	Có 20 lần ngắt bộ định thời. Từ I40** đến I 59**. “**” nghĩa là thời gian ngắt của bộ định thời, đơn vị là ms.
		X3	I0100	I0101	
		X4	I0200	I0201	
		X5	I0300	I0301	
		X6	I0400	I0401	
		X7	I0500	I0501	

Dòng	Name	Phạm vi			Ngắt bộ định thời
		Ngắt ngoài			
		Đầu nối đầu vào	Ngắt sườn lên	Ngắt sườn xuống	
Dòng XD/XL 24-64 điểm	I	X2	I0000	I0001	Có 20 lần ngắt bộ định thời. Từ I40** đến I 59**. “**” nghĩa là thời gian ngắt của bộ định thời, đơn vị là ms.
		X3	I0100	I0101	
		X4	I0200	I0201	
		X5	I0300	I0301	
		X6	I0400	I0401	
		X7	I0500	I0501	
		X10	I0600	I0601	
		X11	I0700	I0701	
		X12	I0800	I0801	
		X13	I0900	I0901	

Ký hiệu P

P thường được sử dụng trong luồng; nó được sử dụng cùng với CJ (lệnh nhảy có điều kiện), CALL (gọi chương trình con), v.v.

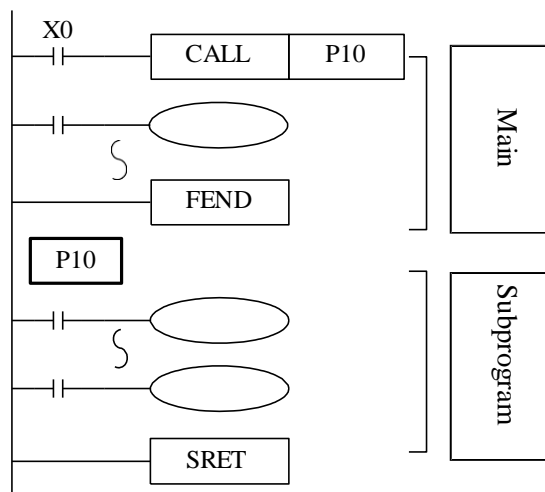
Lệnh nhảy có điều kiện CJ



Nếu cuộn dây X0 ở trạng thái ON, nhảy tới chương trình sau P1;

Nếu cuộn X0 không ở trạng thái ON, không thực thi lệnh nhảy mà chạy chương trình gốc;

Gọi chương trình con (CALL)



Nếu X0 ở trạng thái ON, nhảy tới chương trình con
 Nếu cuộn dây không ở trạng thái ON, hãy chạy chương trình gốc;
 Sau khi thực thi chương trình con, quay lại chương trình chính;

Chương trình con sẽ bắt đầu từ Pn và kết thúc bằng SRET. CALL Pn được sử dụng để gọi chương trình con. n là số nguyên nằm trong khoảng từ 0 đến 9999.

Ký hiệu I

Thẻ I thường được sử dụng trong các trường hợp ngắt, bao gồm ngắt ngoài, ngắt bộ định thời, v.v. Nó thường hoạt động cùng với IRET (kết thúc chương trình ngắt), EI (kích hoạt lệnh ngắt), DI (vô hiệu hóa lệnh ngắt);

- Ngắt ngoài

Nhận tín hiệu đầu vào từ các đầu nối đầu vào đặc biệt, không bị ảnh hưởng bởi chu kỳ quét.

Kích hoạt tín hiệu đầu vào, thực thi lệnh ngắt chương trình con

Với lệnh ngắt ngoài, PLC có thể xử lý tín hiệu ngắn hơn chu kỳ quét; Vì vậy, nó có thể được sử dụng làm phương án xử lý ưu tiên thiết yếu trong điều khiển tuần tự hoặc được sử dụng trong điều khiển xung thời gian ngắn.

- Ngắt bộ định thời

Thực thi chương trình con ngắt tại mỗi thời điểm vòng lặp ngắt được chỉ định. Sử dụng lệnh ngắt này trong các phần tử điều khiển khác với chu trình hoạt động của PLC;

- Trình tự hành động của rơle đầu vào/đầu ra và độ trễ phản hồi

Đầu vào

Trước khi PLC thực hiện chương trình, hãy đọc tất cả trạng thái ON/OFF của đầu vào vào vùng hình ảnh. Trong quá trình thực hiện chương trình, dù đầu vào có thay đổi thì nội dung ở vùng ảnh đầu vào cũng không thay đổi. Tuy nhiên, trong chu kỳ quét tiếp theo, những thay đổi sẽ được đọc.

Đầu ra

Sau khi tất cả các lệnh kết thúc, chuyển trạng thái ON/OFF của vùng hình ảnh đầu ra Y sang vùng bộ nhớ khóa (lock memory) của đầu ra. Đây sẽ là đầu ra thực tế của PLC. Các contactor đầu ra sẽ hoạt động theo thời gian trễ phản hồi của thiết bị.

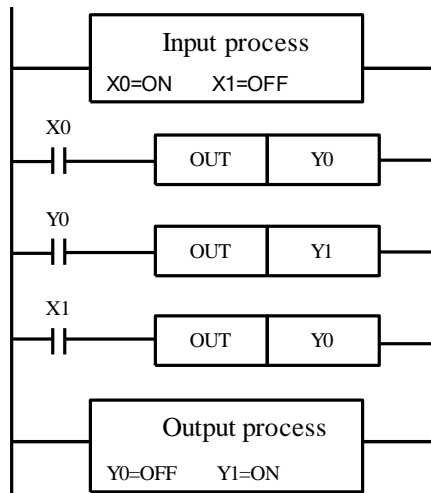
Khi sử dụng chế độ đầu vào/đầu ra hàng loạt, thời gian điều khiển và chu kỳ hoạt động của bộ lọc đầu vào và thiết bị đầu ra cũng sẽ hiển thị độ trễ phản hồi.

- Không nhận tín hiệu xung đầu vào xung hẹp

Thời gian ON/OFF đầu vào của PLC phải dài hơn thời gian vòng lặp của nó. Nếu xem xét độ trễ phản hồi của bộ lọc đầu vào là 10ms, thời gian vòng lặp là 10ms, thì thời gian ON/OFF tối thiểu phải là 20 ms cho từng trạng thái.

Vì vậy, không thể xử lý xung đầu vào lên tới $1.000/(20+20)=25\text{Hz}$. Tuy nhiên, tình trạng này có thể được cải thiện khi sử dụng chức năng đặc biệt của PLC và các lệnh ứng dụng (chẳng hạn như bộ đếm tốc độ cao, ngắt đầu vào, điều chỉnh bộ lọc đầu vào).

- Hành động đầu ra kép (Cuộn dây kép)



Như hình bên trái, hãy xét trường hợp sử dụng cùng một cuộn dây Y0 tại nhiều vị trí:

Ví dụ X0=ON, X1=OFF

Y0 đầu tiên: X0 ở trạng thái ON, vùng hình ảnh của nó ở trạng thái ON, đầu ra Y1 cũng ở trạng thái ON.

Y0 thứ hai: khi đầu vào X1 ở trạng thái OFF, vùng hình ảnh của nó ở trạng thái OFF.

Vậy, đầu ra thực tế là:

Y0=OFF, Y1= ON.

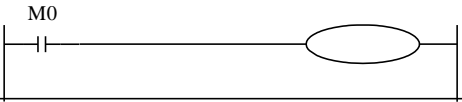
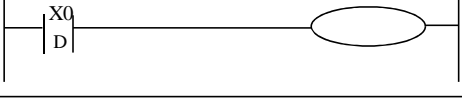
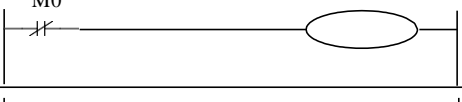
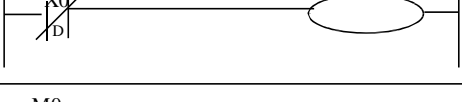
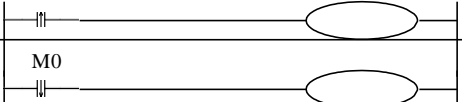
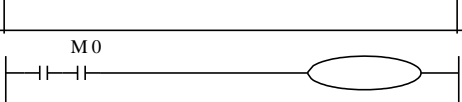
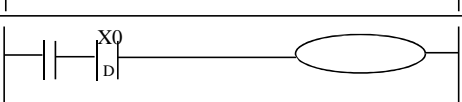
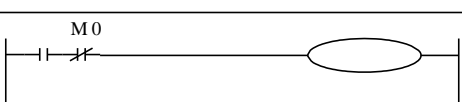
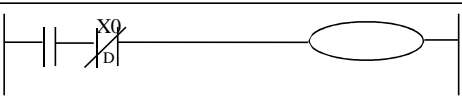
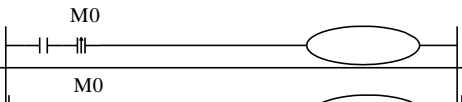
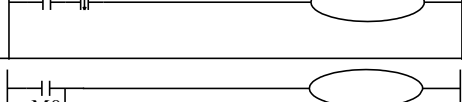
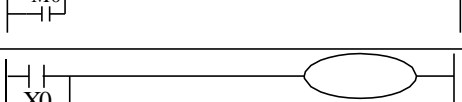
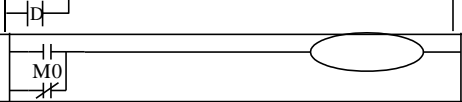


Khi thực thi đầu ra kép (sử dụng cuộn dây kép), cuộn dây sau được ưu tiên hoạt động.

Chương 3: Các lệnh chương trình cơ bản


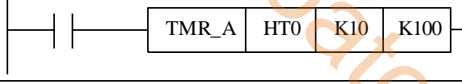



Chương này giới thiệu các lệnh cơ bản và chức năng của chúng.

3-1 Danh sách các lệnh cơ bản

Dòng XD, XL hỗ trợ tất cả các lệnh cơ bản sau :




Tên gọi nhớ	Chức năng	Định dạng và Thiết bị	Chương
LD	Tiếp điểm thao tác logic đầu vào NO (thường mở)		3-2
LDD	Đọc trạng thái trực tiếp từ tiếp điểm		3-6
LDI	Tiếp điểm thao tác logic đầu vào NC (thường đóng)		3-2
LDDI	Đọc trực tiếp tiếp điểm thường đóng		3-6
LDP	Thao tác logic đầu vào-Xung sườn lên		3-5
LDF	Thao tác logic đầu vào-Xung sườn xuống		3-5
AND	Kết nối nối tiếp các tiếp điểm NO (thường mở)		3-3
ANDD	Đọc trạng thái trực tiếp từ tiếp điểm		3-6
ANI	Kết nối nối tiếp các tiếp điểm NC (thường đóng)		3-3
ANDDI	Đọc trạng thái trực tiếp từ tiếp điểm thường đóng		3-6
ANDP	Kết nối nối tiếp xung sườn lên		3-5
ANDF	Kết nối nối tiếp xung sườn xuống		3-5
OR	Kết nối song song các tiếp điểm NO (thường mở)		3-4
ORD	Đọc trạng thái trực tiếp từ tiếp điểm		3-6
ORI	Kết nối song song các tiếp điểm NC (thường đóng)		3-4

ORDI	Đọc trạng thái trực tiếp từ tiếp điểm thường đóng		3-6
ORP	Kết nối song song của xung sườn lên		3-5
ORF	Kết nối song song của xung sườn xuống		3-5
ANB	Kết nối nối tiếp của nhiều mạch song song		3-8
ORB	Kết nối song song của nhiều mạch song song		3-7
OUT	Xuất trạng thái logic phân tử kiểu coil		3-2
OUTD	Xuất đầu ra trực tiếp từ tiếp điểm		3-6
SET	Đặt một thiết bị bit ở trạng thái ON		3-12
RST	Đặt một thiết bị bit ở trạng thái OFF		3-12
CNT	Bộ đếm lên 16-bit không lưu giữ khi tắt nguồn		3-13
CNT_D	Bộ đếm xuống 16-bit có thể lưu giữ khi tắt nguồn		3-13
DCNT	Bộ đếm lên 32-bit không lưu giữ khi tắt nguồn		3-13
DCNT_D	Bộ đếm xuống 32bit có thể lưu giữ khi tắt nguồn		3-13
PLS	Bật chu trình quét khi phát hiện xung sườn lên		3-11
PLF	Bật chu trình quét khi phát hiện xung sườn xuống		3-11
MCS	Kết nối các tiếp điểm nối tiếp chung		3-9
MCR	Xóa các tiếp điểm nối tiếp công cộng		3-9
ALT	Trạng thái của thiết bị được gán bị đảo ngược trong		3-10

	mỗi thao tác lệnh		
TMR	Bộ định thời không lưu giữ khi tắt nguồn		3-14
TMR_A	Bộ định thời có thể lưu giữ khi tắt nguồn		3-14
END	Buộc dừng quá trình quét chương trình hiện tại		3-15
GROUP	Nhóm		3-15
GROUPE	Kết thúc nhóm		3-16

3-2 Lệnh [LD] , [LDI] , [OUT]

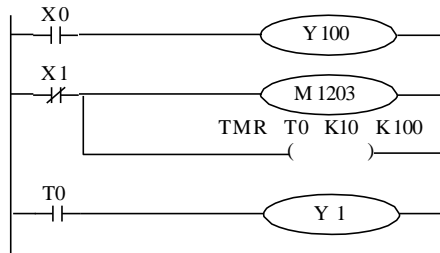
Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Các toán hạng
LD (+)	Tiếp điểm thao tác logic đầu vào NO (thường mở)	 Các toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
LDI (-)	Tiếp điểm thao tác logic đầu vào NC (thường đóng)	 Các thiết bị: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
OUT (OUT)	Loại cuộn dây điều khiển cho thao tác logic cuối cùng	 Các toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

- Kết nối trực tiếp các lệnh LD và LDI với thanh bus bên trái. Nó có thể đi với lệnh ANB và được sử dụng khi bắt đầu nhánh.
- Lệnh OUT có thể điều khiển các role đầu ra, role trung gian, role trạng thái, bộ định thời và bộ đếm. Nhưng lệnh này không thể được sử dụng cho các role đầu vào.

Chương trình

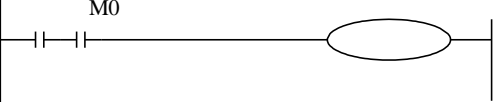
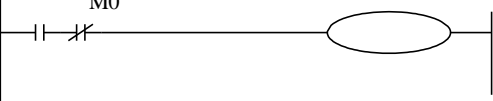


```

LD    X0
OUT   Y100
LDI   X1
OUT   M1203
TMR   T0    K10 K100
LD    T0
OUT   Y1
    
```

3-3 Lệnh [AND] , [ANI]

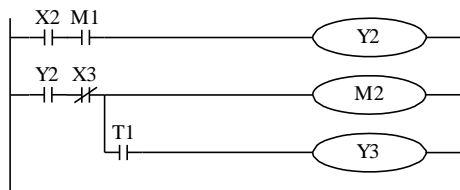
Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và các toán hạng
AND (and)	Mắc nối tiếp các tiếp điểm thường mở	 Các toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ANI (and reverse)	Mắc nối tiếp các tiếp điểm thường đóng	 Các toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

- Sử dụng lệnh AND và ANI để kết nối nối tiếp các công tắc tơ. Không có giới hạn số lượng các contactor được mắc nối tiếp. Các lệnh này có thể được sử dụng nhiều lần.
- Sử dụng lệnh OUT thông qua cuộn dây khác được gọi là đầu ra “follow-on” (Ví dụ, hãy xem chương trình bên dưới: OUT M2 và OUT Y3). Đầu ra follow-on có thể lặp lại miễn là thứ tự đầu ra đúng. Không có giới hạn số lần kết nối nối tiếp các công tắc tơ và đầu ra follow-on.

Chương trình

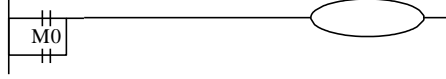
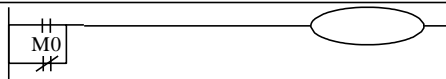


```

LD    X2
AND   M1
OUT   Y2
LD    Y2
ANI   X3
OUT   M2
AND   T1
OUT   Y3
    
```

3-4 Lệnh [OR] , [ORI]

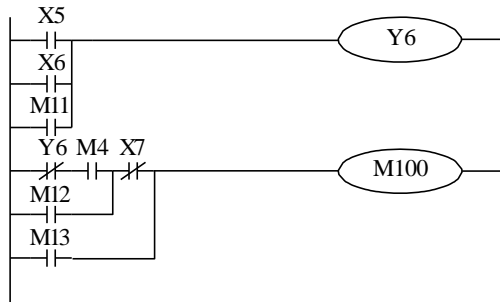
Tên gọi nhớ và Chức năng

Tên gọi nhóe	Chức năng	Định dạng và Các toán hạng
OR (OR)	Kết nối song song các tiếp điểm NO (Thường mở)	 Toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ORI (OR reverse)	Kết nối song song các tiếp điểm NC (Thường đóng)	 Toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

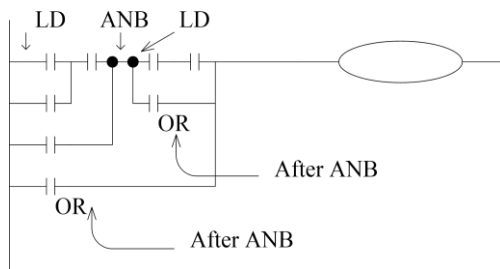
- Sử dụng lệnh OR và ORI để kết nối song song các công tắc tơ. Để kết nối song song một khối chứa nhiều hơn một công tắc tơ được mắc nối tiếp với một khối mạch khác, hãy sử dụng lệnh ORB, lệnh này sẽ được mô tả sau;
- OR và ORI bắt đầu từ bước lệnh, kết nối song song với bước lệnh LD và LDI đã giới thiệu trước đó. Không có giới hạn về số lần kết nối song song.

Chương trình



LD X5
 OR X6
 OR M11
 OUT Y6
 LDI Y6
 AND M4
 OR M12
 ANI X7
 OR M13
 OUT M100

Mối quan hệ với lệnh ANB





Về nguyên tắc, kết nối song song với lệnh OR, ORI phải kết nối với lệnh LD, LDI. Nhưng sau lệnh ANB, bạn vẫn có thể thêm lệnh LD hoặc LDI.

3-5 Lệnh [LDP] , [LDF] , [ANDP] , [ANDF] , [ORP] , [ORF]

Tên gọi nhớ và Chức năng

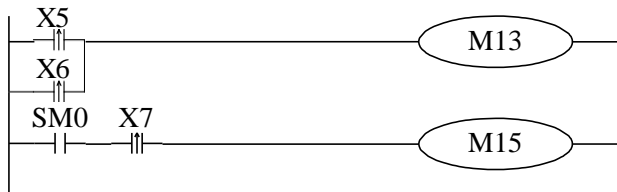
Tên gọi nhớ	Chức năng	Định dạng và các toán hạng
LDP (Load Pulse)	Thao tác logic đầu vào- Xung sườn lên	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
LDF (Load Falling pulse)	Thao tác logic đầu vào- Xung sườn xuống	 M0
ANDP (AND Pulse)	Kết nối nối tiếp xung sườn lên	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ANDF	Kết nối nối tiếp	 M0

(AND Falling pulse)	xung sườn lên	X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ORP (OR Pulse)	Kết nối song song xung sườn lên	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ORF (OR Falling pulse)	Kết nối song song xung sườn xuống	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

- Lệnh LDP, ANDP, ORP sẽ ở trạng thái ON trong khoảng thời gian quét khi sắp có xung sườn lên của tín hiệu (OFF→ON)
- Lệnh LDF, ANDF, ORF sẽ ở trạng thái ON trong khoảng thời gian quét khi sắp có xung sườn xuống của tín hiệu (ON→OFF)

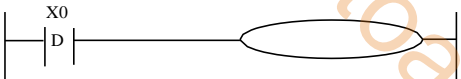
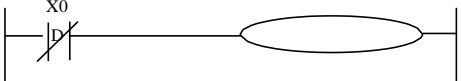
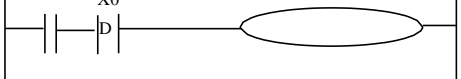
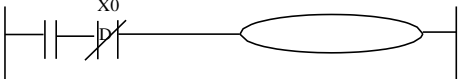
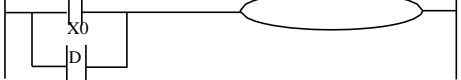
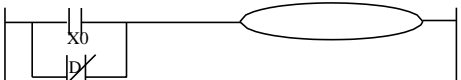
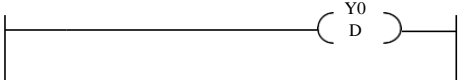
Chương trình



LDP	X5
ORP	X6
OUT	M13
LD	M8000
ANDP	X7
OUT	M15

3-6 Lệnh[LDD] , [LDDI] , [ANDD] , [ANDDI] , [ORD] , [ORDI] , [OUTD]

Tên gọi nhớ và Chức năng

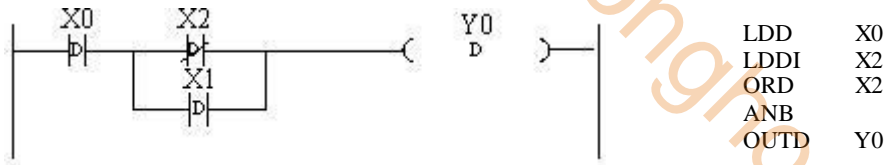
Tên gọi nhớ	Chức năng	Định dạng và Các toán hạng
LDD	Đọc trạng thái trực tiếp từ tiếp điểm	 Thiết bị: X
LDDI	Đọc trạng thái trực tiếp từ tiếp điểm thường đóng	 Thiết bị: X
ANDD	Đọc trạng thái trực tiếp từ tiếp điểm	 Thiết bị: X
ANDDI	Đọc trạng thái trực tiếp từ tiếp điểm thường đóng	 Thiết bị: X
ORD	Đọc trạng thái trực tiếp từ tiếp điểm	 Thiết bị: X
ORDI	Đọc trạng thái trực tiếp từ tiếp điểm thường đóng	 Thiết bị: X
OUTD	Xuất đầu ra trực tiếp	 Thiết bị: Y

Mô tả

Chức năng của các lệnh LDD, ANDD, ORD tương tự như LD, AND, OR; Các lệnh LDDI, ANDDI, ORDI tương tự như các lệnh LDI, ANDI, ORI; nhưng nếu toán hạng là X, các lệnh LDD, ANDD, ORD sẽ đọc tín hiệu trực tiếp từ các đầu nối.


OUTD và OUT là các lệnh đầu ra. OUTD sẽ xuất ra ngay lập tức khi điều kiện được thỏa mãn, không cần đợi chu kỳ quét tiếp theo.

Chương trình



3-7 Lệnh [ORB]

Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và thiết bị
ORB (OR Block)	Kết nối song song các mạch nối tiếp	 Thiết bị: Không

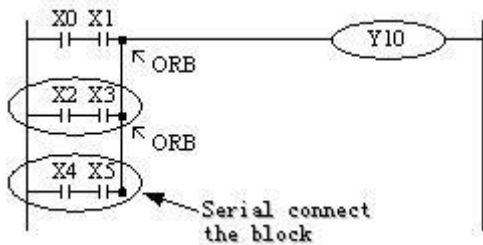
Mô tả

Hai hoặc nhiều công tắc tơ được gọi là "khởi nối tiếp". Nếu kết nối song song khởi nối tiếp, sử dụng lệnh LD, LDI tại điểm bắt đầu nhánh, sử dụng lệnh ORB tại điểm cuối nhánh;

Giống như lệnh ANB, lệnh ORB là lệnh độc lập không liên quan đến bất kỳ phần tử lập trình nào.

Không có giới hạn về số lượng mạch song song khi sử dụng lệnh ORB cho mỗi mạch nối tiếp.

Chương trình



Phương pháp lập trình tốt được khuyến nghị:


```
LD X0
AND X1
LD X2
AND X3
ORB
LD X4
AND X5
ORB
OUT Y10
```

Phương pháp lập trình được khuyến nghị:

```
LD X0
AND X1
LD X2
AND X3
LD X4
AND X5
ORB
ORB
OUT Y10
```

3-8 Lệnh [ANB]

Tên gọi nhớ và Chức năng

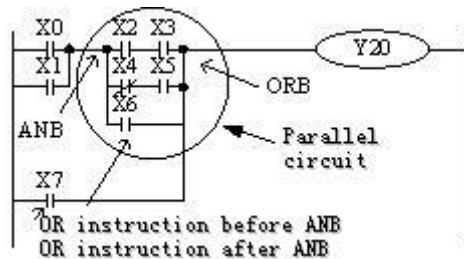
Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
ANB (And Block)	Kết nối nối tiếp của mạch song song	 Thiết bị: Không

Mô tả

Sử dụng lệnh ANB để nối tiếp hai mạch song song. Sử dụng lệnh LD, LDI tại điểm đầu nhánh; sử dụng lệnh ANB tại điểm cuối nhánh.

Không có giới hạn về số lần sử dụng lệnh ANB.


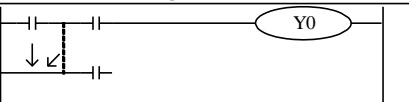
Chương trình



```
LD    X0
OR    X1
LD    X2
AND   X3
LDI   X4
AND   X5
ORB
OR    X6
ANB
OR    X7
OUT   Y20
```

3-9 Lệnh [MCS], [MCR]

Tên gọi nhớ và Chức năng

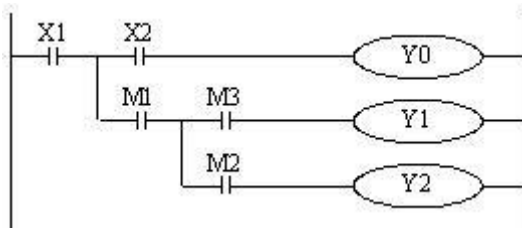
Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
MCS (Master Control)	Bắt đầu đường bus mới	 Thiết bị: Không
MCR (Master control Reset)	Đặt lại đường bus	 Thiết bị: Không

Mô tả

- Sau khi thực thi lệnh MCS, đường bus (LD, LDI) di chuyển đến điểm sau lệnh MCS. Lệnh MCR sẽ đặt lại lệnh này về đường bus ban đầu.
- Các lệnh MCS, MCR nên sử dụng theo cặp.
- Đường bus có thể được lồng vào nhau. Sử dụng các lệnh MCS, MCR giữa các lệnh MCS, MCR. Mức độ lồng nhau tăng lên khi sử dụng lệnh MCS. Mức lồng tối đa là mười. Khi thực thi lệnh MCR, quay trở lại mức cuối cùng của đường bus.
- Khi sử dụng chương trình lồng, quản lý đường bus chỉ có thể được sử dụng trong cùng một luồng. Khi luồng kết thúc, nó phải quay trở lại đường bus chính.

Lưu ý: Các lệnh MCS và MCR không thể được ghi trực tiếp vào sơ đồ bậc thang của phần mềm lập trình PLC dòng XD/XL. Chúng có thể được xây dựng bằng các đường ngang và dọc.

Chương trình



```

LD      X1
MCS
OUT     Y0
LD      X2
MCS
OUT     Y1
LD      M1
MCS
OUT     Y2
MCR
MCR
    
```

3-10 Lệnh [ALT]

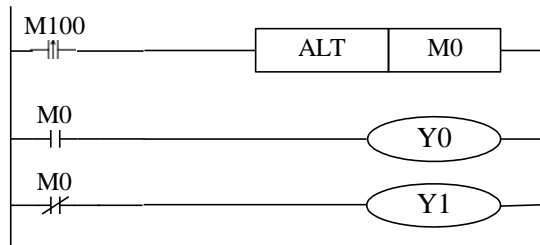
Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
ALT (Đảo)	Đảo trạng thái cuộn dây	<p>Cuộn dây: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m</p>

Mô tả

Trạng thái của cuộn dây bị đảo ngược sau khi sử dụng lệnh ALT (ON chuyển thành OFF, OFF chuyển thành ON).

Chương trình



LDP M100
ALT M0
LD M0
OUT Y0
LDI M0
OUT Y1

3-11 Lệnh [PLS], [PLF]

Tên gọi nhớ và Chức năng

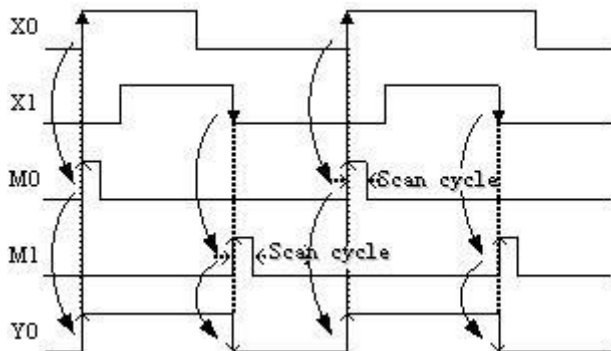
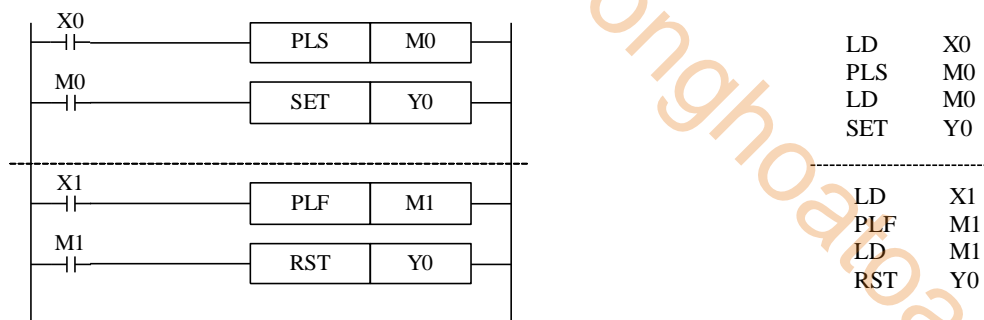
Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
PLS (Xung sườn lên)	Bật chu trình quét khi có xung sườn lên	 Toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
PLF (Xung sườn xuống)	Bật chu trình quét khi có xung sườn xuống	 Toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

Đối với sử dụng lệnh PLS: phần tử lập trình Y và M sẽ hoạt động trong khoảng thời gian quét ngay khi chuyển từ trạng thái OFF sang trạng thái ON.

Đối với sử dụng lệnh PLF: phần tử lập trình Y và M sẽ hoạt động trong khoảng thời gian quét ngay sau khi chuyển từ trạng thái ON sang trạng thái OFF

Chương trình



3-12 Lệnh [SET], [RST]

Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
SET (Set)	Set a bit device permanently ON	 Toán hạng: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
RST (Reset)	Reset a bit device permanently OFF	 Toán hạng : X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

Mô tả

Trong chương trình tiếp theo, Y0 sẽ tiếp tục ở trạng thái ON ngay cả khi X10 ở trạng thái OFF sau khi chuyển sang trạng thái OFF. Y0 sẽ không ở trạng thái ON ngay cả khi X11 OFF sau khi ON. Điều này cũng tương tự với S và M.

Lệnh SET và RST có thể được sử dụng nhiều lần cho cùng một phần tử lập trình. Bất kỳ thứ tự nào cũng được, nhưng chỉ lệnh cuối cùng có hiệu lực.

Lệnh RST có thể được sử dụng để đặt lại bộ đếm, bộ định thời và công tắc tơ.

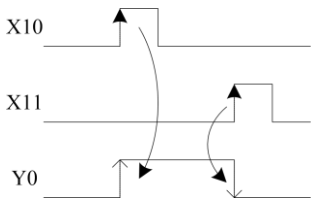
Khi sử dụng lệnh SET hoặc RST, nó không thể sử dụng cùng phần tử lập trình với lệnh OUT.

Chương trình

X10	Y0
	(S)
X11	Y0
	(R)
X12	M50
	(S)
X13	M50
	(R)
X14	S0
	(S)
X15	S0
	(R)
X16	TMR T250 K10 K10
	(R)
X17	T250
	(R)

```



LD X10
SET Y0
LD X11
RST Y0
LD X12
SET M50
LD X13
RST M50
LD X14
SET S0
LD X15
RST S0
LD X16
TMR T250 K10 K10
LD X17
RST T250
    
```



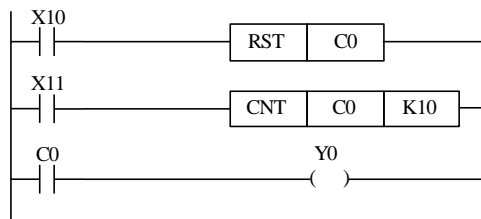
3-13 Lệnh [CNT],[CNT_D],[DCNT],[DCNT_D],[RST] cho bộ đếm

Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
CNT Output	Bộ đếm lên 16 bit không lưu giữ khi tắt nguồn, điều khiển trạng thái logic của đầu ra bộ đếm	<div style="text-align: center;"> </div> <p>Toán hạng: K, D</p>
CNT_D Output	Bộ đếm xuống lên 16 bit có thể lưu giữ khi tắt nguồn, điều khiển trạng thái logic của đầu ra bộ đếm	<div style="text-align: center;"> </div> <p>Toán hạng: K, D</p>
DCNT Output	Bộ đếm lên 32 bit không lưu giữ khi tắt nguồn, điều khiển trạng thái logic của đầu ra bộ đếm	<div style="text-align: center;"> </div> <p>Toán hạng: K, D</p>

DCNT_D Output	Bộ đếm xuống lên 32 bit có thể lưu giữ khi tắt nguồn, điều khiển trạng thái logic của đầu ra bộ đếm	 Toán hạng: K, D
RST Reset	Đặt lại cuộn dây đầu ra, xóa giá trị đếm hiện tại	 Toán hạng: C, HC, HSC

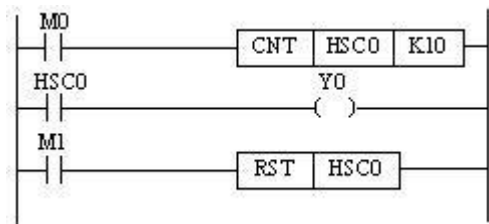
Lập trình bộ đếm bên trong



Bộ đếm tăng C0 đếm số lần X11 chuyển từ OFF thành ON. Khi C0 đạt K10 thì C0 sẽ chuyển từ OFF sang ON. Khi X11 chuyển từ OFF sang ON, giá trị hiện tại C0 sẽ tiếp tục tăng và cuộn dây C0 vẫn ở trạng thái ON. Khi X10 ở trạng thái ON, hãy đặt lại cuộn C0.

Bộ đếm có thể lưu giữ khi tắt nguồn sẽ giữ giá trị hiện tại và trạng thái cuộn dây của bộ đếm khi tắt nguồn

Lập trình bộ đếm tốc độ cao



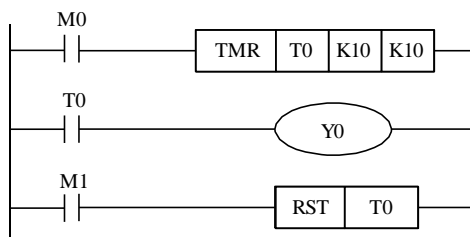
Bộ đếm tăng đếm số lần chuyển OFF thành ON của M0. Khi giá trị đếm đạt đến giá trị cài đặt (giá trị K hoặc D), cuộn dây của bộ đếm sẽ ở trạng thái ON. Khi M1 ở trạng thái ON, cuộn dây bộ đếm của HSC0 được đặt lại, giá trị hiện tại trở thành 0.

3-14 Lệnh [TMR], [TMR_A] cho bộ định thời

Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
TMR output	Bộ định thời 100ms không lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: K, D
TMR output	Bộ định thời 10ms không lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: K, D
TMR output	Bộ định thời 1ms không lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: K, D
TMR_A output	Bộ định thời 100ms có thể lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: K, D
TMR_A output	Bộ định thời 10ms có thể lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: K, D
TMR_A output	Bộ định thời 1ms có thể lưu giữ khi tắt nguồn, biến tần của cuộn dây	 Toán hạng: C, HC, HSC

Lập trình cho bộ định thời bên trong

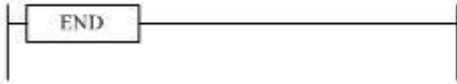


Khi M0 ở trạng thái ON, T0 bắt đầu tính thời gian. Khi T0 đạt đến K10, cuộn dây T0 ở trạng thái ON. Sau đó T0 tiếp tục tính thời gian. Khi M1 ở trạng thái ON, hãy đặt lại T0.

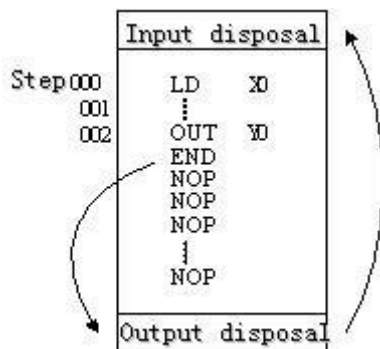
Bộ định thời có thể lưu giữ khi tắt nguồn sẽ lưu giữ giá trị hiện tại và trạng thái cuộn dây bộ đếm khi tắt nguồn.

3-15 Lệnh [END]

Tên gọi nhớ và Chức năng

Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
END (END)	Buộc quá trình quét chương trình hiện tại kết thúc	 Thiết bị: Không

Mô tả



PLC liên tục thực thi việc xử lý đầu vào, thực thi chương trình và xử lý đầu ra. Nếu viết lệnh END vào cuối chương trình thì các lệnh phía sau lệnh END sẽ không được thực thi. Nếu không có lệnh END trong chương trình, PLC sẽ thực hiện bước kết thúc và sau đó lặp lại việc thực thi chương trình từ bước 0.

Khi gỡ lỗi, hãy chèn lệnh END vào từng phân đoạn chương trình để kiểm tra hành động của từng chương trình.

Sau đó, sau khi xác nhận việc sửa hành động của khối trước đó, hãy xóa lệnh END.

Ngoài ra, lần thực thi lệnh RUN đầu tiên bắt đầu bằng lệnh END.

Khi thực thi lệnh END, hãy làm mới bộ định thời. (Kiểm tra xem chu kỳ quét có kéo dài không.)

3-16 Lệnh [GROUP], [GROUPE]

Tên gọi nhớ và Chức năng

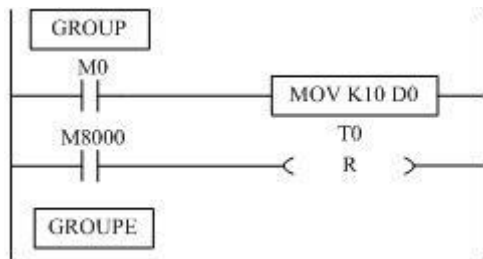
Tên gọi nhớ	Chức năng	Định dạng và Thiết bị
GROUP	NHÓM	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GROUP</div> Thiết bị: Không
GROUPE	KẾT THÚC NHÓM	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GROUPE</div> Thiết bị: Không

Mô tả

Lệnh GROUP và GROUPE nên sử dụng theo cặp.

Lệnh GROUP và GROUPE không có ý nghĩa thực tế; chúng được sử dụng để tối ưu hóa cấu trúc chương trình. Vì vậy, việc thêm hoặc xóa các lệnh này không ảnh hưởng đến việc chạy chương trình;

Phương thức sử dụng lệnh GROUP và GROUPE tương tự như các lệnh luồng (flow instructions); nhập lệnh GROUPE vào đầu phần nhóm; nhập lệnh GROUP ở cuối phần nhóm.



Thông thường, lệnh GROUP và GROUPE có thể được lập trình theo chức năng của nhóm. Trong khi đó, các lệnh được lập trình có thể được CUỘN LẠI (ẩn đi trên màn hình) hoặc HIỆN THỊ TRÊN MÀN HÌNH LẬP TRÌNH. Đối với một dự án dự phòng (redundant project), hai lệnh này khá hữu ích.

3-17 Lưu ý khi lập trình

Cấu trúc liên kết và bước lệnh

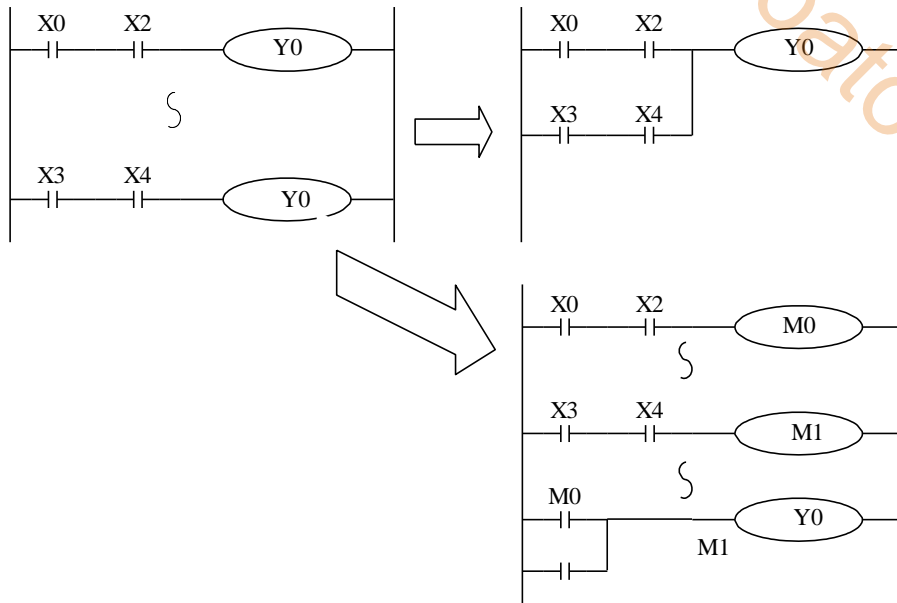
Ngay cả trong mạch điều khiển tuần tự có chức năng tương tự cũng có thể đơn giản hóa chương trình và rút ngắn các bước chương trình theo cấu trúc của contactor. Nguyên tắc lập trình chung là: (a) viết mạch có nhiều tiếp điểm nối tiếp ở trên cùng; (b) Viết mạch có nhiều công tắc tơ song song ở bên trái.

Trình tự thực thi chương trình

Xử lý chương trình điều khiển tuần tự bằng cách **【From top to bottom】** và **【From left to right】**. Các lệnh điều khiển tuần tự cũng được mã hóa theo trình tự này.

Nếu tiếp tục đầu ra kép của cuộn dây (cuộn dây kép) trong chương trình điều khiển tuần tự thì hành động cuối cùng sẽ được thực thi trước.

Đầu ra kép (cuộn dây kép) không đi ngược với quy tắc đầu vào. Nhưng vì thao tác trước rất phức tạp nên vui lòng sửa đổi chương trình như trong ví dụ sau.

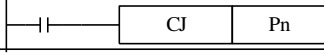

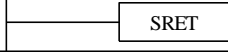

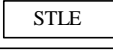
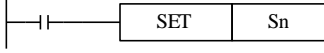
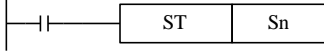

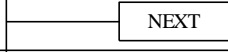
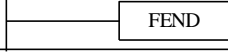
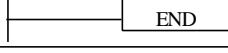
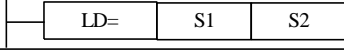
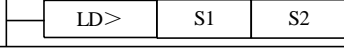
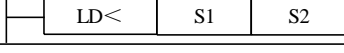
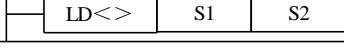
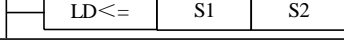
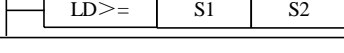
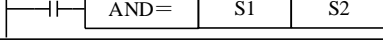
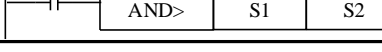




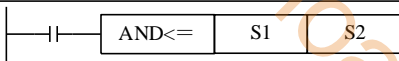
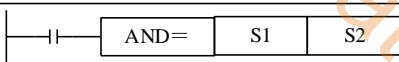
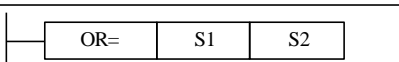
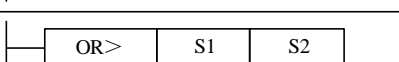

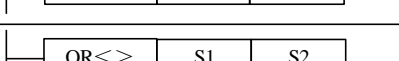
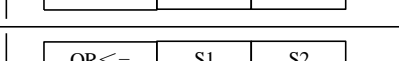
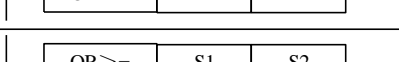
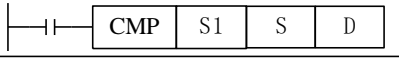

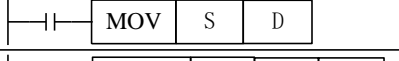
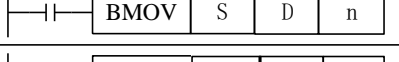
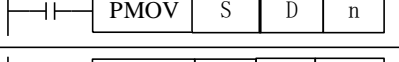
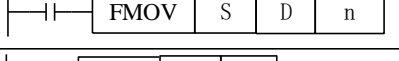
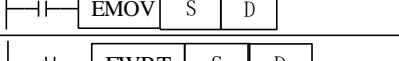
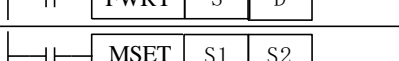
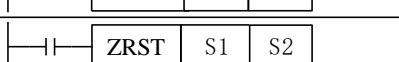
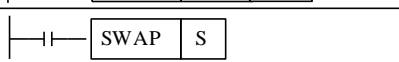
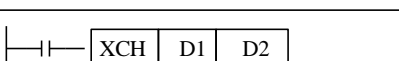
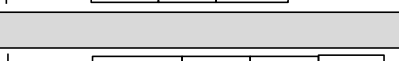
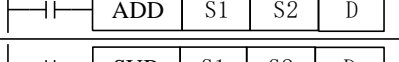
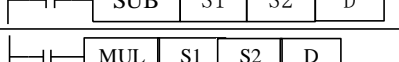
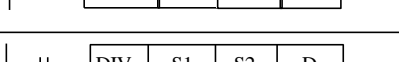
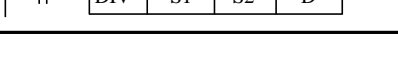
Có những phương pháp khác. Ví dụ. Lệnh nhảy hoặc lệnh luồng.

Chương 4: Các lệnh ứng dụng


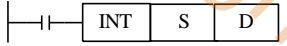




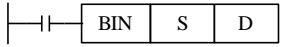
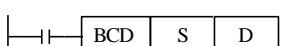
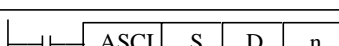
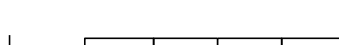
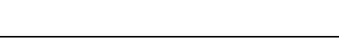
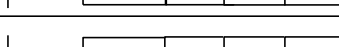
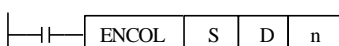
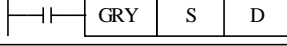
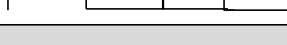
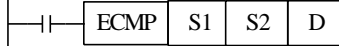
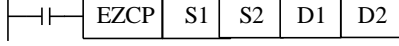
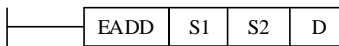
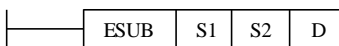

Trong chương này, chúng tôi mô tả chức năng của các lệnh ứng dụng của PLC dòng XD, XL.

4-1 Danh sách các lệnh ứng dụng

Tên gọi nhớ	Chức năng	Sơ đồ bậc thang	Chương
Luồng chương trình			
CJ	Nhảy có điều kiện		4-3-1
CALL	Gọi chương trình con SRET		4-3-2
	Quay lại chương trình con		4-3-2
STL	Bắt đầu luồng		4-3-3
STLE	Kết thúc luồng		4-3-3
SET	Mở luồng được chỉ định, đóng luồng hiện tại		4-3-3
ST	Mở luồng được chỉ định, không đóng luồng hiện tại		4-3-3
FOR	Bắt đầu vòng lặp FOR-NEXT		4-3-4
NEXT	Kết thúc vòng lặp FOR-NEXT		4-3-4
FEND	Kết thúc chương trình chính		4-3-5
END	Kết thúc chương trình		4-3-5
Các lệnh so sánh dữ liệu			
LD=	Lệnh LD kích hoạt nếu (S1) = (S2)		4-4-1
LD>	Lệnh LD kích hoạt nếu (S1) > (S2)		4-4-1
LD<	Lệnh LD kích hoạt khi (S1) < (S2)		4-4-1
LD<>	Lệnh LD kích hoạt nếu (S1) ≠ (S2)		4-4-1
LD<=	Lệnh LD kích hoạt nếu (S1) ≤ (S2)		4-4-1
LD>=	Lệnh LD kích hoạt nếu (S1) ≥ (S2)		4-4-1
AND=	Lệnh AND kích hoạt (S1) = (S2)		4-4-2
AND>	Lệnh AND kích hoạt (S1) > (S2)		4-4-2

AND<	Lệnh AND kích hoạt nếu $(S1) < (S2)$		4-4-2
AND<>	Lệnh AND kích hoạt nếu $(S1) \neq (S2)$		4-4-2
AND<=	Lệnh AND kích hoạt nếu $(S1) \leq (S2)$		4-4-2
AND>=	Lệnh AND kích hoạt nếu $(S1) \geq (S2)$		4-4-2
OR=	Lệnh OR kích hoạt nếu $(S1) = (S2)$		4-4-3
OR>	Lệnh OR kích hoạt nếu $(S1) > (S2)$		4-4-3
OR<	Lệnh OR kích hoạt nếu $(S1) < (S2)$		4-4-3
OR<>	Lệnh OR kích hoạt nếu $(S1) \neq (S2)$		4-4-3
OR<=	Lệnh OR kích hoạt nếu $(S1) \leq (S2)$		4-4-3
OR>=	Lệnh OR kích hoạt nếu $(S1) \geq (S2)$		4-4-3
Các lệnh truyền dữ liệu			
CMP	So sánh dữ liệu		4-5-1
ZCP	So sánh dữ liệu trong vùng nhất định		4-5-2
MOV	Chuyển dữ liệu		4-5-3
BMOV	Chuyển khối dữ liệu		4-5-4
PMOV	Chuyển khối dữ liệu		4-5-5
FMOV	Chuyển dữ liệu đến nhiều thanh ghi dữ liệu		4-5-6
EMOV	Chuyển số thực dấu phẩy động		4-5-7
FWRT	Ghi vào Flash ROM		4-5-8
MSET	Đặt Vùng		4-5-9
ZRST	Đặt lại vùng		4-5-10
SWAP	Hoán đổi byte cao và thấp		4-5-11
XCH	Hoán đổi 2 giá trị		4-5-12
Các lệnh tính toán dữ liệu số học			
ADD	Cộng		4-6-1
SUB	Trừ		4-6-2
MUL	Nhân		4-6-3
DIV	Chia		4-6-4

INC	Tăng giá trị		4-6-5
DEC	Giảm giá trị		4-6-5
MEAN	Lấy giá trị trung bình		4-6-6
WAND	Word And		4-6-7
WOR	Word OR		4-6-7
WXOR	Word eXD3lusive OR		4-6-7
CML	Đảo trạng thái		4-6-8
NEG	Lấy dấu -/Đổi dấu -		4-6-9
Lệnh dịch chuyển dữ liệu			
SHL	Dịch số học sang trái		4-7-1
SHR	Dịch số học sang phải		4-7-1
LSL	Dịch logic sang trái		4-7-2
LSR	Dịch logic sang phải		4-7-2
ROL	Xoay trái		4-7-3
ROR	Xoay phải		4-7-3
SFTL	Dịch bit sang trái		4-7-4
SFTR	Dịch bit sang phải		4-7-5
WSFL	Dịch word sang trái		4-7-6
WSFR	Dịch word sang phải		4-7-7
Lệnh chuyển đổi dữ liệu			
WTD	Lệnh chuyển số nguyên 16 bit thành số nguyên 32 bit		4-8-1
DWTD	Lệnh chuyển số nguyên 32 bit sang số nguyên 64bit		4-8-1
BDWTD	Lệnh chuyển 32 bits integer sang 64 bits integer (chuyển hàng loạt)		4-8-2
FLT	Lệnh chuyển số nguyên 16 bit sang số thực dấu phẩy động		4-8-3
DFLT	Lệnh chuyển 32 bits integer sang số thực dấu phẩy động		4-8-3
FLTD	Lệnh chuyển 64 bits integer sang số thực dấu phẩy động		4-8-3
DFLTD	Lệnh chuyển 32 bits integer sang số thực dấu phẩy động có độ chính xác kép		4-8-4

QFLTD	Lệnh chuyển 64 bits integer sang số thực dấu phẩy động có độ chính xác kép		4-8-4
INT	Lệnh chuyển số thực dấu phẩy động sang số nguyên		4-8-5
DINTD	Lệnh chuyển double - precision floating point sang 32 bits integer		4-8-6
QINTD	Lệnh chuyển double - precision floating point sang 64 bits integer		4-8-6
ECON	Lệnh chuyển single precision floating point sang double precision floating point		4-8-7
BECON	Lệnh chuyển single precision floating point sang double precision floating point (chuyển hàng loạt)		4-8-8
BIN	Lệnh chuyển BCD sang nhị phân		4-8-9
BCD	Lệnh chuyển nhị phân sang BCD		4-8-10
ASCI	Lệnh chuyển Hex. Sang ASCII		4-8-11
HEX	Lệnh chuyển ASCII sang Hex.		4-8-12
DECO	Lệnh giải mã		4-8-13
ENCO	Lệnh mã hóa bit cao		4-8-14
ENCOL	Lệnh mã hóa bit thấp		4-8-15
GRY	Lệnh chuyển Nhị phân sang mã Gray		4-8-16
GBIN	Lệnh chuyển mã Gray sang nhị phân		4-8-17
Các lệnh với số thực dấu phẩy động			
ECMP	Lệnh so sánh số thực dấu phẩy động		4-9-1
EZCP	Lệnh so sánh vùng số thực dấu phẩy động		4-9-2
EADD	Lệnh phép cộng		4-9-3
ESUB	Lệnh phép trừ		4-9-4
EMUL	Lệnh phép nhân		4-9-5

EDIV	Phép chia		4-9-6
ESQR	Căn bậc 2		4-9-7
SIN	Tính Sin		4-9-8
COS	Tính Cos		4-9-9
TAN	Tính Tan		4-9-10
ASIN	Tính arcsin		4-9-11
ACOS	Tính arccos		4-9-12
ATAN	Tính arctan		4-9-13
Lệnh phép toán thời gian			
TRD	Đọc dữ liệu RTC		4-10-1
TWR	Ghi dữ liệu RTC		4-10-2
MOV	Đọc dữ liệu từ bo mạch BD thời gian thực		4-10-3
TO	Ghi dữ liệu thời gian xuống bo mạch BD thời gian thực		4-10-4
TADD	Cộng dữ liệu đồng hồ		4-10-5
TSUB	Trừ dữ liệu đồng hồ		4-10-6
HTOS	Chuyển đổi dữ liệu giờ, phút và giây thành giây		4-10-7
STOH	Chuyển đổi dữ liệu giây thành giờ, phút và giây		4-10-8
TCMP	So sánh thời gian (giờ, phút, giây)		4-10-9
DACMP	So sánh ngày tháng (năm, tháng, ngày)		4-10-10

4-2 Phương pháp đọc các lệnh ứng dụng

Trong sách hướng dẫn này, các lệnh áp dụng được mô tả như sau.

1) Sơ lược

PHÉP CỘNG [ADD]			
16 bits	ADD	32 bits	DADD
Điều kiện thực thi	Thường hở/đóng, Sườn lên/xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Các toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Chỉ định dữ liệu hoặc địa chỉ thanh ghi	16 bits/32 bits, BIN
S2	Chỉ định dữ liệu hoặc địa chỉ thanh ghi	16 bits/32 bits, BIN
D	Chỉ định thanh ghi để lưu trữ kết quả tổng	16 bits/32 bits, BIN

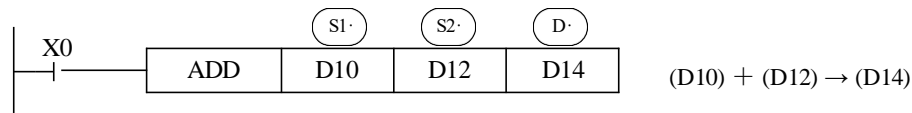
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng Bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•	•	•	•		•	•	•										

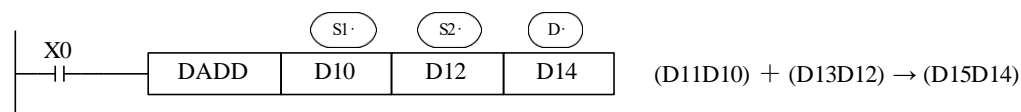
*Lưu ý: D bao gồm D, HD. TD bao gồm TD, HTD. CD bao gồm CD, HCD, HSCD, HSD. DM bao gồm DM, DHM. DS bao gồm DS, DHS. M bao gồm M, HM, SM. S bao gồm S và HS. T bao gồm T và HT. C bao gồm C và HC.

Mô tả

<Lệnh 16 bits >



<Lệnh 32 bits >



- Hai dữ liệu nguồn thực hiện phép cộng nhị phân và dữ liệu kết quả được lưu trữ trong địa chỉ đối tượng.
- Bit cao nhất của mỗi dữ liệu là bit + (0) và bit - (1). Những dữ liệu này sẽ thực hiện phép tính cộng thông qua đại số. Chẳng hạn như $5 + (-8) = -3$.
- Nếu kết quả tính toán là "0", cờ "0" sẽ có hiệu lực. Nếu kết quả vượt quá 323.767 (phép toán 16 bit) hoặc 2.147.483.648 (phép toán 32 bit), cờ mang sẽ hoạt động. (tham khảo trang tiếp theo). Nếu kết quả vượt quá -323.768 (phép toán 16 bit) hoặc -2.147.483.648 (phép toán 32 bit), cờ mượn (borrow flag) sẽ có hiệu lực (Tham khảo trang tiếp theo).
- Khi thực hiện thao tác 32 bit, 16 bit thấp của thanh ghi 32 bit được gán, địa chỉ thanh ghi gần với thanh ghi 16 bit thấp sẽ được gán cho 16 bit cao của thanh ghi 32 bit. Số chẵn được khuyến nghị để gán địa chỉ thanh ghi 16 bit thấp.
- Nguồn và đối tượng có thể có cùng địa chỉ thanh ghi.
- Trong ví dụ trên, khi X0 được ở trạng thái ON, phép cộng sẽ được thực hiện trong mỗi chu kỳ quét.

Các cờ liên quan

Cờ	Tên	Chức năng
SM20	Cờ 0	ON: kết quả tính toán bằng 0 OFF: kết quả tính toán không bằng 0
SM21	Cờ mượn	ON: kết quả tính toán lớn hơn 32767(16bit) hoặc 2147483647(32bit) OFF: kết quả tính toán không quá 32767(16bit) hoặc 2147483647(32bit)
SM22	Cờ nhớ	ON: kết quả tính toán lớn hơn 32767(16bit) hoặc 2147483647(32bit) OFF: kết quả tính toán không quá 32767(16bit) hoặc 2147483647(32bit)

Lưu ý

- Gán dữ liệu

Thanh ghi dữ liệu của PLC dòng XD, XL là thanh ghi dữ liệu kiểu single word (16 bit), dữ liệu kiểu single word chỉ chiếm một thanh ghi được sử dụng cho lệnh single word. Phạm vi xử lý là số thập phân -327,68~327,67 hoặc hex 0000~FFFF.

Lệnh cho đối tượng dạng Single word D(NUM)



Kiểu dữ liệu double word (32 bit) chiếm hai thanh ghi dữ liệu; địa chỉ của hai thanh ghi là liên tục. Phạm vi xử lý là: thập phân -214,748,364,8~214,748,364,7 hoặc hex 00000000~FFFFFFFF.

Lệnh cho đối tượng dạng double word D(NUM+1) D(NUM)



- Cách biểu diễn lệnh 32 bit

Thêm chữ “D” trước lệnh 16 bit để biểu diễn lệnh 32 bit.

Ví dụ:

ADD D0 D2 D4 Lệnh 16 bit

DADD D10 D12 D14 Lệnh 32 bit

※1: Nó hiển thị cờ báo trạng thái dạng bit sau hành động lệnh.

※2: (S) Toán hạng nguồn sẽ không thay đổi khi lệnh hoạt động

※3: (D) Toán hạng đích sẽ thay đổi khi lệnh hoạt động

※4: Nó giới thiệu hành động cơ bản của lệnh, cách sử dụng, ví dụ áp dụng, chức năng mở rộng, các mục ghi chú, v.v.

4-3 Các lệnh luồng chương trình

Tên gọi nhớ	Tên lệnh	Chương
CJ	Lệnh nhảy có điều kiện	4-3-1
CALL	Lệnh gọi chương trình con	4-3-2
SRET	Trở về từ chương trình con	4-3-2
STL	Bắt đầu luồng	4-3-3
STLE	Kết thúc luồng	4-3-3
SET	Mở luồng được chỉ định, đóng luồng hiện tại (nhảy luồng)	4-3-3
ST	Mở luồng được chỉ định, không đóng luồng hiện tại (Mở luồng mới)	4-3-3
FOR	Bắt đầu vòng lặp FOR-NEXT	4-3-4
NEXT	Kết thúc vòng lặp FOR-NEXT	4-3-4
FEND	Kết thúc đoạn chương trình chính (chương trình đầu tiên)	4-3-5
END	Kết thúc chương trình	4-3-5

4-3-1 Lệnh nhảy có điều kiện [CJ]

1) Sơ lược

Là lệnh thực thi một phần chương trình, CJ rút ngắn chu kỳ hoạt động và tránh sử dụng cuộn dây kép

Lệnh nhảy có điều kiện [CJ]			
16 bits	CJ	32 bits	-
Điều kiện thực thi	Cuộn dây Thường hở/đóng	Model phù hợp	XD, XL
Yêu cầu phân cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

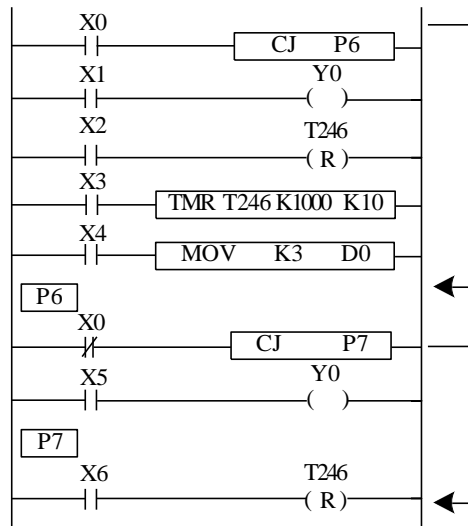
Toán hạng	Chức năng	Kiểu dữ liệu
Pn	Nhảy tới mục tiêu (với con trỏ Nr.) P (P0~P9999)	Pointer's Nr.

3) Phân tử lập trình phù hợp

Khác	Pointer (Con trỏ)	
	P	I
	●	

Mô tả

Trong biểu đồ bên dưới, nếu X0 ở trạng thái ON, hãy chuyển từ bước đầu tiên sang bước tiếp theo sau thẻ P6. Nếu X0 ở trạng thái OFF, không thực thi lệnh nhảy;



- Trong biểu đồ bên trái, Y0 trở thành đầu ra cuộn dây kép, nhưng khi X0=OFF, X1 kích hoạt; khi X0=ON, X5 kích hoạt
- CJ không thể nhảy từ STL này sang STL khác;
- Sau khi điều khiển bộ định thời T0~T575, HT0~HT795 và HSC0~HSC30, nếu thực thi CJ, tiếp tục hoạt động, đầu ra sẽ kích hoạt.
- Thẻ (Tag) phải khớp khi sử dụng lệnh CJ.

4-3-2 Lệnh Gọi chương trình con [CALL] và Trở về từ chương trình con[SRET]

1) Sơ lược

Gọi các chương trình con thực hiện cùng nhau, giảm số bước của chương trình;

Gọi chương trình con [CALL]			
16 bits	CALL	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn logic lên/xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Subroutine Return [SRET]			
16 bits	SRET	32 bits	-
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

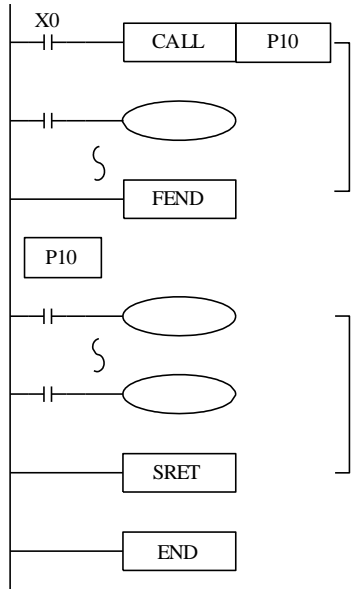
2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
Pn	Jump to the target (with pointer No.) P (P0~P9999)	Pointer's No.

3) Phân tử lập trình phù hợp

Khác	Con trỏ Pointer	
	P	I
	●	

Mô tả

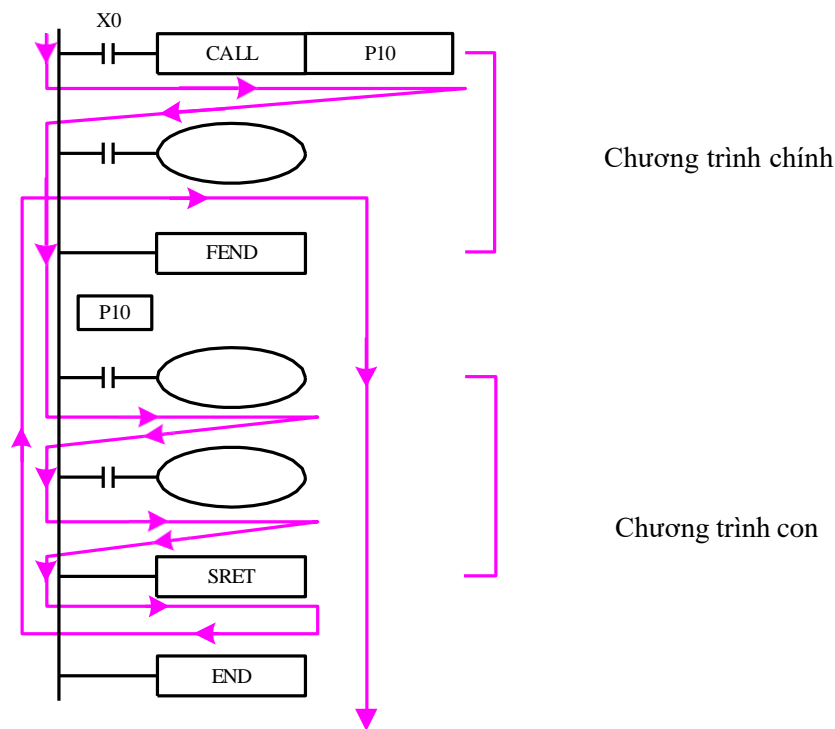


Main

Subroutine

- Nếu X0= ON, thực thi lệnh gọi và nhảy tới P10. Sau khi thực thi chương trình con, quay lại bước ban đầu thông qua lệnh SRET.
- Lập trình thế với lệnh FEND (sẽ mô tả lệnh này sau)
- Trong chương trình con được phép gọi 9 lần, vì vậy tổng cộng có thể có 10 cấp lồng nhau.
- Khi gọi chương trình con, tất cả các bộ định thời, lệnh OUT, PLS, PLF của chương trình chính sẽ giữ nguyên trạng thái.
- Tất cả các lệnh OUT, PLS, PLF, bộ định thời của chương trình con sẽ giữ nguyên trạng thái khi trở về từ chương trình con.
- Không ghi xung, bộ đếm hoặc bộ định thời bên trong chương trình con không thể hoàn thành trong khoảng thời gian quét.

Sơ đồ thực thi chương trình con:



Nếu X0=ON, chương trình sẽ thực thi theo mũi tên.
 Nếu X0=OFF, lệnh CALL sẽ không hoạt động; chỉ có chương trình chính hoạt động.
 Những lưu ý khi viết chương trình con:

Vui lòng lập trình thế sau FEND. Pn là bắt đầu của chương trình con; SRET là điểm kết thúc của chương trình con. Lệnh CALL Pn được sử dụng để gọi chương trình con. Phạm vi của n là từ 0 đến 9999.

Việc gọi chương trình con có thể đơn giản hóa việc lập trình. Nếu chương trình sẽ được sử dụng ở nhiều vị trí, hãy tạo chương trình trong chương trình con và gọi nó.

4-3-3 Luồng [SET], [ST], [STL], [STLE]

1) Sơ lược

Các lệnh xác định điểm bắt đầu, kết thúc, mở, đóng của một luồng;

Mở luồng đã chỉ định, đóng luồng cục bộ [SET]			
16 bits	SET	32 bits	-
Điều kiện thực thi	Thường mở/đóng, Sườn logic lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Mở luồng đã chỉ định, không đóng luồng cục bộ [ST]			
16 bits	ST	32 bits	-
Điều kiện thực thi	Thường mở/đóng, Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Bắt đầu luồng [STL]			
16 bits	STL	32 bits	-
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Kết thúc luồng [STLE]			
16 bits	STLE	32 bits	-
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

Toán hạng	Chức năng	Kiểu dữ liệu
Sn	Nhảy đến luồng mục tiêu S	Flow No.

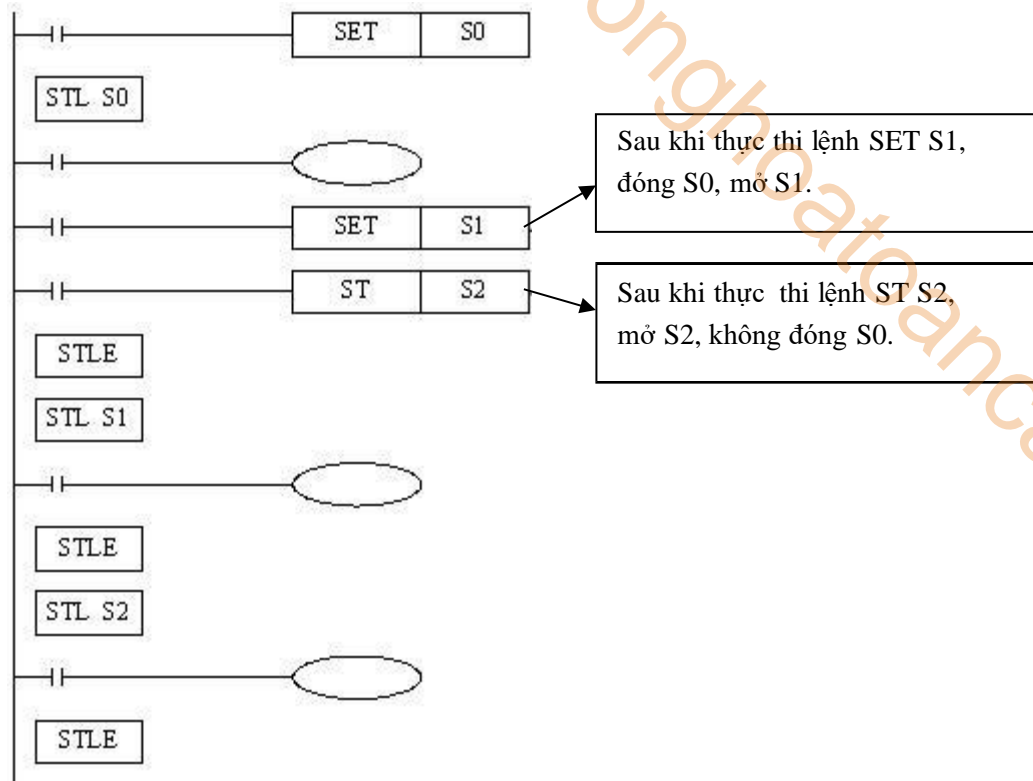
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit								
	Hệ								Hàng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.	
Sn														●					

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

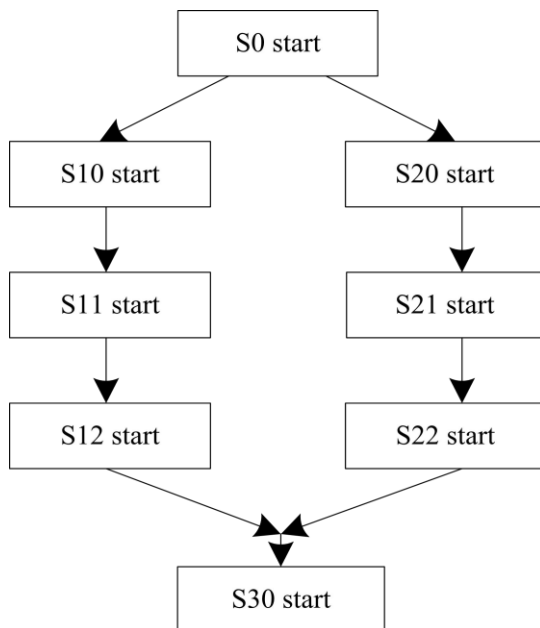
Mô tả

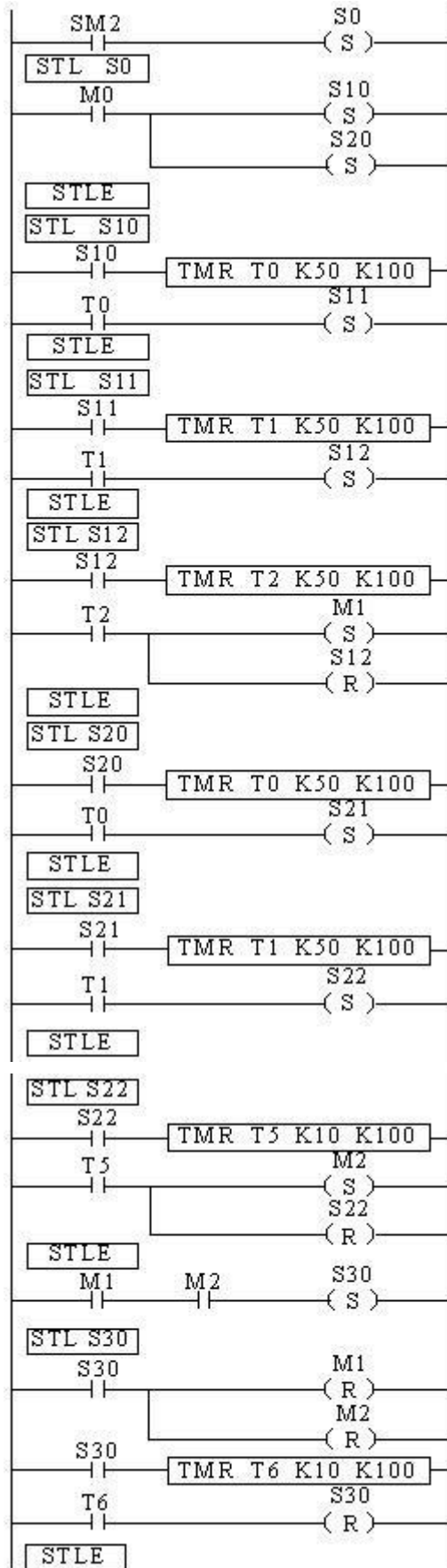
- Lệnh STL và STLE nên được sử dụng theo cặp. Lệnh STL biểu thị cho sự bắt đầu của một luồng; luồng STLE biểu thị cho sự kết thúc của một luồng.
- Mọi luồng đều độc lập. Chúng không thể lồng vào nhau được. Không cần ghi flow theo thứ tự S0, S1, S2... các bạn có thể đặt thứ tự. Ví dụ: thực thi S10, rồi S5, S0.
- Sau khi thực hiện lệnh **SET Sxxx**, luồng được chỉ định bởi các lệnh này sẽ ở trạng thái ON.
- Sau khi thực thi lệnh **RST Sxxx**, luồng được chỉ định sẽ ở trạng thái OFF.
- Trong luồng S0, SET S1 đóng luồng hiện tại tại S0, mở luồng S1. Trong luồng S0, ST S2 mở luồng S2 nhưng không đóng luồng S0.
- Khi luồng chuyển từ trạng thái ON sang OFF, hãy đặt lại lệnh OUT, PLS, PLF, bộ định thời không tích lũy, v.v. trong luồng.
- Lệnh ST thường được sử dụng khi một chương trình cần chạy nhiều luồng cùng một lúc.
- Sau khi thực thi lệnh **SET Sxxx** và chuyển sang luồng tiếp theo, các lệnh xung trong luồng trước sẽ bị đóng. (bao gồm một đoạn, nhiều đoạn, tương đối hoặc tuyệt đối, quay về luồng ban đầu)



Ví dụ

Ví dụ 1: Các luồng chạy trong nhánh rời hợp nhất thành một luồng.
 Sơ đồ chương trình:





Giải thích chương trình:

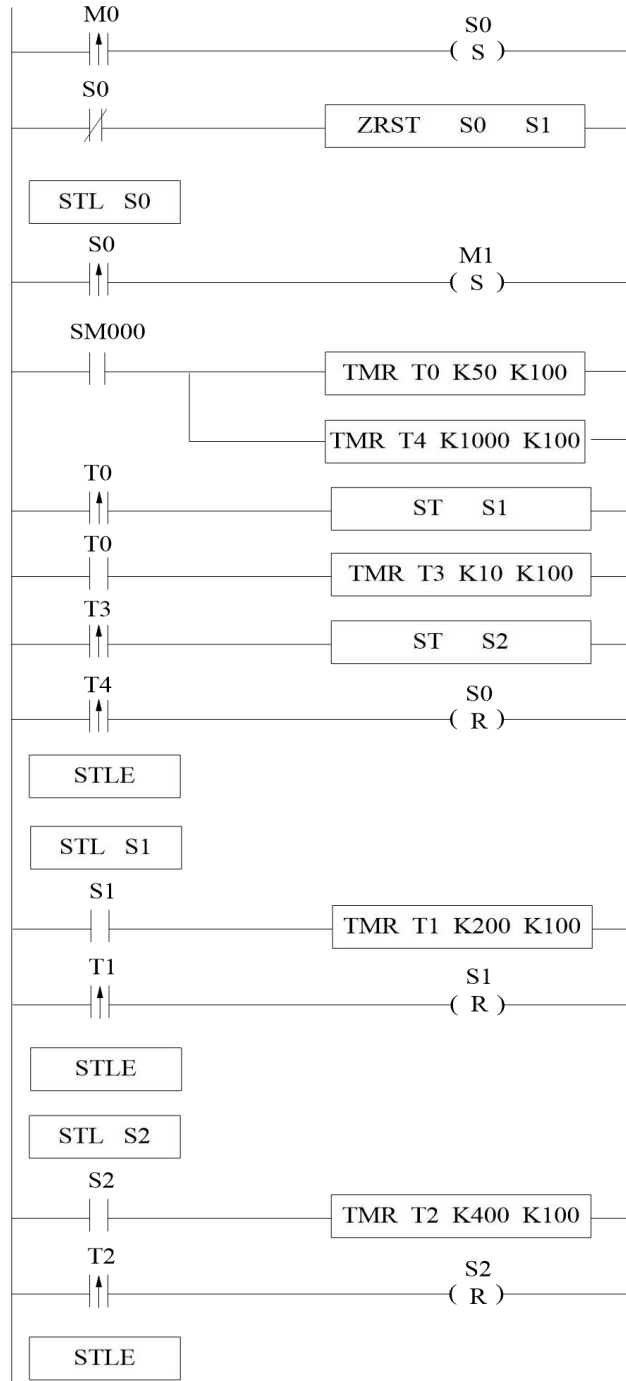
Khi SM2 ở trạng thái ON, đặt trạng thái ON cho luồng S0. Khi M0 ở trạng thái ON, đặt trạng thái ON cho luồng S10 và S20.

Ở nhánh S10 thì chạy S10, S11 và S12. Đặt trạng thái ON cho M1 nghĩa là nhánh S10 đã kết thúc

Ở nhánh S20 thì chạy S20, S21 và S22. Đặt trạng thái On cho M2 nghĩa là nhánh S20 đã kết thúc.

Khi cả hai nhánh S10 và S20 kết thúc, đặt trạng thái on cho S30. Khi S30 kết thúc, hãy đặt lại S30.

Ví dụ 2: lồng luồng. Khi S0 chạy được một lúc thì S1 và S2 bắt đầu chạy; trạng thái hoạt động của S1 được giữ nguyên. Khi S0 đang chạy trong một thời gian nhất định, hãy đóng S0 và buộc đóng S1 và S2.



4-3-4 Lệnh [FOR] và [NEXT]

1) Sơ lược

Vòng lặp thực hiện chương trình giữa **FOR** và **NEXT** với thời gian xác định;

Bắt đầu vòng lặp [FOR]			
16 bits	FOR	32 bits	-
Điều kiện thực thi	Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Kết thúc vòng lặp [NEXT]			
16 bits	NEXT	32 bits	-
Điều kiện thực thi	Thường mở/đóng Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Số lần lặp của chương trình giữa FOR và NEXT	16 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word											Phân tử lập trình dạng bit						
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.
Sn	•							•										m

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

Các lệnh FOR.NEXT phải được lập trình theo một cặp. Được phép lồng lệnh và mức độ lồng ghép tối đa là 8.

Chương trình sau lệnh NEXT sẽ không được thực thi trừ khi chương trình giữa FOR và NEXT được thực thi trong thời gian xác định.

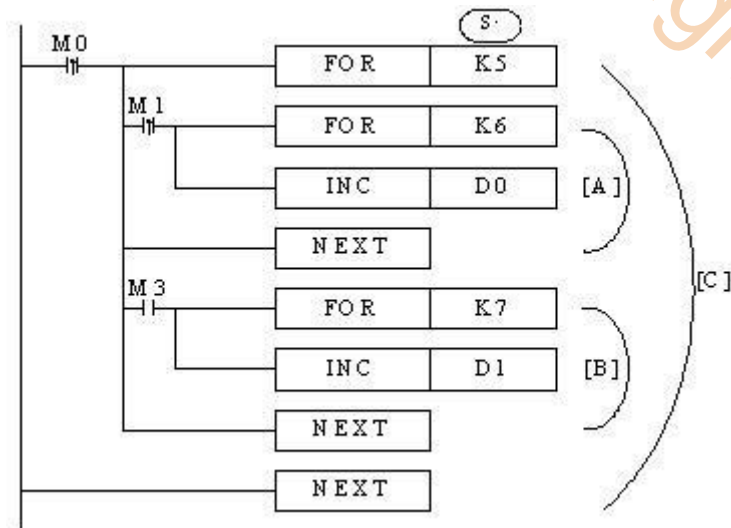
Giữa FOR và NEXT, các lệnh LDP, LDF chỉ có hiệu lực một lần. Mỗi khi M0 chuyển từ trạng thái OFF sang ON và M1 chuyển từ OFF sang ON, vòng lặp [A] được thực thi $5 \times 6 = 30$ lần.

Mỗi lần nếu M0 chuyển từ OFF sang ON và M3 ở trạng thái ON, vòng lặp [B] được thực thi $5 \times 7 = 35$ lần.

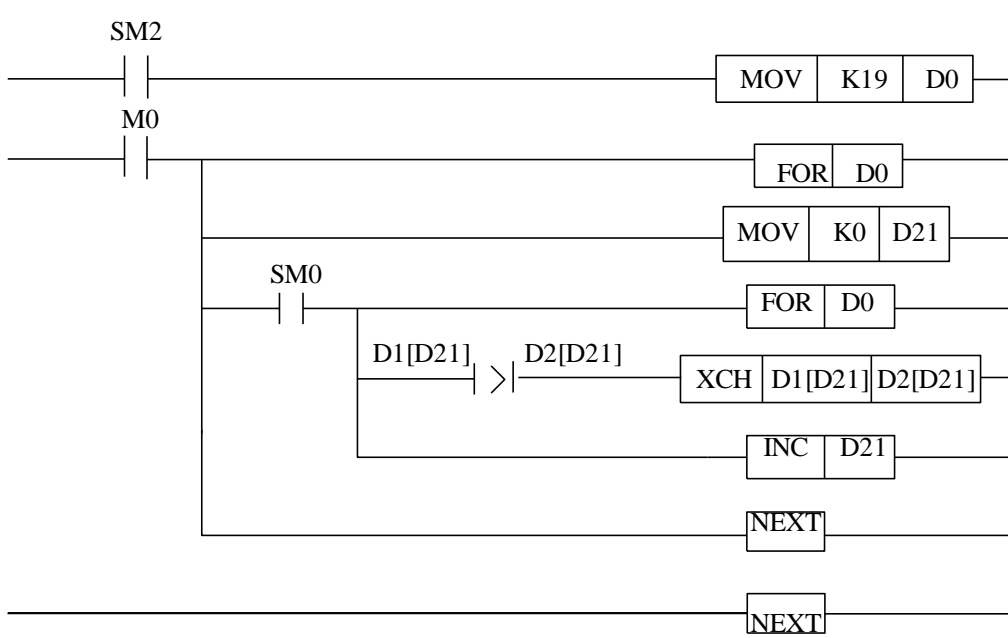
Nếu có nhiều lần lặp, chu kỳ quét sẽ kéo dài. Lỗi bộ định thời có thể xảy ra, hãy lưu ý điều này.

Nếu NEXT ở trước FOR hoặc không có NEXT hoặc NEXT ở sau FEND, END hoặc số lệnh FOR và NEXT không bằng nhau thì sẽ xảy ra lỗi.

Giữa FOR~NEXT, không được phép lồng CJ. FOR~NEXT phải theo cặp trong một STL.



Ví dụ 1: Khi M0 ở trạng thái ON, FOR NEXT bắt đầu sắp xếp các số trong phạm vi từ D1 đến D20 từ nhỏ đến lớn. D21 là giá trị thanh ghi con trỏ. Nếu chương trình có nhiều kiểu sắp xếp, vui lòng sử dụng ngôn ngữ C để tiết kiệm thời gian lập trình và thời gian quét.



```

LD    SM2           //SM2 là cuộn dây ở trạng thái ON ban đầu
MOV   K19  D0       //số lần của vòng lặp FOR
LD    M0            // M0 để kích hoạt vòng lặp FOR
MCS           //
FOR   D0            // Vòng lặp FOR lồng nhau, số lần vòng lặp là D0
MOV   K0  D21       //địa chỉ thanh ghi con trỏ bắt đầu từ 0
LD    SM0           //SM0 là cuộn dây luôn ở trạng thái ON
MCS           //
FOR   D0            //lồng vòng lặp FOR, số lần vòng lặp là D0
LD>  D1[D21]  D2[D21] //nếu dữ liệu hiện tại lớn hơn dữ liệu tiếp theo, nó sẽ
                        //ở trạng thái ON

```

XCH D1[D21] D2[D21] //trao đổi hai dữ liệu lân cận
 LD SM0 //M8000 là cuộn dây luôn ở trạng thái ON
 INC D21 // tăng 1 cho D21
 MCR //

NEXT //khớp với lệnh FOR thứ hai
 MCR //
 NEXT //khớp với lệnh FOR thứ nhất

4-3-5 Lệnh [FEND] và [END]

1) Sơ lược

FEND nghĩa là kết thúc chương trình chính, trong khi END có nghĩa là kết thúc chương trình;

Kết thúc chương trình chính [FEND]			
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Kết thúc chương trình [END]			
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

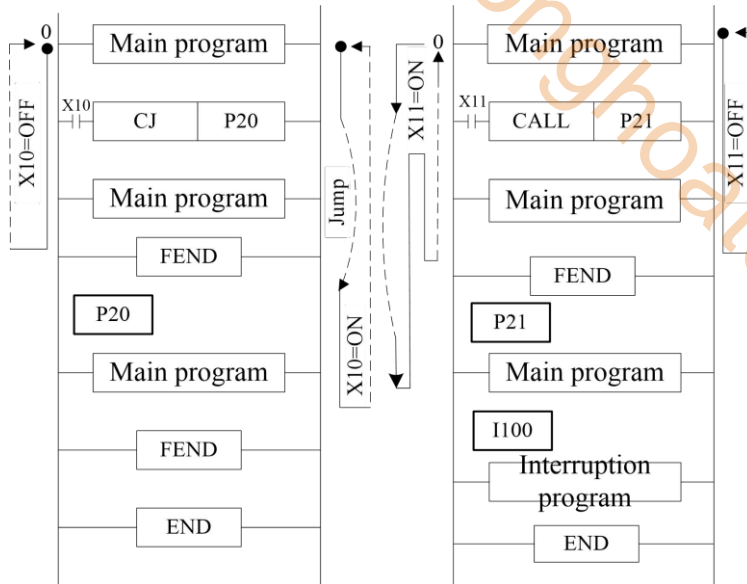
Toán hạng	Chức năng	Kiểu dữ liệu
Không	-	-

3) Phần tử lập trình phù hợp

Không

Mô tả

Mặc dù lệnh [FEND] nghĩa là kết thúc của chương trình chính, chức năng của nó cũng tương tự như END để xử lý đầu ra/đầu vào, giám sát việc làm mới bộ định thời, quay lại bước 0 của chương trình.



Nếu lập trình thẻ lệnh CALL phía sau lệnh FEND thì phải có lệnh SRET. Nếu chương trình the interrupt pointer program nằm sau lệnh FEND thì phải có lệnh IRET. Sau khi thực thi lệnh CALL và trước khi thực thi lệnh SRET, nếu thực thi lệnh FEND; hoặc thực thi lệnh FEND sau khi thực thi lệnh FOR và trước khi thực thi lệnh NEXT, sẽ xảy ra lỗi.

Trong điều kiện sử dụng nhiều lệnh FEND, vui lòng tạo chương trình hoặc chương trình con giữa lệnh FEND cuối cùng và lệnh END.

4-4 Các lệnh so sánh dữ liệu

Tên gọi nhớ	Chức năng	Chương
LD=	LD kích hoạt khi (S1)= (S2)	4-4-1
LD>	LD kích hoạt khi (S1)> (S2)	4-4-1
LD<	LD kích hoạt khi (S1)< (S2)	4-4-1
LD<>	LD kích hoạt khi (S1)≠ (S2)	4-4-1
LD<=	LD kích hoạt khi (S1)≤ (S2)	4-4-1
LD>=	LD kích hoạt khi (S1)≥ (S2)	4-4-1
AND=	AND kích hoạt khi (S1)= (S2)	4-4-2
AND>	AND kích hoạt khi (S1)> (S2)	4-4-2
AND<	AND kích hoạt khi (S1)< (S2)	4-4-2
AND<>	AND kích hoạt khi (S1)≠ (S2)	4-4-2
AND<=	AND kích hoạt khi (S1)≤ (S2)	4-4-2
AND>=	AND kích hoạt khi (S1)≥ (S2)	4-4-2
OR=	OR kích hoạt khi (S1)= (S2)	4-4-3
OR>	OR kích hoạt khi (S1)> (S2)	4-4-3
OR<	OR kích hoạt khi (S1)< (S2)	4-4-3
OR<>	OR kích hoạt khi (S1)≠ (S2)	4-4-3
OR<=	OR kích hoạt khi (S1)≤ (S2)	4-4-3
OR>=	OR kích hoạt khi (S1)≥ (S2)	4-4-3

4-4-1 Lệnh so sánh LD [LD]

1) Sơ lược

Lệnh so sánh dùng lệnh LD kết hợp với phép toán so sánh

LD Compare [LD]			
16 bits	Như sau:	32 bits	Như sau:
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ thanh ghi bị so sánh	16/32bits, BIN
S2	Địa chỉ thanh ghi so sánh	16/32 bits, BIN

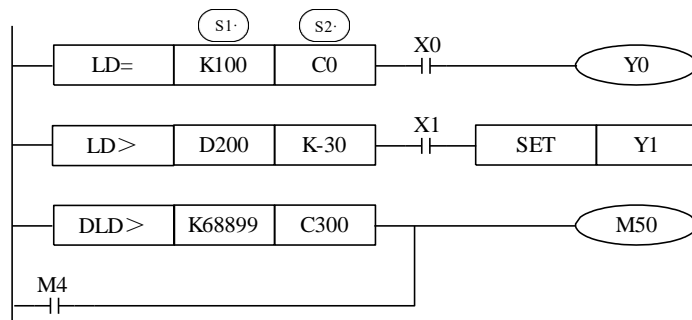
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

Lệnh 16 bit	Lệnh 32 bit	Điều kiện kích hoạt	Điều kiện không kích hoạt
LD=	DLD=	(S1)=(S2)	(S1)≠(S2)
LD>	DLD>	(S1)>(S2)	(S1)≤(S2)
LD<	DLD<	(S1)<(S2)	(S1)≥(S2)
LD<>	DLD<>	(S1)≠(S2)	(S1)=(S2)
LD≤	DLD≤	(S1)≤(S2)	(S1)>(S2)
LD≥	DLD≥	(S1)≥(S2)	(S1)<(S2)



Lưu ý

- Khi bit cao nhất của dữ liệu nguồn (16 bit: b15, 32 bit: b31) là 1, dữ liệu được xem số âm.
- Việc so sánh bộ đếm 32 bit nên sử dụng lệnh 32 bit. Nếu sử dụng lệnh 16 bit, chương trình hoặc thao tác sẽ bị lỗi.

4-4-2 So sánh nối tiếp [AND]

1) Sơ lược

AND: lệnh so sánh kết nối nối tiếp.

AND Compare [AND]			
16 bits	Như sau:	32 bits	Như sau:
Điều kiện thực thi	Thường mở/đóng coil	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu được so sánh	16/32bit, BIN
S2	Địa chỉ dữ liệu so sánh	16/32bit, BIN

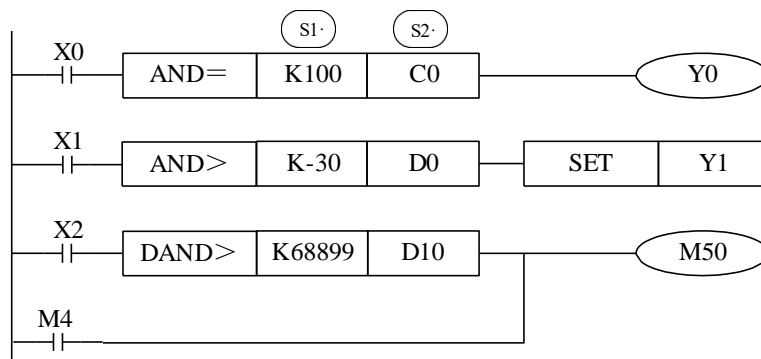
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

Lệnh 16 bit	Lệnh 32 bit	Điều kiện kích hoạt	Điều kiện không kích hoạt
AND=	DAND=	(S1) = (S2)	(S1) ≠ (S2)
AND>	DAND>	(S1) > (S2)	(S1) ≤ (S2)
AND<	DAND<	(S1) < (S2)	(S1) ≥ (S2)
AND<>	DAND<>	(S1) ≠ (S2)	(S1) = (S2)
AND≤	DAND≤	(S1) ≤ (S2)	(S1) > (S2)
AND≥	DAND≥	(S1) ≥ (S2)	(S1) < (S2)



Lưu ý

Khi bit cao nhất của dữ liệu nguồn (16 bit: b15, 32 bit: b31) là 1, nó được xem là số âm.

Việc so sánh bộ đếm 32 bit nên sử dụng lệnh 32 bit. Nếu sử dụng lệnh 16 bit, chương trình hoặc thao tác sẽ bị lỗi.

4-4-3 Lệnh so sánh song song [OR]

1) Sơ lược

OR: lệnh so sánh kết nối song song.

Parallel Compare [OR]			
16 bits	Như sau:	32 bits	Như sau
Điều kiện thực thi	-	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu được so sánh	16/32 bits, BIN
S2	Địa chỉ dữ liệu so sánh	16/32 bits, BIN

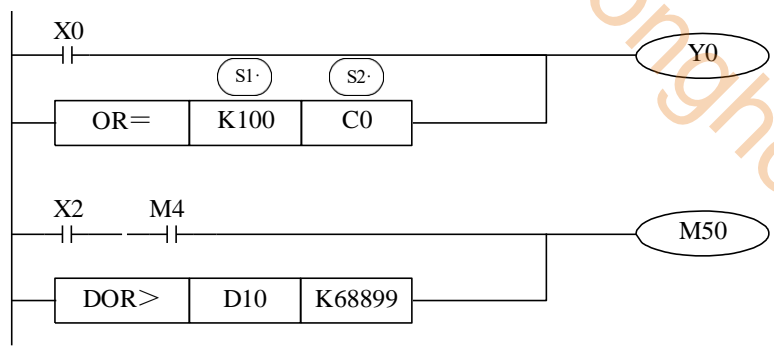
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng word							
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

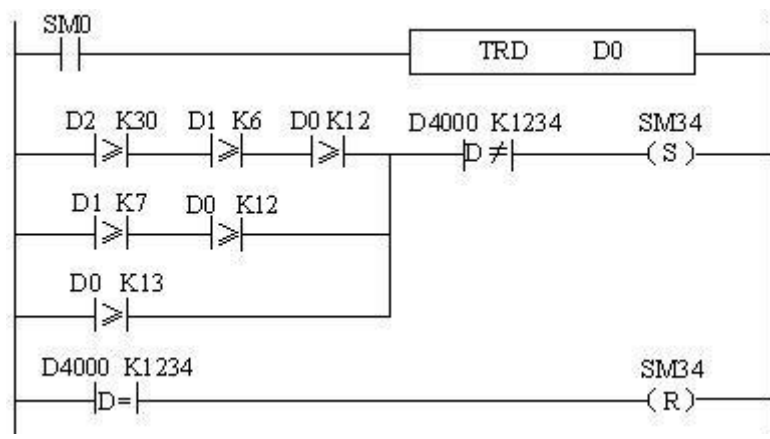
Lệnh 16 bit	Lệnh 32 bit	Điều kiện kích hoạt	Điều kiện không kích hoạt
OR =	DOR =	(S1) = (S2)	(S1) ≠ (S2)
OR >	DOR >	(S1) > (S2)	(S1) ≤ (S2)
OR <	DOR <	(S1) < (S2)	(S1) ≥ (S2)
OR <>	DOR <>	(S1) ≠ (S2)	(S1) = (S2)
OR ≤	DOR ≤	(S1) ≤ (S2)	(S1) > (S2)
OR ≥	DOR ≥	(S1) ≥ (S2)	(S1) < (S2)



Lưu ý

- Khi bit cao nhất của dữ liệu nguồn (16 bit: b15, 32 bit: b31) là 1, nó được xem là số âm.
- Việc so sánh bộ đếm 32 bit nên sử dụng lệnh 32 bit. Nếu sử dụng lệnh 16 bit, chương trình hoặc thao tác sẽ bị lỗi.

Ví dụ: Vô hiệu hóa đầu ra khi đạt đến thời điểm nhất định. Trong chương trình bên dưới, khi ngày là ngày 30 tháng 6 năm 2012, tất cả các đầu ra sẽ bị vô hiệu hóa. Mật khẩu 1234 được lưu trong (D4000, D4001). Khi mật khẩu chính xác, tất cả các đầu ra đều được kích hoạt.



```

LD    SM0                //SM0 là cuộn dây luôn ở vị trí on
TRD   D0                 //đọc giá trị RTC (đồng hồ thời gian thực) và lưu vào D0~D6
LD>=  D2      K30        //Dữ liệu RTC ngày ≥30
AND>=          D1      K6          //Dữ liệu RTC tháng ≥6
AND>=          D0      K12         //Dữ liệu RTC năm ≥12
LD>=  D1      K7          //hoặc dữ liệu RTC tháng ≥ 7
AND>=          D0      K12         //dữ liệu RTC năm ≥ 12
ORB                               //or
OR>=  D0      K13         //Dữ liệu RTC năm ≥ 13
DAND<> D4000 K1234       //và mật khẩu ≠1234
SET   SM34              //đặt M34 ở trạng thái ON, tất cả các đầu ra đều bị vô hiệu hóa
DLD=  D4000 K1234       //mật khẩu =1234, mật khẩu đúng
RST   SM34              //đặt lại M34, tất cả các đầu ra được kích hoạt
    
```

4-5 Lệnh chuyển dữ liệu

Tên gọi nhớ	Chức năng	Chương
CMP	So sánh dữ liệu	4-5-1
ZCP	So sánh vùng dữ liệu	4-5-2
MOV	Chuyển	4-5-3
BMOV	Chuyển khối dữ liệu	4-5-4
PMOV	Chuyển khối dữ liệu (với tốc độ nhanh hơn)	4-5-5
FMOV	Chuyển khối dữ liệu đến nhiều thanh ghi	4-5-6
EMOV	Chuyển số thực dấu phẩy động	4-5-7
FWRT	Ghi vào bộ nhớ FlashROM	4-5-8
MSET	Đặt vùng	4-5-9
ZRST	Đặt lại vùng	4-5-10
SWAP	Byte cao và thấp của thiết bị đích được trao đổi	4-5-11
XCH	Trao đổi hai dữ liệu	4-5-12

4-5-1 Lệnh so sánh dữ liệu [CMP, DCMP, QCMP]

1) Sơ lược

So sánh hai dữ liệu, xuất kết quả.

So sánh dữ liệu [CMP, DCMP, QCMP]			
16 bits	CMP	32 bits	DCMP
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QCMP		
Điều kiện thực thi	Thường đóng/mở, sườn lên/xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

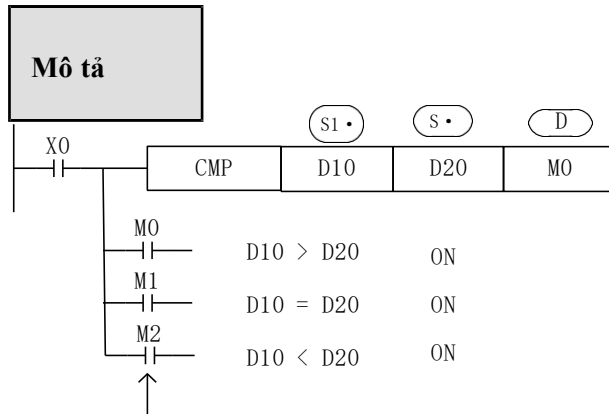
2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Chỉ định dữ liệu (để so sánh) hoặc mã địa chỉ của thành phần phần mềm	16/32/64 bits, BIN
S	Chỉ định giá trị so sánh hoặc mã địa chỉ của phần tử lập trình	16/32/64 bits, BIN
D	Chỉ định mã địa chỉ của kết quả so sánh	bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hãng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•	•										
D													•	•	•				

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S,HS; T gồm T,HT; C gồm C, HC.



Ngay cả khi X0 chuyển sang OFF để dừng lệnh CMP, M0~M2 sẽ giữ nguyên trạng thái ban đầu

- So sánh dữ liệu S1 và S, hiển thị kết quả trong ba phần tử lập trình bắt đầu từ D
- D, D+1, D+2: ba phần tử lập trình sẽ hiển thị kết quả so sánh.
- Lưu ý: Địa chỉ của các toán hạng trong lệnh ZCMP phải là địa chỉ chẵn.

4-5-2 Lệnh so sánh vùng dữ liệu [ZCP, DZCP]

1) Sơ lược

So sánh dữ liệu hiện tại với dữ liệu trong vùng, xuất kết quả.

So sánh vùng dữ liệu [ZCP, DZCP]			
16 bits	ZCP	32 bits	DZCP
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Giới hạn thấp của vùng (giới hạn dưới)	16/32 bits, BIN
S2	Giới hạn cao của vùng (giới hạn trên)	16/32 bits, BIN
S	Địa chỉ dữ liệu hiện tại	16/32 bits, BIN
D	Kết quả so sánh	bit

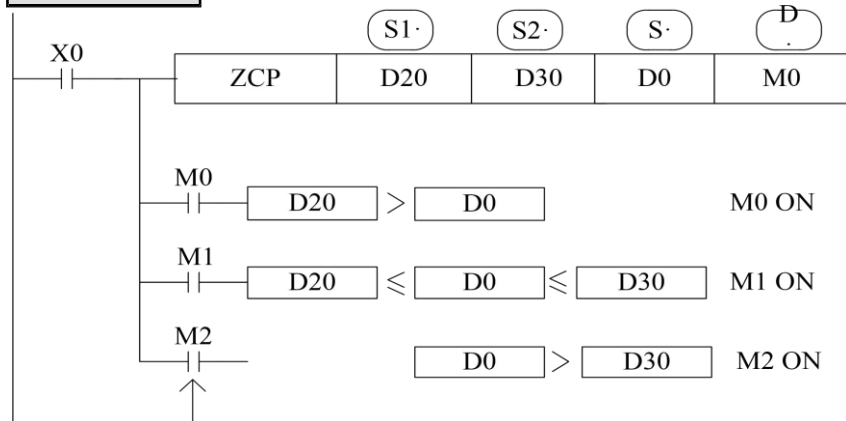
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	●	●	●	●	●	●	●	●	●									

S2	•	•	•	•	•	•	•	•	•										
S	•	•	•	•	•	•	•	•	•										
D													•	•	•				

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



Ngay cả khi X0=OFF để dừng lệnh ZCP, M0~M2 sẽ giữ nguyên trạng thái ban đầu

- So sánh S với S1 và S2, xuất 3 kết quả từ D
- D, D+1, D+2: lưu 3 kết quả

4-5-3 Lệnh MOV [MOV, DMOV, QMOV]

1) Sơ lược

Di chuyển dữ liệu đã chỉ định sang các phần tử lập trình khác

MOV [MOV, DMOV, QMOV]			
16 bits	MOV	32 bits	DMOV
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QMOV		
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định dữ liệu nguồn hoặc mã địa chỉ của thanh ghi	16 bits/32 bits/64 bits, BIN
D	Chỉ định mã địa chỉ của thành phần phần mềm đích	16 bits/32 bits/64 bits, BIN

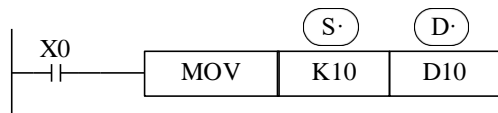
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit										
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S	•	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•			•								

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

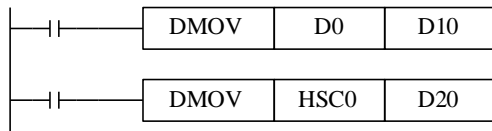
< Chuyển dữ liệu 16 bit >



- Chuyển dữ liệu nguồn đến đích
- Khi X0 ở trạng thái tắt, dữ liệu sẽ không thay đổi
- Chuyển K10 sang D10

< Chuyển dữ liệu 32 bit >

Vui lòng sử dụng DMOV khi giá trị là 32 bit, chẳng hạn như lệnh MUL, bộ đếm tốc độ cao...



(D1, D0) → (D11, D10)
 (Giá trị hiện tại của HSC0) → (D21, D20)

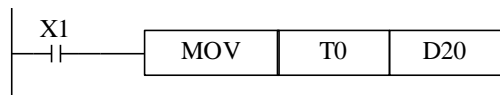
< Chuyển dữ liệu 64 bit >

Vui lòng sử dụng QMOV khi giá trị là 64 bit, chẳng hạn như lệnh DMUL.



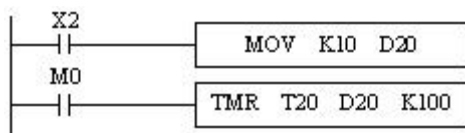
(D3,D2,D1,D0)→(D13,D12,D11,D10)

< đọc giá trị hiện tại của bộ đếm hoặc bộ định thời >



(Giá trị hiện tại của T0)→(D20)

<i> Tương tự với bộ đếm



(K10) (D20)

D20=K10

4-5-4 Lệnh chuyển khối dữ liệu [BMOV]

1) Sơ lược

Chuyển khối dữ liệu sang phần tử lập trình khác

Chuyển khối dữ liệu[BMOV]			
16 bits	BMOV	32 bits	-
Điều kiện thực thi	Cuộn dây Thường hở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định khối dữ liệu nguồn hoặc mã địa chỉ phần tử lập trình	16 bits, BIN; bit
D	Chỉ định mã địa chỉ phần tử lập trình mục tiêu	16 bits, BIN; bit
n	Chỉ định số dữ liệu chuyển	16 bits, BIN;

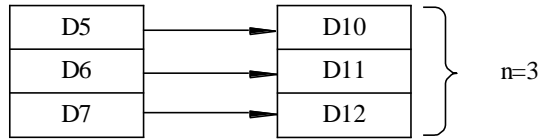
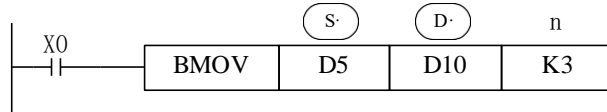
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S	•	•	•	•	•	•	•	•					•	•	•				
D	•		•	•		•	•	•					•	•	•				
n	•		•	•	•		•	•	•										

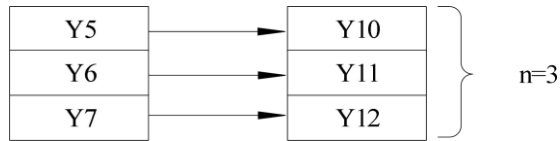
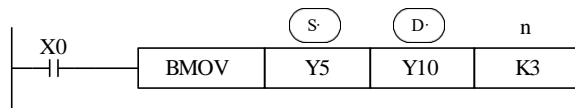
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S,HS; T gồm T, HT; C gồm C, HC.

Mô tả

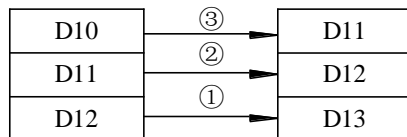
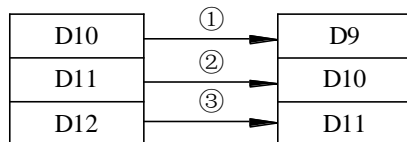
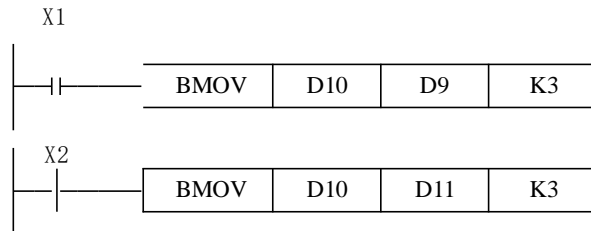
Di chuyển khối dữ liệu nguồn sang khối dữ liệu đích. Số lượng dữ liệu là n.
<chuyển word>



<chuyển bit>



Như chỉ ra trong hình dưới đây, khi dải địa chỉ thanh ghi nguồn và đích dữ liệu có sự trùng lặp nhau, để ngăn chặn dữ liệu nguồn cần truyền bị ghi đè khi chưa được truyền đi, lệnh sẽ được thực hiện theo thứ tự từ 1 đến 3.



4-5-5 Lệnh chuyển khối dữ liệu [PMOV]

1) Sơ lược

Di chuyển khối dữ liệu đã chỉ định sang các phần tử lập trình khác

Data block mov[PMOV]			
16 bits	PMOV	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên/xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định khối dữ liệu nguồn hoặc địa chỉ phần tử lập trình	16 bits, BIN; bit
D	Chỉ định địa chỉ phần tử lập trình mục tiêu	16 bits, BIN; bit
n	Chỉ định số dữ liệu chuyển	16 bits, BIN;

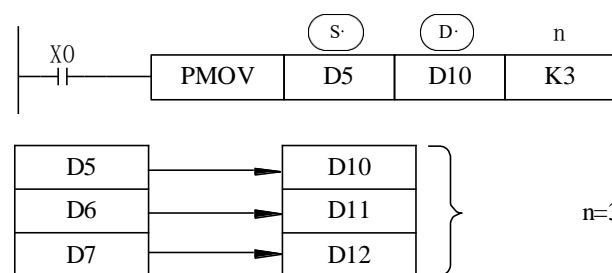
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hàng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S	•																		
D	•																		
n	•		•	•	•		•	•	•										

Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

Chuyển khối dữ liệu nguồn sang khối dữ liệu đích, số lượng dữ liệu (data quantity) là n



Chức năng của lệnh PMOV và BMOV hầu như giống nhau nhưng tốc độ thực thi lệnh PMOV nhanh hơn.

Lệnh PMOV kết thúc trong một chu kỳ quét, khi thực thi lệnh PMOV, hãy đóng tất cả các lệnh ngắt. Lỗi có thể xảy ra nếu địa chỉ nguồn và địa chỉ đích bị trùng lặp.

4-5-6 Lệnh chuyển dữ liệu sang nhiều thanh ghi [FMOV, DFMOV]

1) Sơ lược

Chuyển dữ liệu đã chỉ định sang các phần tử lập trình khác

Lệnh Fill Move [FMOV, DFMOV]			
16 bits	FMOV	32 bits	DFMOV
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn dưới	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định dữ liệu nguồn hoặc địa chỉ phần tử lập trình	16/32 bits, BIN;
D	Chỉ định địa chỉ phần tử lập trình mục tiêu	16/32 bits, BIN;
n	Chỉ định số lượng dữ liệu chuyển	16/32 bits, BIN;

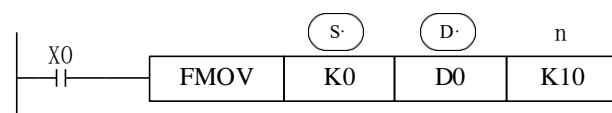
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										
n	•		•	•		•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S,HS; T gồm T,HT; C gồm C, HC.

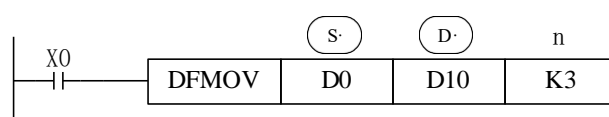
Mô tả

<Lệnh 16 bit>



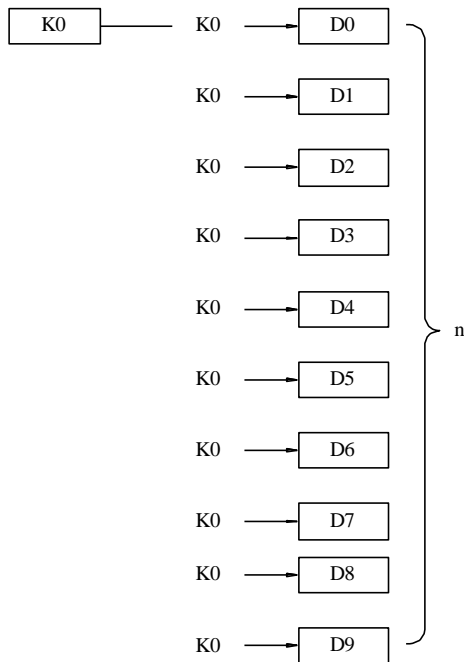
- Di chuyển K0 đến D0~D9, sao chép dữ liệu của một thiết bị sang một phạm vi thiết bị đích
- Di chuyển dữ liệu nguồn sang dữ liệu đích, số lượng dữ liệu đích là n
- Nếu phạm vi đã đặt vượt quá phạm vi mục tiêu, hãy chuyển sang đến phạm vi có thể

<lệnh 32 bit>

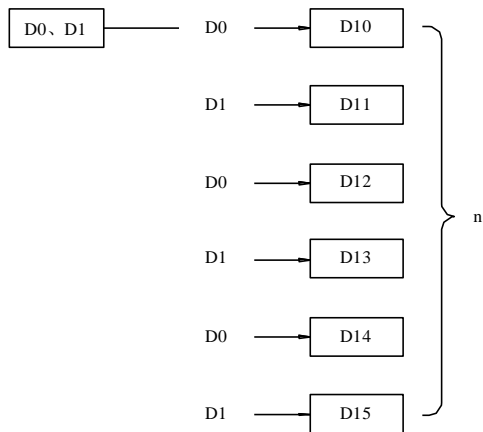


- Chuyển D0.D1 sang D10.D11:D12.D13:D14.D15.

<chuyển dữ liệu 16 bit>



<truyền dữ liệu 32 bit>



4-5-7 Lệnh chuyển dữ liệu số thực dấu phẩy động [EMOV, EDMOV]

1) Sơ lược

Chuyển dữ liệu số thực dấu phẩy động đến địa chỉ đích

Lệnh chuyển số thực dấu phẩy động [EMOV, EDMOV]			
16 bits	-	32 bits	EMOV
Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDMOV		

Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu
S	Địa chỉ phần tử lập trình nguồn	32 /64bits, BIN
D	Địa chỉ phần tử lập trình đích	32 /64bits, BIN

3) Phần tử lập trình phù hợp

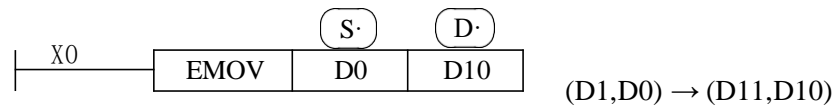
Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S,HS; T gồm T,HT; C gồm C, HC.

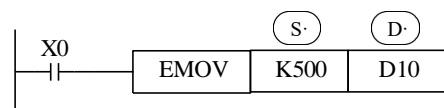
Mô tả

<Lệnh 32 bit>

Địa chỉ số thực này sang địa chỉ số thực khác



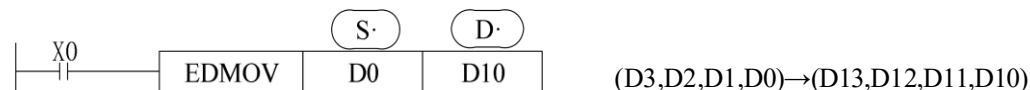
- Nếu X0 ở trạng thái ON, chuyển số thực dấu phẩy động từ (D1, D0) đến (D11, D10).
- Nếu X0 ở trạng thái OFF, lệnh sẽ không được thực thi



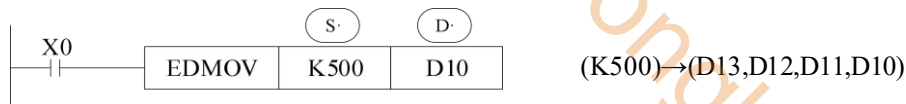
(K500) -> (D11, D10)

- Nếu hàng số K, H là phần tử lập trình nguồn thì sẽ chuyển thành số thực dấu phẩy động
- K500 sẽ được chuyển đổi thành giá trị số thực dấu phẩy động.

<lệnh 64 bit>



- X0 ở trạng thái ON, chuyển số thực dấu phẩy động từ (D3, D2, D1, D0) đến (D13, D12, D11, D10).
- Nếu X0 ở trạng thái OFF, lệnh sẽ không được thực thi



- Nếu hằng số K, H là phân tử lập trình nguồn thì sẽ chuyển thành số thực dấu phẩy động.
- K500 sẽ được chuyển đổi thành giá trị số thực dấu phẩy động.
- Địa chỉ của các toán hạng trong lệnh EDMOV phải là địa chỉ chẵn.

4-5-8 Lệnh Ghi dữ liệu vào FlashROM [FWRT, DFWRT, QFWRT]

1) Sơ lược

Ghi dữ liệu được chỉ định vào thanh ghi FlashROM.

Lệnh ghi dữ liệu vào FlashROM [FWRT,DFWRT,QFWRT]			
16 bits	FWRT	32 bits	DFWRT
Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QFWRT		
Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Dữ liệu ghi vào nguồn hoặc lưu vào phần mềm	16 /32/64 bits, BIN
D	phân tử lập trình mục tiêu	16 /32/64 bits
D1	địa chỉ bắt đầu phân tử lập trình mục tiêu	16 /32/64 bits
D2	số lượng dữ liệu ghi vào	16 /32/64 bits, BIN

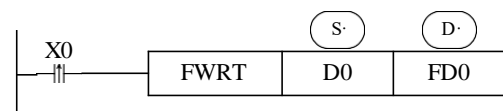
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•	•	•	•	•	•	•	•	•										
S2		•																	
S		•																	
D	•		•	•	•	•	•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

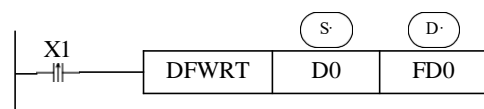
Mô tả

< Ghi giá trị 16 bit (1 word = 16 bit)>



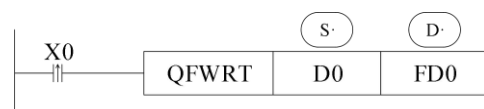
Ghi giá trị từ D0 sang FD0

<Ghi giá trị 32 bit (2 word = 32 bit)>



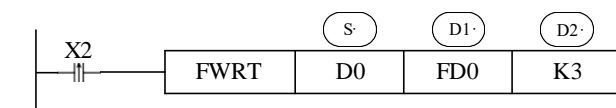
Ghi giá trị từ D0,D1 sang FD0,FD1

<Ghi giá trị 64 bit (4 word = 64 bit)>



Ghi giá trị từ D0,D1,D2,D3 sang FD0,FD1,FD2,FD3.

<Ghi đa word>



Ghi giá trị từ D0, D1, D2 sang FD0, FD1,FD2

LƯU Ý:

- ※1: Lệnh FWRT chỉ có thể ghi dữ liệu vào thanh ghi FlashROM. FlashROM có thể lưu giữ dữ liệu ngay cả khi nguồn điện bị tắt. Nó có thể lưu trữ các thông số kỹ thuật quan trọng.
- ※2: Việc ghi dữ liệu vào FWRT cần thời gian dài, khoảng 500ms, vì vậy không nên ghi thường xuyên
- ※3: Thời gian ghi của FlashROM là khoảng 1.000.000 lần. Vì vậy, chúng tôi khuyên bạn nên sử dụng tín hiệu sườn (LDP, LDF, v.v.) để kích hoạt lệnh.
- ※4: Việc ghi vào thường xuyên sẽ làm hỏng FlashROM.

4-5-9 Lệnh Cài đặt vùng [MSET]

1) Sơ lược

Đặt phần tử lập trình trong phạm vi nhất định

Lệnh Multi-set [MSET]			
16 bits	MSET	32 bits	-
Điều kiện thực thi	Thường mở/đóng; tín hiệu xung sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

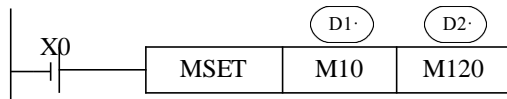
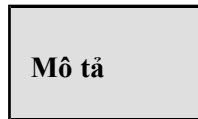
2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D1	Địa chỉ phần tử phần mềm nguồn	bit
D2	Địa chỉ phần tử phần mềm đích	bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit									
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D1												•	•	•	•	•	•	
D2												•	•	•	•	•	•	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.



Đặt M10~M120
sang trạng thái ON

- Đặt cuộn dây từ M10 thành M120.
- Khi D1, D2 được chỉ định làm phần tử lập trình và D1<D2
- Khi D1>D2, D2 sẽ không chạy lệnh Zone set mà đặt SM409 SD409 = 2

4-5-10 Lệnh đặt lại Vùng [ZRST]

1) Sơ lược

Đặt lại phần tử lập trình trong phạm vi nhất định

Lệnh Multi-reset [ZRST]			
16 bits	ZRST	32 bits	-
Điều kiện thực thi	Thường mở/đóng; xung sườn lên/sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

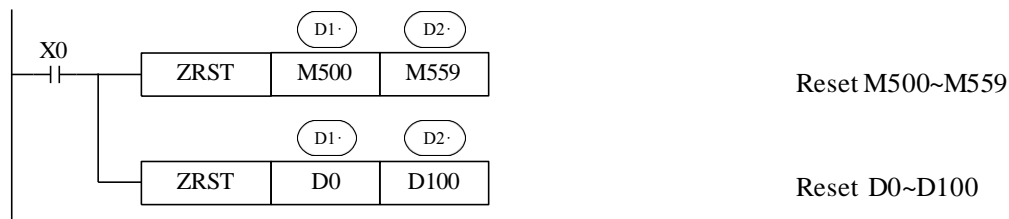
Toán hạng	Chức năng	Kiểu dữ liệu
D1	Địa chỉ bắt đầu phần tử lập trình	Bit, 16 bits, BIN
D2	Địa chỉ kết thúc phần tử lập trình	Bit, 16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.
		D	D	D	X	Y	M	S		D	D							m
D1	•				•	•	•					•	•	•	•	•	•	
D2	•			•	•	•	•					•	•	•	•	•	•	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



- (D1), (D2) được chỉ định là cùng loại phần tử lập trình và <(D1) (D2)>
- Khi D1>D2, chỉ đặt lại phần tử lập trình được chỉ định và đặt SM409, SD409 = 2.

Lệnh đặt lại (reset) khác

Lệnh RST có thể đặt lại một phần tử lập trình. Toán hạng có thể là Y, M, HM, S, HS, T, HT, C, HC, TD, HTD, CD, HCD, D, HD

Lệnh FMOV có thể chuyển 0 sang các phần tử lập trình này: DX, DY, DM, DS, T(TD), HT(HTD), C(CD), HC(HCD), D, HD.

4-5-11 Lệnh hoán đổi byte cao và thấp [SWAP]

1) Sơ lược

Hoán đổi byte cao và thấp của thanh ghi được chỉ định

Hoán đổi byte cao và thấp [SWAP]			
16 bits	SWAP	32 bits	-
Điều kiện thực thi	Falling or rising pulse edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

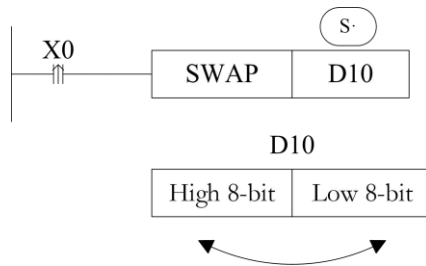
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình	16 bits; BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit								
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•		•	•														

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- Trao đổi thanh ghi 16 bit cao và 8 bit thấp.
- Nếu lệnh này được kích hoạt bởi cuộn dây Thường hở/đóng, lệnh sẽ được thực thi trong mỗi chu kỳ quét khi X0 ở trạng thái ON. Nên sử dụng xung sườn lên/ sườn xuống để kích hoạt lệnh.

4-5-12 Lệnh trao đổi [XCH, DXCH]

1) Sơ lược

Trao đổi dữ liệu trong hai phân tử lập trình

Lệnh trao đổi [XCH, DXCH]			
16 bits	XCH	32 bits	DXCH
Điều kiện thực thi	Xung sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D1	Địa chỉ phân tử lập trình	16 bits/32 bits, BIN
D2	Địa chỉ phân tử lập trình	16 bits/32 bits, BIN

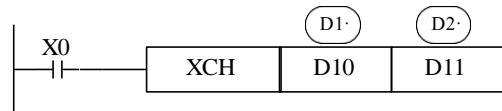
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
	D	D	D	X	Y	M	S		D	D								
D1	•		•	•		•	•	•										
D2	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

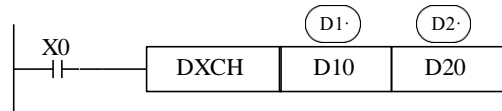
<Lệnh 16 bit>



Before (D10)=100 → After (D10)=101
 (D11)=101 (D11)=100

- Nội dung của hai thiết bị đích D1 và D2 được hoán đổi,
- Khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét. Nên sử dụng xung sườn lên/sườn xuống để kích hoạt lệnh.

<lệnh 32 bit>



Lệnh 32 bit [DXCH] hoán đổi giá trị dword của D10,D11 và D20, D21.

Trước (D10)=100	→ Sau (D10)=200
(D11)=1 (D11D10)=65636	(D11)=10 (D11D10)=655460
(D20)=200	(D20)=100
(D21)=10 (D21D20)=655460	(D21)=1 (D21D20)=65636

4-6 Lệnh phép toán dữ liệu

Tên gọi nhớ	Chức năng	Chương
ADD	Cộng	4-6-1
SUB	Trừ	4-6-2
MUL	Nhân	4-6-3
DIV	Chia	4-6-4
INC	Tăng giá trị	4-6-5
DEC	Giảm giá trị	4-6-5
MEAN	Lấy giá trị trung bình	4-6-6
WAND	Logic Word And	4-6-7
WOR	Logic Word Or	4-6-7
WXOR	Logic Exclusive Or	4-6-7
CML	Phép đảo trạng thái	4-6-8
NEG	Lấy dấu -	4-6-9

4-6-1 Lệnh phép cộng [ADD, DADD, QADD]

1) Sơ lược

Cộng 2 số và lưu kết quả

Phép cộng [ADD, DADD, QADD]			
16 bits	ADD	32 bits	DADD
Điều kiện thực thi	Thường mở/đóng; xung sườn lên/sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QADD		
Điều kiện thực thi	Thường mở/đóng; xung sườn lên/sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
3 toán hạng		
S1	Địa chỉ dữ liệu phép cộng	16 bits/32 bits/64 bits, BIN
S2	Địa chỉ dữ liệu phép cộng	16 bits/32bit/64 bits, BIN
D	Địa chỉ kết quả	16 bits/32bit/64 bits, BIN
Hai toán hạng		
D	Dữ liệu được cộng thêm và địa chỉ dữ liệu kết quả	16 bits/32 bits/64 bits, BIN
S1	Địa chỉ dữ liệu cộng	16 bits/32 bits/64 bits, BIN

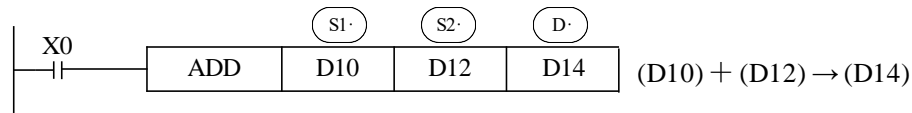
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
3 toán hạng																		
S1	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										
2 toán hạng																		
D	•																	
S1	•	•							•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

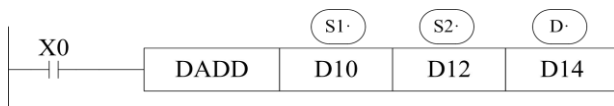
Mô tả

<3 toán hạng>



- Hai dữ liệu nguồn thực hiện phép cộng nhị phân và gửi kết quả đến địa chỉ đích. Bit cao nhất của mỗi dữ liệu là bit dấu, 0 là +, 1 là -. Tất cả các tính toán đều được xử lý đại số. $(5 + (-8) = -3)$
- Nếu kết quả tính toán là “0”, cờ “0” sẽ hoạt động. Nếu kết quả vượt quá 323767 (phép toán 16 bit) hoặc 2147483647 (phép toán 32 bit) hoặc 9223372036854775807 (phép toán 64 bit), cờ nhớ (carry flag) sẽ hoạt động (tham khảo Cờ liên quan- Related flag). Nếu kết quả vượt quá -323768 (hoạt động 16 bit) hoặc -2147483648 (phép toán 32 bit) hoặc -9223372036854775808 (phép toán 64 bit), cờ mượn (borrow flag) sẽ hoạt động (tham khảo Cờ liên quan).
- Khi thực hiện phép tính 32/64 bit, phía dưới 16 bit của phần tử lập trình dạng word được chỉ định và phần tử lập trình được gán địa chỉ tiếp theo sẽ được sử dụng làm vị trí cao. Để tránh lặp lại ID, nên gán địa chỉ phần tử lập trình bằng số chẵn.

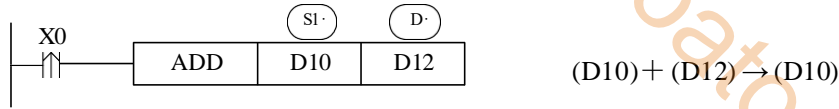
Ví dụ: ký hiệu 32 bit của ví dụ trước được hiển thị trong hình sau. Trong phép toán 32 bit, địa chỉ của số hạng thứ hai phải bắt đầu từ D12 vì số hạng thứ nhất chiếm các thanh ghi D10 và D11. Để tránh việc các thanh ghi bị chiếm giữ nhiều lần, nên gán địa chỉ các phần tử lập trình là số chẵn.



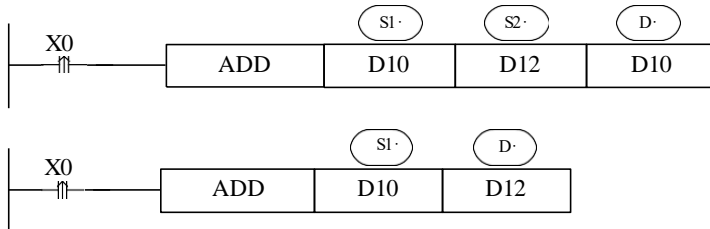
$(D11, D10) + (D13, D12) \rightarrow (D15, D14)$

- Địa chỉ nguồn và đích có thể giống nhau. Trong ví dụ trên, khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét.

<Hai toán hạng>



- Hai dữ liệu nguồn thực hiện phép cộng nhị phân và gửi kết quả đến địa chỉ dữ liệu số hạng. Bit cao nhất của mỗi dữ liệu là bit dấu, 0 là dương, 1 là âm. Tất cả các tính toán đều được xử lý đại số. $(5 + (-8) = -3)$
- Nếu kết quả tính toán là “0”, cờ “0” sẽ hoạt động. Nếu kết quả vượt quá 323767 (phép toán 16 bit) hoặc 2147483647 (phép toán 32 bit) hoặc 9223372036854775807 (phép toán 64 bit), cờ nhớ sẽ hoạt động (tham khảo Cờ liên quan). Nếu kết quả vượt quá -323768 (phép toán 16 bit) hoặc -2147483648 (phép toán 32 bit) hoặc -9223372036854775808 (phép toán 64 bit), cờ mượn sẽ hoạt động (tham khảo Cờ liên quan).
- Khi thực hiện phép tính 32/64 bit, phía dưới 16 bit của phần tử lập trình dạng word được chỉ định và phần tử lập trình được gán địa chỉ tiếp theo sẽ được sử dụng làm vị trí cao. Để tránh lặp lại ID, nên gán địa chỉ phần tử lập trình bằng số chẵn.
- Lưu ý: Địa chỉ của các toán hạng trong lệnh QADD phải là số chẵn.
- Trong ví dụ trên, khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét. Nên sử dụng xung sườn lên hoặc xung sườn giảm để kích hoạt lệnh.



Hai lệnh đều giống nhau.

Cờ liên quan

Ý nghĩa của Cờ

Cờ	Tên Cờ	Chức năng
SM020	Zero (0)	ON: kết quả tính toán bằng 0 OFF: kết quả tính toán khác 0
SM021	Borrow (Cờ mượn)	ON: kết quả tính toán vượt quá -32768 (16 bit) hoặc -2147483648 (32 bit) hoặc $-9,223,372,036,854,775,808$ (64 bit), bit cờ mượn sẽ hoạt động OFF: kết quả tính toán nhỏ hơn -32768 (16 bit) hoặc -2147483648 (32 bit) hoặc $-9,223,372,036,854,775,808$ (64 bit)
SM022	Carry (Cờ nhớ)	ON: kết quả tính toán lớn hơn 32768 (16 bit) hoặc 2147483648 (32 bit) hoặc $9,223,372,036,854,775,807$ (64 bit), bit cờ nhớ sẽ hoạt động. OFF: kết quả tính toán nhỏ hơn 32768 (16 bit) hoặc 2147483648 (32 bit) hoặc $9,223,372,036,854,775,807$ (64 bit)

4-6-2 Lệnh phép trừ [SUB,DSUB,QSUB]

1) Sơ lược

Thực hiện phép trừ hai số, lưu kết quả

Phép trừ [SUB, DSUB, QSUB]			
16 bits	SUB	32 bits	DSUB
Điều kiện thực thi	Thường mở/đóng; xung sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QSUB		
Điều kiện thực thi	Thường mở/đóng; xung sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

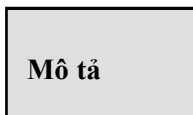
2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
3 toán hạng		
S1	Địa chỉ dữ liệu phép trừ	16 bits /32 bits/64 bits, BIN
S2	Địa chỉ dữ liệu phép trừ	16 bits /32 bits/64 bits, BIN
D	Địa chỉ dữ liệu kết quả	16 bits /32 bits/64 bits, BIN
2 toán hạng		
D	Dữ liệu bị trừ và địa chỉ kết quả	16 bits /32 bits/64 bits, BIN
S1	Địa chỉ dữ liệu trừ	16 bits /32 bits/64 bits, BIN

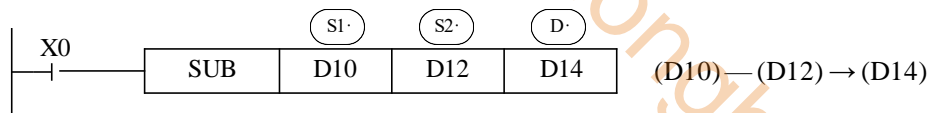
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun	Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C
3 toán hạng																	
S1	•	•	•	•	•	•	•	•	•								
S2	•	•	•	•	•	•	•	•	•								
D	•		•	•		•	•	•									
2 toán hạng																	
D	•																
S1	•	•							•								

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

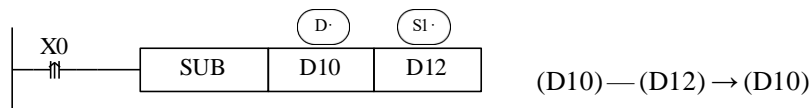


<3 toán hạng>

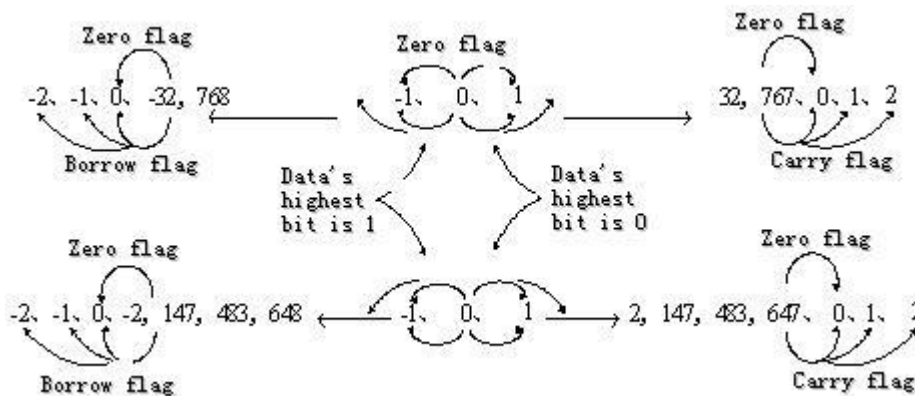


- S1 chỉ định dữ liệu phần tử lập trình, trừ đi dữ liệu phần tử lập trình do S2 chỉ định theo phương pháp đại số. Kết quả sẽ được lưu trữ trong phần tử lập trình do D chỉ định.
- Hoạt động của mỗi cờ, phương pháp đặt các phần tử lập trình của phép toán 32/64 bit tương tự như lệnh ADD trước đó.
- Lưu ý quan trọng: trong chương trình trước, nếu X0 ở trạng thái ON, lệnh SUB sẽ được thực thi sau mỗi chu kỳ quét.
- Tham khảo chương 4-6-1 để tìm hiểu về hành động và chức năng của cờ.

<2 toán hạng>



- D chỉ định dữ liệu của phần tử lập trình, trừ đi dữ liệu của phần tử lập trình do S1 chỉ định theo phương pháp đại số. Kết quả sẽ được lưu trữ trong phần tử lập trình do D chỉ định.
- Hoạt động của mỗi cờ, phương pháp đặt các phần tử lập trình của phép toán 32/64 bit tương tự như lệnh ADD trước đó.
- Lưu ý quan trọng: trong chương trình trước, nếu X0 ở trạng thái ON, lệnh SUB sẽ được thực thi sau mỗi chu kỳ quét. Nên sử dụng xung sườn lên/xuống để kích hoạt lệnh.
- Tham khảo chương 4-6-1 để tìm hiểu về hành động và chức năng của cờ. Mối quan hệ giữa hành động của cờ và +/- của giá trị được hiển thị bên dưới:



Lưu ý: Địa chỉ của các toán hạng trong lệnh QSUB phải là địa chỉ chẵn.

4-6-3 Lệnh phép nhân [MUL, DMUL, QMUL]

1) Sơ lược

Nhân hai số, lưu kết quả

Phép nhân [MUL, DMUL, QMUL]			
16 bits	MUL	32 bits	DMUL

Điều kiện thực thi	Thường mở/đóng; xung sườn coil	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QMUL		
Điều kiện thực thi	Thường mở/đóng, xung sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép nhân	16 bits /32 bits/64 bits, BIN
S2	Địa chỉ dữ liệu phép nhân	16 bits /32 bits/64 bits, BIN
D	Địa chỉ kết quả	16 bits /32 bits/64 bits, BIN

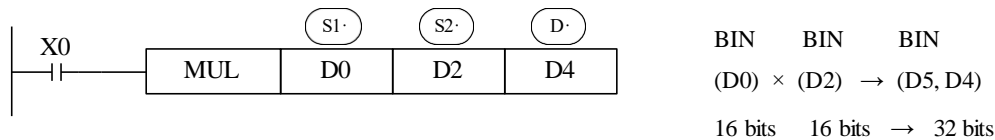
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hãng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•											

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

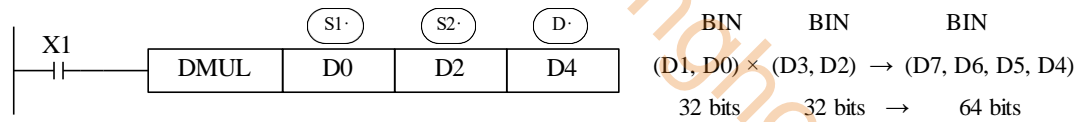
Mô tả

<Phép toán 16 bit>



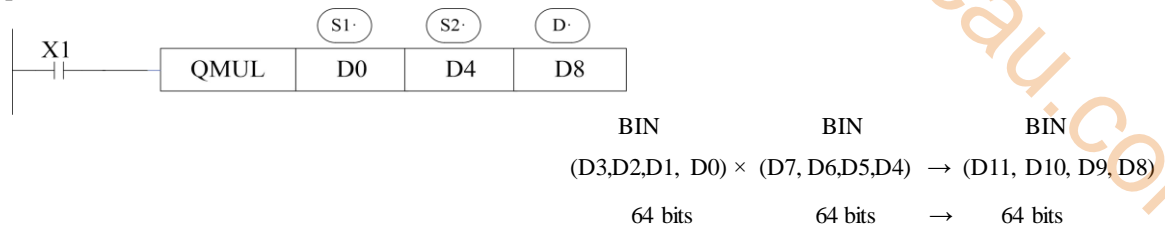
- Dữ liệu của hai thiết bị nguồn được nhân với nhau và kết quả được lưu trữ tại thiết bị đích ở định dạng 32 bit. Như sơ đồ trên: khi (D0)=8, (D2)=9, (D5, D4)=72.
- Bit cao nhất của kết quả là bit dấu: + (0), - (1).
- Trong ví dụ trên, khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét.

<Phép toán 32 bit>



- Khi sử dụng phép toán 32 bit, kết quả được lưu trữ
- Ngay cả khi sử dụng phần tử lập trình dạng word cũng không thể kiểm soát được kết quả 64 bit. Vui lòng chuyển sang phép toán giá trị số thực dấu phẩy động cho trường hợp này.

<Phép toán 64 bit>



- Trong phép toán 64 bit, địa chỉ đích sử dụng phần tử mềm bit để nhận kết quả 64 bit (chiếm bốn thanh ghi liên tiếp, vì vậy đừng sử dụng lại chúng). Khi sử dụng thành phần dạng word, không thể giám sát trực tiếp kết quả của phép toán dữ liệu 64-bit. Số thực dấu phẩy động được khuyến nghị trong trường hợp này.
- Lưu ý: Địa chỉ của các toán hạng trong lệnh QMUL phải là địa chỉ chẵn.

4-6-4 Lệnh phép chia [DIV, DDIV, QDIV]

1) Sơ lược

Chia 2 số và lưu kết quả

Phép chia [DIV, DDIV, QDIV]			
16 bits	DIV	32 bits	DDIV
Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QDIV		
Điều kiện thực thi	Thường mở/đóng; sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép chia	16 bits /32 bits/64 bits, BIN
S2	Địa chỉ dữ liệu phép chia	16 bits /32 bits/64 bits, BIN
D	Địa chỉ kết quả	16 bits /32 bits/64 bits, BIN

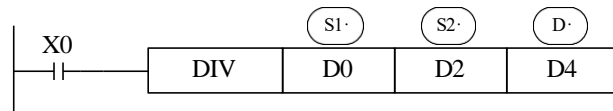
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

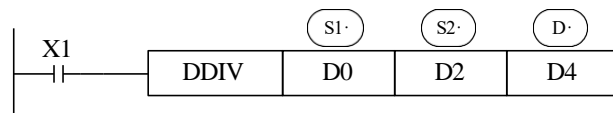
Mô tả

< Phép toán 16 bit >



Số bị chia	Số chia	Thương	Số dư
BIN	BIN	BIN	BIN
(D0)	÷ (D2)	→ (D4)	--- (D4)
16 bits	16 bits	16 bits	16 bits

- S1 chỉ định địa chỉ phần tử lập trình (số bị chia), S2 chỉ định địa chỉ phần tử lập trình (số chia) và D chỉ định địa chỉ thành phần phần mềm và địa chỉ tiếp theo của thành phần phần mềm sẽ làm kết quả và số dư.
 - Trong ví dụ trên, đầu vào X0 ở trạng thái ON, lệnh chia được thực hiện sau mỗi chu kỳ quét.
- < Phép toán 32 bit >

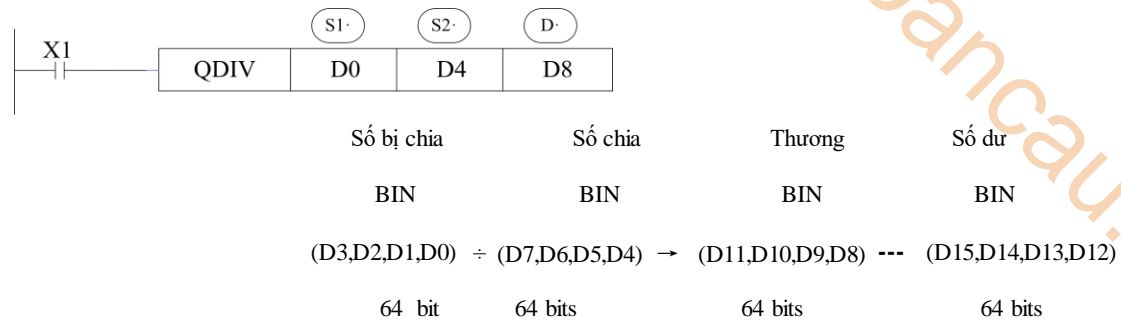


Số bị chia	Số chia	Thương	Số dư
BIN	BIN	BIN	BIN
(D1,D0)	÷ (D3,D2)	→ (D5,D4)	--- (D7,D6)
32 bits	32 bits	32 bits	32 bits

- Số bị chia được chỉ định bởi phần tử lập trình do S1 chỉ định và phần tử lập trình tiếp theo. Số chia được chỉ định bởi phần tử lập trình do S2 chỉ định và phần tử lập trình tiếp theo. Thương (kết quả) và số dư được lưu trữ trong bốn phần tử lập trình liên tiếp, phần tử lập trình đầu tiên do D chỉ định.

- Nếu giá trị của số chia là 0 thì lệnh sẽ bị lỗi.
- Bit cao nhất của kết quả và số dư là bit dấu (+: 0, -: 1). Khi bất kỳ số bị chia hoặc số chia nào âm thì kết quả sẽ âm. Khi số bị chia âm thì số dư sẽ âm.

<Phép toán 64 bit>



- Số bị chia được tạo thành bởi thiết bị do S1 chỉ định và thiết bị tiếp theo. Số chia được tạo bởi thiết bị do S2 chỉ định và thiết bị tiếp theo. Kết quả và số dư được lưu trữ trong bốn thiết bị liên tiếp, thiết bị đầu tiên do D chỉ định.
- Nếu giá trị của số chia là 0 thì lệnh sẽ bị lỗi.
- Bit cao nhất của kết quả và số dư là bit dấu (+: 0, -: 1). Khi bất kỳ số bị chia hoặc số chia nào âm thì kết quả sẽ âm. Khi số bị chia âm thì số dư sẽ âm.
- Lưu ý: Địa chỉ của các toán hạng trong lệnh QDIV phải là địa chỉ chẵn.

4-6-5 Lệnh tăng giá trị [INC, DINC, QINC] & giảm giá trị [DEC, DDEC, QDEC]

1) Sơ lược

Tăng hoặc giảm giá trị

Increase one [INC,DINC,QINC]			
16 bits	INC	32 bits	DINC
Điều kiện thực thi	Thường đóng/ mở; sườn lên/sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	QINC		
Điều kiện thực thi	Thường mở/đóng; sườn lên/sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên
Decrease one [DEC,DDEC,QDEC]			
16 bits	DEC	32 bits	DDEC
Điều kiện thực thi	Thường mở/đóng, sườn tăng/ sườn giảm	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

64 bits	QDEC		
Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ dữ liệu tăng hoặc giảm	16 bits / 32 bits/64 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.
D	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

<Lệnh tăng [INC]>



- D sẽ tăng 1 khi X0 ở trạng thái ON.
- Đối với phép toán 16 bit, khi +32767 tăng 1, nó sẽ trở thành -32768; Bit cờ sẽ hoạt động. Đối với lệnh 32 bit, +2147483647 tăng 1 là -2147483648. Bit cờ sẽ hoạt động. Đối với phép toán 64 bit, +9223372036854775807 tăng 1 là -9223372036854775808. Bit cờ sẽ hoạt động.

<Lệnh giảm [DEC]>



- D sẽ giảm đi 1 khi X1 ở trạng thái ON.
- Đối với phép toán 16 bit, -32767 hoặc -2147483647 giảm đi 1, kết quả sẽ là +32767 hoặc +2147483647. Bit cờ sẽ hoạt động. Đối với phép toán 64 bit, -9223372036854775808 giảm đi 1 sẽ là +9223372036854775807. Bit cờ sẽ hoạt động.
- Địa chỉ của các toán hạng trong lệnh QINC và QDEC phải là địa chỉ chẵn.

Lưu ý: Khi lệnh được kích hoạt bởi xung sườn, thao tác cộng và trừ tự động được thực hiện cho mỗi kích hoạt. Nếu nó được kích hoạt bởi cuộn dây thường mở/ đóng, thao tác tự động cộng và trừ tự động sẽ được thực hiện trong mỗi chu kỳ quét sau khi dẫn truyền.

4-6-6 Lệnh lấy giá trị trung bình [MEAN, DMEAN]

1) Sơ lược

Lấy giá trị trung bình của dữ liệu

Lệnh lấy giá trị trung bình [MEAN,DMEAN]			
16 bits	MEAN	32 bits	DMEAN
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

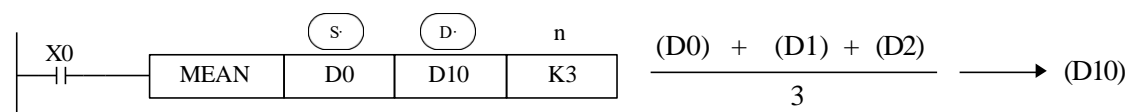
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ dữ liệu nguồn	16 bits/32 bits, BIN
D	Địa chỉ kết quả trung bình	16 bits/32 bits, BIN
n	Số lượng dữ liệu data quantity	16 bits/32 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.
S	•	•	•	•														
D	•	•	•	•														
n									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



- Lưu trữ giá trị trung bình của dữ liệu nguồn (tổng nguồn chia cho số lượng nguồn n). đưa số dư.
- n không thể lớn hơn số lượng phân tử lập trình, nếu không sẽ xảy ra lỗi.

4-6-7 Logic AND [WAND, DWAND], Logic OR[WOR, DWOR], Logic Exclusive OR [WXOR, DWXOR]

1) Sơ lược

Thực hiện phép toán logic AND, OR, XOR cho dữ liệu

Logic AND [WAND, DWAND]			
16 bits	WAND	32 bits	DWAND
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Logic OR[WOR,DWOR]			
16 bits	WOR	32 bits	DWOR
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Logic Exclusive OR [WXOR,DWXOR]			
16 bits	WXOR	32 bits	DWXOR
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép toán	16bits/32bits,BIN
S2	Địa chỉ dữ liệu phép toán	16bits/32bits,BIN
D	Địa chỉ kết quả	16bits/32bits,BIN

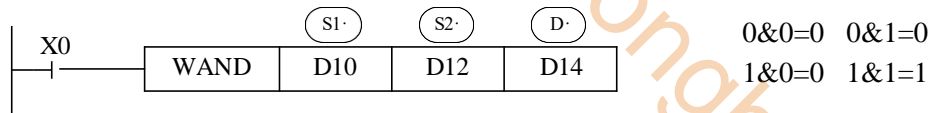
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

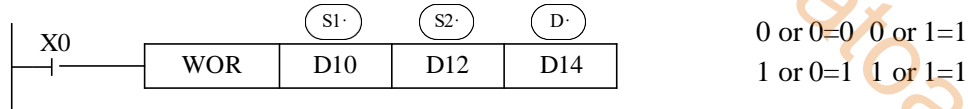
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

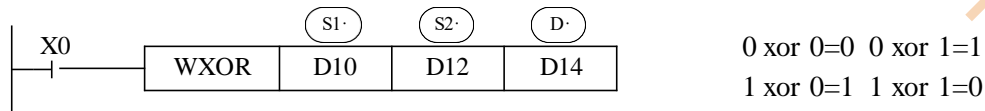
<Logic AND >



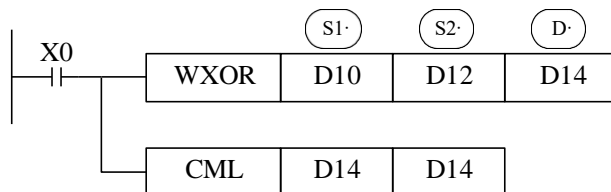
<Logic OR >



< Logic WXOR >

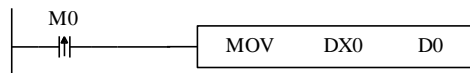


Nếu sử dụng lệnh này cùng với lệnh CML, lệnh XOR NOT cũng có thể được thực thi.



Ví dụ 1:

Dữ liệu 16 bit được tạo bởi X0~X7 và được lưu trữ trong D0.

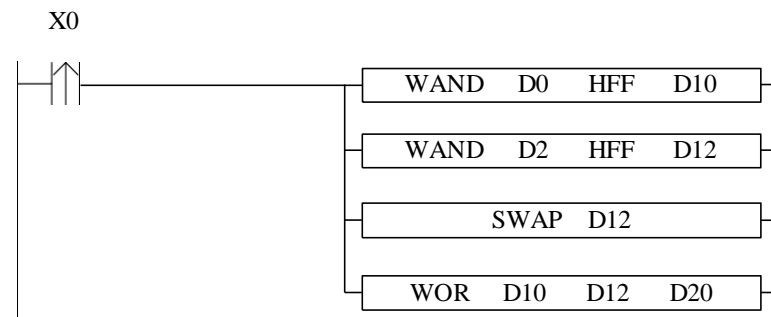


Chuyển đổi trạng thái của mã X0, X1, X2, X3 sang mã 8421 và lưu vào D0.



Ví dụ 2:

Kết hợp 8 bit thấp của D0 và D2 thành một word.



```
LDP X0
WAND D0 HFF D10 //X0 sườn tăng
WAND D2 HFF D12 //Logic and, lấy 8 bit thấp của D0 và lưu vào D10
SWAP D12 // Logic and, lấy 8 bit thấp của D2 và lưu vào D12
WOR D10 D12 D20 //hoán đổi 8 bit thấp và 8 bit cao của D12
//kết hợp 8 bit thấp của D10 và 8 bit cao của D12, và
lưu vào D20
```

4-6-8 Lệnh đảo logic dữ liệu [CML, DCML]

1) Sơ lược

Logic đảo dữ liệu

Converse [CML,DCML]			
16 bits	CML	32 bits	DCML
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

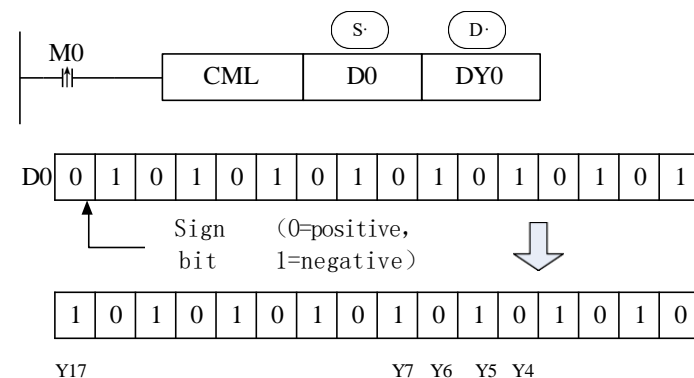
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ dữ liệu nguồn	16 bits/32 bits, BIN
D	Địa chỉ kết quả	16 bits/32 bits, BIN

3) Phân tử lập trình phù hợp

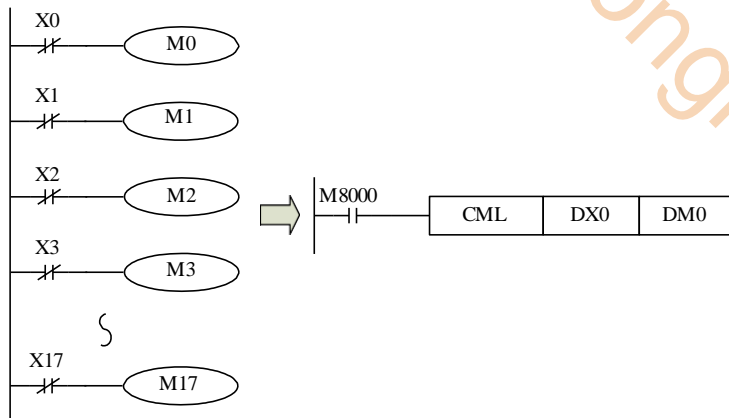
Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- Mỗi bit dữ liệu trong thiết bị nguồn được đảo ngược (1→0, 0→1) và gửi đến thiết bị đích. Nếu sử dụng hằng số K trong thiết bị nguồn, nó có thể tự động chuyển đổi thành nhị phân.
- This instruction is fit for PLC logical converse output.
- Lệnh này phù hợp với đầu ra đảo ngược logic PLC



- Lệnh điều khiển tuần tự ở bên trái có thể được biểu thị bằng lệnh CML sau.

4-6-9 Lệnh lấy dấu âm [NEG, DNEG]

1) Sơ lược

Lấy dữ liệu âm

Negative [NEG,DNEG]			
16 bits	NEG	32 bits	DNEG
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

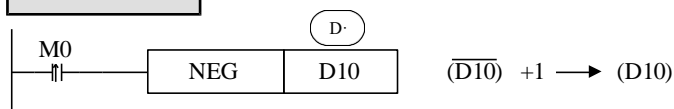
Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ dữ liệu nguồn	16 bits/ 32 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

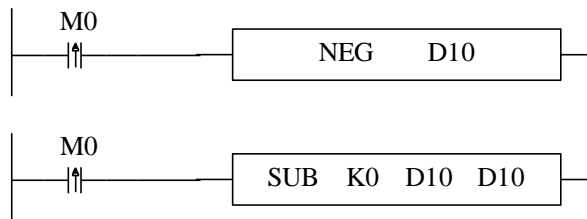


- Đảo ngược từng bit của dữ liệu nguồn (1→0, 0→1), sau đó cộng 1 và lưu kết quả vào địa

chỉ dữ liệu nguồn.

- Ví dụ: dữ liệu nguồn D10 là 20, khi xung sườn lên M0 xuất hiện, D10 trở thành -20.

Hai lệnh sau đây đều giống nhau.



4-7 Các lệnh dịch chuyển dữ liệu

Tên gọi nhớ	Chức năng	Chương
SHL	Dịch dữ liệu số học sang trái	4-7-1
SHR	Dịch dữ liệu số học sang phải	4-7-1
LSL	Dịch chuyển logic sang trái	4-7-2
LSR	Dịch chuyển logic sang phải	4-7-2
ROL	Xoay trái	4-7-3
ROR	Xoay phải	4-7-3
SFTL	Dịch chuyển bit sang trái	4-7-4
SFTR	Dịch chuyển bit sang phải	4-7-5
WSFL	Dịch chuyển word sang trái	4-7-6
WSFR	Dịch chuyển word sang phải	4-7-7

4-7-1 Lệnh dịch chuyển số học sang trái [SHL,DSHL], sang phải [SHR,DSHR]

1) Sơ lược

Thực hiện dịch chuyển số học sang trái/phải cho các số

Dịch chuyển số học sang trái [SHL,DSHL]			
16 bits	SHL	32 bits	DSHL
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Dịch chuyển số học sang phải [SHR,DSHR]			
16 bits	SHR	32 bits	DSHR
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa điểm dữ liệu nguồn	16 bits/32 bits, BIN
n	Số lần dịch sang phải/ sang trái	16 bits/32bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit										
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
D	•	•	•	•		•	•	•											
n									•										

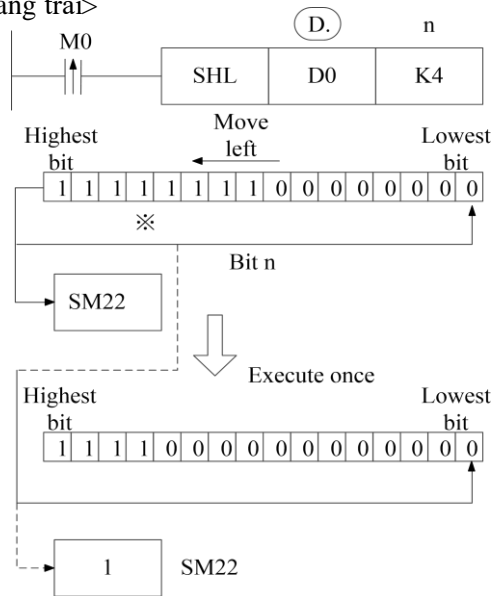
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

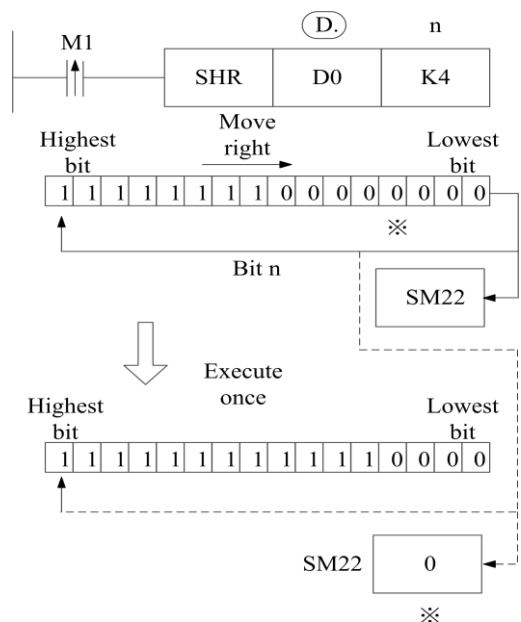
Sau khi thực thi lệnh SHL một lần, bit thấp nhất được điền bằng 0, bit cuối cùng được lưu trong cờ nhớ.

Sau khi thực thi lệnh SHR một lần, bit cao nhất vẫn giữ nguyên; bit cuối cùng được lưu trong cờ nhớ.

< Dịch chuyển số học sang trái >



< Dịch chuyển số học sang phải >



4-7-2 Lệnh dịch chuyển logic sang trái [LSL], sang phải [LSR]

1) Sơ lược

Thực hiện dịch chuyển logic sang phải/trái cho dữ liệu

Logic shift left [LSL, DLSL]			
16 bits	LSL	32 bits	DLSL
Điều kiện thực thi	Thường mở/đóng, rising/falling edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Logic shift right [LSR, DLSR]			
16 bits	LSR	32 bits	DLSR
Điều kiện thực thi	Thường mở/đóng, rising/falling edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ dữ liệu nguồn	16 bits/32 bits, BIN
n	Arithmetic shift left/right times	16 bits/32bit, BIN

3) Phần tử lập trình phù hợp

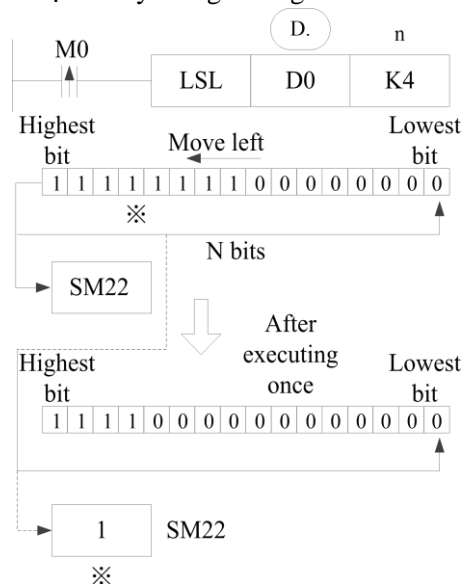
Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	•	•	•	•		•	•	•										
n									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

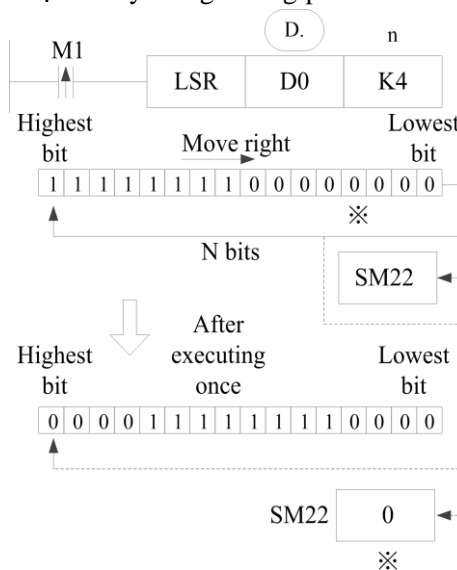
Mô tả

- Sau khi thực hiện LSL một lần, bit thấp nhất được điền bằng 0; bit cuối cùng được lưu trong cờ nhớ.
- Ý nghĩa và hoạt động của lệnh LSL giống với lệnh SHL.
- Sau khi thực thi lệnh LSR một lần, bit cao nhất được điền bằng 0; bit cuối cùng được lưu trong cờ nhớ.
- Lệnh LSR và SHR khác nhau, lệnh LSR thêm 0 vào bit cao nhất khi dịch chuyển, lệnh SHR tắt cả các bit đều được dịch chuyển.

< Dịch chuyển logic sang trái >



< Dịch chuyển logic sang phải >



4-7-3 Xoay trái [ROL, DROL], Xoay phải [ROR, DROR]

1) Sơ lược

Xoay trái, xoay phải

Xoay trái [ROL, DROL]			
16 bits	ROL	32 bits	DROL
Điều kiện thực thi	Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
Rotation shift right [ROR, DROR]			
16 bits	ROR	32 bits	DROR
Điều kiện thực thi	Sườn lên/ sườn xuống	Model phù hợp	XD, XL

Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
-------------------	---	------------------	---

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ dữ liệu nguồn	16 bits/32 bits, BIN
n	Số lần xoay sang trái/phải	16 bits/32 bits, BIN

3) Phần tử lập trình phù hợp

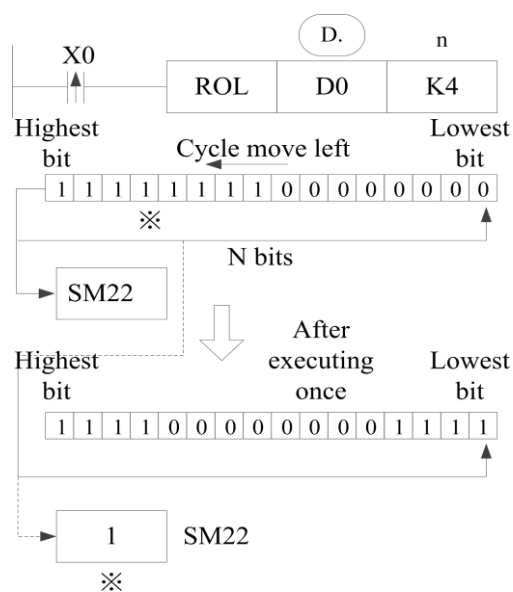
Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	•	•	•	•		•	•	•										
n									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

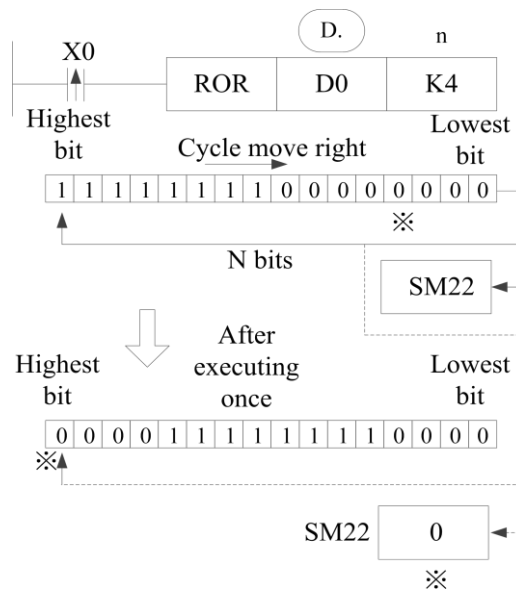
Mô tả

- Khi X0 thay đổi từ trạng thái OFF sang ON, giá trị sẽ được xoay sang trái hoặc sang phải, bit cuối cùng được lưu trong cờ nhớ.

< Xoay sang trái >



< Xoay sang phải >



4-7-4 Dịch chuyển bit sang trái [SFTL]

1) Sơ lược

Dịch chuyển bit sang trái

Dịch chuyển bit sang trái [SFTL]			
16 bits	SFTL	32 bits	-
Điều kiện thực thi	Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Bit địa chỉ bắt đầu phần tử nguồn	bit
D	Bit địa chỉ bắt đầu phần tử đích	bit
n1	Số lượng dữ liệu nguồn (không quá 1024)	16 bits, BIN
n2	Số lần chuyển sang trái (không quá 1024)	16 bits, BIN

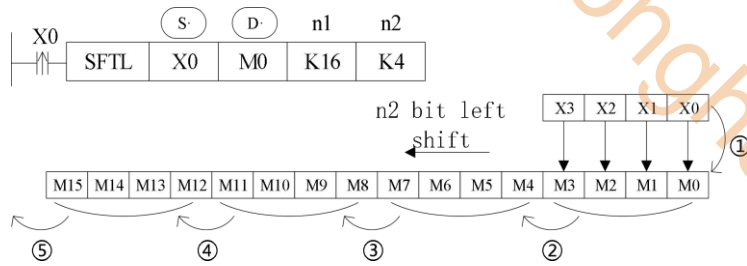
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hãng số	Mô-đun		Hệ						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S												•	•	•	•	•	•	
D													•	•	•	•	•	
n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Di chuyển n2 bit sang trái cho đối tượng chứa n1 bit.
- Khi X0 thay đổi từ trạng thái OFF sang ON, lệnh sẽ di chuyển n2 bit cho đối tượng.
- Ví dụ: nếu n2 là K1, đối tượng sẽ di chuyển sang trái 1 bit khi lệnh thực thi một lần.



- ① X 3~X 0→M3~M0
- ② M 3~M 0→M7~M4
- ③ M 7~M 4→M11~M8
- ④ M11~M 8→M15~M 12
- ⑤ M15~M12→Overflow

4-7-5 Dịch chuyển bit sang phải [SFTR]

1) Sơ lược

Dịch chuyển bit sang phải

Dịch chuyển bit sang phải [SFTR]			
16 bits	SFTR	32 bits	-
Điều kiện thực thi	rising/falling edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Bit địa chỉ bắt đầu phần tử nguồn	bit
D	Bit địa chỉ bắt đầu phần tử đích	bit
n1	Số lượng dữ liệu nguồn (không quá 1024)	16 bits, BIN
n2	Số lần chuyển sang phải (không quá 1024)	16 bits, BIN

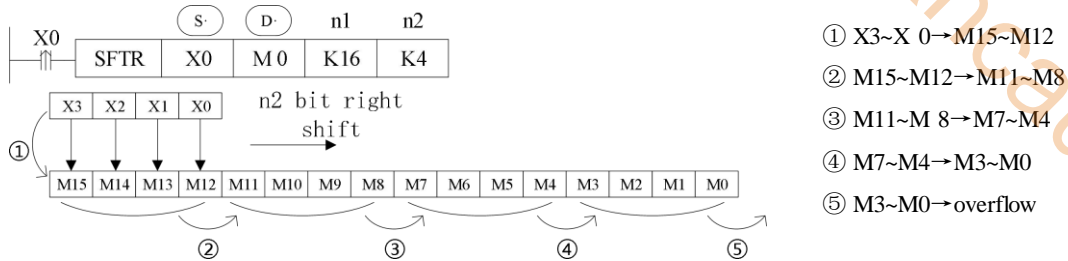
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit										
	Hệ								Hàng số	Mô-đun		Hệ									
	D	FD	TD	CD	DX	DY	DM	DS				K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S															•	•	•	•	•	•	
D																•	•	•	•	•	
n1	•		•	•	•	•	•	•	•												
n2	•		•	•	•	•	•	•	•												

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Di chuyển n2 bit sang phải đối tượng chứa n1 bit
- Khi X0 thay đổi từ trạng thái OFF sang ON, lệnh sẽ di chuyển n2 bit cho đối tượng
- Ví dụ: nếu n2 là 1, đối tượng sẽ di chuyển sang phải 1 bit khi lệnh thực thi một lần.



4-7-6 Dịch chuyển word sang trái [WSFL]

1) Sơ lược

Dịch chuyển word sang trái

Dịch chuyển word sang trái [WSFL]			
16 bits	WSFL	32 bits	-
Điều kiện thực thi	rising/falling edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Bit địa chỉ bắt đầu phần tử nguồn	16 bits, BIN
D	Bit địa chỉ bắt đầu phần tử đích	16 bits, BIN
n1	Số lượng dữ liệu nguồn (không quá 512)	16 bits, BIN
n2	Số lần dịch chuyển word sang trái (không quá 512)	16 bits, BIN

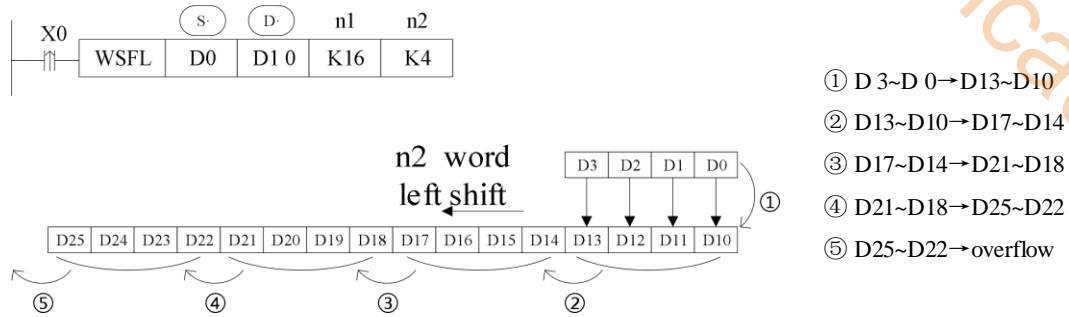
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hãng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•	•	•														
D	•	•	•	•														
n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Di chuyển n2 word sang trái cho đối tượng chứa n1 word.
- Khi X0 thay đổi từ trạng thái OFF sang ON, lệnh sẽ di chuyển n2 word cho đối tượng.



4-7-7 Dịch chuyển word sang phải [WSFR]

1) Sơ lược

Dịch chuyển word sang phải

Dịch chuyển word sang phải [WSFR]			
16 bits	WSFR	32 bits	-
Điều kiện thực thi	rising/falling edge	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Bit địa chỉ bắt đầu phần tử nguồn	16 bits, BIN
D	Bit địa chỉ bắt đầu phần tử đích	16 bits, BIN
n1	Số lượng dữ liệu nguồn (không quá 512)	16 bits, BIN
n2	Số lần dịch chuyển sang phải (không quá 512)	16 bits, BIN

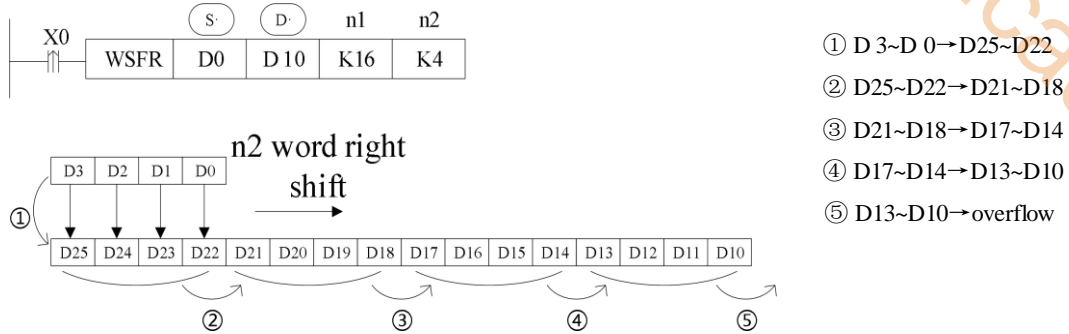
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•	•	•						D	D							
D	•	•	•	•														
n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Di chuyển n2 word sang phải cho đối tượng chứa n1 word.
- Khi X0 thay đổi từ trạng thái OFF sang ON, lệnh sẽ di chuyển n2 word cho đối tượng.



4-8 Các lệnh chuyển đổi dữ liệu

Tên gọi nhớ	Chức năng	Chương
WTD	Chuyển single word integer thành double word integer	4-8-1
DWTD	Chuyển double word integer thành four word integer	4-8-1
BDWTD	Chuyển 32 bits integer thành 64 bits integer chuyển đổi hàng loạt	4-8-2
FLT	Chuyển 16 bits integer converts thành số thực dấu phẩy động	4-8-3
DFLT	Chuyển 32 bits integer thành số thực dấu phẩy động	4-8-3
FLTD	Chuyển 64 bits integer thành số thực dấu phẩy động	4-8-3
DFLTD	Chuyển 32 bits integer thành double precision floating point	4-8-4
QFLTD	Chuyển 64 bits integer thành double precision floating point	4-8-4
INT	Chuyển float point thành integer	4-8-5
DINTD	Chuyển double - precision floating point thành 32 bits integer	4-8-6
QINTD	Chuyển double - precision floating point thành 64 bits integer	4-8-6
ECON	Chuyển single precision floating point thành double precision floating point	4-8-7
BECON	Chuyển single precision floating point thành double precision floating point batch conversion	4-8-8
BIN	Chuyển mã BCD thành nhị phân	4-8-9
BCD	Chuyển nhị phân thành mã BCD	4-8-10

ASCI	Chuyển Hex thành ASCII	4-8-11
HEX	Chuyển ASCII thành Hex	4-8-12
DECO	Lệnh giải mã	4-8-13
ENCO	Mã hóa bit cao	4-8-14
ENCOL	Mã hóa bit thấp	4-8-15
GRY	Chuyển nhị phân thành mã gray	4-8-16
GBIN	Chuyển mã gray thành nhị phân	4-8-17

4-8-1 Lệnh chuyển đổi single word integer thành double word integer [WTD.DWTD]

1) Sơ lược

Single word integer chuyển đổi thành double word integer [WTD.DWTD]			
16 bits	WTD		
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
32 bits	DWTD		
Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Version V3.7.1 or later	Yêu cầu phần mềm	Version V3.7.4a or later

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	16 bits/32 bits, BIN
D	Địa chỉ phần tử lập trình đích	32 bits/64 bits, BIN

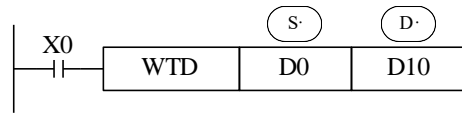
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										

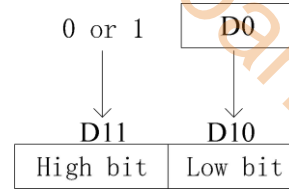
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

<lệnh 16 bit>

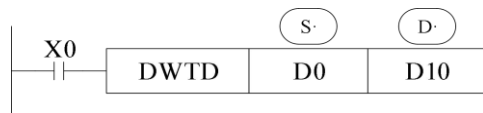


(D0) → (D11, D10)
Single Word Double Word

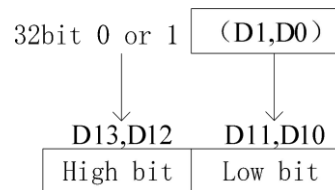


- Khi single word D0 là số nguyên dương, sau khi thực hiện lệnh này, bit cao của double word D10 là 0
- Khi single word D0 là số nguyên dương, sau khi thực thi lệnh này, bit cao của double word D10 là 0
- Khi single word D0 là số nguyên âm, sau khi thực thi lệnh này, bit cao của double word D10 là 1.
- bit cao 0 và 1 là giá trị nhị phân.

<lệnh 32 bit>



(D1,D0) → (D13,D12,D11,D10)
Double word four word



- Khi single word D0 là số nguyên dương, sau khi thực hiện lệnh này, bit cao của four word D10 là 0.
- Khi single word D0 là số nguyên âm, sau khi thực hiện lệnh này, bit cao của four word D10 là 1.
- bit cao 0 và 1 là giá trị nhị phân.

4-8-2 Lệnh chuyển đổi hàng loạt số nguyên 32 bit thành số nguyên 64 bit [BDWTD]

1) Sơ lược

Lệnh chuyển đổi hàng loạt 32 bits integer thành 64 bits integer [BDWTD]			
32 bits	BDWTD		
Điều kiện thực thi	Thường mở/đóng/sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định dữ liệu nguồn hoặc mã địa chỉ của thanh ghi	32 bits, BIN
D	Chỉ định mã địa chỉ của thành phần phần mềm mục tiêu	64 bits, BIN
N	Chỉ định giá trị của điểm chuyển	16 bits, BIN

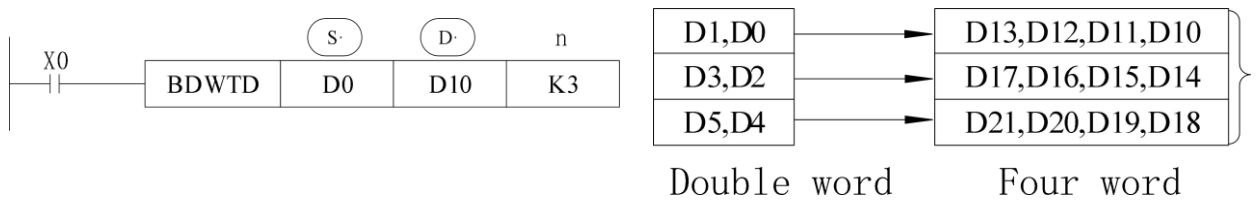
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•																
D	•																	
n	•		•	•	•			•	•									

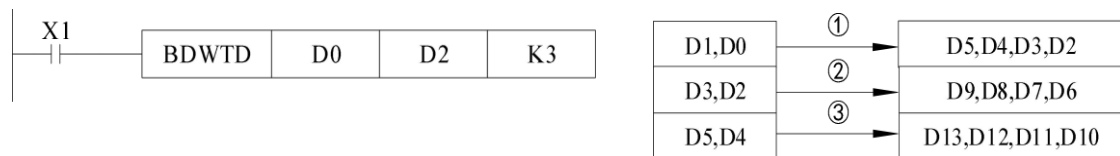
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Chuyển đổi dữ liệu n điểm bắt đầu bằng phần tử lập trình nguồn được chỉ định đến thành phần tử mềm n điểm bắt đầu bằng phần tử lập trình đích được chỉ định dưới dạng khối dữ liệu. (Khi vượt quá phạm vi số phần tử lập trình, hãy chuyển đổi trong phạm vi có thể).



- Theo phương pháp đánh số trùng lặp, các lệnh được truyền tự động theo thứ tự từ ① đến ③ nhằm tránh dữ liệu nguồn được chuyển bị ghi đè khi phạm vi số truyền trùng nhau như minh họa trong hình bên dưới.



Lưu ý: Địa chỉ của thanh ghi số nguyên 64 bit trong lệnh BDWTD phải là địa chỉ chẵn.

4-8-3 Lệnh chuyển số nguyên sang số thực dấu phẩy động [FLT, DFLT,FLTD]

1) Sơ lược

Chuyển đổi nguyên sang số thực dấu phẩy động [FLT, DFLT,FLTD]					
16 bits	FLT	32 bits	DFLT	64 bits	FLTD
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống		Model phù hợp	XD, XL	
Yêu cầu phần cứng	-		Yêu cầu phần mềm	-	

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình nguồn	16 bits/32 bits/64 bits, BIN
D	Địa chỉ phân tử lập trình đích	32 bits/64 bits, BIN

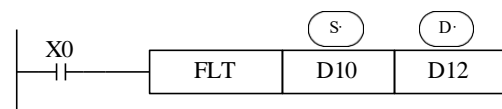
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•																
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

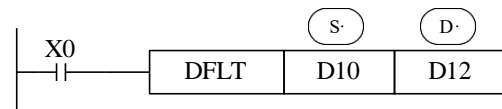
<Lệnh 16 bit>



(D10) → (D13,D12)

BIN integer Binary float point

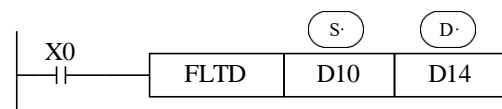
<Lệnh 32 bit>



(D11,D10) → (D13,D12)

BIN integer Binary float point

<Lệnh 64 bit>



(D13,D12,D11,D10) → (D15,D14)

BIN integer Binary float point

- Chuyển đổi dữ liệu dạng BIN integer thành binary floating point. Vì hằng số K, H sẽ tự động chuyển đổi bằng lệnh thao tác với số thực dấu phẩy động nên lệnh FLT này không thể sử dụng được.

- Lệnh chuyển đổi ngược của lệnh này là lệnh INT
- Lệnh FLTD có thể thay đổi số nguyên 64 bit thành giá trị số thực dấu phẩy động 32 bit.
- Toán hạng S của lệnh FLTD không hỗ trợ hằng số K/H.



D0 là số nguyên 20, sau khi thực hiện lệnh D10 có giá trị số thực dấu phẩy động là 20.

Lưu ý: Trước khi sử dụng các lệnh thao tác với số thực dấu phẩy động như lệnh EADD, ESUB, EMUL, EDIV, EMOV và ECMP, hãy đảm bảo rằng tất cả các tham số thao tác đều là số thực dấu phẩy động.

4-8-4 Lệnh chuyển integer thành double precision floating point [DFLTD, QFLTD]

1) Sơ lược

Chuyển integer thành double precision floating point [DFLTD, QFLTD]			
32 bits	DFLTD	64 bits	QFLTD
Điều kiện thực thi	Thường mở/đóng/, sườn lên/sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình nguồn	32 bits/64 bits, BIN
D	Địa chỉ phân tử lập trình đích	64 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun	Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C
S	•	•															
D	•																

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

<Lệnh 32 bit>



(D11, D10) → (D15, D14, D13, D12)
BIN integer Binary float point

<Lệnh 64 bit>



- Lệnh chuyển đổi các giá trị số nguyên nhị phân (binary integer) thành giá trị số thực dấu phẩy động nhị phân. Các hằng số K và H được tự động chuyển đổi trong mỗi lệnh thao tác với số thực dấu phẩy động và FLT không thể được sử dụng lệnh.
- Lệnh chuyển đổi ngược của lệnh này là lệnh DINTD/QINTD.
- Lệnh QFLTD chuyển đổi số nguyên 64 bit thành số thực dấu phẩy động 64 bit. Lưu ý địa chỉ của toán hạng trong lệnh QFLTD phải là số chẵn.)
- Toán hạng S của lệnh QFLTD không hỗ trợ hằng số K/H.

4-8-5 Lệnh chuyển số thực dấu phẩy động thành số nguyên [INT, DINT]

1) Sơ lược

Lệnh chuyển số thực dấu phẩy động thành số nguyên [INT, DINT]			
16 bits	INT	32 bits	DINT
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình nguồn	16 bits/32 bits, BIN
D	Địa chỉ phân tử lập trình đích	16 bits/32 bits, BIN

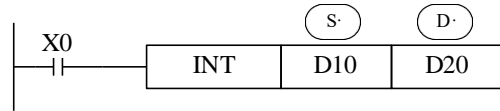
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun	Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C
S	•	•															
D	•																

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

<lệnh 16 bit>

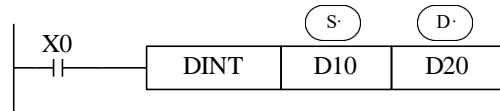


(D11,D10) → (D20)

Binary Float BIN integer

Bỏ dữ liệu sau dấu phẩy thập phân

<lệnh 32 bit >



(D11,D10) → (D20,D21)

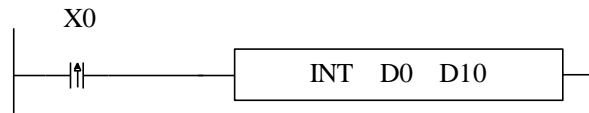
Binary Float BIN integer

Bỏ dữ liệu sau dấu phẩy thập phân

- Dữ liệu nguồn dạng nhị phân được chuyển đổi thành số nguyên BIN và được lưu trữ tại thiết bị đích. Bỏ giá trị đằng sau dấu thập phân.
- Lệnh đảo ngược của lệnh này là lệnh FLT.
- Khi kết quả là 0, bit cờ sẽ ở trạng thái ON.
- Khi kết quả là vượt quá giá trị dưới đây, cờ nhớ ở trạng thái ON.

Lệnh 16 bit: -32,768~32,767

Lệnh 32 bit: -2,147,483,648~2,147,483,647



Ví dụ: nếu D0 là giá trị số thực dấu phẩy động 130.2 thì sau khi thực hiện lệnh INT, giá trị D10 là số nguyên 130.

4-8-6 Lệnh chuyển double - precision floating point thành số nguyên [DINTD,QINTD]

1) Sơ lược

Chuyển floating point đến integer [DINTD,QINTD]			
32 bits	DINTD	64 bits	QINTD
Điều kiện thực thi	Thường mở/đóng/sườn lên/ sườn xuống	Model phù hợp	XDH、XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	64 bits, BIN
D	Địa chỉ phần tử lập trình đích	32 bits/64 bits, BIN

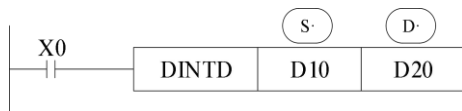
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun	Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•																
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

<lệnh 32 bit>



(D13,D12,D11,D10) → (D21,D20)
Binary Float BIN integer
Bỏ dữ liệu sau dấu phẩy thập phân

<lệnh 64 bit>



(D13,D12,D11,D10)→ (D23,D22,D21,D20)
Binary Float BIN integer Bỏ dữ liệu
sau dấu phẩy thập phân

- Dữ liệu nguồn dạng nhị phân được chuyển đổi thành số nguyên BIN và được lưu trữ tại thiết bị đích. Bỏ giá trị đằng sau dấu thập phân.
- Lệnh đảo ngược của lệnh này là lệnh DFLTD/QFLTD.
- Đối với lệnh 64-bit, số địa chỉ thanh ghi phải là số chẵn.
- Khi kết quả là 0, bit cờ sẽ ở trạng thái ON.
- Khi kết quả là vượt quá giá trị dưới đây, cờ nhớ ở trạng thái ON.
Lệnh 64 bit: -9223372036854775808~9223372036854775807.

4-8-7 Lệnh chuyển single precision floating point sang double precision floating point[ECON]

1) Sơ lược

Chuyển single precision floating point sang double precision floating point [ECON]			
32 bits	DINTD		
Điều kiện thực thi	Thường mở/đóng/sườn lên/ sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	32 bits, BIN
D	Địa chỉ phần tử lập trình đích	64 bits, BIN

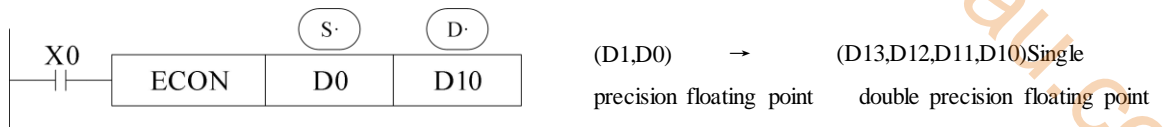
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word			Phần tử lập trình dạng bit
	Hệ	Hằng số	Mô-đun	Hệ

	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•																
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- Khi X0 ở trạng thái ON, giá trị single-precision floating-point trong địa chỉ dữ liệu nguồn được chuyển đổi thành giá trị double-precision floating-point và được lưu trong địa chỉ đích.
- Địa chỉ thanh ghi cho dữ liệu double-precision floating-point phải bắt đầu bằng số chẵn.

4-8-8 Lệnh chuyển hàng loạt single precision floating point sang double precision floating point [BECON]

1) Sơ lược

Chuyển hàng loạt single precision floating point sang double precision floating point [BECON]			
32 bits	BECON		
Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Chỉ định dữ liệu nguồn hoặc mã địa chỉ của thanh ghi	32 bits, BIN
D	Chỉ định mã địa chỉ của phần tử lập trình mục tiêu	64 bits, BIN
N	Chỉ định giá trị của điểm chuyển	16 bits, BIN

3) Phần tử lập trình phù hợp

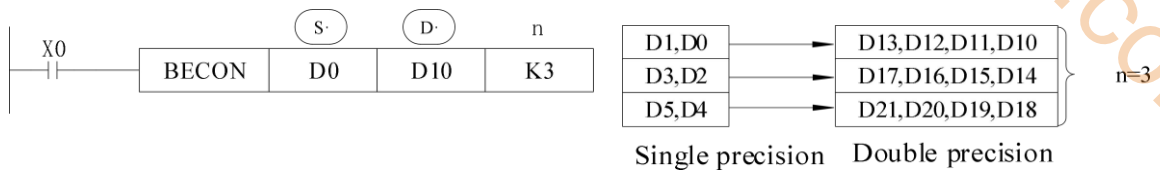
Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S	•	•																
D	•																	



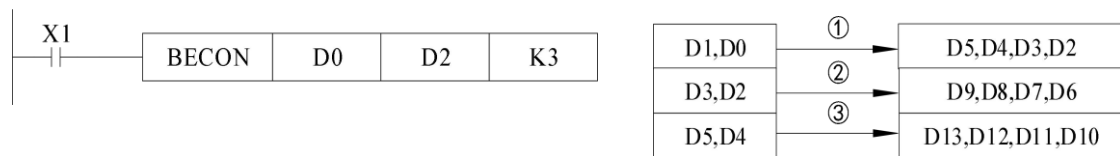
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

- Theo sơ đồ, dữ liệu n điểm bắt đầu bằng phần tử lập trình do nguồn chỉ định sẽ được truyền dưới dạng khối dữ liệu đến phần tử lập trình N điểm bắt đầu bằng phần tử lập trình do đích (mục tiêu) chỉ định. (Khi vượt quá phạm vi số phần tử lập trình, hãy chuyển đổi trong phạm vi có thể).



- Khi phạm vi số truyền bị lặp lại, để tránh việc ghi đè dữ liệu nguồn truyền mà không chuyển đổi, lệnh sẽ được truyền tự động theo thứ tự ①~③.



Lưu ý: Địa chỉ bắt đầu của thanh ghi dữ liệu double - precision floating-point phải là số chẵn

4-8-9 Lệnh chuyển BCD sang binary [BIN]

1) Sơ lược

Chuyển BCD sang binary [BIN]			
16 bits	BIN	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toáng hạng

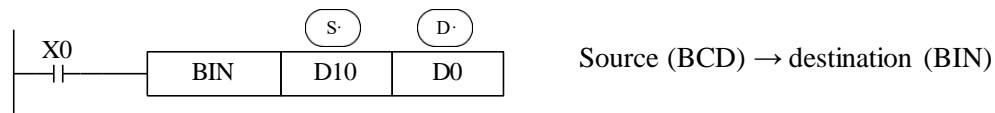
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	BCD
D	Địa chỉ phần tử lập trình đích	16 bits, BIN

3) Phần tử lập trình phù hợp

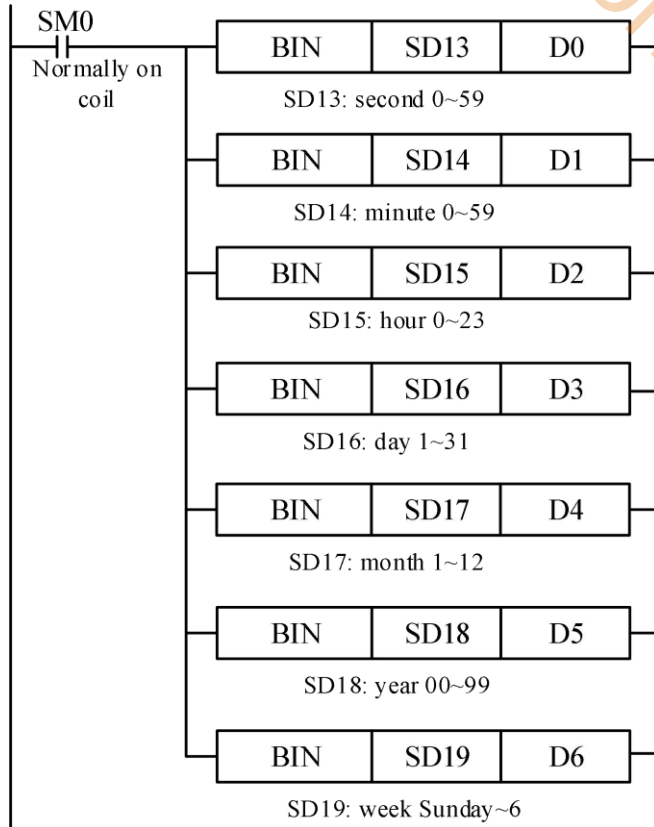
Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hãng số	Mô-đun		Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•	•	•	•	•	•	•											
D	•		•	•		•	•	•											

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- Nếu dữ liệu nguồn không phải là mã BCD, SM409 sẽ ở trạng thái ON (Lỗi thao tác), SD409=4 (xảy ra lỗi).
- Vì hãng số K tự động chuyển sang dạng nhị phân nên không phù hợp với lệnh này
- Ví dụ: tất cả thông tin được lưu trong thanh ghi thông tin thời gian (clock) SD13~SD19 của PLC là mã BCD, nhưng chúng ta đã quen sử dụng giá trị thập phân. Thông tin thời gian có thể được chuyển đổi từ thông tin mã BCD sang nhị phân:



4-8-10 Lệnh chuyển binary sang BCD [BCD]

1) Sơ lược

Chuyển dữ liệu Nhị phân sang mã BCD

Chuyển dữ liệu Nhị phân sang mã BCD [BCD]			
16 bits	BCD	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

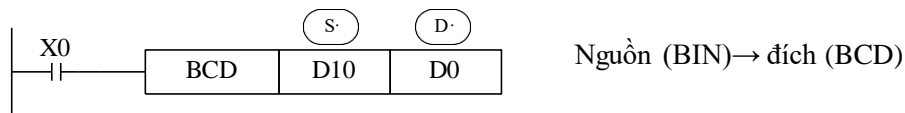
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	16 bits, BIN
D	Địa chỉ phần tử lập trình đích	BCD code

3) Phần tử lập trình phù hợp

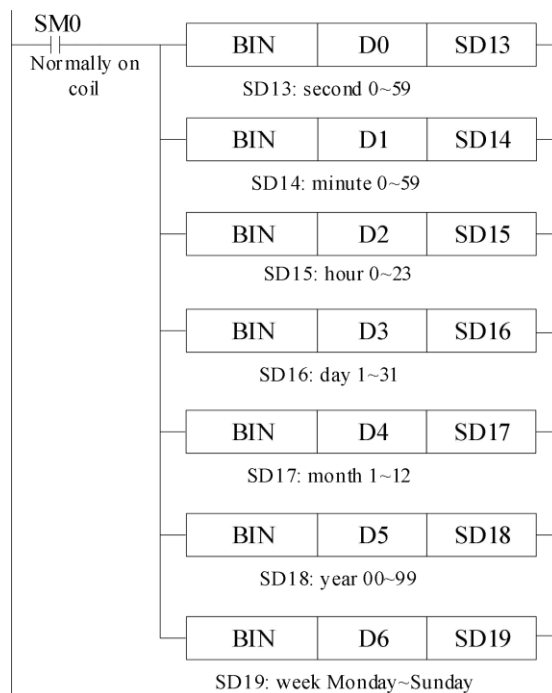
Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hàng số	Mô-đun	Hệ							
	D	F	TD	C	DX	DY	DM	DS	K/H	I	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- Lệnh này có thể chuyển đổi giá trị nhị phân thành mã BCD
- Ví dụ: thanh ghi thông tin thời gian PLC SD13 đến SD19 lưu dữ liệu dưới dạng mã BCD, mà chúng ta thường sử dụng các giá trị thập phân, do đó bạn có thể sử dụng lệnh BCD để chuyển đổi thông tin thời gian trong các thanh ghi SD13 đến SD19.



4-8-11 Lệnh chuyển Hex sang ASCII [ASCII]

1) Sơ lược

Lệnh chuyển Hex sang ASCII [ASCII]			
16 bits	ASCII	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

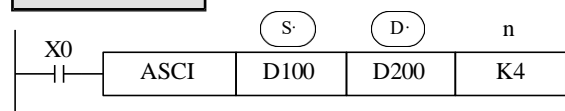
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình nguồn	2 bits, HEX
D	Địa chỉ phần tử lập trình đích	ASCII code
n	Số lượng ký tự chuyển đổi	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit										
	Hệ								Hằng số	Mô-đun		Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•	•	•	•	•	•	•											
D	•		•	•		•	•	•											
n	•		•	•		•	•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



- D chuyển đổi dữ liệu Hex nguồn thành mã ASCII và lưu trữ trong S . Các ký tự biến đổi là n.
- D sẽ lưu trữ một mã ASCII.
- Quá trình chuyển đổi như sau

Gán thiết bị nguồn:

(D100)=0ABCH

(D101)=1234H

(D102)=5678H

[0]=30H [1]=31H

[5]=35H [A]=41H

[2]=32H [6]=36H

[B]=42H [3]=33H

[7]=37H [C]=43H

[4]=34H [8]=38H

D \ n	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 down	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 up		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 down			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 up				[C]	[B]	[A]	[0]	[4]	[3]
D202 down					[C]	[B]	[A]	[0]	[4]
D202 up						[C]	[B]	[A]	[0]
D203 down							[C]	[B]	[A]
D203 up								[C]	[B]
D204 down									[C]

4-8-12 Lệnh chuyển ASCII sang Hex [HEX]

1) Sơ lược

ASCII chuyển sang Hex [HEX]			
16 bits	HEX	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

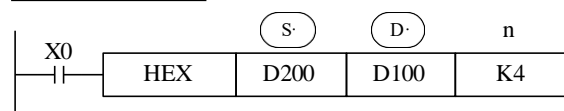
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình nguồn	ASCII
D	Địa chỉ phân tử lập trình đích	2 bits, HEX
n	Số lượng ký tự ASCII	16 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun	Hệ							
	D	F	T	CD	D	D	D	DS	K/H	I	QD	X	Y	M	S	T	C	Dn
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										
n									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

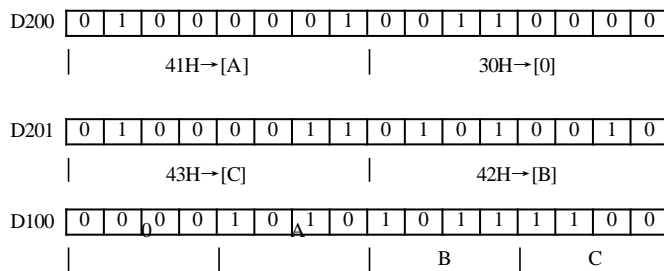


- Chuyển đổi 8 bit cao và 8 bit thấp trong dữ liệu nguồn S thành HEX. Di chuyển 4 bit mỗi lần tới đích D
- Số ký tự chuyển đổi được gán bởi n.
Quá trình chuyển đổi như sau:

(S)	Mã ASCII	Chuyển đổi HEX
D200 low	30H	0
D200 high	41H	A
D201 low	42H	B
D201 high	43H	C
D202 low	31H	1
D202 high	32H	2
D203 low	33H	3
D203 high	34H	4
D204 low	35H	5

n=k4

n (D)	D102	D101	D100
1	Không thay đổi thành 0		...0H
2			...0AH
3			...0ABH
4			0ABCH
5		...0H	ABC1H
6		...0AH	BC12H
7		...0ABH	C123H
8		0ABCH	1234H
9	...0H	ABC1H	2345H



4-8-13 Lệnh giải mã [DECO]

1) Sơ lược

Thay dữ liệu hoặc bit thành 1.

Giải mã [DECO]			
16 bits	DECO	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ dữ liệu nguồn	16 bits, BIN
D	Địa chỉ bắt đầu kết quả giải mã	16 bits, BIN
n	Số bit giải mã của phần tử lập trình	16 bits, BIN

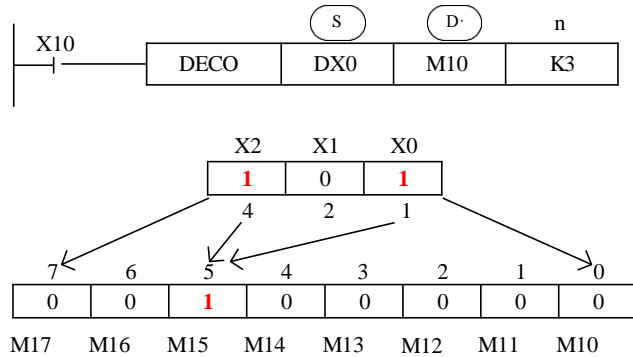
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun		Hệ						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D												•	•	•	•	•		
n									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

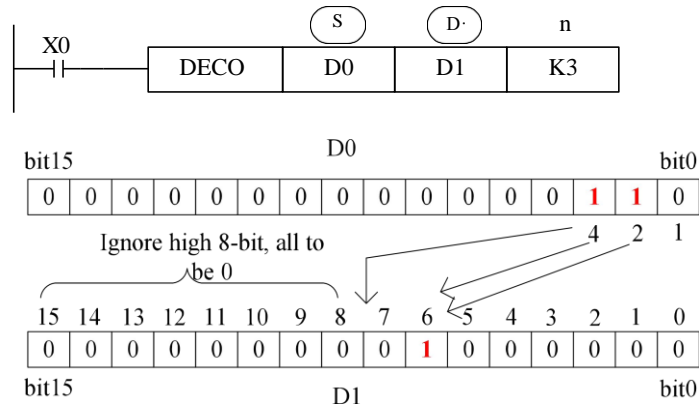
Mô tả

< Khi D là thiết bị dạng bit > $n \leq 16$



- $n = 3$, do đó đối tượng giải mã là ba bit thấp hơn trong DX0, đó là X2 ~ X0.
- $n = 3$ nên kết quả giải mã cần biểu thị bằng $2^3 = 8$ bit, tức là M17 ~ M10.
- Khi X2 = 1, X1 = 0, X0 = 1 thì giá trị mà nó biểu thị là $4 + 1 = 5$ nên M15 ở vị trí thứ năm từ M10 chuyển thành 1; khi X2 ~ X0 đều bằng 0 thì giá trị là 0 nên M10 là 1 (M10 ở vị trí thứ 0).
- Nếu $n = 0$ thì lệnh sẽ không được thực thi. Nếu n là giá trị nằm ngoài 0 ~ 16 thì lệnh sẽ không được thực thi.
- Khi $n = 16$, nếu lệnh giải mã là phần tử lập trình dạng thì số điểm là $2^{16} = 65536$.
- Khi đầu vào trình điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra giải mã của hành động được duy trì.

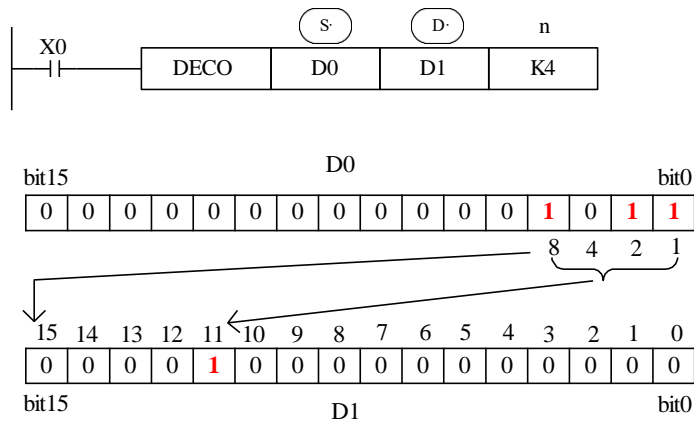
< Khi D là thiết bị dạng word > $n \leq 4$



- n -bit thấp ($n \leq 4$) của địa chỉ nguồn được giải mã đến địa chỉ đích. Khi $n \leq 3$, 8 bit cao của địa chỉ đích chuyển thành 0
- Nếu $n = 0$ thì lệnh sẽ không được thực thi. Nếu n nằm ngoài 0 ~ 4 thì lệnh sẽ không được thực thi.

- $N = 3$, do đó đối tượng giải mã trong D0 là bit2-bit0 và giá trị tối đa mà nó đại diện là $4 + 2 + 1 = 7$.
- $N = 3$, vì vậy trong D1 cần $2^3 = 8$ bit để thể hiện kết quả giải mã, tức là bit7 ~ bit0.
- Khi bit2 và bit1 đều là 1 và bit0 là 0, giá trị là $4+2=6$, do đó bit6 trong D1 ở trạng thái ON..

< D là phần tử lập trình dạng word> $n \leq 4$



- n -bit thấp ($n \leq 4$) của địa chỉ nguồn được giải mã đến địa chỉ đích. Khi $n \leq 3$, 8 bit cao của địa chỉ đích chuyển thành 0.
- Nếu $n = 0$ thì lệnh sẽ không được thực thi. Nếu n nằm ngoài $0 \sim 4$ thì lệnh sẽ không được thực thi.
- $N = 4$ nên đối tượng giải mã trong D0 là bit3 ~ bit0, đại diện cho giá trị lớn nhất là $8 + 4 + 2 + 1 = 15$.
- $N = 4$, vì vậy trong D1, cần $2^4 = 16$ bit để biểu diễn kết quả giải mã, nghĩa là bit15 ~ bit0.
- Khi bit3, bit1 và bit0 đều là 1 và bit2 là 0, giá trị là $8+2+1=11$, do đó bit11 trong D1 ở trạng thái ON.

4-8-14 Lệnh mã hóa bit cao [ENCO]

1) Sơ lược

Xác định bit cao nhất là 1.

Mã hóa bit cao [ENCO]			
16 bits	ENCO	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ dữ liệu mã hóa	16 bits, BIN
D	Địa chỉ kết quả mã hóa	16 bits, BIN
n	Số lượng bit của kết quả mã hóa	16 bits, BIN

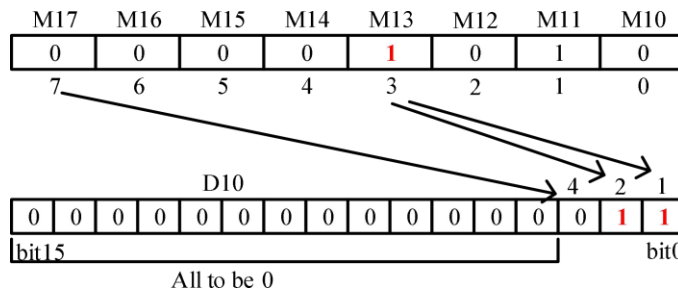
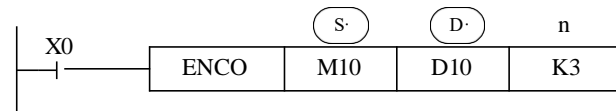
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit										
	Hệ								Hãng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	QD	X	Y	M	S	T	C	Dn.m	
S	•	•	•	•	•	•	•					•	•	•	•	•	•		
D	•		•	•		•	•												
n								•											

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

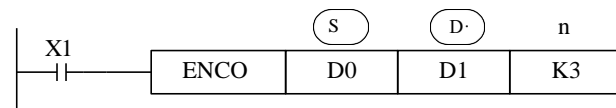
Mô tả

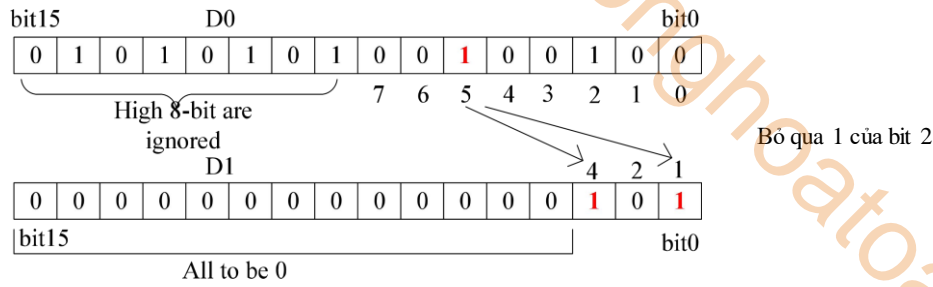
< Khi S là thiết bị dạng bit > $n \leq 16$



- Nếu số bit trong địa chỉ nguồn là 1 thì phía thấp sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi điều kiện điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa không thay đổi.
- Khi $n = 16$, nếu lệnh mã hóa là thành phần dạng bit thì số điểm của nó là $2^{16} = 65536$.
- $N = 3$, đối tượng được mã hóa có $2^3 = 8$ bit, là M17 ~ M10 và kết quả mã hóa được lưu trữ ở ba bit thấp hơn của D10, là bit2 ~ bit0.
- M13 và M11 đều là 1. Bỏ qua M11, M13 được mã hóa, bit2-bit0 đại diện cho 3, trong khi bit0 và bit1 là 1.

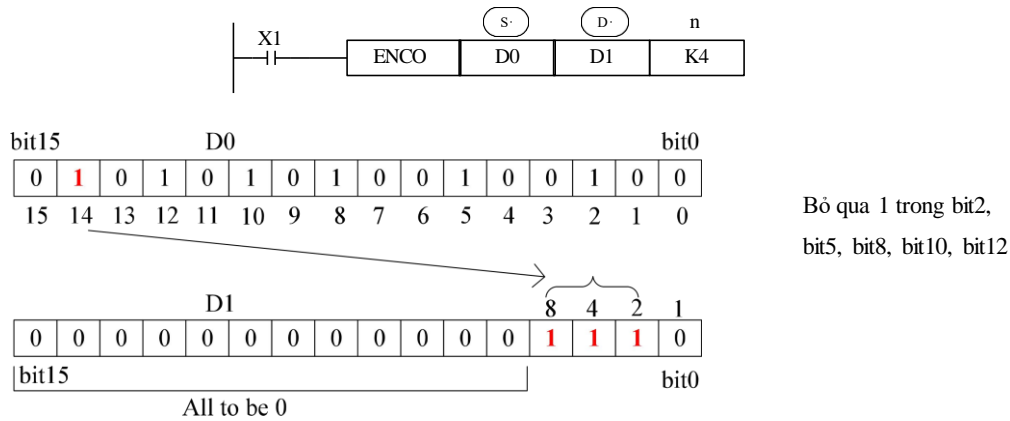
< Khi S là thiết bị dạng word $n \leq 4$





- Nếu nhiều bit trong địa chỉ nguồn là 1 thì phía thấp sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi đầu vào trình điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa không thay đổi.
- Khi $n \leq 3$, 8 bit cao trong D0 bị bỏ qua.
- Khi $n=3$, đối tượng mã hóa có $2^3 = 8$ bit, tức là bit7 ~ bit0 trong D0. Kết quả mã hóa được lưu trữ ở 3 bit thấp hơn trong D1, nghĩa là bit2 ~ bit0.
- Khi bit5 và bit2 trong D0 đều là 1, bit2 bị bỏ qua và bit5 được mã hóa, bit2-bit0 đại diện cho 5, bit2 và bit0 là 1.

< S là phần tử lập trình dạng word > $n \leq 4$



- Nếu số bit trong địa chỉ nguồn là 1 thì phía thấp sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi đầu vào trình điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa không thay đổi.
- $n = 4$, đối tượng được mã hóa có $2^4 = 16$ bit, tức là bit15 ~ bit0 trong D0. Kết quả mã hóa được lưu trữ ở 4 bit thấp hơn trong D1, nghĩa là bit3 ~ bit0.
- Bit cao nhất của 1 trong D0 là bit14, bỏ qua tất cả các bit thấp 1 và mã hóa bit14, bit3-bit0 đại diện cho 14, bit3, bit2 và bit1 là 1.

4-8-15 Lệnh mã hóa bit thấp [ENCOL]

1) Sơ lược

Xác định vị trí mà bit thấp ở trạng thái ON.

Mã hóa bit thấp [ENCOL]			
16 bits	ENCOL	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình cần mã hóa	16bit, BIN
D	Địa chỉ phân tử mềm để lưu kết quả mã hóa	16bit, BIN
n	Số lượng bit của kết quả mã hóa	16bit, BIN

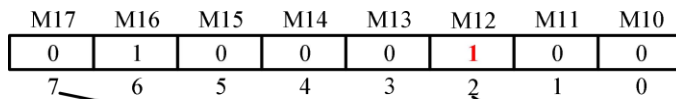
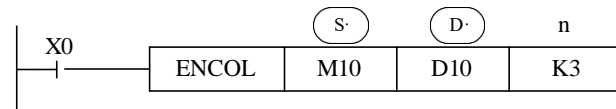
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word								Phân tử lập trình dạng bit								
	Hệ							Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	ID	Q	X	Y	M	S	T	C
S	•	•	•	•	•	•	•				•	•	•	•	•	•	
D	•		•	•		•	•										
n								•									

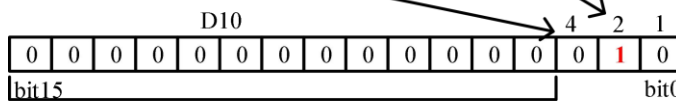
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

<Nếu S là thiết bị bit > $n \leq 16$



Bỏ qua 1 của M16



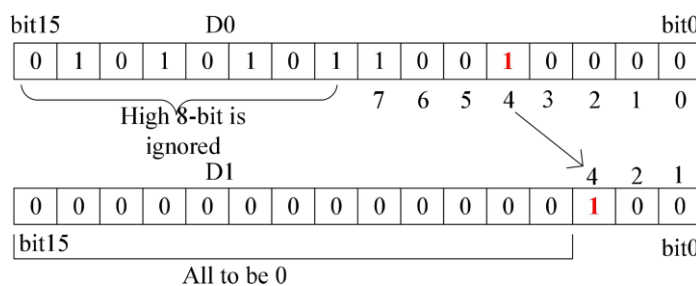
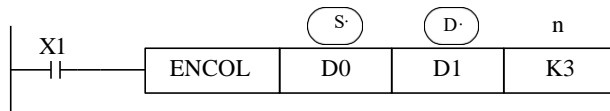
All to be 0

- Nếu số bit trong địa chỉ nguồn là 1 thì phía bit cao sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi điều kiện điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa

không thay đổi

- Khi $n = 16$, nếu S của lệnh mã hóa là phần tử bit thì điểm của nó là $2^{16} = 65536$.
- $n = 3$, đối tượng được mã hóa có $2^3 = 8$ bit, là $M17 \sim M10$ và kết quả mã hóa được lưu trữ ở ba bit thấp hơn của $D10$, là bit2 ~ bit0.
- $M12$ và $M16$ đều là 1. Bỏ qua $M16$, $M12$ được mã hóa, bit2-bit0 đại diện cho 2, còn bit1 là 1.

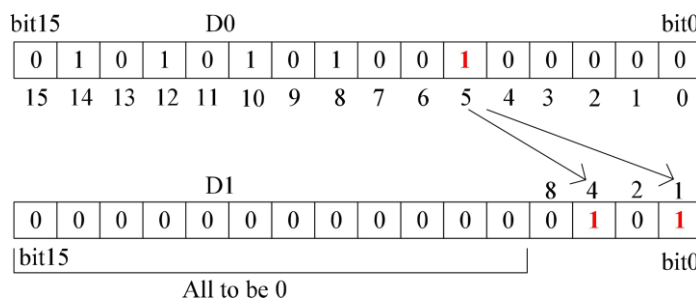
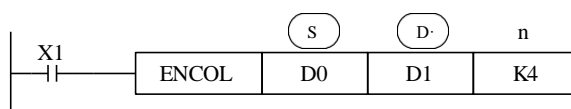
< Nếu S là thiết bị dạng word > $n \leq 4$



Bỏ qua 1 của b7

- Nếu nhiều bit trong địa chỉ nguồn là 1 thì phía bit cao sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi đầu vào trình điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa không thay đổi.
- Khi $n \leq 3$, 8 bit cao trong $D0$ bị bỏ qua.
- Đối tượng mã hóa có $2^3 = 8$ bit, tức là bit7 ~ bit0 trong $D0$. Kết quả mã hóa được lưu trữ ở 3 bit thấp hơn trong $D1$, nghĩa là bit2 ~ bit0.
- Khi bit7 và bit4 trong $D0$ đều là 1, bit7 bị bỏ qua và bit4 được mã hóa. Bit 2 là 1 khi bit2-bit0 được biểu thị bằng 4.

< S là phần tử lập trình dạng word > $n \leq 4$



Bỏ qua 1 trong
bit14, bit12, bit10,
bit8

- Nếu nhiều bit trong địa chỉ nguồn là 1 thì phía bit cao sẽ bị bỏ qua và nếu địa chỉ nguồn là 0 thì lệnh sẽ không được thực thi.
- Khi đầu vào trình điều khiển ở trạng thái OFF, lệnh không được thực thi và đầu ra mã hóa không thay đổi.
- $n = 4$, đối tượng được mã hóa có $2^4 = 16$ bit, tức là bit15 ~ bit0 trong D0. Kết quả mã hóa được lưu trữ ở 4 bit thấp hơn trong D1, nghĩa là bit3 ~ bit0.
- Bit thấp nhất của 1 trong D0 là bit5, bỏ qua tất cả các bit cao 1 và mã hóa bit5 với bit3-bit0 là 5, bit2 và bit0 là 1.

4-8-16 Lệnh chuyển nhị phân sang mã gray [GRY]

1) Sơ lược

Chuyển đổi dữ liệu nhị phân sang mã gray

Lệnh chuyển nhị phân sang mã gray[GRY,DGRY]			
16 bits	GRY	32 bits	DGRY
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

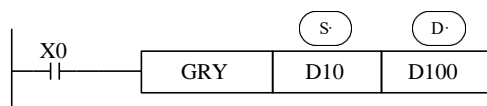
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình cần mã hóa	16bits/32bits,BIN
D	Địa chỉ phần tử mềm để lưu kết quả mã hóa	16bits/32bits,BIN

3) Phần tử lập trình phù hợp

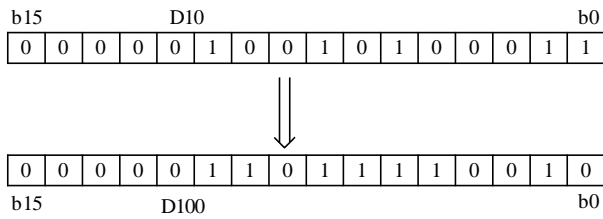
Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun	Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	●	●	●	●	●	●	●	●	●									
D	●		●	●		●	●	●										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



Nguồn (BIN) → đích (GRY)



Mỗi bit của D10 sẽ thực hiện lệnh XOR với bit ở bên trái của nó. Như mã gray liên quan, bit bên trái sẽ không thay đổi (bit bên trái là 0); kết quả chuyển đổi được lưu trữ trong D100.

- Chuyển đổi giá trị nhị phân sang mã gray.
- GRY có DGRY chế độ 32 bit , có thể chuyển đổi mã gray 32 bit.
- Phạm vi S là 0~32.767 (lệnh 16 bit); 0~2,147,483,647 (lệnh 32 bit).

4-8-17 Lệnh chuyển mã Gray thành Nhị phân [GBIN,DGBIN]

1) Sơ lược

Chuyển đổi dữ liệu mã gray thành nhị phân.

Chuyển mã Gray thành Nhị phân [GBIN,DGBIN]			
16 bits	GBIN	32 bits	DGBIN
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

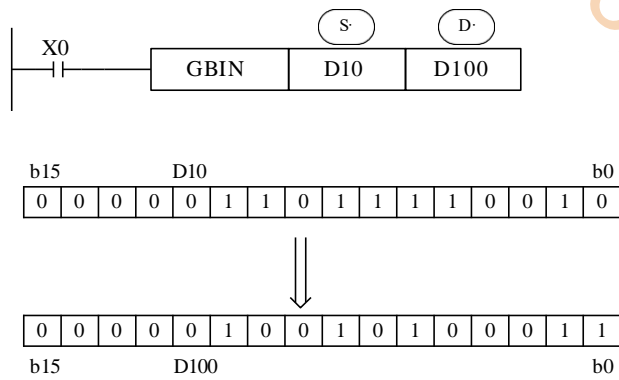
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình cần mã hóa	16bits/32bits, BIN
D	Địa chỉ phân tử mềm để lưu kết quả mã hóa	16bits/32bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit								
	Hệ									Hằng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	D	K/H	ID	QD	X	Y	M	S	T	C	Dn. m
S	•	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•											

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



Nguồn (GRY) → đích (BIN)

Từ bit thứ 2 bên trái của D10, sẽ XOR với từng bit với giá trị sau khi giải mã (bit bên trái sẽ không thay đổi). Giá trị chuyển đổi sẽ được lưu trữ trong D100.

- Chuyển đổi mã gray thành giá trị nhị phân.
- GBIN có DBIN chế độ 32 bit, có thể chuyển đổi giá trị nhị phân 32 bit.
- Phạm vi S là 0~32.767 (lệnh 16 bit); 0~2,147,483,647 (lệnh 32 bit).

4-9. Các lệnh với số thực dấu phẩy động

Tên gọi nhớ	Chức năng	Chương
ECMP	So sánh số thực dấu phẩy động	4-9-1
EZCP	So sánh vùng số thực dấu phẩy động	4-9-2
EADD	Phép cộng số thực dấu phẩy động	4-9-3
ESUB	Phép trừ số thực dấu phẩy động	4-9-4
EMUL	Phép nhân số thực dấu phẩy động	4-9-5
EDIV	Phép chia số thực dấu phẩy động	4-9-6
ESQR	Phép căn bậc 2 số thực dấu phẩy động	4-9-7
SIN	Tính Sine	4-9-8
COS	Tính Cosine	4-9-9
TAN	Tính Tangent	4-9-10
ASIN	Tính ASIN	4-9-11
ACOS	Tính ACOS	4-9-12
ATAN	Tính ATAN	4-9-13

4-9-1 Lệnh so sánh số thực dấu phẩy động [ECMP,EDCMP]

1) Sơ lược

So sánh số thực dấu phẩy động [ECMP,EDCMP]			
16 bits	-	32 bits	ECMP
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDCMP		
Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ phần tử lập trình cần so sánh	32/64 bits, BIN
S2	Địa chỉ phần tử lập trình cần so sánh	32/64 bits, BIN
D	Kết quả so sánh	bit

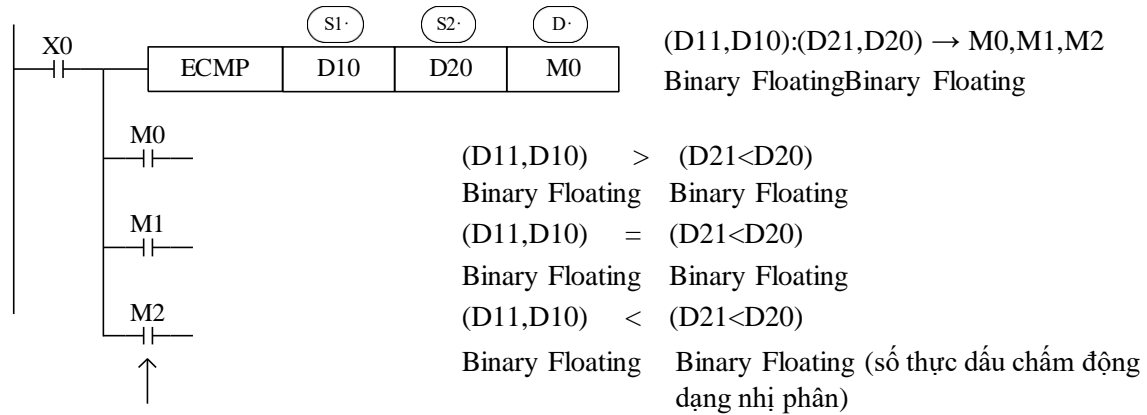
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	ID	Q	X	Y	M	S	T	C	Dn.m	
S1	•	•			•	•	•	•	•										
S2	•	•			•	•	•	•	•										
D											•	•	•						

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

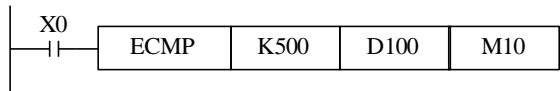
Mô tả

<lệnh 32 bit>



Khi X0 ở trạng thái OFF, ngay cả lệnh ECMP cũng không hoạt động, M0~M2 sẽ giữ trạng thái trước khi X0 về trạng thái OFF.

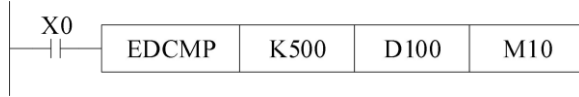
- Lệnh sẽ so sánh hai dữ liệu nguồn S1 và S2. Kết quả được lưu trữ trong ba bit từ D.
- Trước khi lệnh được thực thi, dữ liệu so sánh phải là ở dạng số thực dấu phẩy động (nếu là số nguyên thì có thể chuyển đổi bằng lệnh FLT); nếu không, kết quả thực thi lệnh sẽ sai.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị đó sẽ được chuyển đổi thành giá trị số thực dấu phẩy động.



(K500) : (D101, D100) → M10, M11, M12

Nhị phân chuyển
sang số thực dấu
phẩy động Số thực dấu
phẩy động

<lệnh 64 bit>



(K500) : (D103, D102, D101, D100) → M10, M11, M12

Nhị phân chuyển
sang số thực dấu
phẩy động Số thực dấu
phẩy động

4-9-2 Lệnh so sánh vùng số thực dấu phẩy động [EZCP]

1) Sơ lược

So sánh vùng số thực dấu phẩy động [EZCP]			
16 bits	-	32 bits	EZCP
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ phần tử lập trình cần so sánh	32 bits, BIN
S2	Giới hạn trên của dữ liệu so sánh	32 bits, BIN
S3	Giới hạn dưới của dữ liệu so sánh	32 bits, BIN
D	Địa chỉ phần tử mềm để lưu kết quả so sánh	bit

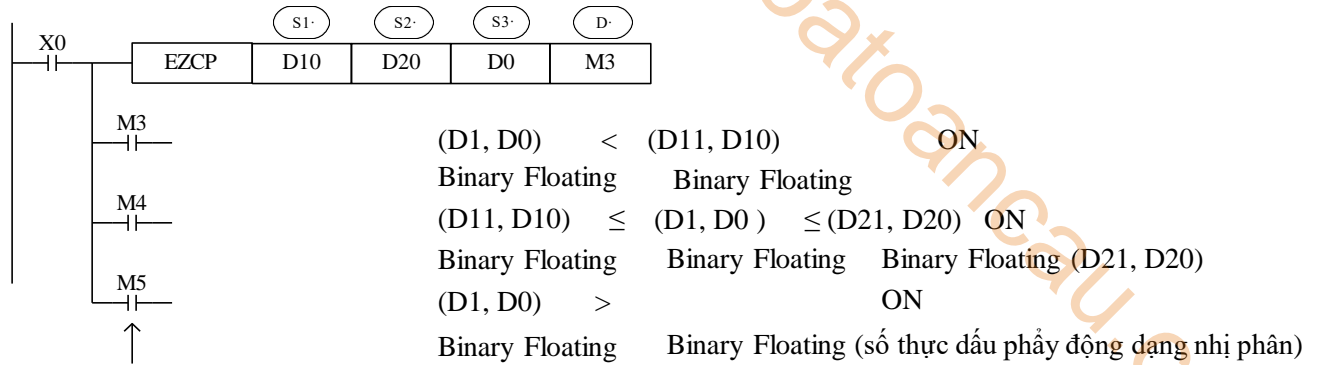
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
S3	•	•			•	•	•	•	•									
D												•	•	•				

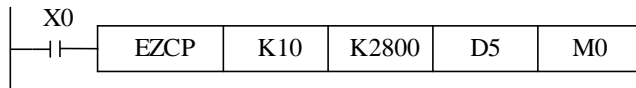
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

So sánh dữ liệu nguồn với phạm vi



- Khi X0 ở trạng thái OFF, ngay cả lệnh EZCP cũng không chạy, M3~M5 sẽ giữ trạng thái trước khi X0 ở trạng thái OFF.
- Kết quả sẽ được lưu trữ trong ba cuộn dây bắt đầu từ D.
- Hằng số K và H sẽ chuyển đổi thành giá trị số thực dấu phẩy động dạng nhị phân khi chúng là dữ liệu nguồn.



$(K10) : [D6, D5] : (K2800) \rightarrow M0, M1, M2$
 Nhị phân chuyển sang số thực dấu phẩy động Số thực dấu phẩy động

Hãy đặt $S1 < S2$, khi $S2 < S1$, đặt S2 có cùng giá trị với S1.

Lưu ý: giá trị so sánh phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

4-9-3 Lệnh cộng số thực dấu phẩy động [EADD, EDADD]

1) Sơ lược

Phép cộng số thực dấu phẩy động [EADD, EDADD]			
16 bits	-	32 bits	EADD
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDADD		
Điều kiện thực thi	Thường mở/đóng/ sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép cộng	32/64 bits, BIN
S2	Địa chỉ dữ liệu phép cộng	32/64 bits, BIN
D	Địa chỉ kết quả	32/64 bits, BIN

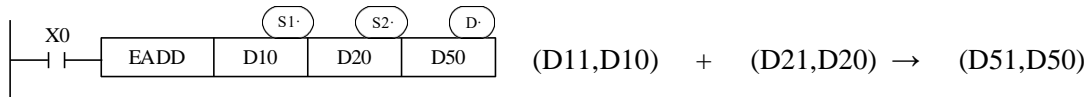
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

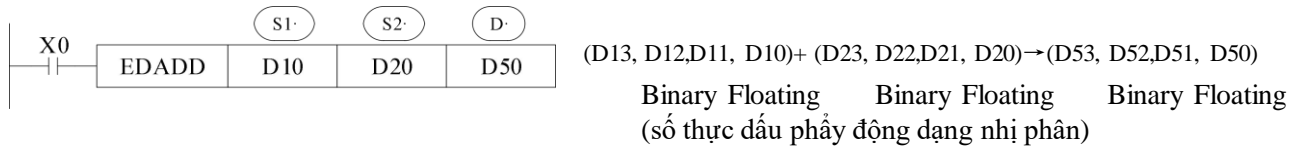
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

<lệnh 32 bit>



<lệnh 64 bit>



- Hai dữ liệu nguồn dạng số thực dấu phẩy động dạng nhị phân thực hiện phép cộng, kết quả sẽ được lưu trữ ở địa chỉ đích.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị sẽ được chuyển đổi thành giá trị số thực dấu phẩy động trước khi thực thi phép cộng.
- The registers in EDADD must start with an even address.
- Địa chỉ các thanh ghi trong kênh EDADD phải bắt đầu bằng số chẵn.

<lệnh 32 bit>



(K1234) + (D101,D100) -> (D111,D110)

Nhị phân sang
số thực dấu
phẩy động

Số thực dấu
phẩy động
dạng nhị phân

Số thực dấu
phẩy động
dạng nhị phân

<64 bits operation>



(K1234) + (D103, D102, D101, D100) → (D113, D112, D111, D110)
 Nhị phân sang số thực dấu phẩy động Số thực dấu phẩy động dạng nhị phân Số thực dấu phẩy động dạng nhị phân

- Địa chỉ dữ liệu nguồn và địa chỉ có thể giống nhau. Xin lưu ý rằng khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét.

Lưu ý: giá trị phép cộng phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

4-9-4 Lệnh trừ số thực dấu phẩy động [ESUB,EDSUB]

1) Sơ lược

Phép trừ số thực dấu phẩy động [ESUB,EDSUB]			
16 bits	-	32 bits	ESUB
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDSUB		
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép trừ	32/64 bits, BIN
S2	Địa chỉ dữ liệu phép trừ	32/64 bits, BIN
D	Địa chỉ kết quả	32/64 bits, BIN

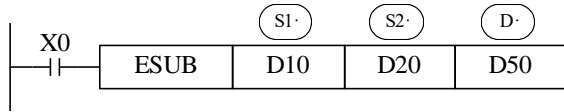
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S1	●	●			●	●	●	●	●									
S2	●	●			●	●	●	●	●									
D	●					●	●	●										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

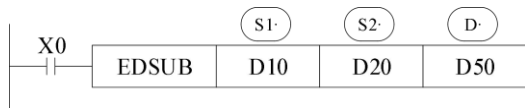
Mô tả

< lệnh 32 bit >



(D11,D10) – (D21,D20) → (D51,D50)
 Binary Floating Binary Floating Binary Floating (số thực dấu phẩy động dạng thập phân)

< lệnh 64 bit >



(D13, D12,D11, D10) – (D23,D22,D21,D20) → (D53, D52,D51, D50)
 Binary Floating Binary Floating Binary Floating

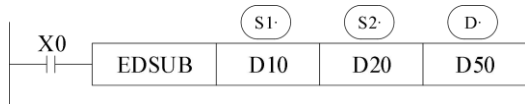
- Giá trị số thực dấu phẩy động dạng nhị phân S1 trừ S2, kết quả được lưu trong địa chỉ đích.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị sẽ được chuyển đổi thành số thực dấu phẩy động trước thực hiện phép trừ.

< lệnh 32 bit >



(K1234) – (D101, D100) → (D111, D110)
 Nhị phân chuyển sang số thực dấu phẩy động Số thực dấu phẩy động dạng nhị phân Số thực dấu phẩy động dạng nhị phân

< lệnh 64 bit >



(D13, D12,D11, D10) – (D23,D22,D21,D20) → (D53, D52,D51, D50)
 Nhị phân chuyển sang số thực dấu phẩy động Số thực dấu phẩy động dạng nhị phân Số thực dấu phẩy động dạng nhị phân

- Địa chỉ dữ liệu nguồn và kết quả có thể giống nhau. Xin lưu ý rằng khi X0 ở trạng thái ON, lệnh sẽ được thực thi trong mỗi chu kỳ quét.
- Lưu ý: giá trị toán hạng phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

4-9-5 Lệnh nhân số thực dấu phẩy động [EMUL,EDMUL]

1) Sơ lược

Phép nhân số thực dấu phẩy động [EMUL, EDMUL]			
16 bits	-	32 bits	EMUL
Điều kiện thực thi	Thường mở/đóng, sừon lên, sừon xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDMUL		
Điều kiện thực thi	Thường mở/đóng, sừon lên, sừon xuống	Model phù hợp	XDH, XLH
Yêu cầu phần cứng	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép nhân	32 /64bits, BIN
S2	Địa chỉ dữ liệu phép nhân	32 /64bits, BIN
D	Địa chỉ kết quả	32 /64bits, BIN

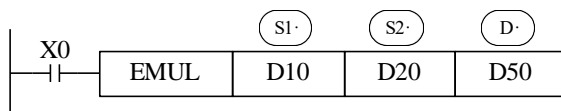
3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

<lệnh 32 bit>



$(D11, D10) \times (D21, D20) \rightarrow (D51, D50)$

Binary Floating Binary Floating Binary Floating
(số thực dấu phẩy động dạng nhị phân)

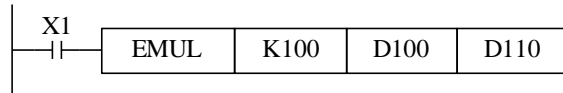
<lệnh 64 bit>



(D13, D12, D11, D10) × (D23, D22, D21, D20) → (D53, D52, D51, D50)
 Binary Floating Binary Floating Binary Floating

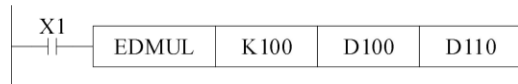
- Giá trị số thực dấu phẩy động của S1 được nhân với giá trị số thực dấu phẩy động của S2. Kết quả của phép nhân được lưu tại D dưới dạng giá trị số thực dấu phẩy động.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị sẽ được chuyển đổi thành số thực dấu phẩy động trước khi thực hiện phép nhân.
- Địa chỉ các thanh ghi trong lệnh EDMUL phải bắt đầu bằng số chẵn.

<lệnh 32 bit>



(K100) × (D101, D100) → (D111, D110)
 Nhị phân chuyển Số thực dấu phẩy Số thực dấu phẩy
 sang số thực dấu động dạng nhị phân động dạng nhị phân
 phẩy động

<lệnh 64 bit>



(K00) × (D103, D102, D101, D100) → (D113, D112, D111, D110)
 Nhị phân chuyển Số thực dấu phẩy Số thực dấu phẩy
 sang số thực dấu động dạng nhị phân động dạng nhị phân
 phẩy động

Lưu ý: giá trị toán hạng phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

4-9-6 Lệnh chia số thực dấu phẩy động [EDIV, EDDIV]

1) Sơ lược

Phép chia số thực dấu phẩy động [EDIV, EDDIV]			
16 bits	-	32 bits	EDIV
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-
64 bits	EDDIV		
Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống	Điều kiện thực thi	Thường mở/đóng/sườn lên, sườn xuống

Yêu cầu phần mềm	Phiên bản V3.7.1 trở lên	Yêu cầu phần mềm	Phiên bản V3.7.4a trở lên
------------------	--------------------------	------------------	---------------------------

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ dữ liệu phép chia	32/64 bits, BIN
S2	Địa chỉ dữ liệu phép chia	32/64 bits, BIN
D	Địa chỉ kết quả	32/64 bits, BIN

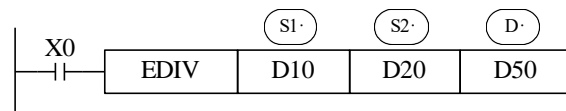
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

<lệnh 32 bit>



$$(D11, D10) \div (D21, D20) \rightarrow (D51, D50)$$

Binary Floating Binary Floating Binary Floating
(số thực dấu phẩy động dạng nhị phân)

<lệnh 64 bit>

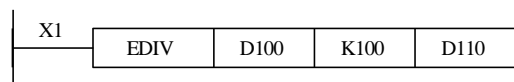


$$(D13, D12, D11, D10) \div (D23, D22, D21, D20) \rightarrow (D53, D52, D51, D50)$$

Binary Floating Binary Floating Binary Floating

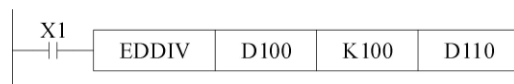
- Giá trị số thực dấu phẩy động của S1 bị chia cho giá trị số thực dấu phẩy động của S2. Kết quả của phép chia được lưu trong D dưới dạng giá trị số thực dấu phẩy động.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị sẽ được chuyển đổi thành số thực dấu phẩy động trước khi thực hiện phép chia.
- Nếu dữ liệu nguồn S2 bằng 0, phép tính sẽ bị lỗi. Lệnh này sẽ không được thực thi.
- Giá trị toán hạng phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

<lệnh 32 bit>



(D101, D100) ÷ (K100) → (D111, D110)
 Nhi phân sang số Số thực dấu phẩy Số thực dấu phẩy
 thực dấu phẩy động dạng nhị động dạng nhị
 động phân phân

<lệnh 64 bit>



(D103, D102, D101, D100) ÷ (K100) → (D113, D112, D111, D110)
 Nhi phân sang số Số thực dấu phẩy Số thực dấu phẩy
 thực dấu phẩy động dạng nhị động dạng nhị
 động phân phân

4-9-7 Lệnh căn bậc 2 số thực dấu phẩy động [ESQR]

1) Sơ lược

Căn bậc 2 số thực dấu phẩy động [ESQR]			
16 bits	-	32 bits	ESQR
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

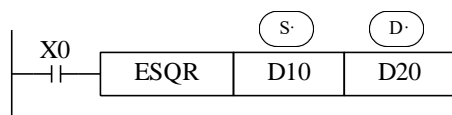
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình căn thực hiện căn bậc hai	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả

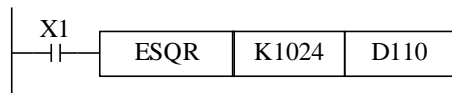


(D11,D10) → (D21,D20)

Binary Floating Binary Floating

(số thực dấu phẩy động dạng nhị phân)

- Căn bậc hai được thực hiện trên giá trị số thực dấu phẩy động của S; kết quả được lưu trữ trong D.
- Nếu hằng số K hoặc H được sử dụng làm dữ liệu nguồn thì giá trị sẽ được chuyển đổi thành số thực dấu phẩy động trước khi thực hiện phép toán.



(K1024) → (D111, D110)

Binary converts to Floating Binary Floating

- Khi kết quả bằng 0, cờ 0 sẽ được kích hoạt.
- Chỉ khi dữ liệu nguồn + thì lệnh mới được thực thi. Nếu S âm thì xảy ra lỗi và cờ lỗi SM409 được ở trạng thái ON, SD409=7, lệnh không được thực thi.
- Giá trị toán hạng phải là số thực dấu phẩy động, nếu không kết quả sẽ bị lỗi.

4-9-8 Lệnh tính Sine[SIN]

1) Sơ lược

Tính sin số thực dấu phẩy động [SIN]			
16 bits	-	32 bits	SIN
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

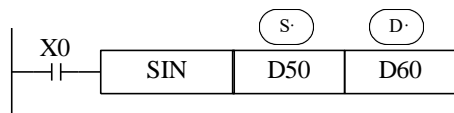
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình cần tính sin	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



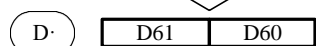
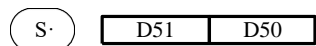
(D51,D50) → (D61,D60)SIN

Binary Floating Binary Floating (số thực dấu phẩy động dạng nhị phân)

- Lệnh này thực hiện phép toán SIN toán học trên giá trị số thực dấu phẩy động trong S (đơn vị góc RAD). Kết quả được lưu trữ trong D.

Giá trị RAD ($\text{góc} \times \pi / 180$)

Gán giá trị số thực dấu phẩy động dạng nhị phân



Giá trị SIN

Giá trị số thực dấu phẩy động dạng nhị phân

Lưu ý: giá trị toán hạng phải là số thực, nếu không kết quả sẽ bị lỗi.

4-9-9 Lệnh tính Cosine[COS]

1) Sơ lược

Tính cosine số thực dấu phẩy động [COS]			
16 bits	-	32 bits	COS
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

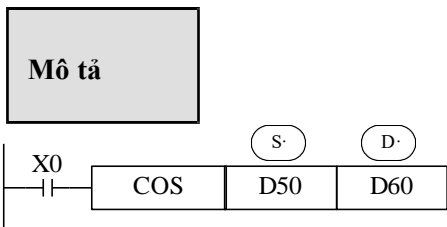
2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình cần tính cos	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

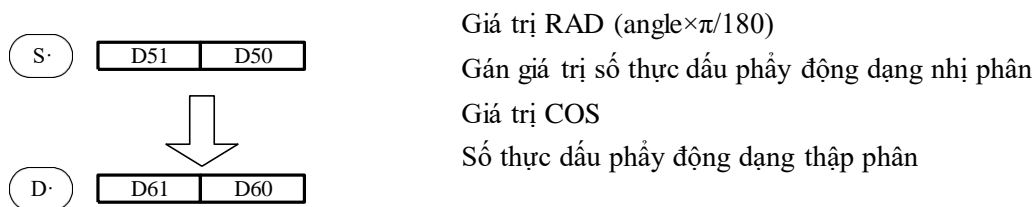
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.



$$(D51, D50)RAD \rightarrow (D61, D60)COS$$

Binary Floating Binary Floating (số thực dấu phẩy động dạng nhị phân)

- Lệnh này thực hiện phép toán COS toán học trên giá trị số thực dấu phẩy động trong S (đơn vị góc RAD). Kết quả được lưu trữ trong D.



Lưu ý: Trước khi lệnh được thực thi, dữ liệu trong tham số S phải là số thực dấu phẩy động; nếu không, kết quả thực hiện sẽ sai.

4-9-10 Lệnh tính TAN [TAN]

1) Sơ lược

TAN [TAN]			
16 bits	-	32 bits	TAN
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phân cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

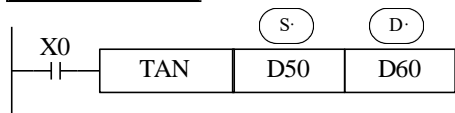
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử lập trình cần tính tan	32bit, BIN
D	Địa chỉ kết quả	32bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.	
S	●	●			●	●	●	●	●										
D	●					●	●	●											

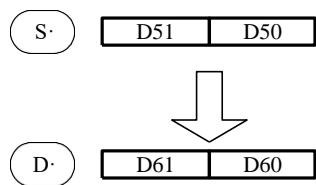
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



(D51,D50) RAD → (D61,D60) TAN
 Binary Floating Binary Floating
 (số thực dấu phẩy động dạng thập phân)

- Lệnh này thực hiện phép toán TAN toán học trên giá trị số thực dấu phẩy động trong S. Kết quả được lưu trong D.



Giá trị RAD ($\text{angle} \times \pi / 180$)
 Gán giá trị số thực dấu phẩy động dạng nhị phân
 Giá trị TAN
 Số thực dấu phẩy động dạng thập phân

Lưu ý: Trước khi lệnh được thực thi, dữ liệu trong tham số S phải là số thực dấu phẩy động; nếu không, kết quả thực hiện sẽ sai.

4-9-11 Lệnh tính ASIN [ASIN]

1) Sơ lược

ASIN [ASIN]			
16 bits	-	32 bits	ASIN
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phân cứng		Yêu cầu phần mềm	-

2) Toán hạng

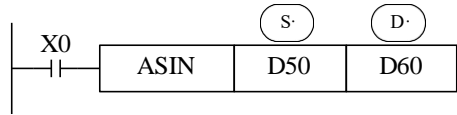
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phân tử mềm cần tính arcsin	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit									
	Hệ								Hằng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

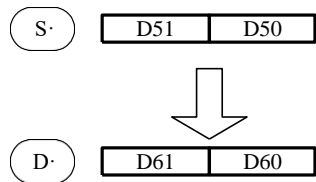
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



(D51,D50)ASIN → (D61,D60)RAD
 Binary Floating Binary Floating
 (số thực dấu phẩy động dạng nhị phân)

Lệnh này thực hiện phép toán ASIN toán học trên giá trị số thực dấu phẩy động trong S.
 Kết quả được lưu trong D.



Giá trị ASIN
 Số thực dấu phẩy động
 dạng nhị phân
 Giá trị RAD ($\text{angle} \times \pi / 180$)
 Gán giá trị số thực dấu phẩy động

Lưu ý: Trước khi lệnh được thực thi, dữ liệu trong tham số S phải là số thực dấu phẩy động; nếu không, kết quả thực hiện sẽ sai.

4-9-12 Lệnh tính ACOS [ACOS]

1) Sơ lược

ACOS [ACOS]			
16 bits	-	32 bits	ACOS
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng		Yêu cầu phần mềm	-

2) Toán hạng

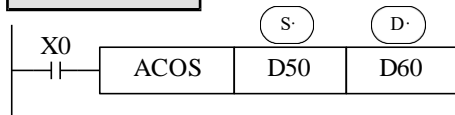
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử mềm cần tính arccos	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

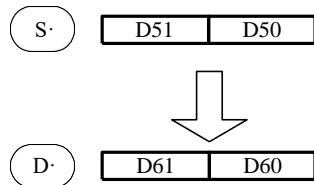
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS; M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Description



(D51,D50)ACOS → (D61,D60)RAD
 Binary Floating Binary Floating
 (số thực dấu phẩy động dạng nhị phân)

Tính giá trị arcsin (radian), lưu kết quả vào địa chỉ đích



Giá trị ACOS
 Số thực dấu phẩy động dạng nhị phân
 Giá trị RAD (angle×π/180)
 Gán giá trị số thực dấu phẩy động

Lưu ý: Trước khi lệnh được thực thi, dữ liệu trong tham số S phải là số thực dấu phẩy động; nếu không, kết quả thực hiện sẽ sai.

4-9-13 Lệnh tính ATAN [ATAN]

1) Summary

ATAN [ATAN]			
16 bits	-	32 bits	ACOS
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng		Yêu cầu phần mềm	-

2) Toán hạng

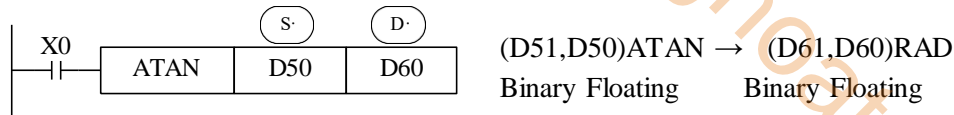
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử mềm cần tính arctan	32 bits, BIN
D	Địa chỉ kết quả	32 bits, BIN

3) Phân tử lập trình phù hợp

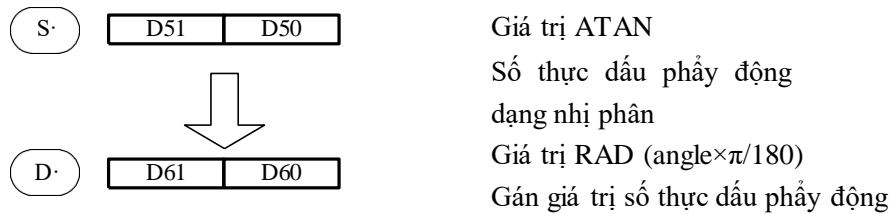
Toán hạng	Phân tử lập trình dạng word										Phân tử lập trình dạng bit							
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	QD	X	Y	M	S	T	C	Dn. m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Description



Tính giá trị arctan (radian), lưu kết quả vào địa chỉ đích



Lưu ý: Trước khi lệnh được thực thi, dữ liệu trong tham số S phải là số thực dấu phẩy động; nếu không, kết quả thực hiện sẽ sai.

4-10 Các lệnh RTC

Tên gọi nhớ	Chức năng	Chương
TRD	Đọc dữ liệu đồng hồ thời gian	4-10-1
TWR	Ghi giá trị đồng hồ thời gian	4-10-2
MOV	Đọc dữ liệu đồng hồ chính xác của bo mạch BD	4-10-3
TO	Ghi dữ liệu đồng hồ chính xác của bo mạch BD	4-10-4
TADD	Phép cộng dữ liệu thời gian	4-10-5
TSUB	Phép trừ dữ liệu thời gian	4-10-6
HTOS	Chuyển đổi dữ liệu giờ, phút và giây thành giây	4-10-7
STOH	Chuyển đổi dữ liệu giây thành giờ, phút và giây	4-10-8
TCMP	So sánh thời gian (giờ, phút, giây)	4-10-9
DACMP	So sánh ngày (năm, tháng, ngày)	4-10-10

※1: Để sử dụng lệnh, Model phải được trang bị chức năng RTC;

※2: Có một chút chênh lệch (sai số) trong đồng hồ thời gian của PLC dòng XD/XL, khoảng ±5 phút mỗi tháng. Nó có thể được hiệu chuẩn thường xuyên bằng HMI hoặc trong chương trình PLC.

※3: Nếu cần độ chính xác về thời gian cao, XD-RTC-BD có thể được sử dụng cùng nhau với sai số khoảng 13 giây mỗi tháng.

4-10-1 Lệnh đọc dữ liệu đồng hồ thời gian [TRD]

1) Sơ lược

Đọc dữ liệu đồng hồ

Đọc dữ liệu đồng hồ: [TRD]			
16 bits	TRD	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ thanh ghi để lưu dữ liệu thời gian	16 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word									Phân tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	•		•	•														

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



Ngày giờ hiện tại của đồng hồ thời gian thực được đọc và lưu trữ trong 7 thiết bị dữ liệu được chỉ định bởi địa chỉ đầu D.

- Đọc đồng hồ thời gian thực của PLC theo định dạng sau.
Đọc thanh ghi dữ liệu đặc biệt (SD013~SD019).

Thanh ghi dữ liệu đặc biệt cho đồng hồ thời gian thực	Thiết bị	Đơn vị	Dữ liệu đồng hồ	→	Thiết bị	Đơn vị
	SD018	Năm	0-99	→	D0	Năm
	SD017	Tháng	1-12	→	D1	Tháng
	SD016	Ngày	1-31	→	D2	Ngày
	SD015	Giờ	0-23	→	D3	Giờ
	SD014	Phút	0-59	→	D4	Phút
	SD013	Giây	0-59	→	D5	Giây
	SD019	Thứ	0 (Sun.)-6 (Sat.)	→	D6	Thứ

- The Giá trị RTC (đồng hồ thời gian thực) ở định dạng mã BCD (SD013 đến SD019).

- Sau khi đọc lệnh RTC bằng TRD, giá trị sẽ hiển thị ở định dạng thập phân.
- Sau khi đọc RTC bằng TRD, giá trị sẽ trở thành giá trị thập phân.
- Sau khi thực hiện lệnh TRD, D0 đến D6 bị chiếm dụng.

4-10-2 Ghi dữ liệu đồng hồ [TWR]

1) Sơ lược

Ghi dữ liệu đồng hồ

Ghi dữ liệu đồng hồ[TWR]			
16 bits	-	32 bits	TWR
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng		Yêu cầu phần mềm	-

2) Toán hạng

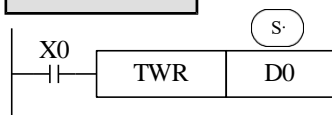
Toán hạng	Chức năng	Kiểu dữ liệu
S	Ghi dữ liệu đồng hồ vào thanh ghi	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hãng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	●		●	●	●	●	●	●										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



Ghi giá trị RTC vào PLC.

- Ghi dữ liệu đồng hồ đã đặt vào đồng hồ thời gian thực của PLC.
- Để ghi dữ liệu đồng hồ thời gian thực, vui lòng đặt giá trị 7 thanh ghi từ D0 đến D6.

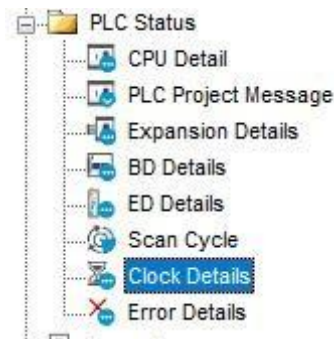
	Thiết bị	Đơn vị	Dữ liệu đồng hồ		Thiết bị	Đơn vị	
Dữ liệu cài đặt đồng hồ	D0	Năm	0-99	→	SD018	Năm	Thanh ghi dữ liệu đặc biệt cho đồng hồ thời gian thực
	D1	Tháng	1-12	→	SD017	Tháng	
	D2	Ngày	1-31	→	SD016	Ngày	
	D3	Giờ	0-23	→	SD015	Giờ	
	D4	Phút	0-59	→	SD014	Phút	
	D5	Giây	0-59	→	SD013	Giây	
	D6	Tuần	0 (Sun.)-6 (Sat.)	→	SD019	Thứ	

Sau khi thực hiện lệnh TWR, thời gian trong đồng hồ thời gian thực sẽ ngay lập tức thay đổi thành thời gian mới. Bạn nên đặt thời gian trễ vài phút so với thời gian hiện tại, sau đó điều khiển lệnh khi thời gian thực đạt đến giá trị này.

Lưu ý: khi chọn chế độ nâng cao của chương trình tải xuống bảo mật trong phần mềm XDPpro, RTC chỉ có thể được thay đổi thông qua lệnh TWR.

Có một phương thức khác để viết dữ liệu RTC.

Trong phần mềm XDPpro, hãy nhấp vào chi tiết đồng hồ (clock details) trong thanh dự án ở bên trái.



Sau đó nhấp vào “write the current time”, PC sẽ tự động ghi thời gian hiện tại vào PLC.



Sau đó nhấp vào “write the current time”, PC sẽ tự động ghi thời gian hiện tại vào PLC.

4-10-3 Lệnh đọc dữ liệu đồng hồ chính xác của bo mạch BD [MOV]

1) Sơ lược

Đọc dữ liệu đồng hồ chính xác của BD board [MOV]			
16 bits	MOV	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

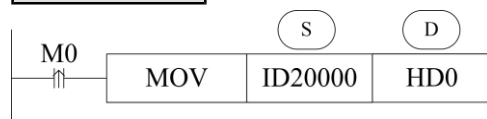
Toán hạng	Chức năng	Kiểu dữ liệu
S	Địa chỉ phần tử lập trình của dữ liệu đồng hồ để đọc	16 bits, BIN
D	Địa chỉ thanh ghi để lưu dữ liệu đồng hồ	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S	•																	
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Mô tả



Khi M0 được ở trạng thái on, dữ liệu “giờ” trong XD-RTC-BD của đồng hồ #1 được đọc vào thanh ghi HD0 của PLC.

- Địa chỉ dữ liệu của bảng BD XD-RTC-BD được hiển thị như sau:

Địa chỉ BD 1	Địa chỉ BD 2	Mô tả	Dữ liệu đồng hồ	Ghi chú
ID20000	ID20100	Giây	0~59	Hệ thập phân
ID20001	ID20101	Phút	0~59	Hệ thập phân
ID20002	ID20102	Giờ	0~23	Hệ thập phân
ID20003	ID20103	Ngày	1~31	Hệ thập phân
ID20004	ID20104	Tháng	1~12	Hệ thập phân
ID20005	ID20105	Năm	00~99	Hệ thập phân
ID20006	ID20106	Thứ	0 (Sun.)-6 (Sat.)	Hệ thập phân

- Vì thời gian trong thanh ghi ID được lưu theo thứ tự giây, phút, giờ, ngày, tháng, năm,

và thứ nên không nên sử dụng lệnh BMOV hoặc PMOV để đọc dữ liệu đồng hồ hàng loạt nếu dữ liệu đồng hồ đọc được dùng để so sánh và tính toán.

4-10-4 Lệnh ghi dữ liệu đồng hồ chính xác của BD board [TO]

1) Sơ liệu

Ghi dữ liệu đồng hồ chính xác của BD board [TO]			
16 bits	TO	32 bits	-
Điều kiện thực thi	Sườn lên/ sườn xuống	Model phù hợp	XD, XL
Yêu cầu phân cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

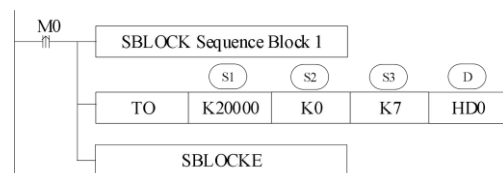
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Số BD board	16 bits, BIN
S2	Địa chỉ bắt đầu phần tử cho dữ liệu clock	16 bits, BIN
S3	Địa chỉ dữ liệu đồng hồ	16 bits, BIN
D	Địa chỉ bắt đầu phần tử cho dữ liệu clock nội bộ	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word								Phần tử lập trình dạng bit									
	Hệ								Hãng số	Mô-đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1								•										
S2									•									
S3								•										
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả

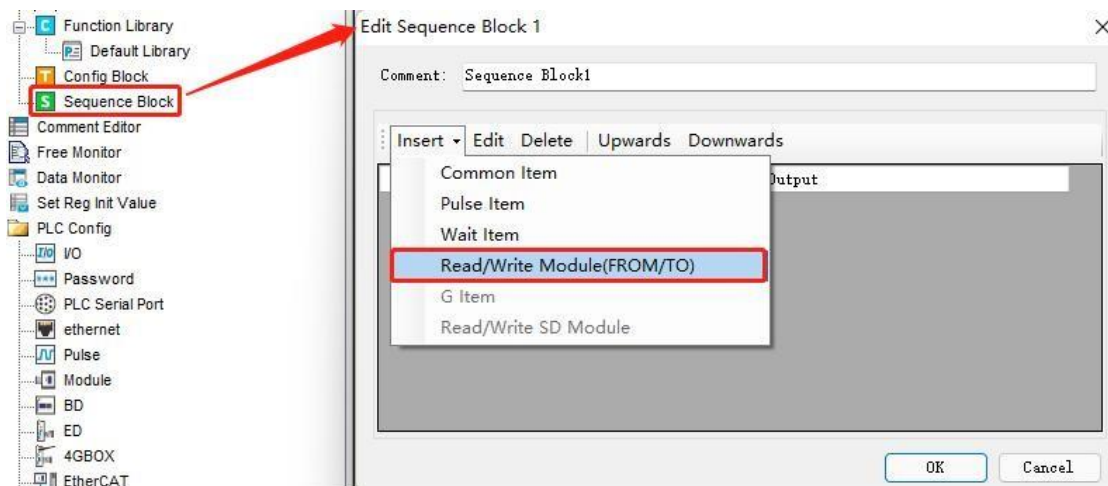


- Khi M0 ở trạng thái on, ghi các giá trị trong 7 thanh ghi liên tiếp bắt đầu bằng HD0 vào thanh ghi ID trong xD-RTC-BD của đồng hồ #1 của BD board.

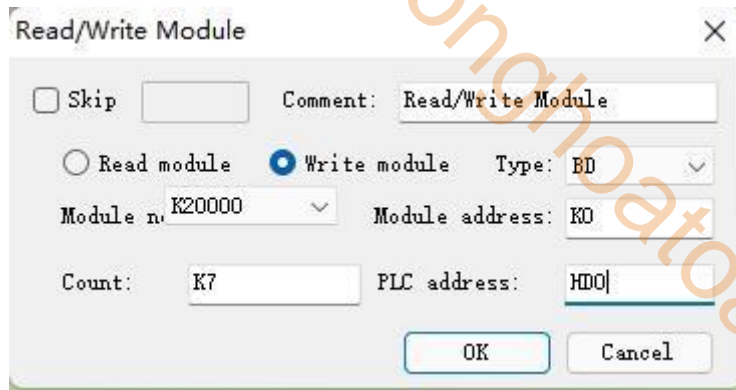
Dữ liệu nguồn	Địa chỉ BD board 1	Mô tả	Dữ liệu đồng hồ	Ghi chú
HD0	ID20000(K0)	Second	0~59	Hệ thập phân
HD1	ID20001(K1)	Minute	0~59	Hệ thập phân
HD2	ID20002(K2)	Hour	0~23	Hệ thập phân
HD3	ID20003(K3)	Date	1~31	Hệ thập phân
HD4	ID20004(K4)	Month	1~12	Hệ thập phân
HD5	ID20005(K5)	Year	00~99	Hệ thập phân
HD6	ID20006(K6)	Week	0 (Sun.)-6 (Sat.)	Hệ thập phân

- Lệnh TO cần được nhập vào Khối tuần tự. Quy trình thao tác như sau:

Cửa sổ hiển thị, nhấp vào "Insert" - "Read/Write Module (FROM/TO)":



Trong cửa sổ cấu hình hiển thị, thiết lập các thông số như sau:



Lưu ý: Địa chỉ mô-đun K20000 viết tắt là #1 BD, K20001 viết tắt là #2 BD; Địa chỉ module được đánh số từ K0 tương ứng với ID20000, ID20001..... ID20006.

4-10-5 Lệnh cộng dữ liệu đồng hồ[TADD]

1) Sơ lược

Phép cộng dữ liệu đồng hồ[TADD]			
16 bits	TADD	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ bắt đầu phần tử của dữ liệu đồng hồ (giờ, phút, giây)	16 bits, BIN
S2	Soft element header address của dữ liệu đồng hồ (giờ, phút, giây)	16 bits, BIN
D	Địa chỉ kết quả	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit									
	Hệ								Hàng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•																		
S2	•																		
D	•																		

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, H; T gồm T, HT. C gồm C, HC.

Mô tả



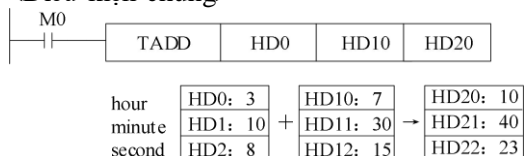
Lưu ý: sự tương ứng của các thanh ghi là cố định, nghĩa là chúng được lưu trữ theo thứ tự giờ, phút và giây.

Nếu kết quả của thao tác là 0 giờ, 0 phút, 0 giây, SM20 sẽ ở trạng thái ON.

Các toán hạng S1, S2 và D, mỗi toán hạng chiếm ba thanh ghi. Không sử dụng chúng cho các mục đích khác

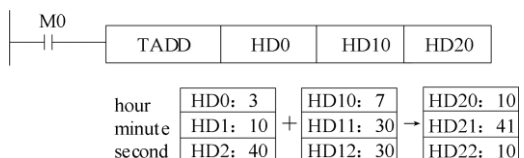
Ví dụ 1:

<Điều kiện chung>



Ví dụ 2:

< Lớn hơn 59 giây>



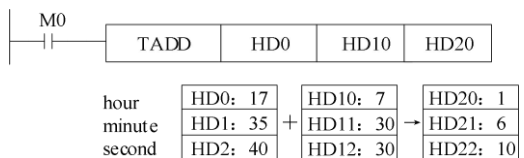
Ví dụ 3:

< Lớn hơn 59 giây>



Ví dụ 4:

< Lớn hơn 23 giờ>



4-10-6 Lệnh trừ dữ liệu đồng hồ[TSUB]

1) Sơ lược

Phép trừ dữ liệu đồng hồ[TSUB]			
16 bits	TSUB	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ bắt đầu phần tử của dữ liệu đồng hồ (giờ, phút, giây)	16 bits, BIN
S2	Địa chỉ bắt đầu phần tử của dữ liệu đồng hồ (giờ, phút, giây)	16 bits, BIN
D	Địa chỉ kết quả	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word									Phần tử lập trình dạng bit								
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•																	
S2	•																	
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, H; T gồm T, HT. C gồm C, HC.

Mô tả



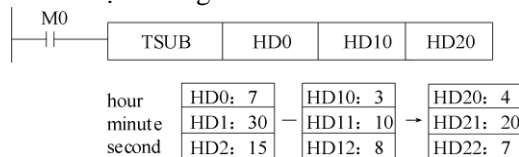
Lưu ý: sự tương ứng của các thanh ghi là cố định, nghĩa là chúng được lưu trữ theo thứ tự giờ, phút và giây.

Nếu kết quả thao tác là 0 giờ, 0 phút, 0 giây, SM20 sẽ ở trạng thái ON.

Các toán hạng S1, S2 và D, mỗi toán hạng chiếm ba thanh ghi. Không sử dụng chúng cho các mục đích khác.

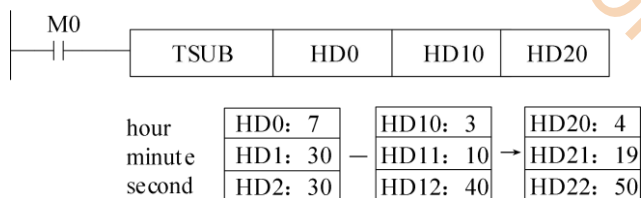
Ví dụ 1:

<Điều kiện chung>



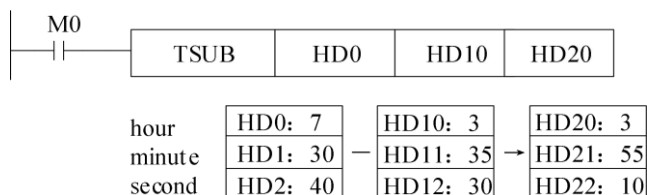
Ví dụ 2:

«Nhỏ hơn 0 giây»



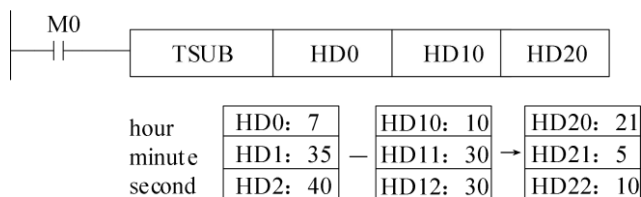
Ví dụ 3:

«Nhỏ hơn 0 phút»



Ví dụ 4:

«Nhỏ hơn 0 giờ»



4-10-7 Lệnh chuyển đổi dữ liệu giờ, phút và giây thành giây [HTOS]

1) Sơ lược

Chuyển đổi dữ liệu giờ, phút và giây thành giây [HTOS]			
16 bits	HTOS	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

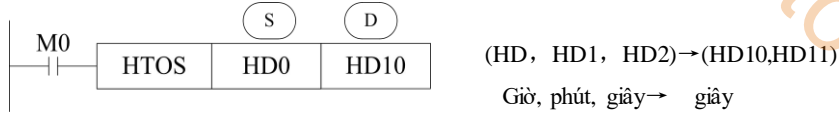
Toán hạng	Chức năng	Kiểu dữ liệu
S	Dữ liệu đồng hồ trước khi chuyển đổi	16 bits, BIN
D	Dữ liệu đồng hồ sau khi chuyển đổi	16 bits, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình dạng word											Phân tử lập trình dạng bit						
	Hệ								Hằng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.m
S	•																	
D	•																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD. CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM. DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS. T gồm T, HT. C gồm C, HC.

Mô tả



- Khi M0 ở trạng thái on, nó sẽ chuyển đổi dữ liệu đồng hồ (giờ, phút và giây) trong ba thanh ghi liên tiếp bắt đầu bằng HD0 thành dữ liệu giây, được lưu trong thanh ghi HD10 (dạng double word).
- Lưu ý: sự tương ứng của các thanh ghi là cố định, nghĩa là chúng được lưu trữ theo thứ tự giờ, phút và giây.
- Các toán hạng S chiếm ba thanh ghi. Không sử dụng chúng cho các mục đích khác.

4-10-8 Lệnh chuyển đổi dữ liệu giây thành giờ, phút và giây[STOH]

1) Sơ lược

Chuyển đổi dữ liệu giây thành giờ, phút và giây [STOH]			
16 bits	-	32 bits	STOH
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

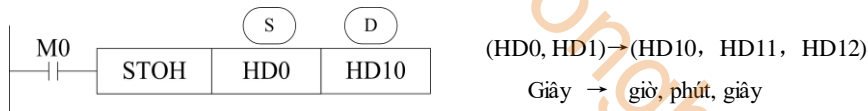
Toán hạng	Chức năng	Kiểu dữ liệu
S	Dữ liệu đồng hồ trước khi chuyển đổi	16 bits, BIN
D	Dữ liệu đồng hồ sau khi chuyển đổi	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hàng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S	●																	
D	●																	

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, H; T gồm T, HT. C gồm C, HC.

Mô tả



- Khi M0 ở trạng thái ON, sẽ thực hiện việc chuyển đổi giá trị thành ghi 32 bit bắt đầu từ HD0 (HD0+HD1) thành trường dữ liệu giờ, phút, giây lưu vào 3 thanh ghi 16 bit bắt đầu từ HD10 (HD10,HD11,HD12)
- Lưu ý: sự tương ứng của các thanh ghi là cố định, nghĩa là chúng được lưu trữ theo thứ tự giờ, phút và giây.
- Các toán hạng S chiếm ba thanh ghi. Không sử dụng chúng cho các mục đích khác.

4-10-9 Lệnh so sánh dữ liệu đồng hồ [TCMP]

1) Sơ lược

So sánh thời gian ba đồng hồ liên tục.

So sánh dữ liệu đồng hồ [TCMP]			
16 bits	TCMP	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

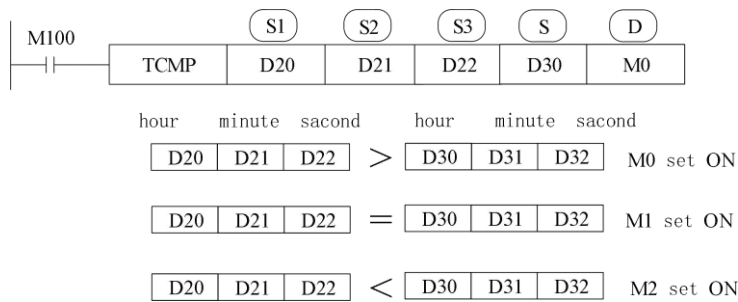
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ phân tử cho dữ liệu giờ	16 bits, BIN
S2	Địa chỉ phân tử cho dữ liệu phút	16 bits, BIN
S3	Địa chỉ phân tử cho dữ liệu giây	16 bits, BIN
S4	Địa chỉ bắt đầu của dữ liệu thời gian thực trên PLC	16 bits, BIN
D2	Địa chỉ bit bắt đầu báo kết quả so sánh giờ, phút, giây	bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình dạng word											Phần tử lập trình dạng bit						
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•																
S2	•	•																
S3	•	•																
S	•	•																
D													•	•				

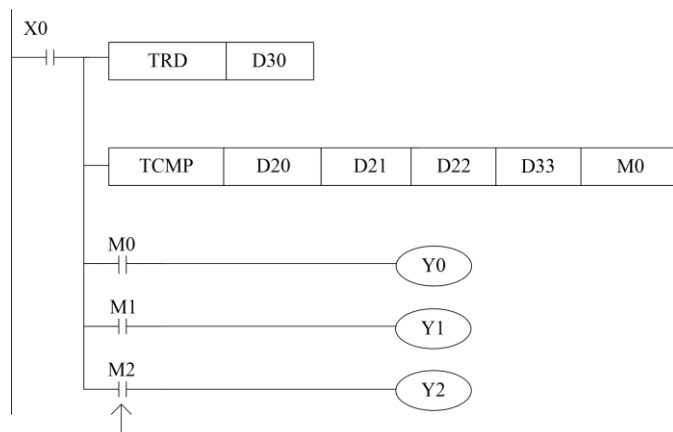
*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, H; T gồm T, HT. C gồm C, HC.

Mô tả



- M100 chuyển từ OFF sang ON, lệnh TCMP hoạt động. So sánh ba thanh ghi bắt đầu từ S4 với ba thanh ghi S1, S2, S3 (năm, tháng, ngày). Khi S1, S2, S3 lớn hơn dữ liệu đồng hồ S4, M0 sẽ ở trạng thái ON. Khi S1, S2, S3 bằng dữ liệu đồng hồ S4 thì M1 ở trạng thái ON. Khi S1, S2, S3 nhỏ hơn dữ liệu đồng hồ S4, M2 ở trạng thái ON.

Ví dụ:



Dữ liệu đồng hồ hiện tại là 15:32:49 7,30,2014 Thứ Tư. Vậy D33=15, D34=32, D35=49 . Nếu thời gian cài đặt là 15:32:49, D20=15, D21=32, D22=49, thì Y1=ON. Nếu thời gian cài đặt là 17:32:49, D20=17, D21=32, D22=49, thì Y0=ON. Nếu thời gian cài đặt là 2:32:5, D20=2, D21=32, D22=5, thì Y2=ON.

4-10-10 Lệnh so sánh dữ liệu ngày tháng (ngày, tháng, năm)[DACMP]

1) Sơ lược

Lệnh so sánh dữ liệu ngày tháng (ngày, tháng, năm)[DACMP]			
16 bits	DACMP	32 bits	-
Điều kiện thực thi	Thường mở/đóng, sườn lên, sườn xuống	Model phù hợp	XD, XL
Yêu cầu phần cứng	Phiên bản V3.4.6 (hoặc V3.5.3a) trở lên	Yêu cầu phần mềm	Phiên bản V3.5.3 trở lên

2) Toán hạng

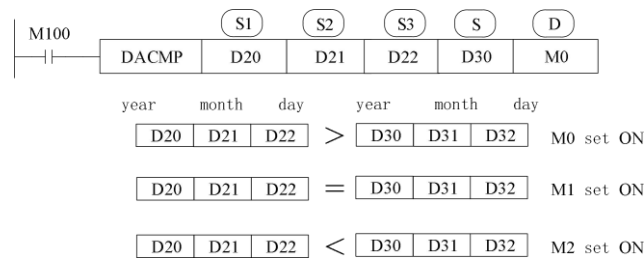
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ phần tử lập trình cho dữ liệu năm	16 bits, BIN
S2	Địa chỉ phần tử lập trình cho dữ liệu tháng	16 bits, BIN
S3	Địa chỉ phần tử lập trình cho dữ liệu ngày	16 bits, BIN
S4	Địa chỉ bắt đầu của dữ liệu thời gian thực trên PLC	16 bits, BIN
D2	Địa chỉ bit bắt đầu báo kết quả so sánh ngày, tháng, năm	bit

3) Phần tử lập trình phù hợp

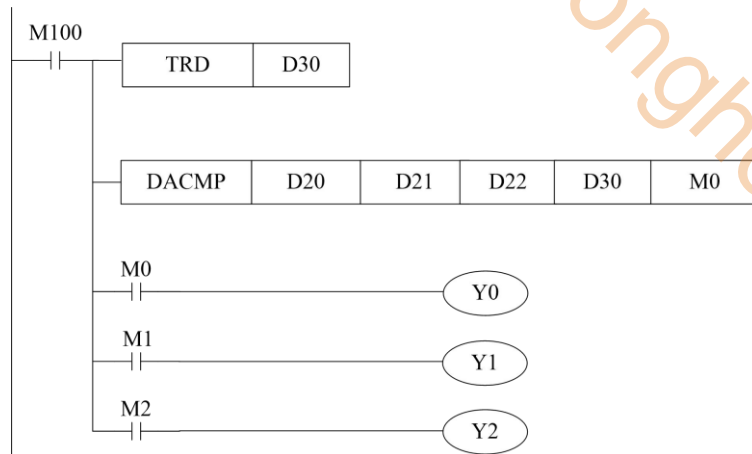
Toán hạng	Phần tử lập trình dạng word										Phần tử lập trình dạng bit							
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•																
S2	•	•																
S3	•	•																
S	•	•																
D													•	•				

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T,HT; C gồm C, HC.

Mô tả



Ví dụ:



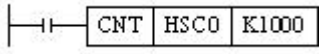
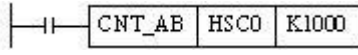
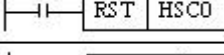
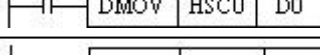
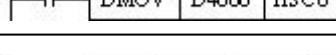
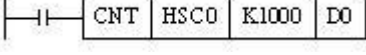
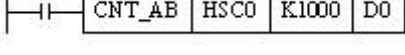
Ví dụ:

Dữ liệu đồng hồ hiện tại là 15:32:49 7,30,2014 Thứ Tư. Vậy D30=14, D31=7, D32=30. Nếu thời gian cài đặt là 1, 6, 2015, D20=15, D21=1, D22=6, thì Y0=ON. Nếu thời gian cài đặt là 7, 30, 2014, D20=14, D21=7, D22=31, thì Y1=BẬT. Nếu thời gian cài đặt là 6, 31, 2014, D20=14, D21=6, D22=31 thì Y2=ON.

Chương 5: BỘ ĐẾM TỐC ĐỘ CAO (HSC)

Chương này sẽ giới thiệu các chức năng của bộ đếm tốc độ cao, bao gồm mô hình đếm tốc độ cao, phương pháp nối dây, đọc/ghi giá trị HSC, đặt lại, v.v.

Danh sách lệnh cho HSC

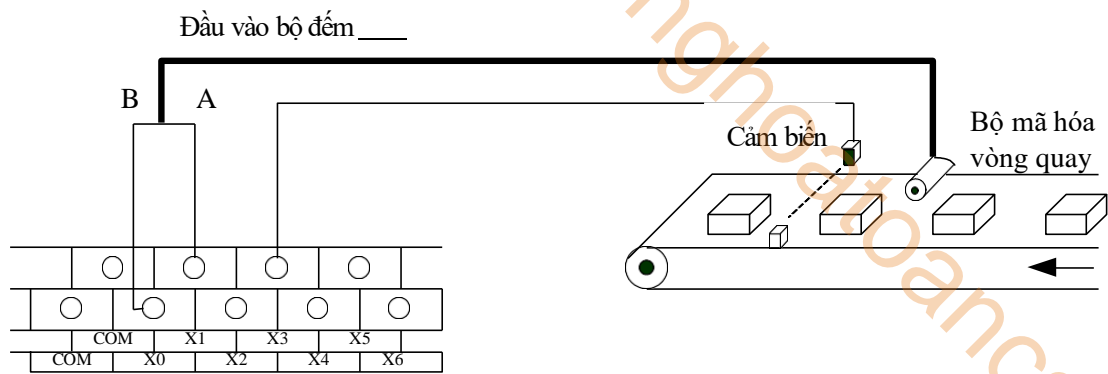
Tên lệnh	Chức năng	Instruction	Chương
Đọc/ghi HSC			
CNT	Bộ đếm dạng single phase		5-7-1
CNT_AB	Bộ đếm dạng AB phase		5-7-2
RST	HSC reset		5-7-3
DMOV	HSC read		5-7-4
DMOV	HSC write		5-7-5
CNT	Single-phase 100-segments high-speed counting (with interruption)		5-9-2
CNT_AB	AB phase 100-segments high speed counting (with interruption)		5-9-3

5-1 Giới thiệu sơ lược về chức năng

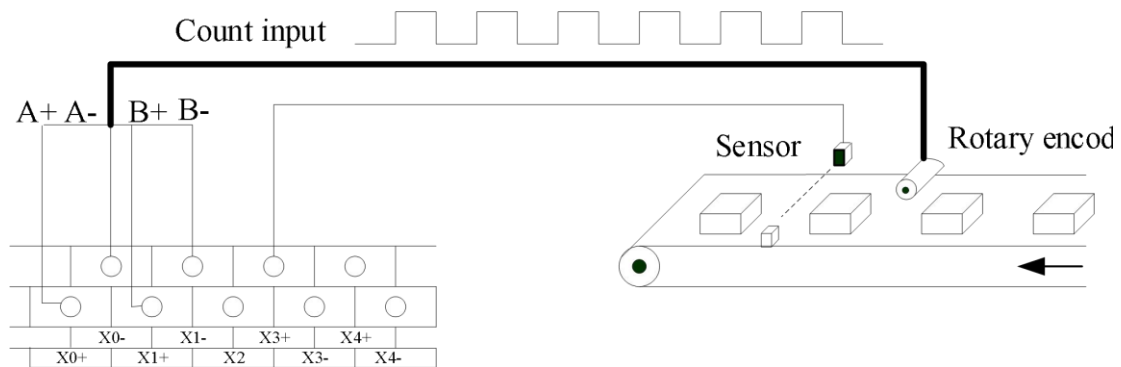
PLC dòng XD, XL có chức năng HSC (Bộ đếm tốc độ cao) sẽ không ảnh hưởng đến chu kỳ quét. Thông qua việc chọn bộ đếm khác nhau, hãy kiểm tra tín hiệu đầu vào tốc độ cao bằng cảm biến phát hiện và bộ mã hóa vòng quay. Tần số thử nghiệm cao nhất có thể đạt tới 80KHz.

Lưu ý:

- (1) Đối với PLC có chế độ đầu vào NPN, vui lòng chọn bộ mã hóa có ngõ ra open collector NPN (OC) sử dụng nguồn DC24V; đối với PLC có chế độ đầu vào PNP, vui lòng chọn bộ mã hóa có ngõ ra open collector PNP (OC) sử dụng nguồn DC24V.



(2) Đầu vào đếm tốc độ cao XD5-48D4T4 có thể nhận tín hiệu vi sai (DIFF), hãy đảm bảo chọn bộ mã hóa tín hiệu vi sai (DIFF).



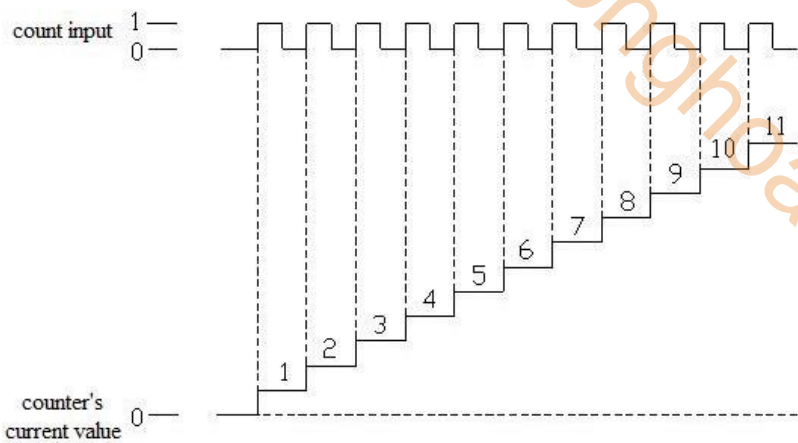
(3) Khi tần số đếm cao hơn 25Hz, vui lòng chọn bộ đếm tốc độ cao.

5-2 Chế độ làm việc của HSC

Bộ đếm tốc độ cao dòng XD và XL có hai chế độ làm việc: Chế độ tăng một pha và chế độ pha AB.

Chế độ tăng một pha

Trong chế độ này, giá trị đếm tăng ở sườn lên của mỗi xung;

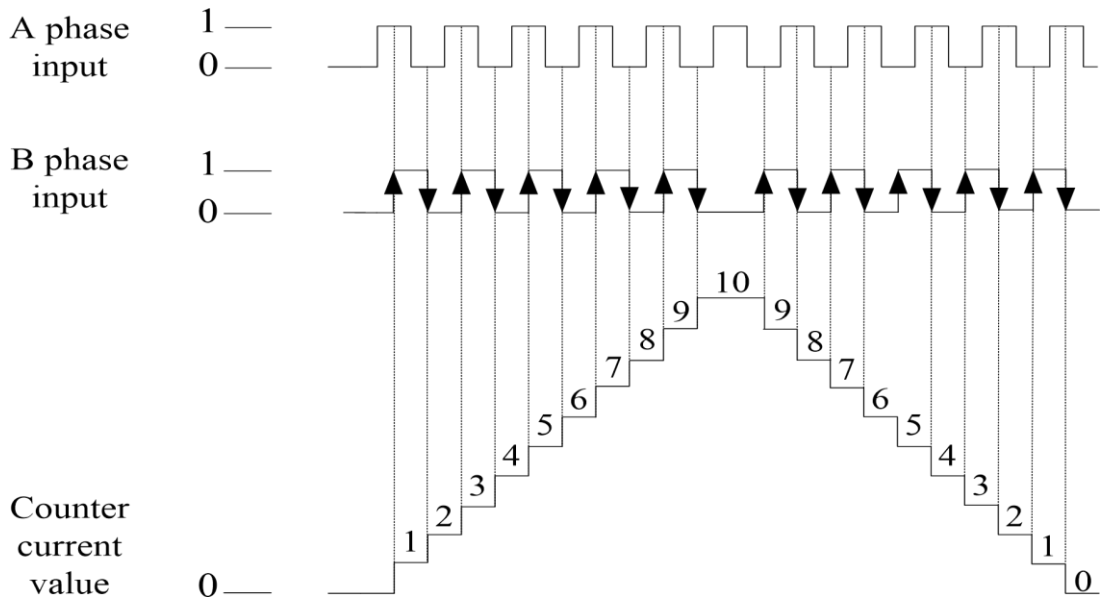


Chế độ pha AB

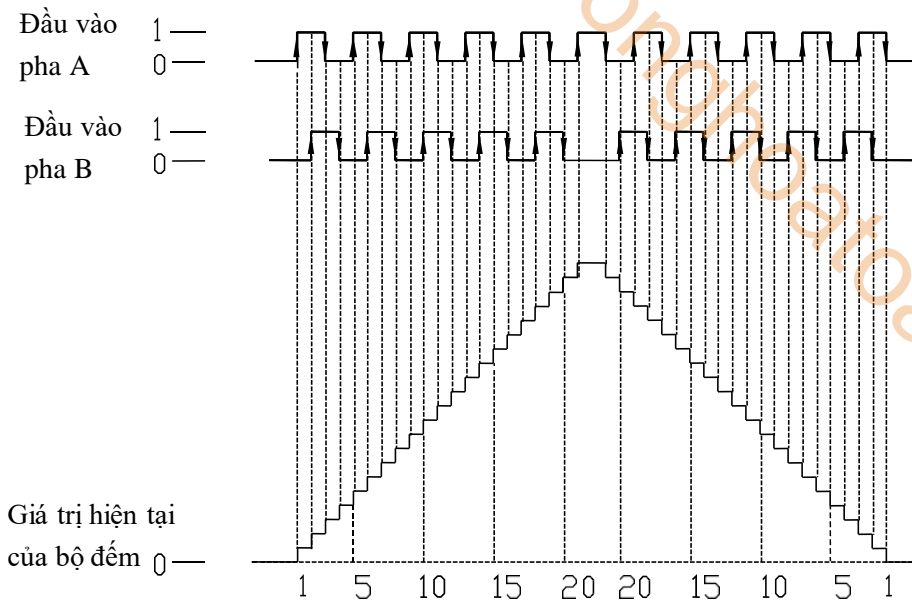
Ở chế độ này, giá trị HSC tăng hoặc giảm theo hai tín hiệu vi sai (pha A và pha B). Theo phép nhân ta có tần số 2 lần và tần số 4 lần nhưng chế độ đếm mặc định là chế độ tần số 4 lần.

Chế độ tần số 2 lần và tần số 4 lần được thể hiện trong hình dưới đây:

Tần số 2 lần



Tần số 4 lần

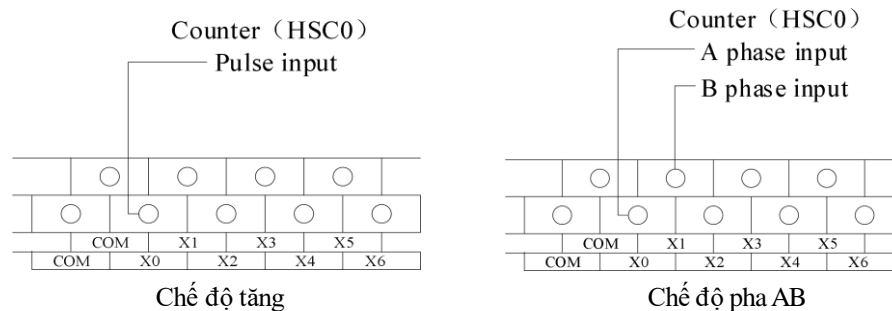


5-3 Phạm vi đếm của HSC

Phạm vi đếm của HSC là: $K-2,147,483,648 \sim K+2,147,483,647$. Nếu giá trị đếm vượt quá phạm vi này thì sẽ xảy ra hiện tượng overflow hoặc underflow; Overflow có nghĩa là giá trị đếm nhảy từ $K+2,147,483,647$ đến $K-2,147,483,648$, sau đó tiếp tục đếm; underflow có nghĩa là giá trị đếm nhảy từ $-2,147,483,648$ đến $+2,147,483,647$ rồi tiếp tục đếm.

5-4 Dây đầu vào HSC

Đối với hệ thống dây đầu vào xung của bộ đếm, sẽ có sự khác biệt tùy thuộc vào model PLC và model bộ đếm khác nhau; một số sơ đồ nối dây đầu vào điển hình được thể hiện như hình dưới: (lấy XD3-60 HSC0 làm ví dụ):



5-5 Phân bổ cổng HSC

1) Danh sách kênh HSC PLC dòng XD:

Model PLC		Kênh HSC	
		Chế độ tăng	Chế độ pha AB
XD1	16/32	0	0
XD2/XD3	16/24/32/42/48/60	3	3
XD5	16/24/32/42/48/60	3	3
	24T4/32T4/48T4/60T4	4	4
	24D2T2	2	2
	48D4T4	8	8
	48T6/60T6	6	6
XDM	60T10	10	10
	24T4/32T4/48T4/60T4	4	4
XDC	24/32/48/60	4	4
XD5E	24/30/48/60	3	3
	30T4	4	4
	60T4	4	4
	60T6	6	6
	60T10	10	10
XDME	30T4/60T4	4	4
	60T10	10	10
XDH	30A16(L)/60T4/60A32	4	4
XL1	16	0	0
XL3	16/32	3	3
XL5	16/32	3	3
	32T4	4	4
XL5E	16/32	3	3
	32T4	4	4
	64T6	6	6
	64T110	10	10
XLME	32T4	4	4
	64T10	10	10
XLH	24A16(L)	4	4

Lưu ý: Các phiên bản phần cứng của XL5E-64T6 bắt đầu với H4 hỗ trợ đếm tốc độ cao sáu kênh, trong khi các phiên bản bắt đầu với H3 chỉ hỗ trợ đếm tốc độ cao bốn kênh.

2) Ý nghĩa của mỗi chữ cái:

U	A	B	Z
Đầu vào xung	Đầu vào pha A	Đầu vào pha B	Bắt xung pha Z

Lưu ý: Chức năng đếm tín hiệu pha Z đang được phát triển.

Trong điều kiện bình thường, tần số đầu vào của X0 và X1 có thể lần lượt đạt 80KHz và 50KHz ở chế độ một pha và pha AB. Các thiết bị đầu cuối khác có tần số tối đa lần lượt là 10KHz và 5KHz ở chế độ một pha và pha AB.

X có thể sử dụng làm ngõ vào thông thường khi không có đầu vào xung tốc độ cao.
 Trong bảng sau, 2 có nghĩa là tần số gấp đôi; 4 có nghĩa là tần số gấp bốn lần; 2/4 có nghĩa là tần số gấp đôi và tần số gấp bốn có thể được điều chỉnh.

XD2-16												
	Chế độ tăng							Chế độ pha AB				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Tần số tối đa	10K	10K	10K					5K	5K	5K		
Tần số gấp bốn								2/4	2/4	2/4		
Gián đoạn bộ đếm	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		

XD3-16, XL3-16												
	Chế độ tăng							Chế độ pha AB				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Tần số tối đa	80K	10K	10K					50K	5K	5K		
Tần số gấp bốn								2/4	2/4	2/4		
Gián đoạn bộ đếm	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		

XD2-24/32/42/48/60, XD3-24/32/48/60, XD5-16/24/32/42/48/60, XD5E-24/30/48/60, XL3-32, XL5-16/32, XL5E-16/32												
	Chế độ tăng							Chế độ pha AB				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Tần số tối đa	80K	80K	10K					50K	50K	5K		
Tần số gấp bốn								2/4	2/4	2/4		
Gián đoạn bộ đếm	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		

XD5-24T4/32T4/48T4/60T4, XDM-24T4/32T4/60T4/60T4L, XDC-24/32/48/60T XD5E-30T4/60T4, XDME-30T4/60T4, XL5-32T4, XL5E-32T4, XLME-32T4, XLH- 24A16/24A16L												
	Chế độ tăng						Chế độ pha AB					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Tần số tối đa	80K	80K	80K	80K			50K	50K	50K	50K		
Tần số gấp bốn							2/4	2/4	2/4	2/4		
Gián đoạn bộ đếm	√	√	√	√			√	√	√	√		
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U							A	
X012											B	
X013											Z	

XD5-48D4T4																
	Chế độ tăng								Chế độ pha AB							
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14
Tần số tối đa	1M	1M	1M	1M	80K	80K	80K	80K	1M	1M	1M	1M	50K	50K	50K	50K
Tần số gấp bốn									2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4
Gián đoạn bộ đếm	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
X0+	U+								A+							
X0-	U-								A-							
X1+									B+							
X1-									B-							
X2																
X3+		U+								A+						
X3-		U-								A-						
X4+										B+						
X4-										B-						
X5																
X6+			U+								A+					
X6-			U-								A-					
X7+											B+					
X7-											B-					
X10																
X11+				U+								A+				
X11-				U-								A-				
X12+												B+				
X12-												B-				
X13																
X14					U								A			
X15													B			
X16																
X17						U								A		
X20														B		

XD5-48D4T4															
	Chế độ tăng							Chế độ pha AB							
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12
X21															
X22							U							A	
X23														B	
X24															
X25								U							A
X26															B
X27															

XDH-30A16/30A16L/60T4/60A32												
	Chế độ tăng						Chế độ pha AB					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Tần số tối đa	200K	200K	200K	200K			100K	100K	100K	100K		
Tần số gấp bốn							2/4	2/4	2/4	2/4		
Gián đoạn bộ đếm	√	√	√	√			√	√	√	√		
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		

XD5-24D2T2																
	Chế độ tăng								Chế độ pha AB							
	HSC 0	HSC 2	HSC 4	HSC 6	HSC 8	HSC1 0	HSC1 2	HSC1 4	HSC 0	HSC 2	HSC 4	HSC 6	HSC 8	HSC1 0	HSC1 2	HSC1 4
Tần số tối đa	1M	1M	80K	80K					1M	1M	50K	50K				
Tần số gấp bốn									2/4	2/4	2/4	2/4				
Gián đoạn bộ đếm	√	√	√	√					√	√	√	√				
X0+	U+								A+							
X0-	U-								A-							
X1+									B+							
X1-									B-							
X2																
X3+		U+							A+							
X3-		U-							A-							
X4+									B+							
X4-									B-							
X5																
X6			U							A						
X7										B						
X10																
X11				U							A					
X12											B					
X13																
X14																
X15																

XD5-48T6/60T6, XD5E-60T6, XL5E-64T6												
	Chế độ tăng						Chế độ pha AB					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Tần số tối đa	80K	80K	80K	80K	80K	80K	50K	50K	50K	50K	50K	50K
Tần số gấp bốn							2/4	2/4	2/4	2/4	2/4	2/4
Gián đoạn bộ đếm	√	√	√	√	√	√	√	√	√	√	√	√
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		
X014					U						A	
X015											B	
X016											Z	
X017						U						A
X020												B
X021												Z

Lưu ý: Các phiên bản phần cứng của XL5E-64T6 bắt đầu với H4 hỗ trợ đếm tốc độ cao sáu kênh, trong khi các phiên bản bắt đầu với H3 chỉ hỗ trợ đếm tốc độ cao bốn kênh.

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10, XL5E-64T10, XLME-64T10												
	Chế độ tăng											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Tần số tối đa	80K	80K	80K	80K	80K	80K	80K	80K	80K	80K		
Tần số gấp bốn												
Gián đoạn bộ đếm	√	√	√	√	√	√	√	√	√	√		
X000	U											
X001												
X002												
X003		U										
X004												
X005												
X006			U									
X007												
X010												
X011				U								
X012												
X013												
X014					U							
X015												
X016							U					
X017												
X020												
X021												
X022								U				
X023												
X024												
X025									U			
X026												
X027												

X030										U		
X031												
X032												
X033										U		
X034												

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10, XL5E-64T10, XLME-64T10												
	Chế độ pha AB											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Tần số tối đa	50K	50K	50K	50K	50K	50K	50K	50K	50K	50K	50K	
Tần số gấp bốn	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	
Gián đoạn bộ đếm	√	√	√	√	√	√	√	√	√	√		
X000	A											
X001	B											
X002	Z											
X003		A										
X004		B										
X005		Z										
X006			A									
X007			B									
X010			Z									
X011				A								
X012				B								
X013				Z								
X014					A							
X015					B							
X016					Z							
X017						A						
X020						B						
X021						Z						
X022							A					
X023							B					
X024							Z					
X025								A				
X026								B				
X027								Z				
X030									A			
X031									B			
X032									Z			
X033										A		
X034										B		
X035										Z		

5-6 Cài đặt nhân đôi tần số đếm pha AB

Đối với chế độ đếm pha AB, số tần số kép có thể được đặt trong các thanh ghi dữ liệu FLASH đặc biệt SFD321, SFD322, SFD323... SFD330, 2 có nghĩa là tần số kép; 4 có nghĩa là tần số gấp bốn lần.

Tên thanh ghi	Chức năng	Giá trị cài đặt	Ý nghĩa
SFD320	Tăng gấp đôi tần số HSC0	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD321	Tăng gấp đôi tần số HSC2	2	Tăng gấp đôi tần số 2 lần

		4	Tăng gấp đôi tần số 4 lần
SFD322	Tăng gấp đôi tần số HSC4	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD323	Tăng gấp đôi tần số HSC6	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD324	Tăng gấp đôi tần số HSC8	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD325	Tăng gấp đôi tần số HSC10	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD326	Tăng gấp đôi tần số HSC12	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD327	Tăng gấp đôi tần số HSC14	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD328	Tăng gấp đôi tần số HSC16	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần
SFD329	Tăng gấp đôi tần số HSC18	2	Tăng gấp đôi tần số 2 lần
		4	Tăng gấp đôi tần số 4 lần

Lưu ý: Sau khi sửa đổi thanh ghi SFD, cần phải khởi động lại bộ đếm tốc độ cao (tức là ngắt kết nối và khởi động lại tình trạng ổ đĩa) để cấu hình mới có hiệu lực!

5-7 Lệnh HSC

Phần này giới thiệu cách sử dụng lệnh đếm tốc độ cao một pha (CNT), lệnh đếm tốc độ cao pha AB (CNT_AB), đặt lại chế độ đếm tốc độ cao, đọc và ghi giá trị đếm tốc độ cao.

5-7-1 Lệnh HSC một pha [CNT]

1) Tóm tắt về lệnh

Lệnh HSC một pha

HSC một pha [CNT]			
Lệnh 16 bit	-	Lệnh 32 bit	CNT
Điều kiện thực thi	Cuộn dây Thường hở/đóng	Model phù hợp	XD, XL (trừ XD1 / XL1)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

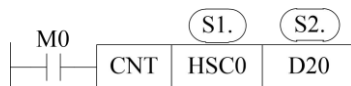
Toán hạng	Chức năng	Loại
S	Xác định mã HSC (VD: HSC0)	32 bit, BIN
D	Xác định giá trị so sánh (VD: K100, D0)	32 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit								
	Hệ								Hàng		Mô đun		Hệ						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	Chỉ có thể là HSC																		
S2	●																		

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

CHỨC NĂNG VÀ HÀNH ĐỘNG



- Khi M0 bật, HSC0 đếm tín hiệu X0 ở chế độ một pha, so sánh giá trị đếm tốc độ cao với giá trị được đặt trong thanh ghi D20. Khi giá trị đếm tốc độ cao bằng giá trị cài đặt, cuộn dây HSC0 được bật ngay lập tức và giá trị đếm được tích lũy trong HSCD0 (dữ liệu dạng double word).
- Nếu quá trình đếm hoàn tất và điều kiện điều khiển M0 không bị ngắt, HSC0 sẽ giữ nguyên trạng thái ON và tiếp tục đếm, đồng thời giá trị đếm trong HSCD0 sẽ tiếp tục tích lũy.
- Nếu quá trình đếm hoàn tất và điều kiện điều khiển M0 bị ngắt, HSC0 sẽ vẫn ở trạng thái ON và giá trị đếm trong HSCD0 sẽ không thay đổi.
- Trong quá trình đếm, nếu ngắt kết nối M0 và kết nối lại thì các giá trị trong HSCD0 sẽ tiếp tục tích lũy sau giá trị đếm gần nhất.
- Trong quá trình đếm, nếu giá trị cài đặt trong D20 thay đổi và giá trị đếm hiện tại nhỏ hơn giá trị cài đặt mới thì giá trị cài đặt mới sẽ được so sánh.
- Chế độ sườn tín hiệu của bộ đếm tốc độ cao một pha có thể được thiết lập bằng cách sử dụng SFD310 đến SFD313 (tương ứng với HSC0 đến HSC6). Lấy HSC0 làm ví dụ, với SFD310 thì 0: đếm sườn lên; 1: biểu thị việc đếm sườn xuống; 2: biểu thị việc đếm cả sườn lên và sườn xuống.

Lưu ý: Chức năng này chỉ được hỗ trợ bởi phiên bản phần mềm PLC V3.4.6 trở lên.

5-7-2 Lệnh HSC pha AB [CNT_AB]

1) Tóm tắt lệnh

Lệnh HSC pha AB

HSC pha AB [CNT_AB]			
Lệnh 16 bit	-	Lệnh 32 bit	CNT_AB
Điều kiện thực thi	Cuộn dây Thường hở/đóng	Model phù hợp	XD, XL(trừ XD1, XL1)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

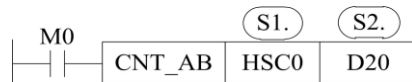
Toán hạng	Chức năng	Loại
S	Xác định mã HSC (VD: HSC0)	32 bit, BIN
D	Xác định giá trị so sánh (VD K100, D0)	32 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit									
	Hệ								Hàng		Mô đun		Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m		
S1	Chỉ có thể là HSC																			
S2	●																			

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S, HS; T gồm T, HT; C bao gồm C, HC.

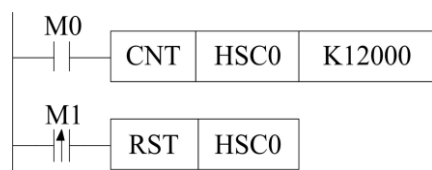
CHỨC NĂNG VÀ HÀNH ĐỘNG



- Khi M0 bật, HSC0 đếm tín hiệu X0, X1 ở chế độ pha AB, so sánh giá trị đếm tốc độ cao với giá trị đặt trong thanh ghi D20. Khi giá trị đếm tốc độ cao bằng giá trị cài đặt, cuộn dây HSC0 được bật ngay lập tức và giá trị đếm được tích lũy trong HSCD0 (dữ liệu dạng double word).
- Nếu điều kiện điều khiển M0 không bị ngắt, HSC0 sẽ giữ nguyên trạng thái ON và tiếp tục đếm, đồng thời giá trị đếm trong HSCD0 sẽ tiếp tục tích lũy.
- Nếu điều kiện điều khiển M0 bị ngắt, HSC0 sẽ vẫn ở trạng thái ON và giá trị đếm trong HSCD0 sẽ không thay đổi.
- Trong quá trình đếm, nếu ngắt kết nối M0 và kết nối lại thì các giá trị trong HSCD0 sẽ tiếp tục tích lũy sau giá trị đếm gần nhất.
- Trong quá trình đếm, nếu giá trị cài đặt trong D20 thay đổi và giá trị đếm hiện tại nhỏ hơn giá trị cài đặt mới thì giá trị cài đặt mới sẽ được so sánh.

5-7-3 Lệnh Reset HSC [RST]

Chế độ đặt lại (reset) của bộ đếm tốc độ cao là chế độ reset bằng phần mềm.



Như được thể hiện ở trên, khi M0 ở trạng thái ON, HSC0 bắt đầu đếm xung đầu vào của cổng X0; khi M1 thay đổi từ OFF sang ON, HSC0 được đặt lại và giá trị đếm trong HSCD0 (dữ liệu dạng double word) bị xóa.

5-7-4 Đọc giá trị HSC [DMOV]

1) Tóm tắt lệnh

Đọc giá trị HSC vào thanh ghi được chỉ định;

Đọc giá trị HSC [DMOV]			
Lệnh 16 bit	-	Lệnh 32 bit	DMOV
Điều kiện thực thi	Thường hở/đóng, sườn lên/xuống	Model phù hợp	XD, XL (trừ XD1, XL1)
Yêu cầu phần cứng		Yêu cầu phần mềm	-

2) Toán hạng

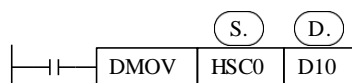
Toán hạng	Chức năng	Loại
S	Xác định mã HSC	32 bit, BIN
D	Xác định thanh ghi đọc/ghi	32 bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình Word											Phân tử lập trình Bit						
	Hệ								Hằng	Mô đun		Hệ						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Chỉ có thể là HSC																	
S2	●																	

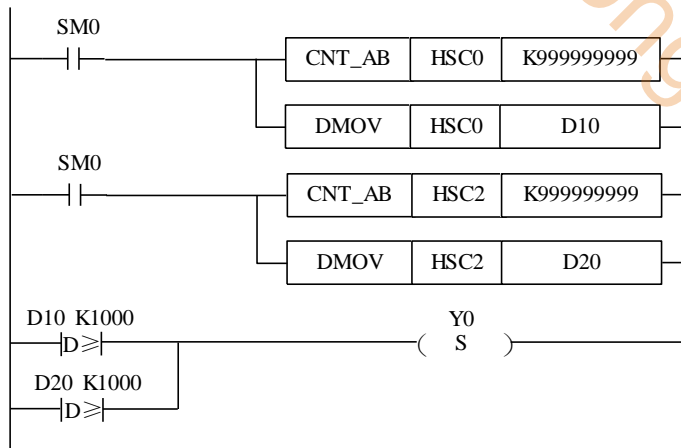
* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S, HS; T gồm T, HT; C bao gồm C, HC.

CHỨC NĂNG VÀ HÀNH ĐỘNG



- Khi điều kiện kích hoạt được thiết lập, giá trị đếm tốc độ cao trong thanh ghi tích lũy HSCD0 (dữ liệu dạng double word) tương ứng với HSC0 của bộ đếm tốc độ cao được đọc vào thanh ghi dữ liệu D10 (dữ liệu dạng double word).
- Bộ đếm tốc độ cao không thể tham gia trực tiếp vào bất kỳ lệnh ứng dụng hoặc lệnh so sánh dữ liệu nào (như DMUL, LD > v.v.) ngoại trừ DMOV, mà chỉ có thể được thực hiện sau khi đọc và ghi vào các thanh ghi khác.
- Vì bộ đếm tốc độ cao là bộ đếm với dữ liệu dạng double word nên nó phải sử dụng lệnh DMOV 32 bit.
- DMOV thường được sử dụng cùng với bộ đếm tốc độ cao.

Ví dụ về chương trình:



5-7-5 Ghi giá trị HSC [DMOV]

1) Tóm tắt lệnh

Ghi giá trị thanh ghi được chỉ định vào HSC;

Ghi giá trị HSC [DMOV]			
Lệnh 16 bit	-	Lệnh 32 bit	DMOV
Điều kiện thực thi	Thường hở/đóng, sườn lên/xuống	Model phù hợp	XD, XL (trừ XD1, XL1)
Điều kiện phần cứng		Điều kiện phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Loại
S	Xác định mã HSC	32 bit, BIN
D	Xác định thanh ghi đọc/ghi	32 bit, BIN

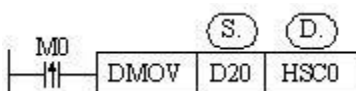
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word								Phần tử lập trình Bit									
	Hệ								Hàng	Mô đun	Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Chỉ có thể là HSC																	
S2	●																	

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD;

DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S, HS; T gồm T, HT; C bao gồm C, HC.

CHỨC NĂNG VÀ HÀNH ĐỘNG



- Khi điều kiện kích hoạt được thiết lập, giá trị trong thanh ghi dữ liệu double word D20 được ghi vào thanh ghi tích lũy HSCD0 (double word) tương ứng với HSC0 của bộ đếm tốc độ cao và dữ liệu gốc được thay thế.
- Bộ đếm tốc độ cao không thể tham gia trực tiếp vào bất kỳ lệnh ứng dụng hoặc lệnh so sánh dữ liệu nào (như DMUL, LD > v.v.) ngoại trừ DMOV, mà chỉ có thể được thực hiện sau khi đọc và ghi vào các thanh ghi khác.
- Vì bộ đếm tốc độ cao là bộ đếm với dữ liệu dạng double word nên nó phải sử dụng lệnh DMOV 32 bit.
- DMOV thường được sử dụng cùng với bộ đếm tốc độ cao.

5-7-6 Sự khác biệt giữa HSC và bộ đếm thông thường

Mặc dù các lệnh của bộ đếm tốc độ cao sử dụng "CNT" giống như các lệnh của bộ đếm thông thường nhưng chức năng của chúng khá khác nhau.

Khi M0 được thay đổi từ OFF sang ON một lần, giá trị của bộ đếm chung được cộng thêm 1. Khi đếm, điều kiện kích hoạt bộ đếm tốc độ cao phải ở trạng thái đóng, tương đương với việc bộ đếm số cao được kích hoạt nhưng giá trị của bộ đếm số cao không thay đổi. Bộ đếm số cao sẽ đếm chỉ khi đầu vào tín hiệu bên ngoài tương ứng nhận được tín hiệu. Nếu ngõ vào tín hiệu bên ngoài có đầu vào tín hiệu và điều kiện kích hoạt của nó không đóng, bộ đếm số cao sẽ không đếm.

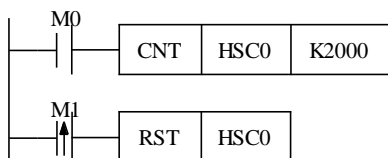
Sự khác biệt được thể hiện trong bảng sau:

Loại bộ đếm	Format lệnh	Chức năng
Bộ đếm thông thường		Đếm số lần chuyển từ OFF sang ON của M0, khi giá trị đếm đạt 2000 thì C0 ở trạng thái ON.
Bộ đếm tốc độ cao		Khi M0 BẬT, đếm tín hiệu đầu vào X0, khi giá trị đếm đạt 2000, HSC0 BẬT, M0 phải luôn BẬT khi đếm.

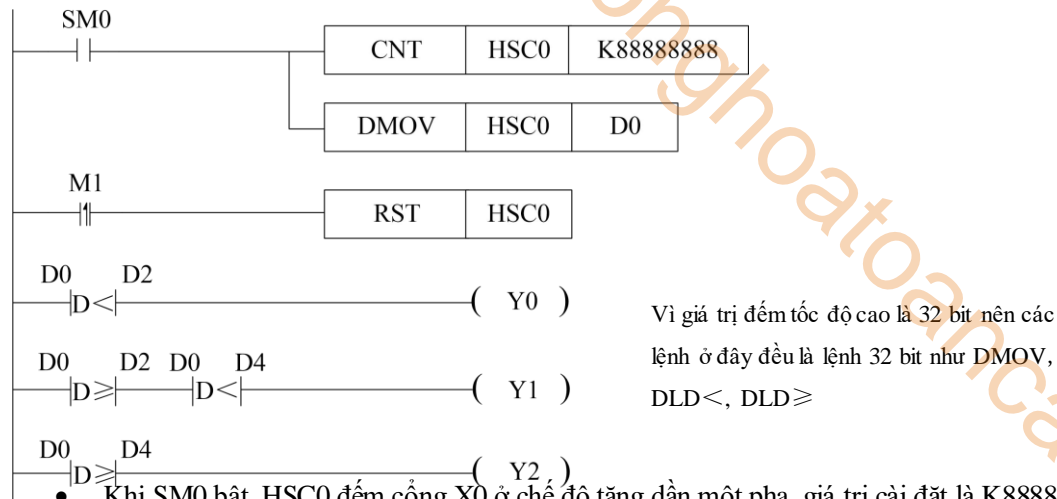
5-8 Ví dụ về HSC

Phần sau đây lấy XD3-60 làm ví dụ minh họa phương pháp lập trình của HSC.

Chế độ tăng một pha

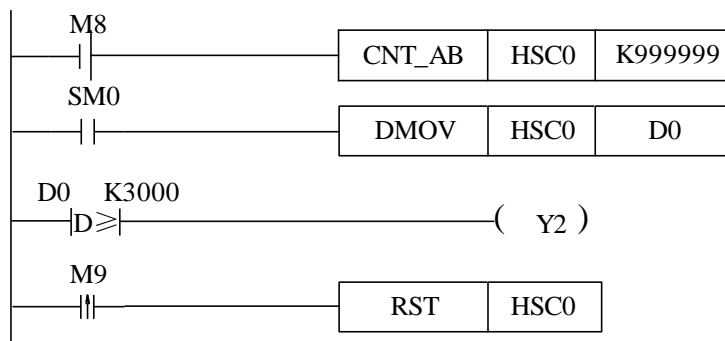


- Khi M0 BẬT, HSC0 đếm sườn tăng của số lần chuyển từ OFF sang ON của cổng đầu vào X0 ở tốc độ cao.
- Khi sườn lên M1 xuất hiện, reset bộ đếm tốc độ cao HSC0 và HSCD0 (dữ liệu dạng double word).

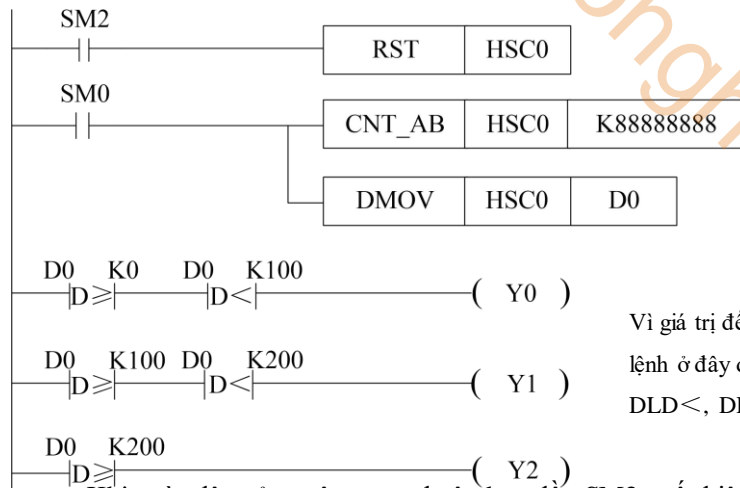


- Khi SM0 bật, HSC0 đếm công X0 ở chế độ tăng dần một pha, giá trị cài đặt là K8888888 và đọc giá trị đếm tốc độ cao tới D0 (double word) trong thời gian thực.
- Khi D0 (double word) nhỏ hơn D2 (double word), Y0 BẬT, khi D0 (double word) bằng hoặc lớn hơn D2 (double word) và nhỏ hơn D4 (double word), Y1 BẬT. Khi D0 (double word) bằng hoặc lớn hơn D4 (double word), Y2 BẬT.
- Khi sườn tăng M1 sắp đến, reset HSC0 và HSCD0 (double word).
- Vì bộ đếm tốc độ cao là bộ đếm với dữ liệu dạng double word nên hãy sử dụng các lệnh double word DLD < và DLD ≥.

Chế độ đầu vào pha AB



- Khi M8 BẬT, HSC0 bắt đầu đếm. Tín hiệu đầu vào từ X0 (pha A) và X1 (pha B).
- Khi SM0 BẬT, giá trị trong HSCD0 (double word) liên quan đến HSC0 được ghi vào D0 (double word) trong thời gian thực.
- Khi giá trị đếm hiện tại lớn hơn 3000, Y2 BẬT.
- Khi sườn lên của M9 sắp đến, reset HSC0 và HSCD0 (double word).




Vì giá trị đếm tốc độ cao là 32 bit nên các lệnh ở đây đều là lệnh 32 bit như DMOV, DLD<, DLD≥

- Khi sườn lên của cuộn xung thuận ban đầu SM2 xuất hiện, tức là ở đầu mỗi chu kỳ quét, HSC0 được đặt lại và giá trị đếm trong HSCD0 bị xóa.
- Khi cuộn dây SM0 bật, HSC0 bắt đầu đếm các công X0 và X1 ở chế độ pha AB. Giá trị cài đặt đếm là K8888888. Đồng thời, giá trị đếm trong HSCD0 (dữ liệu dạng double words) được ghi vào D0 (double words) theo thời gian thực.
- Khi giá trị đếm trong D0 (double words) lớn hơn K0 và nhỏ hơn K100, cuộn dây đầu ra Y0 ở trạng thái ON; khi giá trị đếm trong D0 (double words) lớn hơn hoặc bằng K100 và nhỏ hơn K200, cuộn dây đầu ra Y1 ở trạng thái BẬT; và khi giá trị đếm trong D0 (double words) lớn hơn hoặc bằng K200 thì cuộn dây đầu ra Y2 ở trạng thái ON.
- Vì bộ đếm tốc độ cao là bộ đếm sử dụng dữ liệu dạng double word nên cần sử dụng lệnh so sánh double word DLD≥ và DLD< để so sánh.

5-9 Ngắt HSC

5-9-1 Tổng quan về chức năng và cấu hình bảng điều khiển

Đối với PLC dòng XD/XL, một số bộ đếm tốc độ cao (tham khảo bảng phân bố cổng đầu vào đếm tốc độ cao ở chương 5-5 của từng loại PLC) có giá trị cài đặt là 32 bit trong 1-100 phân đoạn. Khi chênh lệch của thao tác đếm tốc độ cao bằng với giá trị cài đặt của 100 phân đoạn tương ứng, hiện tượng ngắt sẽ xảy ra theo dấu ngắt tương ứng. Nếu giá trị của đoạn N được thiết lập thì phải có dấu ngắt và chương trình ngắt tương ứng với đoạn N. Các dấu ngắt tương ứng với từng bộ đếm tốc độ cao được trình bày ở chương 5-9-4. Khi sử dụng chức năng ngắt đếm tốc độ cao, các lệnh có thể được viết trực tiếp (xem chương 5-9-2 và 5-9-3) hoặc có thể được cấu hình trên bảng phân mềm. Vui lòng nhấp vào biểu tượng  trong phần mềm XDPPro, cửa sổ sau đây sẽ xuất hiện.

High Speed Count 24 Section Config

Single phase 100 segment high speed counting

High Speed C: **HSC0** Compare Value: **D500** Interrupt Address: **HD100**

Opposite Absolute Circulate Cam

Config Value

Compare Value: **99999999** Section Num: **3**

Section Num	Value
Segment1 Count Num:	500
Segment2 Count Num:	20000
Segment3 Count Num:	50000

Read From PLC Write To PLC OK Cancel

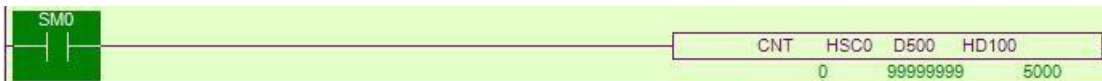
Trong bảng này, chúng ta có thể cấu hình các thông số liên quan đến ngắt quy trình đếm tốc độ cao. Lấy các cài đặt trong hình trên làm ví dụ để giải thích từng chức năng thông số.

Thông số		Chức năng
Single phase 100 segment high speed counting	Đếm tốc độ cao 100 đoạn một pha	Đếm tốc độ cao ở chế độ tăng dần một pha
	Đếm tốc độ cao pha AB 100 đoạn	Đếm tốc độ cao ở chế độ pha AB
High Speed C: HSC0	HSC0~HSC18(32-bit)	Số bộ đếm tốc độ cao tương ứng với cổng đầu vào tốc độ cao
Compare Value: D500	Tùy chỉnh	HSC0 BẬT khi giá trị đếm bằng giá trị trong thanh ghi.
Compare Value: 99999999	Tùy chỉnh	Khi đếm đến giá trị so sánh, HSC0 được BẬT , giá trị so sánh có thể được đặt ở đây hoặc nhập vào thanh ghi so sánh D500.
<input type="checkbox"/> Opposite <input checked="" type="checkbox"/> Absolute	Tương đối	Ngắt đoạn N khi giá trị đếm = giá trị đếm ngắt đoạn N-1 + giá trị cài đặt đoạn N.
	Tuyệt đối	Hiện tượng ngắt sẽ xảy ra khi giá trị đếm bằng giá trị cài đặt.
Interrupt Address: HD100	Tùy chỉnh	Giá trị cài đặt của 100 phân đoạn ngắt đếm tốc độ cao được lưu trong các thanh ghi bắt đầu từ

		HD100 và các giá trị cài đặt được lưu trong các thanh ghi double word HD100, HD102, HD104....
<input type="checkbox"/> Circulate <input type="checkbox"/> Cam	Chu kỳ ngắt	Phải được sử dụng ở chế độ tương đối. Khi tất cả các lần ngắt đã kết thúc, các lần ngắt quy trình đếm tốc độ cao vẫn có thể được tạo theo vòng tròn.
	CAM	Phải sử dụng ở chế độ tuyệt đối. Khi giá trị đếm bằng bất kỳ giá trị cài đặt nào thì sẽ xảy ra hiện tượng ngắt.
Section Num: 3	Tùy chọn từ 1 ~ 100	Nếu được đặt thành 3, có nghĩa là thực hiện ba lần ngắt quy trình đếm tốc độ cao
Value	Tùy chỉnh	Mỗi đoạn tương ứng với một giá trị số lần ngắt, được ghi vào khối địa chỉ bắt đầu từ HD100; thời gian ngắt được xác định bởi chế độ đếm tương đối/tuyệt đối

Để biết cách sử dụng chi tiết các thông số trên, vui lòng xem các chương sau.

Sau khi ghi vào PLC và nhập "OK", cấu hình lệnh ngắt quy trình đếm tốc độ cao được hoàn thành, như thể hiện trong hình sau:



5-9-2 Lệnh HSC 1 pha 100 đoạn [CNT]

1) Tóm tắt

Lệnh HSC 100 đoạn một pha.

HSC 1 pha 100 đoạn [CNT]			
Lệnh 16 bit	-	Lệnh 32 bit	CNT
Điều kiện thực thi	Thường hở/đóng	Model phù hợp	XD, XL (trừ XL1, XD1)
Yêu cầu phân cứng	-	Yêu cầu phân mềm	-

2) Toán hạng

Toán hạng	Chức năng	Loại
S1	Đặt HSC (ví dụ: HSC0)	32 bit, BIN
S2	Đặt giá trị so sánh (ví dụ: K100, D0)	32 bit, BIN
S3	Đặt giá trị cài đặt 100 đoạn	32 bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình Word										Phân tử lập trình Bit								
	Hệ								Hàng	Mô đun	Hệ								
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	Chỉ có thể là HSC																		
S2	●								●										
S3	●																		

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S, HS; T gồm T, HT; C bao gồm C, HC.

Mô tả



- Khi bộ đếm tốc độ cao HSC0 đếm ở chế độ một pha, giá trị đếm tốc độ cao được so sánh với khối dữ liệu bắt đầu từ HD100 (chẳng hạn như HD102, HD102, HD104 và các thanh ghi từ kép khác), nó sẽ ngay lập tức tạo ra hiện tượng ngắt đếm tốc độ cao tương ứng khi đáp ứng điều kiện, tham khảo chương 5-9-4 để biết thêm về mỗi phần của dấu ngắt tương ứng.
- Trong quá trình đếm tốc độ cao, việc sửa đổi giá trị cài đặt của 100 đoạn sẽ không có hiệu lực.
- Trong quá trình đếm tốc độ cao, điều kiện điều khiển M0 không thể bị ngắt kết nối. Nếu M0 bị ngắt kết nối và sau đó được khởi động lại, sẽ không có hiện tượng ngắt nào xảy ra. Bộ đếm tốc độ cao phải được đặt lại trước, sau đó BẬT lại M0 để có thể ngắt.
- Khi quy trình ngắt kết thúc trong một lần thực thi, nếu cần bắt đầu lại quy trình ngắt thì trước tiên bộ đếm tốc độ cao phải được đặt lại, sau đó điều kiện phải BẬT lại.
- Trong chế độ vòng lặp ngắt, các lần ngắt có thể được tạo theo trình tự miễn là M0 vẫn giữ nguyên trạng thái.

5-9-3 Lệnh HSC pha AB 100 đoạn [CNT_AB]

1) Tóm tắt

Lệnh HSC 100 đoạn pha AB.

HSC 100 đoạn pha AB [CNT_AB]			
Lệnh 16 bit	-	Lệnh 32 bit	CNT_AB
Điều kiện thực thi	Thường hở/đóng	Model phù hợp	XD, XL (trừ XL1, XD1)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Loại
S1	Đặt HSC (chẳng hạn như: HSC0)	32 bit, BIN
S2	Đặt giá trị so sánh (chẳng hạn như: K100, D0)	32 bit, BIN
S3	Đặt giá trị cài đặt 100 đoạn	32 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Thành phần mềm Word										Phân tử lập trình Bit									
	Hệ								Hãng		Mô đun		Hệ							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m		
S1	Chỉ có thể là HSC																			
S2	•								•											
S3	•																			

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S, HS; T gồm T, HT; C bao gồm C, HC.

Mô tả



- Khi bộ đếm tốc độ cao HSC0 đếm ở chế độ pha AB, giá trị đếm tốc độ cao được so sánh với khối dữ liệu bắt đầu từ HD100 (chẳng hạn như HD102, HD102, HD104 và các thanh ghi double word khác), nó sẽ ngay lập tức tạo ra hiện tượng ngắt đếm tốc độ cao tương ứng khi đáp ứng điều kiện, vui lòng tham khảo chương 5-9-4 để biết thêm về mỗi phần của dấu ngắt tương ứng.
- Trong quá trình đếm tốc độ cao, việc sửa đổi giá trị cài đặt của 100 đoạn là không có hiệu lực.
- Trong quá trình đếm tốc độ cao, điều kiện điều khiển M0 không thể bị ngắt kết nối. Nếu M0 bị ngắt kết nối và sau đó được khởi động lại, sẽ không có hiện tượng ngắt nào xảy ra. Bộ đếm tốc độ cao phải được đặt lại trước, sau đó BẬT lại M0 để tạo ra hiện tượng ngắt.
- Khi quá trình ngắt kết thúc trong một lần thực thi, nếu cần bắt đầu lại quá trình ngắt thì trước tiên bộ đếm tốc độ cao phải được đặt lại, sau đó điều kiện điều khiển phải BẬT lại.
- Trong chế độ vòng lặp ngắt, các lần ngắt có thể được tạo ra theo trình tự miễn là M0 vẫn giữ nguyên trạng thái.

5-9-4 Cờ ngắt của HSC

Cờ ngắt 100 đoạn của mỗi HSC được thể hiện trong bảng sau. Ví dụ: cờ ngắt 100 đoạn của HSC0 là I2000, I2001, I2002..... I2099.

HSC	Cờ ngắt					
	Đoạn 1	Đoạn 2	Đoạn 3	...	Đoạn N	Đoạn 100
HSC0	I2000	I2001	I2002	...	I(2000+N-1)	I2099
HSC2	I2100	I2101	I2102	...	I(2100+N-1)	I2199
HSC4	I2200	I2201	I2202	...	I(2200+N-1)	I2299
HSC6	I2300	I2301	I2302	...	I(2300+N-1)	I2399
HSC8	I2400	I2401	I2402	...	I(2400+N-1)	I2499
HSC10	I2500	I2501	I2502	...	I(2500+N-1)	I2599
HSC12	I2600	I2601	I2602	...	I(2600+N-1)	I2699
HSC14	I2700	I2701	I2702	...	I(2700+N-1)	I2799
HSC16	I2800	I2801	I2802	...	I(2800+N-1)	I2899
HSC18	I2900	I2901	I2902	...	I(2900+N-1)	I2999

5-9-5 Ý nghĩa giá trị cài đặt ở chế độ tuyệt đối hoặc tương đối

Ý nghĩa của giá trị cài đặt là khác nhau ở chế độ tuyệt đối và tương đối. Chế độ tương đối/tuyệt đối có thể được đặt trong bảng điều khiển phần mềm. Nó cũng có thể được sửa đổi bằng thanh ghi Flash đặc biệt SFD330.

(Lưu ý: Điều kiện điều khiển phải OFF và ON lại để cấu hình có hiệu quả.)

0: Chế độ tương đối

1: Chế độ tuyệt đối

- **Chế độ tương đối**

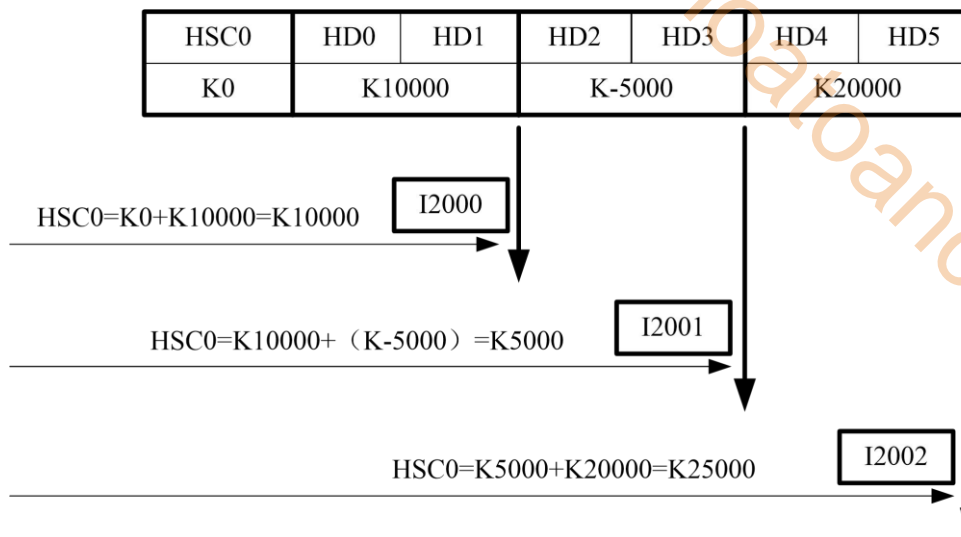
Trong chế độ tương đối, giá trị cài đặt của việc đếm 100 đoạn tốc độ cao là giá trị tích lũy tương đối. Khi giá trị đặt của bộ đếm bằng tổng giá trị số lượng ngắt của phân đoạn N-1 và giá trị đặt của phân đoạn N, thì lệnh ngắt phân đoạn N được tạo ra.

Ngắt thứ N sẽ được làm tương ứng với cài đặt ngắt thứ N đã lập trình. Thanh ghi cài đặt ngắt N+1 được dành riêng cho các mục đích khác.

Ví dụ 1:

Giá trị hiện tại của HSC0 là 0, giá trị đặt trước của phân đoạn một là 10000, giá trị đặt trước trong phân đoạn 2 là -5000, giá trị đặt trước trong phân đoạn 3 là 20000. Khi bắt đầu đếm, khi giá trị hiện tại của bộ đếm là 10000, nó sẽ tạo ra đoạn 1 lệnh ngắt I2000; khi giá trị hiện tại của bộ đếm là 5000, nó tạo ra lệnh ngắt đoạn 2 I2001; khi giá trị hiện tại của bộ đếm là 25000, nó tạo ra lệnh ngắt đoạn 3 I2002.

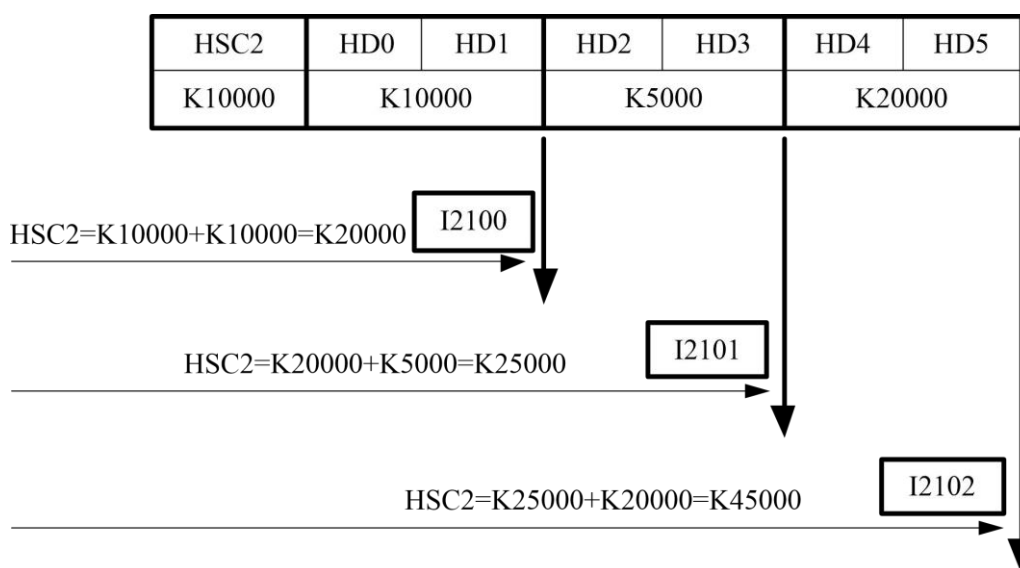
Xem biểu đồ bên dưới:



Ví dụ 2:

Giá trị hiện tại của HSC2 là 10000, giá trị đặt trước của đoạn 1 là 10000, giá trị đặt trước của đoạn 2 là 5000, giá trị đặt trước của đoạn 3 là 20000. Khi bắt đầu đếm, khi giá trị hiện tại của bộ đếm là 20000, nó sẽ tạo ra ngắt đoạn 1 I2100; khi giá trị hiện tại của bộ đếm là 25000, nó tạo ra hiện tượng ngắt đoạn 2 I2101; khi giá trị hiện tại của bộ đếm là 45000, nó tạo ra hiện tượng ngắt đoạn 3 I2102.

Xem biểu đồ bên dưới:

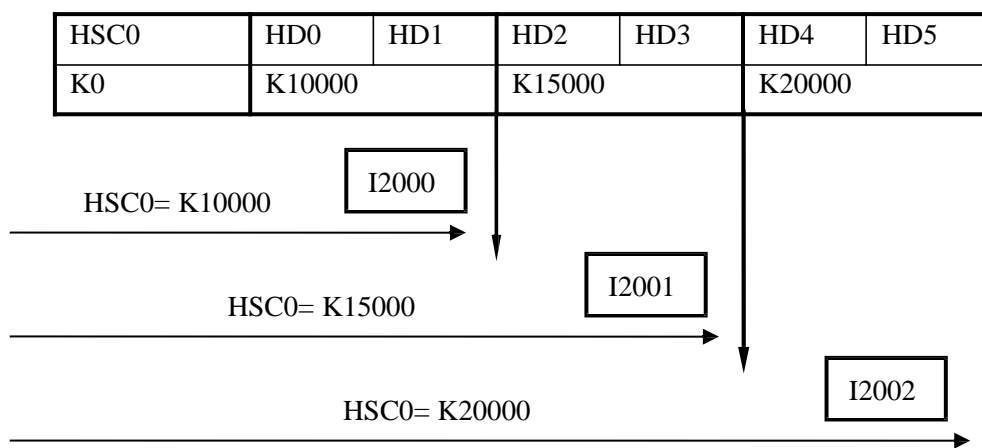


- **Chế độ Tuyệt đối**

Ở chế độ tuyệt đối, hiện tượng ngắt xảy ra khi giá trị đếm bằng giá trị cài đặt của từng phần của bộ đếm. Ngắt thứ N sẽ được làm tương ứng với cài đặt ngắt thứ N đã lập trình. Thanh ghi cài đặt ngắt N+1 được dành riêng cho các mục đích khác.

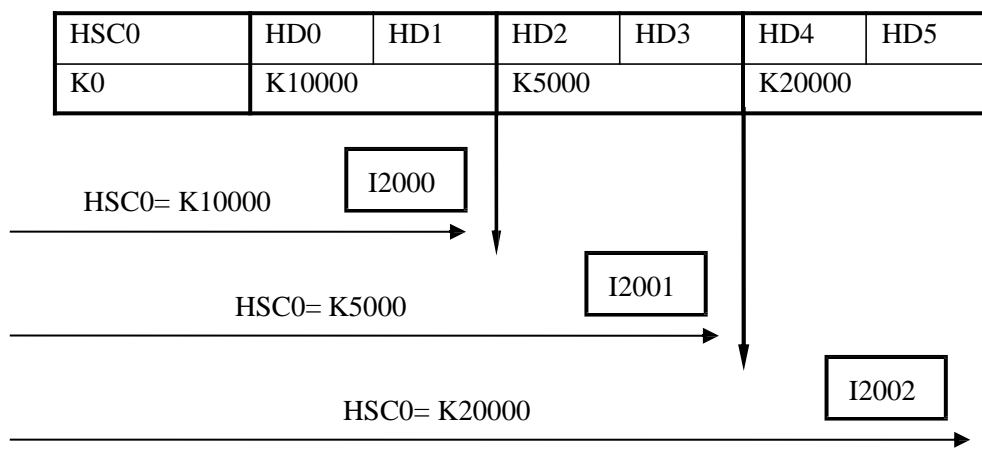
Ví dụ 1:

Giá trị hiện tại của bộ đếm HSC0 là 0, giá trị cài đặt của đoạn 1 là 10000, giá trị cài đặt của đoạn 2 là 15000 và giá trị cài đặt của đoạn 3 là 20000. Khi bắt đầu đếm, nếu giá trị hiện tại của bộ đếm là 10000, ngắt đoạn 1 I2000 được tạo ra; khi giá trị hiện tại của bộ đếm là 15000, ngắt đoạn 2 I2001 được tạo ra; khi giá trị hiện tại của bộ đếm bằng 20000, ngắt đoạn 3 I2002 được tạo ra.



Ví dụ 2:

Giá trị hiện tại của bộ đếm HSC2 là 5000, giá trị cài đặt đoạn 1 là 10000, giá trị cài đặt đoạn 2 là 5000 và giá trị cài đặt đoạn 3 là 20000. Khi bắt đầu đếm, nếu giá trị hiện tại của bộ đếm là 10000, ngắt đoạn 1 I2100 được tạo ra; khi giá trị hiện tại của bộ đếm là 5000, ngắt đoạn 2 I2101 được tạo ra; khi giá trị hiện tại của bộ đếm bằng 20000, ngắt đoạn 3 I2102 được tạo ra.



Lưu ý: Khi quá trình đếm tuyệt đối được thực hiện ở chế độ non-cam, các lần ngắt đếm được tạo ra tuần tự, tức là ngắt đoạn 1, ngắt đoạn 2, ngắt đoạn 3... Khi xảy ra ngắt đoạn, không lần ngắt nào xảy ra ngay cả khi giá trị đếm đạt đến giá trị đã cài đặt của đoạn một lần nữa.

Như trong ví dụ trên, nếu giá trị đếm được tăng từ 4000 lên 5000 và 10000 sau khi xảy ra ngắt đoạn 1 và 2 thì hiện tượng ngắt đoạn 1 và 2 sẽ không xảy ra nữa và ngắt đoạn 3 sẽ xảy ra khi giá trị đếm tiếp tục tăng lên 20000.

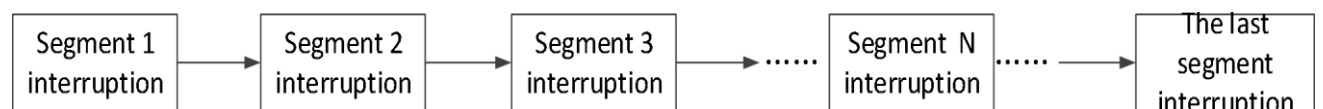
5-9-6 Chế độ chu kỳ ngắt HSC

Chế độ 1: Vòng lặp đơn (chế độ bình thường)

Hiện tượng ngắt HSC sẽ không xảy ra sau khi quá trình đếm kết thúc. Các điều kiện sau đây có thể bắt đầu lại hiện tượng ngắt.

- (1) đặt lại HSC
- (2) Khởi động lại điều kiện kích hoạt HSC

Hiện tượng ngắt được tạo ra theo trình tự sau đây trong quá trình thực hiện vòng lặp đơn:

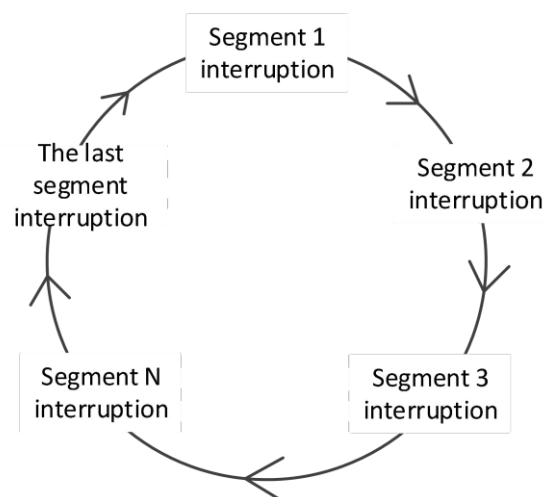


Chế độ 2: Vòng lặp liên tục

Ngắt vòng lặp liên tục chỉ phù hợp với chế độ đếm tương đối. Ở chế độ vòng lặp liên tục, quá trình ngắt sẽ bắt đầu lại sau khi hoàn thành. Chế độ này đặc biệt phù hợp cho ứng dụng sau:

- (1) Chuyển động qua lại liên tục.
- (2) Tạo hiện tượng ngắt chu kỳ theo xung cố định.

Khi thực hiện gián đoạn vòng lặp liên tục (không kích hoạt chức năng cam), các lần ngắt xảy ra theo thứ tự sau :



Thông qua cài đặt SFD331, người dùng có thể chuyển đổi giữa chế độ vòng lặp đơn hoặc chế độ vòng lặp liên tục. Thông số cài đặt chi tiết được thể hiện dưới đây:

(Lưu ý: cài đặt sẽ có hiệu lực sau khi TẮT và BẬT lại điều kiện điều khiển)

Địa chỉ	HSC	Cài đặt
Bit0	Chu kỳ ngắt HSC 100 đoạn (HSC0)	0: vòng lặp đơn 1: vòng lặp liên tục
Bit1	Chu kỳ ngắt HSC 100 đoạn (HSC2)	
Bit2	Chu kỳ ngắt HSC 100 đoạn (HSC4)	
Bit3	Chu kỳ ngắt HSC 100 đoạn (HSC6)	
Bit4	Chu kỳ ngắt HSC 100 đoạn (HSC8)	
Bit5	Chu kỳ ngắt HSC 100 đoạn (HSC10)	
Bit6	Chu kỳ ngắt HSC 100 đoạn (HSC12)	
Bit7	Chu kỳ ngắt HSC 100 đoạn (HSC14)	
Bit8	Chu kỳ ngắt HSC 100 đoạn (HSC16)	
Bit9	Chu kỳ ngắt HSC 100 đoạn (HSC18)	

5-9-7 Chức năng CAM ngắt bộ đếm tốc độ cao

Cam đếm tốc độ cao: Sau khi thiết lập tất cả giá trị cài đặt ngắt, chức năng cam đếm tốc độ cao được chọn. Khi giá trị đếm tốc độ cao bằng bất kỳ giá trị cài đặt ngắt nào, thì quá trình ngắt đếm tốc độ cao tương ứng (giống như dấu ngắt đếm tốc độ cao 100 đoạn) được thực hiện ngay lập tức. Khi giá trị đếm tốc độ cao thay đổi liên tục, việc ngắt tốc độ cao tương tự của cam có thể được thực hiện lặp đi lặp lại.

Cam đếm tốc độ cao không chỉ có thể thực hiện đầy đủ chức năng ngắt chuỗi tuần hoàn của cam điện tử thông thường mà còn có thể tạo ra nhiều lần ngắt điểm dương và âm trong một chu kỳ. Nó được sử dụng rộng rãi trong các hệ thống điều khiển của máy cuộn dây và máy đóng gói tốc độ cao.

Lưu ý: Chức năng CAM chỉ phù hợp với chế độ đếm tuyệt đối.

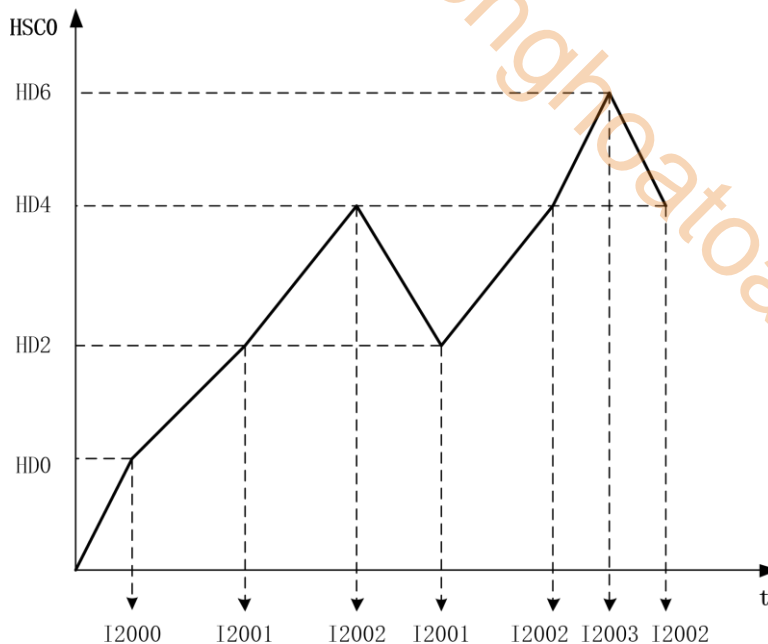
Chức năng cam có thể được thiết lập bằng bảng cấu hình trong phần mềm XINJE PLC hoặc bằng thanh ghi Flash đặc biệt SFD332: (Lưu ý: Điều kiện điều khiển phải được TẮT và BẬT lại để cấu hình có hiệu lực)

0: Không kích hoạt chức năng cam

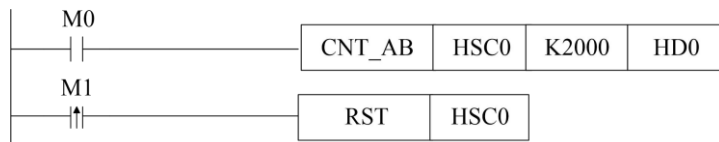
1: Kích hoạt chức năng cam

Ví dụ:

Bốn giá trị được lưu trữ trong bốn thanh ghi double word liên tiếp bắt đầu bằng thanh ghi HD0. Khi HSC0 bắt đầu đếm, nếu giá trị đếm HSC0 bằng bất kỳ thanh ghi nào trong bốn thanh ghi, tín hiệu ngắt tương ứng sẽ được tạo ngay lập tức như được thể hiện trong hình sau:



5-9-8 Ngắt sử dụng ghi chú và địa chỉ thông số



```
LD M0 // Điều kiện kích hoạt HSC M0 (cũng là điều kiện đếm ngắt)
CNT_AB HSC0 K2000 HD0 // Cài đặt địa chỉ đầu HSC và 100 đoạn
LDP M1 // Điều kiện kích hoạt đặt lại HSC
RST HSC0 // Đặt lại HSC và 100 đoạn (cũng đặt lại quá trình ngắt)
```

Như trong ví dụ trên (lưu ý: chương trình con cho quá trình ngắt bị bỏ qua, xem ví dụ về ứng dụng ở chương 5-9-9). Thanh ghi dữ liệu HD0 thiết lập địa chỉ bắt đầu vùng cho giá trị thiết lập của 100 đoạn và sau đó lưu giá trị thiết lập của 100 đoạn ở dạng từ kép (double word).

Cần chú ý khi sử dụng ngắt đếm tốc độ cao:

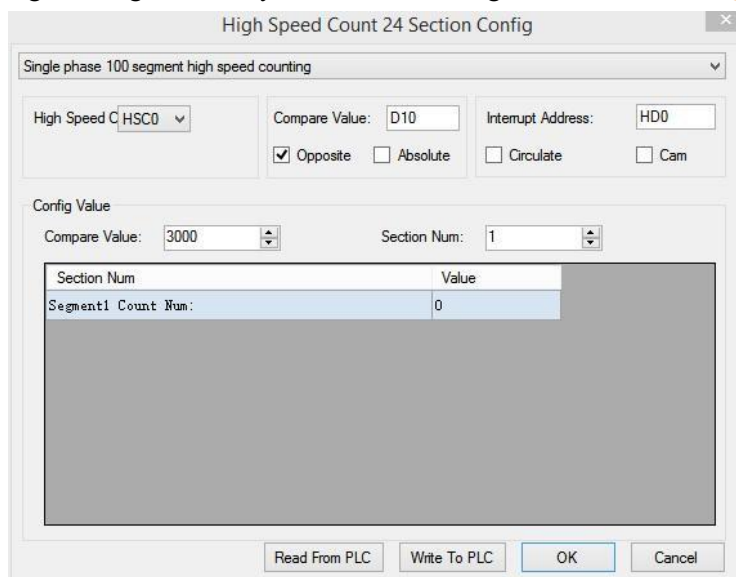
- Thanh ghi sau đoạn cuối cùng không cần đặt thành 0, nhưng phải được lưu lại và không thể sử dụng cho mục đích khác. Ví dụ có 3 đoạn, đoạn 1 là HD0, đoạn 2 là HD2, đoạn 3 là HD4 thì HD6 được lưu.
- Không được phép đặt giá trị cài đặt ngắt mà không ghi chương trình ngắt, nếu không sẽ xảy ra lỗi.
- Quá trình ngắt 100 đoạn của bộ đếm tốc độ cao lần lượt được tạo ra, nghĩa là nếu lần ngắt đầu tiên không xảy ra thì lần ngắt thứ hai sẽ không xảy ra.
- Trong quá trình đếm tốc độ cao, nếu giá trị đếm hiện tại bị thay đổi bởi lệnh DMOV, ADD (DMOV K1000 HSCD0), thì giá trị ngắt sẽ không thay đổi tại thời điểm này. Vui lòng không thay đổi giá trị HSCD khi bộ đếm tốc độ cao đang chạy.

Một số thông số có thể được sửa đổi trong các thanh ghi Flash đặc biệt, như trong bảng sau:

Thông số	Địa chỉ thanh ghi	Giá trị cài đặt
Chế độ đếm	SFD330	0: tương đối 1: tuyệt đối
Chế độ thực thi	SFD331	0: thực thi 1 lần 1: chu kỳ ngắt
Chức năng CAM	SFD332	0: không kích hoạt 1: kích hoạt chức năng cam

Các thông số trên cũng có thể được cấu hình trên bảng cấu hình như sau:

Di chuyển chuột qua lệnh đếm tốc độ cao và nhấp chuột phải vào nó. Chọn "CNT_AB Instruction Parameter Configuration" từ menu thả xuống. Một bảng cấu hình sẽ xuất hiện để cấu hình các thông số trong cửa sổ này, như thể hiện trong hình sau:

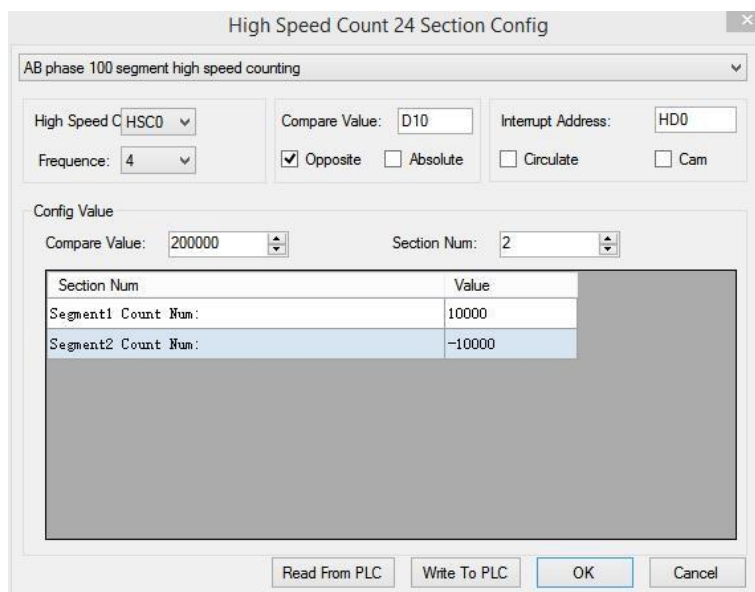


5-9-9 Ứng dụng ngắt HSC

Ứng dụng 1:

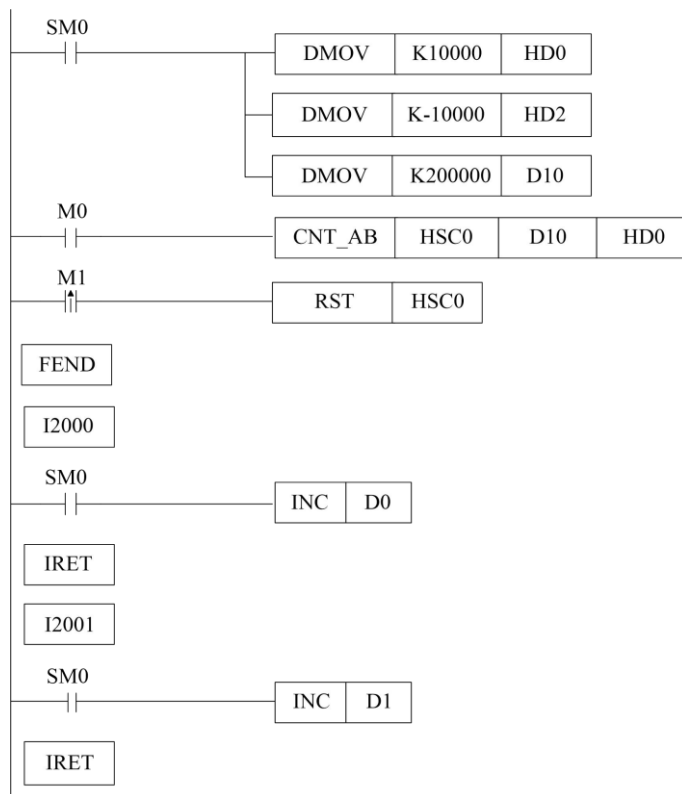
Khi M0 BẬT, HSC0 bắt đầu đếm. Giá trị đếm được lưu trữ trong địa chỉ bắt đầu từ HD0. Khi đạt đến giá trị cài đặt thì sẽ xảy ra ngắt. Khi sườn lên của M1 sắp đến, hãy xóa HSC0.

Phương pháp 1: Cấu hình các thông số thông qua phần mềm XDPpro :



Mục cấu hình	Chức năng
High speed counter	Chọn HSC, phạm vi từ HSC0 đến HSC18
Frequency	Chọn tần số HSC nhân đôi (2 hoặc 4)
Compare value	Giá trị có thể là thanh ghi hoặc hằng số, trong ví dụ này, khi giá trị đếm đạt đến giá trị so sánh, HSC0 sẽ BẬT. Ở đây giá trị so sánh là 200000 được lưu trong D10.
Relative and absolute	HSC là chế độ tương đối hoặc chế độ tuyệt đối
Interrupt address	Các thanh ghi khởi động để lưu trữ giá trị đặt trước ngắt 100 đoạn
Circulate	Chế độ ngắt 100 đoạn có theo chu kỳ hay không
Cam	Chức năng cam được thực thi khi bất kỳ giá trị thiết lập nào của quá trình ngắt đếm tốc độ cao 100 đoạn bằng với giá trị đếm.

Phương pháp 2: Tạo chương trình



Lệnh:

```

LD    SM0           //SM0 là cuộn dây thường ON
DMOV K10000HD0     //giá trị đặt trước đoạn 1 HD0 là 10000
DMOV K-10000 HD2   // Giá trị đặt trước đoạn 2 HD2 là -10000
DMOV K200000  D10  // đặt giá trị so sánh HSC
LD    M0           // Điều kiện kích hoạt HSC M0
CNT_AB HSC0 D10 HD0 //Lệnh ngắt HSC
LDP   M1           //Điều kiện đặt lại HSC M1
RST   HSC0        //đặt lại HSC và ngắt 100 đoạn
FEND                                     //chương trình chính kết thúc
I2000                                    //cờ ngắt đoạn 1
LD    SM0         //SM0 là cuộn dây thường ON

```

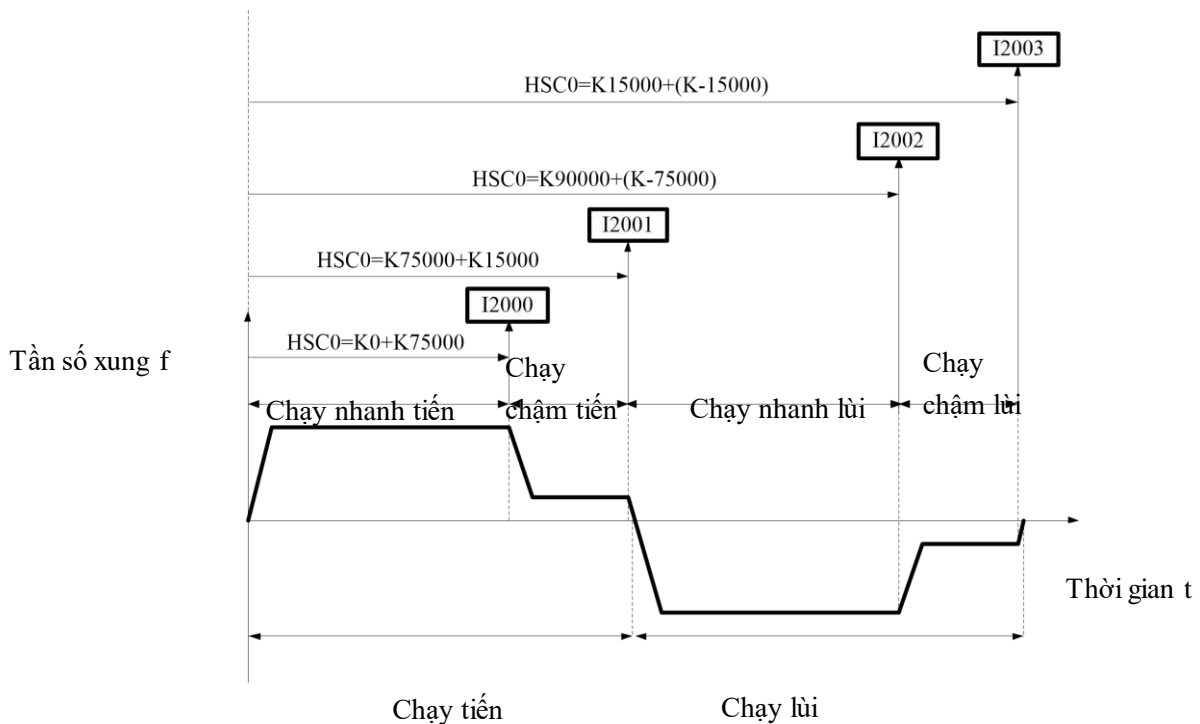
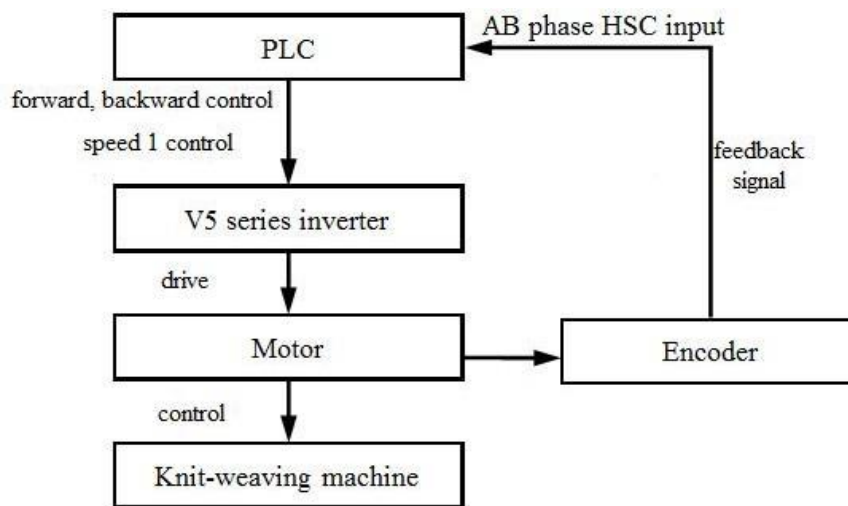
```

INC    D0          //D0= D0+1
IRET                   //cờ kết thúc lệnh ngắt
I2001                   //cờ ngắt đoạn 2
LD     SM0         //SM0 là cuộn dây thường ON
INC    D1          //D1= D1+1
IRET                   //cờ kết thúc lệnh ngắt

```

Ứng dụng 2: máy dệt kim (chế độ vòng lặp liên tục)

Nguyên lý máy: Điều khiển biến tần thông qua PLC, từ đó điều khiển động cơ. Trong khi đó, thông qua tín hiệu phản hồi từ bộ mã hóa, điều khiển máy dệt kim và vị trí chính xác.



Dưới đây là chương trình PLC:

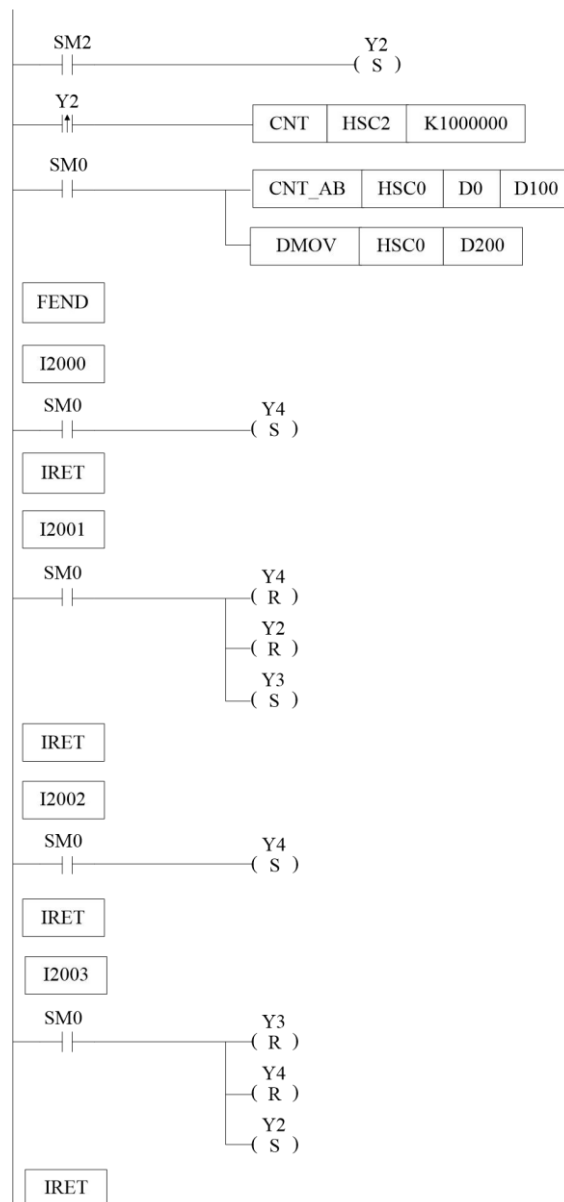
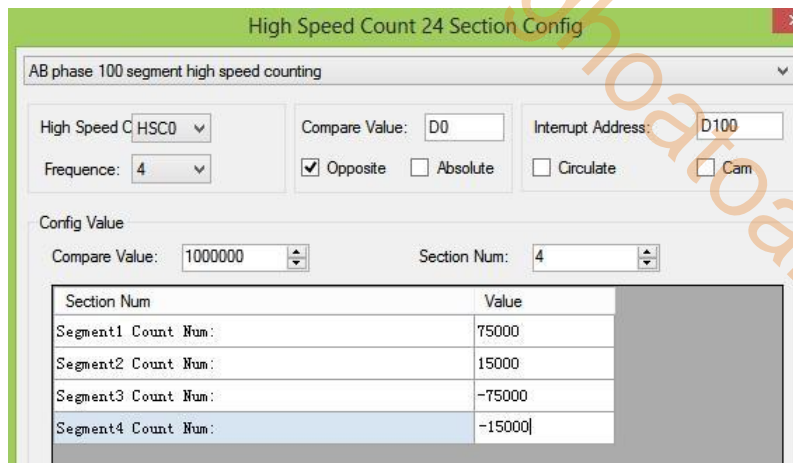
Y2 biểu thị tín hiệu đầu ra thuận;

Y3 biểu thị tín hiệu đầu ra ngược;

Y4 biểu thị tín hiệu đầu ra của tốc độ 1;

HSC2: Bộ đếm tích lũy số lần qua lại;

HSC0: HSC pha AB;



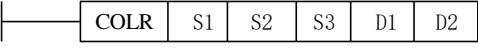
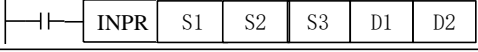
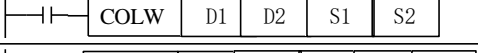
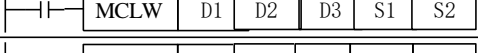
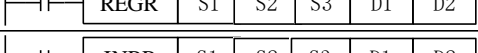
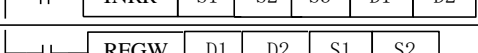

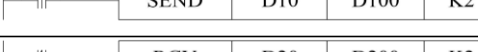

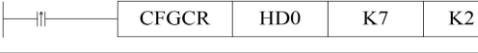

Danh sách lệnh:

LD SM2 // SM2 là cuộn dây ON mặc định
SET Y2 // đặt ON Y2 (chạy tiến)
LDP Y2 // Điều kiện kích hoạt số lần qua lại Y2
CNT HSC2 K1000000 //HSC2 bắt đầu đếm
LD SM0 //SM000 là cuộn dây thường ON
CNT_AB HSC0 D0 D100 //Địa chỉ đầu tiên của đoạn HSC 100
DMOV HSC0 D200 //đọc giá trị đếm HSC0 tới D200
FEND //chương trình chính kết thúc
I2000 //Cờ ngắt 1
LD SM0 //SM0 là cuộn dây thường ON
SET Y4 //đặt ON Y4 (chạy ở tốc độ 1)
IRET //kết thúc lệnh ngắt
I2001 //Cờ ngắt 2
LD SM0 //SM0 là cuộn dây thường ON
RST Y4 //đặt lại Y4 (dừng chạy ở tốc độ 1)
RST Y2 //đặt lại Y2 (dừng chạy tiến)
SET Y3 //đặt ON Y3 (chạy lùi)
IRET //trở về trạng thái trước khi ngắt
I2002 //cờ ngắt 3
LD SM0 //SM0 là cuộn dây thường ON
SET Y4 //đặt ON Y4 (chạy ở tốc độ 1)
IRET //kết thúc lệnh ngắt
I2003 //cờ kết thúc lệnh ngắt 4
LD SM0 //SM0 là cuộn dây thường ON
RST Y3 //đặt lại Y3 (dừng chạy lùi)
RST Y4 //đặt lại Y4 (dừng chạy ở tốc độ chậm)
SET Y2 //đặt ON Y2 (chạy tiến)
IRET //kết thúc lệnh ngắt

Chương 6 : Chức năng giao tiếp

Chương này chủ yếu bao gồm: khái niệm cơ bản về giao tiếp, giao tiếp Modbus và giao tiếp tự do.

Các lệnh liên quan

Ký hiệu	Chức năng	Mạch và phần tử lập trình	Chương
Giao tiếp MODBUS Communication			
COLR	Đọc cuộn dây		6-2-3
INPR	Đọc cuộn dây đầu vào		6-2-3
COLW	Ghi một cuộn dây		6-2-3
MCLW	Ghi nhiều cuộn dây		6-2-3
REGR	Đọc thanh ghi		6-2-3
INRR	Đọc thanh ghi đầu vào		6-2-3
REGW	Ghi một thanh ghi		6-2-3
MRGW	Ghi nhiều thanh ghi		6-2-3
Giao tiếp tự do			
SEND	Gửi dữ liệu		6-3-4
RCV	Nhận dữ liệu		6-3-4
Đọc và ghi dữ liệu công nối tiếp			
CFGCR	Đọc công nối tiếp		6-5-1
CFGCW	Ghi công nối tiếp		6-5-2

6-1 Tổng quan về truyền thông

Các thiết bị chính của PLC dòng XD, XL có thể đáp ứng yêu cầu của bạn về giao tiếp và mạng. Chúng không chỉ hỗ trợ Modbus RTU mà còn hỗ trợ Modbus ASCII và field bus X-NET. PLC dòng XD, XL cung cấp nhiều phương thức giao tiếp giúp bạn giao tiếp với các thiết bị (chẳng hạn như máy in và các thiết bị khác v.v.) có giao thức Modbus.

6-1-1 Cổng COM



PLC dòng XD, XL có nhiều cổng giao tiếp, như cổng USB, cổng Ethernet, cổng 0~cổng 5, cổng 2-RS232, cổng 2-RS485.

× không hỗ trợ √ hỗ trợ

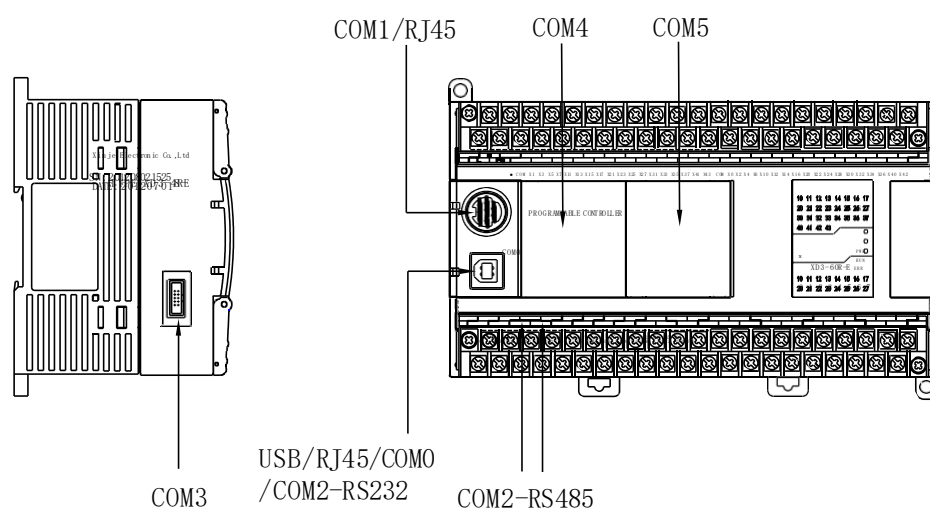
	USB	RJ45	COM0	COM1	COM2- RS232	COM2- RS485	COM3	COM4	COM5
XD1	×	×	√	√	×	√	×	×	×
XD2	×	×	√	√	×	√	√	√	√
XD3	√	×	×	√	×	√	√	√	√
XD5	√	×	×	√	×	√	√	√	√
XDM	√	×	×	√	×	√	√	√	√
XDC	×	×	×	√	√	√	√	√	√
XD5E	×	√	×	√	×	√	√	√	√
XDME	×	√	×	√	×	√	√	√	√
XDH	×	√	×	√	×	√	√	×	×
XL1	×※1	×	√	√	×	√	×	×	×
XL3	√	×	×	√	×	√	√	×	×
XL5	√	×	×	√	×	√	√	×	×
XL5E	×	√	×	√	×	√	√	×	×
XLME	×	√	×	√	×	√	√	×	×
XLH	×	√	×	√	×	√	√	×	×

Lưu ý:

※1: XL1-16T-U có cổng USB.

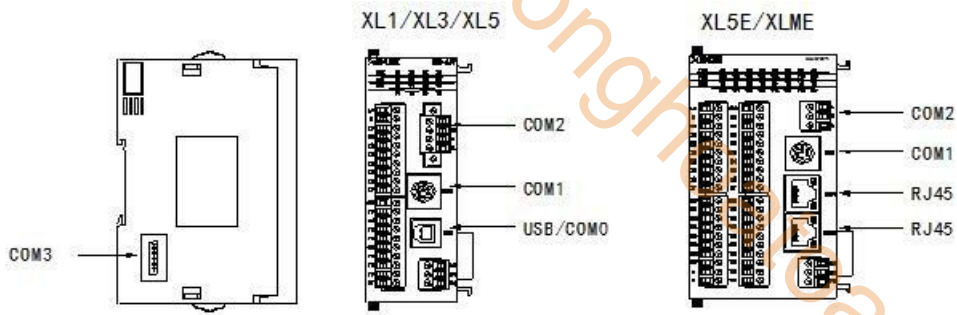
※2: Trong các dòng PLC được hỗ trợ "√", có thể có một số model không hỗ trợ COM2-COM5. Xem Phụ lục 5 để biết thêm chi tiết.

Các cổng giao tiếp dòng XD được phân bố như sau:



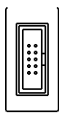
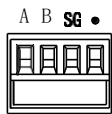
Lưu ý: Phía bên trái khối đầu ra của XD5E/XDME/XDH là cổng RS232.

Các cổng giao tiếp dòng XL được phân bố như sau:



Định nghĩa và chức năng của từng cổng giao tiếp như sau:

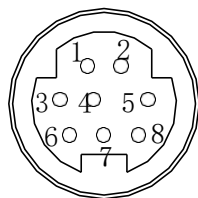
Cổng	Ngoại quan	Định nghĩa	Giao thức	Chức năng
COM0		Cổng RS232	X-NET Modbus	Tải xuống chương trình, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM1		Cổng RS232	Modbus RTU Modbus ASCII Giao tiếp tự do X-NET	Tải xuống chương trình và kết nối các thiết bị bên ngoài, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM2- RS232		Cổng RS232	Modbus RTU Modbus ASCII Giao tiếp tự do X-NET	Tải xuống chương trình và kết nối các thiết bị bên ngoài, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM2- RS485	Cổng A, B	Cổng RS485	Modbus RTU Modbus ASCII Giao tiếp tự do X-NET	Tải xuống chương trình và kết nối các thiết bị bên ngoài, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM2	 Cổng A, B	Cổng RS485		
USB		Cổng USB	X-NET	Cổng tải xuống tốc độ cao, vui lòng cài đặt trình điều khiển USB trước
RJ45		Cổng Ethernet	Giao tiếp TCP/IP dựa trên Ethernet	Chương trình và dữ liệu tải xuống/tải lên ổn định tốc độ cao, giám sát từ xa, giao tiếp với thiết bị TCP/IP trong mạng LAN, thiết lập các thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje. Chỉ có cổng LAN2 dòng XDH hỗ trợ EtherCAT, có thể điều khiển đồng bộ động cơ 32 trục.

COM3		Cổng ED mở rộng bên trái (để mở rộng cổng RS232/RS485)	Modbus RTU Modbus ASCII Giao tiếp tự do X-NET	Kết nối thiết bị bên ngoài, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM4		Cổng BD mở rộng phía trên/RS232/RS485/	Modbus RTU Modbus ASCII Giao tiếp tự do X-NET	Kết nối thiết bị bên ngoài, thiết lập thông số cổng thông qua phần mềm hoặc công cụ cấu hình Xinje
COM5		Cổng cáp quang (xem chi tiết bên dưới)		

Lưu ý:

- (1) Cổng COM0 là chế độ giao tiếp X-NET theo mặc định; COM1 của XDC là chế độ giao tiếp X-NET theo mặc định.
- (2) Không thể sử dụng đồng thời COM2-RS232 và COM2-RS485 của dòng XDC; khi cấu hình trong phần mềm lập trình thì số cổng là COM2.
- (3) Nếu COM1 không thể giao tiếp với PC sau khi thay đổi các thông số, vui lòng nhấp vào [stop PLC when reboot] (dừng PLC khi khởi động lại) trong phần mềm rồi bật lại nguồn để giải quyết vấn đề; nếu không cần thiết, tốt hơn hết là không sửa đổi các thông số giao tiếp COM1.
- (4) Cổng COM3 của PLC dòng XDH không hỗ trợ mô-đun ED mở rộng giao tiếp, cổng LAN1 hỗ trợ giao tiếp Ethernet, cổng LAN2 hỗ trợ chức năng bus EtherCAT.
- (5) Chức năng giao tiếp X-NET không nằm trong phạm vi của sách hướng dẫn này, vui lòng tham khảo sách hướng dẫn sử dụng X-NET.
- (6) Nội dung giao tiếp Ethernet không nằm trong phạm vi của sách hướng dẫn này, vui lòng tham khảo sách hướng dẫn sử dụng giao tiếp TCP IP dựa trên Ethernet.
- (7) bus Ethernet không nằm trong phạm vi của sách hướng dẫn này. Vui lòng tham khảo sách hướng dẫn sử dụng điều khiển chuyển động EtherCAT.

1. Cổng RS232 (COM0, COM1, COM2-RS232)

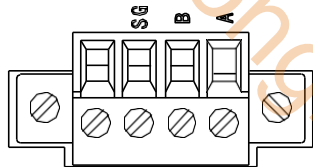


- 4: RxD
- 5: TxD
- 8: GND

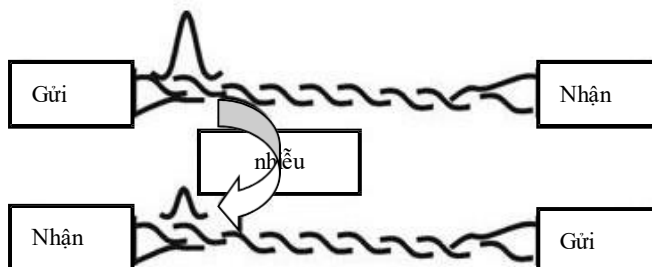
Phích cắm Mini Din 8 chân (lỗ)

2. Cổng RS485 (COM2, COM2-RS485)

Về cổng RS485, A là tín hiệu "+", B là tín hiệu "-". Cổng PLC RS485 dòng XL được đặt bên ngoài. Cầu đầu nối SG là hình thức tiếp địa signal ground. Sơ đồ các đầu nối được hiển thị như dưới đây:



Vui lòng sử dụng cáp xoắn đôi cho RS485. (Xem sơ đồ dưới đây). Tuy nhiên, cáp xoắn đôi có vỏ bọc sẽ tốt hơn và cáp một đầu sẽ kết nối với mặt đất.

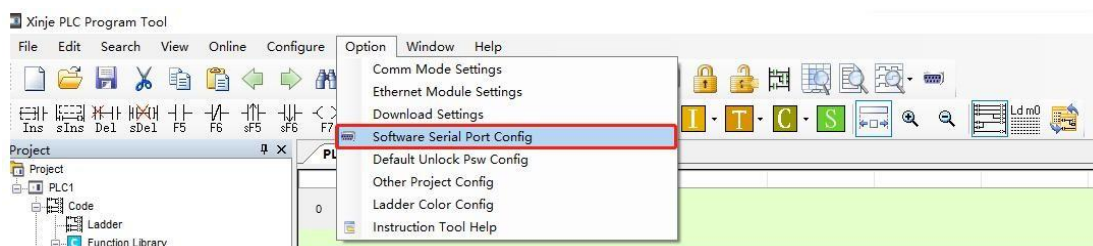


3. Cổng USB

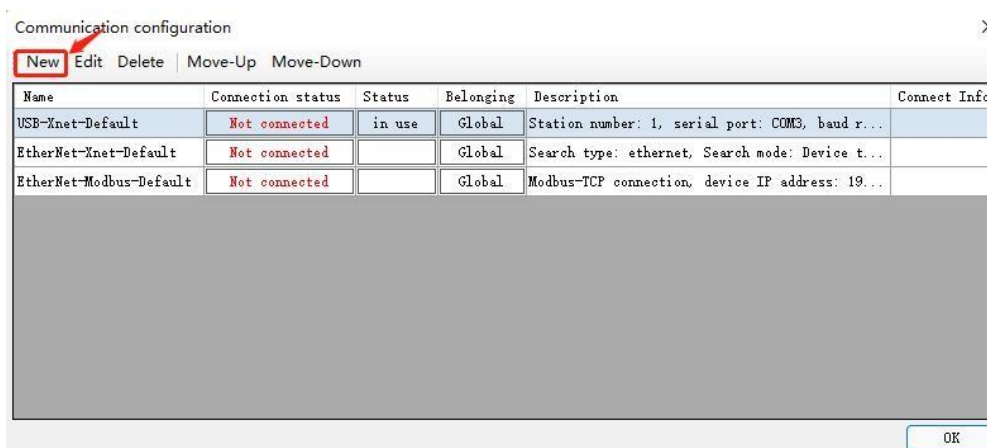
Khi tải chương trình và dữ liệu qua cổng USB, trước tiên phải cài đặt trình điều khiển USB và công cụ cấu hình XINJEConfig. Vì trình điều khiển USB hiện tại đã được tích hợp sẵn trong phần mềm XINJEConfig nên trình điều khiển USB sẽ được cài đặt tự động sau khi cài đặt phần mềm XINJEConfig.

Sau khi cài đặt công cụ XinjeConfig và USB driver, vui lòng chuyển sang chế độ Xnet trong phần mềm PLC :

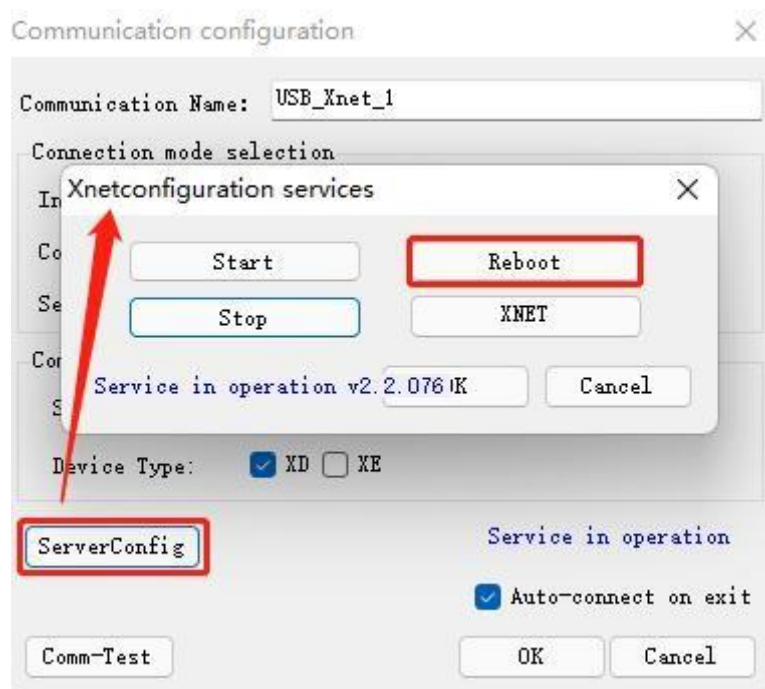
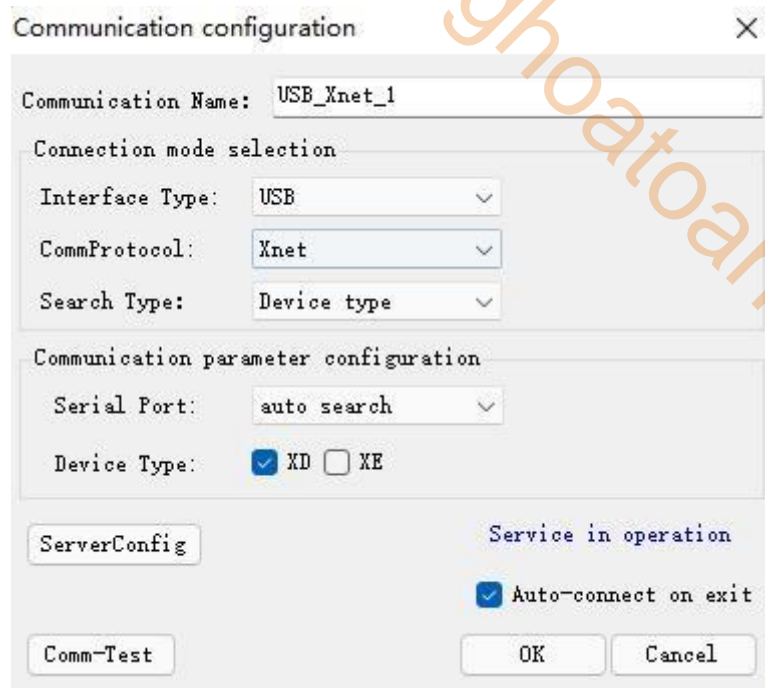
- (1) Mở phần mềm XDPPro, nhấp vào option/software serial port config (tùy chọn/cấu hình cổng nối tiếp phần mềm)



- (2) Cửa sổ "Communication Configuration" (Cấu hình giao tiếp) như hình dưới bật lên, nhấp vào 'New' và giao diện cấu hình sẽ xuất hiện như dưới đây:



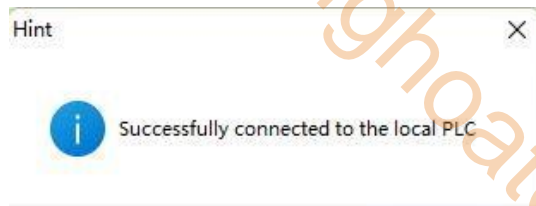
(3) Chọn USB làm giao diện giao tiếp cổng vuông, XNET làm giao thức giao tiếp và loại thiết bị làm phương thức tìm kiếm. Sau khi khởi động lại dịch vụ, nhấp vào OK.



(4) Sau khi trạng thái kết nối được thay đổi thành 'in use' (đang sử dụng), nhấp vào OK.

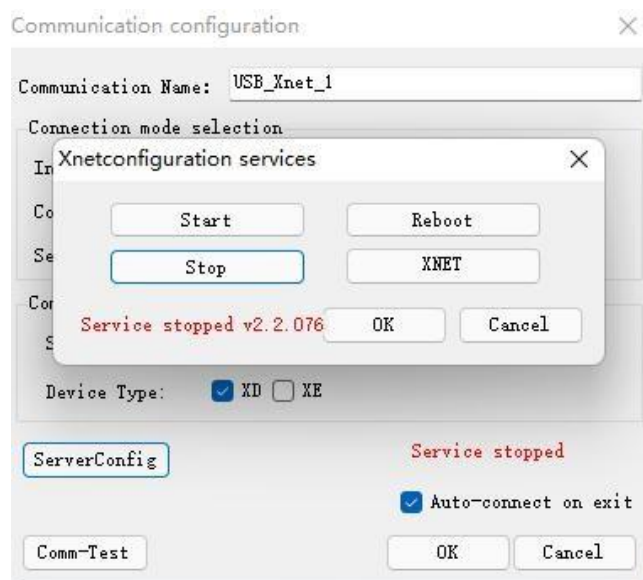
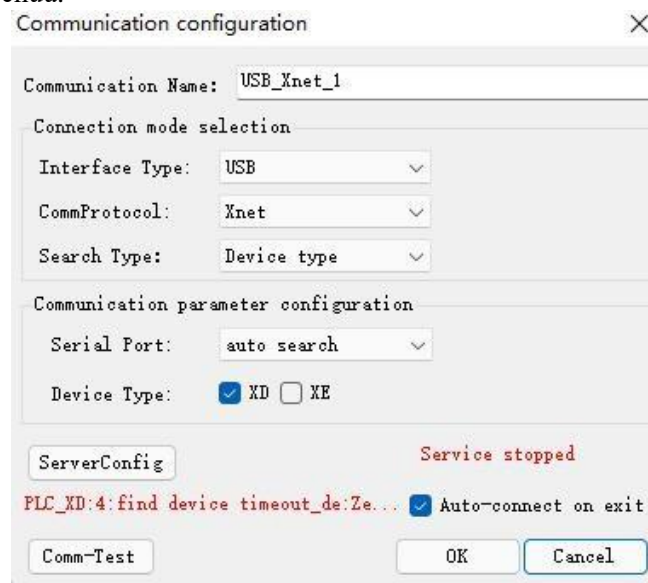
Connection status	Status	Belonging	Description
The connected	in use	Global	Station number: 1,
Not connected		Global	Search type: Autom

(5) Nếu hiển thị thông báo "Successfully connected to the local PLC" tức là đã kết nối thành công.



Lưu ý:

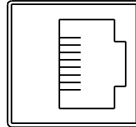
- (1) Nếu xuất hiện lỗi "find device timeout" (hết thời gian chờ tìm thiết bị), bạn có thể nhấp vào "Restart Service" để thử kết nối lại hoặc khởi động lại phần mềm lập trình và kết nối lại PLC. Nếu vẫn không kết nối được, bạn cần kiểm tra xem PLC đã bật nguồn chưa, cáp tải USB đã kết nối đúng cách chưa, driver USB và phần mềm XINJConfig đã được cài đặt đúng cách chưa.



4. Cổng Ethernet (RJ45)

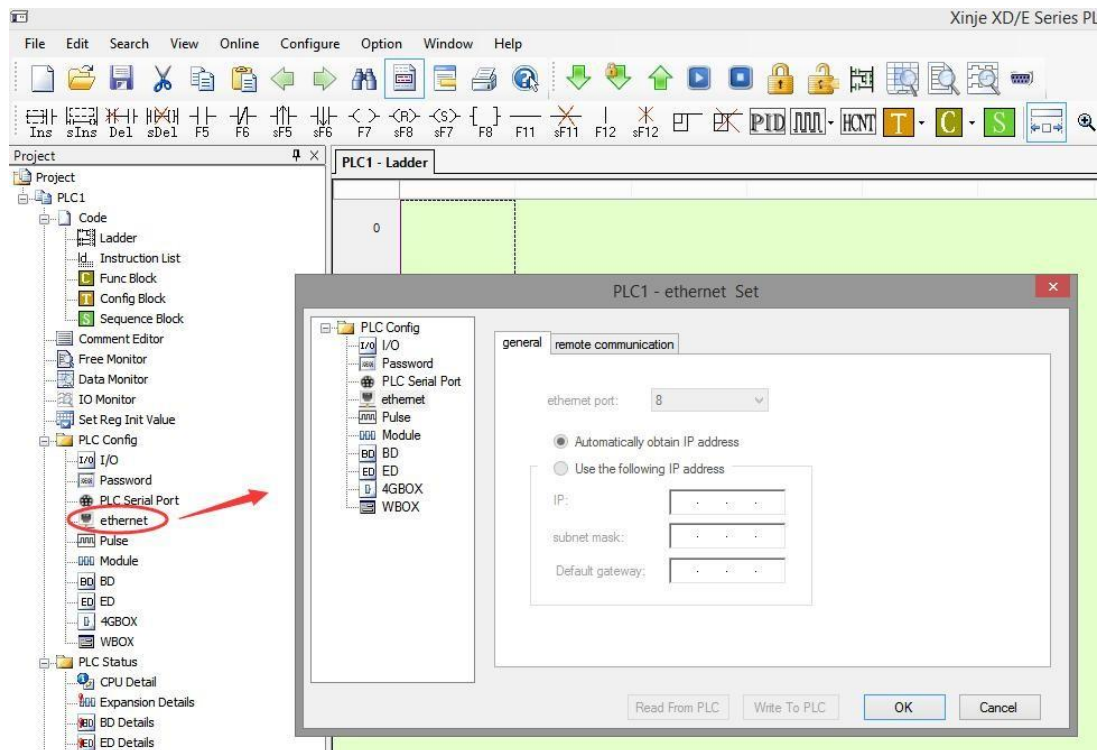
Cổng RJ45 dành riêng cho Ethernet PLC, hỗ trợ giao tiếp Ethernet TCP/IP, cổng nhanh hơn và ổn định hơn giao tiếp USB, khả năng giám sát dữ liệu theo thời gian thực tốt hơn, tải xuống và tải lên chương trình nhanh hơn. Bản thân chế độ kết nối của giao tiếp Ethernet có những ưu điểm rõ ràng so với RS485 và USB. Trong nhiều tình huống giao tiếp PLC, người dùng có thể giao tiếp với bất kỳ PLC nào tại chỗ chỉ thông qua một công tắc.

Ngoài ứng dụng trong mạng LAN, Ethernet còn hỗ trợ tìm kiếm, giám sát và vận hành PLC từ xa, chức năng tải xuống và giao tiếp với các thiết bị TCP IP khác trong mạng thông qua Internet.



Cổng RJ45 có thể được cấu hình trong "PLC Config-Ethernet" của phần mềm lập trình PLC XINJE hoặc thông qua công cụ XINJEConfig. Tham khảo sách hướng dẫn liên quan để biết thêm chi tiết.

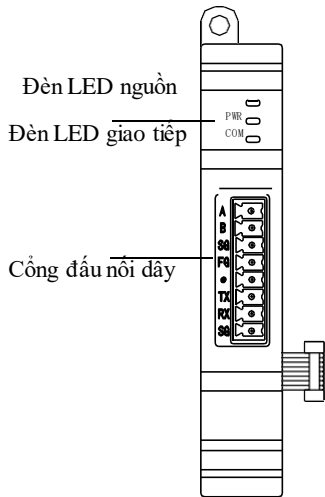
Cổng LAN2 của PLC dòng XDH hỗ trợ chức năng điều khiển bus EtherCAT. Số lượng trục lên tới 32 và chu kỳ điều khiển nhỏ hơn 1ms. Vui lòng tham khảo hướng dẫn sử dụng điều khiển chuyển động EtherCAT để biết cụ thể cách sử dụng chức năng.



5. Cổng ED mở rộng bên trái (COM3)

Cổng ED mở rộng bên trái có thể kết nối thẻ ED để mở rộng cổng RS232 và RS485. Các model ED bao gồm XD-NES-ED (có thể mở rộng một cổng RS232 và một cổng RS485, nhưng cả hai không thể giao tiếp cùng một lúc).

XD-NES-ED



Tên mỗi bộ phận được hiển thị như dưới đây:

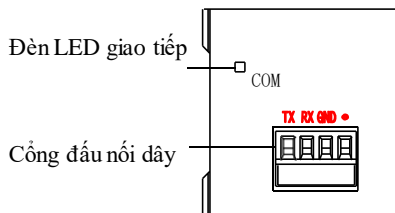
Tên	Chức năng	
Đèn LED nguồn	Đèn BẬT khi mô-đun ED bật nguồn	
Đèn LED giao tiếp	Đèn BẬT khi giao tiếp mô-đun ED bình thường	
Cổng đầu nối dây	A	RS485+
	B	RS485-
	SG	Mặt đất
	FG	Kết nối với cầu đầu dây nối đất
	-	Trống
	TX	Gửi RS232
	RX	Nhận RS232
	SG	Tiếp địa

6. Cổng BD mở rộng phía trên (COM4, COM5)

Cổng mở rộng ở trên có thể kết nối thẻ BD có chế độ RS232 (XD-NS-BD), chế độ RS485 (XD-NE-BD) và chế độ cáp quang (XD-NO-BD).

PLC dòng XD 24/32 I/O có thể mở rộng một thẻ BD, PLC dòng XD 48/60 I/O có thể mở rộng 2 thẻ BD, PLC dòng XD 16 I/O không thể mở rộng thẻ BD.

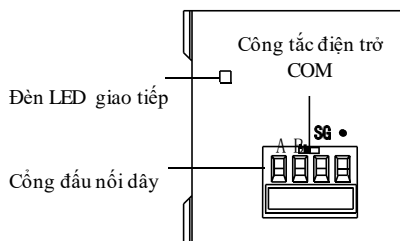
(1) XD-NS-BD



Tên mỗi bộ phận được hiển thị như dưới đây:

Tên	Chức năng	
Đèn LED giao tiếp	Không hỗ trợ chức năng này	
Cổng đầu nối dây	TX	Gửi tín hiệu
	RX	Nhận tín hiệu
	GND	Mặt đất
	•	Trống

(2) XD-NE-BD

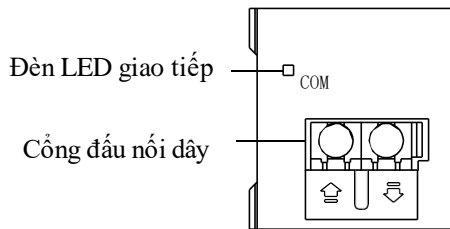


Tên mỗi bộ phận được hiển thị như dưới đây:

Tên	Chức năng	
Đèn LED giao tiếp	The light is flashing when the BD card communication is successful	
Cổng đầu nối dây	A	485+
	B	485-
	S	Tiếp địa tín hiệu
	•	Trống
Công tắc điện trở	Chọn có sử dụng điện trở hay không (120Ω)	

XD-NE-BD có công tắc để chọn xem có lắp điện trở hay không. Cài đặt mặc định của công tắc là TẮT, có nghĩa là không lắp điện trở. Nếu XD-NE-BD ở đầu hoặc cuối bus thì cần lắp điện trở 120Ω ở cả hai bên và bật công tắc (phải).

(3) XD-NO-BD



Tên mỗi bộ phận được hiển thị như dưới đây:

Tên	Chức năng
Đèn LED giao tiếp	Không hỗ trợ chức năng này
Cổng đấu nối dây	Bên trái là ngõ vào tín hiệu, bên phải là ngõ ra tín hiệu

6-1-2 Thông số giao tiếp

Thông số truyền thông

Station	Số trạm Modbus: 1~254
Baud Rate	300bps~9Mbps
Data Bit	5, 6, 7, 8, 9
Stop Bit	1, 1.5, 2
Parity	Chẵn, Lẻ, chẵn, trống, mặt nạ

Các thông số mặc định: Số trạm là 1, tốc độ baud là 19200bps, 8 bit dữ liệu, 1 bit stop, chẵn.

Có nhiều cách cài đặt thông số cổng giao tiếp PLC:

Có hai cách thiết lập thông số giao tiếp Modbus: (1) thiết lập thông số bằng phần mềm lập trình; (2) cài đặt thông số bằng công cụ XINJEConfig, tham khảo chương 6-2-6 để biết thêm chi tiết.

Các thông số giao tiếp format tự do có thể được thiết lập bằng phần mềm lập trình, tham khảo chương 6-3-2 để biết thêm chi tiết.

Các thông số giao tiếp X-NET có thể được thiết lập bằng công cụ Xinje Config. Tham khảo hướng dẫn sử dụng fieldbus X-NET để biết thêm chi tiết.

Lưu ý: Đối với cầu đấu dây A, B trên thân PLC, tốc độ truyền 1Mbps trở lên chỉ phù hợp với chế độ giao tiếp X-NET.

6-2 Giao tiếp MODBUS

6-2-1 Tổng quan về Modbus

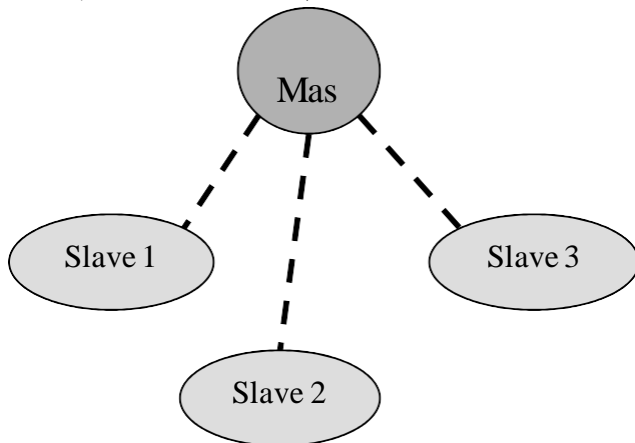
PLC dòng XD, XL hỗ trợ cả Modbus master và Modbus Slave.

Chế độ master: Khi PLC được cài đặt thành master (chủ), nó có thể giao tiếp với các thiết bị slave (tớ) khác có giao thức MODBUS-RTU hoặc MODBUS-ASCII thông qua các lệnh Modbus; nó cũng có thể thay đổi dữ liệu với các thiết bị khác.

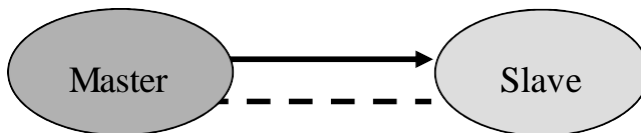
Ví dụ: PLC dòng XD3 của Xinje có thể điều khiển biến tần bằng Modbus.

Chế độ slave: Khi PLC được cài đặt thành slave, nó chỉ có thể phản hồi với các thiết bị master khác.

Master và Slave: Trong mạng RS485, có thể có một master và nhiều slave cùng một lúc (xem sơ đồ bên dưới). Trạm master có thể đọc và ghi bất kỳ trạm slave nào. Hai trạm slave không thể giao tiếp với nhau. Trạm master viết chương trình và đọc hoặc ghi một trạm slave; trạm slave không có chương trình mà chỉ phản hồi trạm master. (Đầu dây: kết nối tất cả 485+, kết nối tất cả 485-)



Trong mạng RS232 (xem sơ đồ bên dưới), cùng một lúc chỉ có thể có một master và một slave.



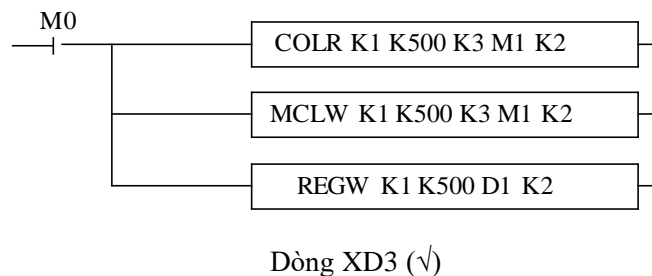
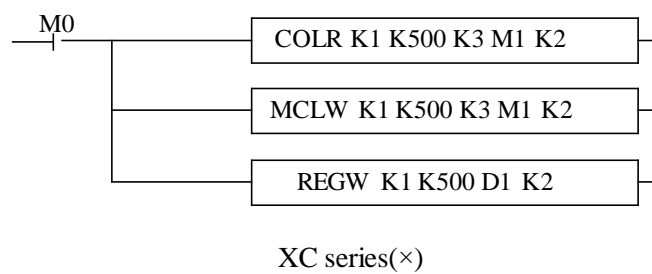
Có đường nét đứt trong sơ đồ. Điều đó có nghĩa là bất kỳ PLC nào cũng có thể là trạm master khi tất cả PLC trong mạng không gửi dữ liệu. Vì PLC không có chuẩn xung clock thống nhất nên việc giao tiếp sẽ không thành công khi có nhiều PLC gửi dữ liệu cùng một lúc. Trường hợp này không khuyến khích sử dụng.

Lưu ý:

1. Đối với PLC dòng XD/XL, RS232 và RS485 chỉ hỗ trợ bán song công.
2. Đối với PLC dòng XC, nếu PLC master gửi một dữ liệu đến PLC slave và PLC master gửi lại dữ liệu trước khi PLC slave nhận đầy đủ dữ liệu cuối cùng thì có thể xảy ra lỗi dữ liệu cuối ở PLC slave; Đối với PLC dòng XD/XL, chúng tôi giải quyết vấn đề này bằng cách thêm thời gian chờ trước khi giao tiếp, có nghĩa là PLC slave sẽ chỉ nhận dữ liệu tiếp theo sau một thời gian dữ liệu cuối cùng kết thúc.

6-2-2 Thay đổi lệnh Modbus

Chế độ xử lý lệnh Modbus đã thay đổi trong PLC dòng XD/XL, người dùng có thể viết lệnh Modbus trực tiếp trong chương trình, trạm giao thức sẽ xếp hàng các yêu cầu Modbus, nhiệm vụ không giống với việc giao tiếp; Điều đó có nghĩa là người dùng có thể sử dụng một điều kiện kích hoạt để kích hoạt nhiều lệnh Modbus cùng một lúc. PLC sẽ xếp hàng các yêu cầu Modbus theo trạm giao thức, điều này sẽ dẫn đến lỗi giao tiếp trong PLC dòng XC.



Lưu ý: Sequence block (Khối tuần tự) PLC dòng XD/XL đã hủy các lệnh giao tiếp Modbus, được thay thế bằng chế độ xử lý lệnh Modbus hiện tại.

6-2-3 Địa chỉ giao tiếp Modbus

Mã phân tử lập trình trong PLC tương ứng với số ID Modbus, xem bảng sau:

1) Địa chỉ Modbus PLC dòng XD1, XD2, XD3, XL1, XL3 và bảng phân tử lập trình bên trong:

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)
Coil bit (Bit cuộn dây)	M	M0~M7999	8000	0~1F3F	0~7999
	X	X0~X77 (thiết bị chính)	64	5000~503F	20480~20543
		X10000~X10077 (Mô đun #1)	64	5100~513F	20736~20799
		X10100~X10177 (Mô đun #2)	64	5140~517F	20800~20863
		X10200~X10277 (Mô đun #3)	64	5180~51BF	20864~20927
		X10300~X10377 (Mô đun #4)	64	51C0~51FF	20928~20991
		X10400~X10477 (Mô đun #5)	64	5200~523F	20992~21055

		X10500~X10577 (Mô đùn #6)	64	5240~527F	21056~21119	
		X10600~X10677 (Mô đùn #7)	64	5280~52BF	21120~21183	
		X10700~X10777 (Mô đùn #8)	64	52C0~52FF	21184~21247	
		X11000~X11077 (Mô đùn #9)	64	5300~533F	21248~21311	
		X11100~X11177 (Mô đùn #10)	64	5340~537F	21312~21375	
		X20000~X20077(#1 BD)	64	58D0~590F	22736~22799	
	Y	Y0~77 (thiết bị chính)	64	6000~603F	24576~24639	
		Y10000~Y10077 (Mô đùn #1)	64	6100~613F	24832~24895	
		Y10100~Y10177 (Mô đùn #2)	64	6140~617F	24896~24959	
		Y10200~Y10277 (Mô đùn #3)	64	6180~61BF	24960~25023	
		Y10300~Y10377 (Mô đùn #4)	64	61C0~61FF	25024~25087	
		Y10400~Y10477 (Mô đùn #5)	64	6200~623F	25088~25151	
		Y10500~Y10577 (Mô đùn #6)	64	6240~627F	25152~25215	
		Y10600~Y10677 (Mô đùn #7)	64	6280~62BF	25216~25279	
		Y10700~Y10777 (Mô đùn #8)	64	62C0~62FF	25280~25343	
		Y11000~Y11077 (Mô đùn #9)	64	6300~633F	25344~25407	
		Y11100~Y11177 (Mô đùn #10)	64	6340~637F	25408~25471	
		Y20000~Y20077(#1 BD)	64	68D0~690F	26832~26895	
		S	S0~S1023	1024	7000~73FF	28672~29695
		SM	SM0~SM2047	2048	9000~97FF	36864~38911
	T	T0~T575	576	A000~A23F	40960~41535	
	C	C0~C575	576	B000~B23F	45056~45631	
	ET	ET0~ET31	32	C000~C01F	49152~49183	
	SEM	SEM0~SEM31	32	C080~C09F	49280~49311	
	HM※1	HM0~HM959	960	C100~C4BF	49408~50367	
	HS※1	HS0~HS127	128	D900~D97F	55552~55679	
	HT※1	HT0~HT95	96	E100~E15F	57600~57695	
	HC※1	HC0~HC95	96	E500~E55F	58624~58719	
	HSC※1	HSC0~HSC31	32	E900~E91F	59648~59679	
Register word (Từ thanh ghi)	D	D0~D7999	8000	0~1F3F	0~7999	
	ID	ID0~ID99 (thiết bị chính)	100	5000~5063	20480~20579	
		ID10000~ID10099 (Mô đùn #1)	100	5100~5163	20736~20835	
		ID10100~ID10199 (Mô đùn #2)	100	5164~51C7	20836~20935	

	ID10200~ID10299 (Mô đun #3)	100	51C8~522B	20936~21035
	ID10300~ID10399 (Mô đun #4)	100	522C~528F	21036~21135
	ID10400~ID10499 (Mô đun #5)	100	5290~52F3	21136~21235
	ID10500~ID10599 (Mô đun #6)	100	52F4~5357	21236~21335
	ID10600~ID10699 (Mô đun #7)	100	5358~53BB	21336~21435
	ID10700~ID10799 (Mô đun #8)	100	53BC~541F	21436~21535
	ID10800~ID10899 (Mô đun #9)	100	5420~5483	21536~21635
	ID10900~ID10999 (Mô đun #10)	100	5484~54E7	21636~21735
	ID20000~ID20099 (#1 BD)	100	58D0~5933	22736~22835
QD	QD0~QD99 (thiết bị chính)	100	6000~6063	24576~24675
	QD10000~QD10099 (Mô đun #1)	100	6100~6163	24832~24931
	QD10100~QD10199 (Mô đun #2)	100	6164~61C7	24932~25031
	QD10200~QD10299 (Mô đun #3)	100	61C8~622B	25032~25131
	QD10300~QD10399 (Mô đun #4)	100	622C~628F	25132~25231
	QD10400~QD10499 (Mô đun #5)	100	6290~62F3	25232~25331
	QD10500~QD10599 (Mô đun #6)	100	62F4~6357	25332~25431
	QD10600~QD10699 (Mô đun #7)	100	6358~63BB	25432~25531
	QD10700~QD10799 (Mô đun #8)	100	63BC~641F	25532~25631
	QD10800~QD10899 (Mô đun #9)	100	6420~6483	25632~25731
	QD10900~QD10999 (Mô đun #10)	100	6484~64E7	25732~25831
	QD20000~QD20099 (#1 BD)	100	68D0~6933	26832~26931
	SD	SD0~SD2047	2048	7000~77FF
TD	TD0~TD575	576	8000~823F	32768~33343
CD	CD0~CD575	576	9000~923F	36864~37439
ETD	ETD0~ETD31	32	A000~A01F	40960~40991
HD* ¹	HD0~HD999	1000	A080~A467	41088~42087
HSD* ¹	HSD0~HSD499	500	B880~BA73	47232~47731
HTD* ¹	HTD0~HTD95	96	BC80~BCDF	48256~48351
HCD* ¹	HCD0~HCD95	96	C080~C0DF	49280~49375
HSCD* ¹	HSCD0~HSCD31	32	C480~C49F	50304~50335
FD* ²	FD0~FD5119	5120	C4C0~D8BF	50368~55487
SFD* ²	SFD0~SFD1999	2000	E4C0~EC8F	58560~60559

	FS※2	FS0~FS47	48	F4C0~F4EF	62656~62703
--	------	----------	----	-----------	-------------

2) Địa chỉ Modbus PLC dòng XD5, XDM, XDC, XD5E, XDME, XL5, XL5E, XLME và bảng phần tử lập trình bên trong:

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (thập lục phân)	Địa chỉ Modbus (Thập phân)	
Coil bit (Bit cuộn dây)	M	M0~M20479	20480	0~4FFF	0~20479	
	X	X0~X77 (thiết bị chính)	64	5000~503F	20480~20543	
		X10000~X10077 (Mô đun #1)	64	5100~513F	20736~20799	
		X10100~X10177 (Mô đun #2)	64	5140~517F	20800~20863	
		X10200~X10277 (Mô đun #3)	64	5180~51BF	20864~20927	
		X10300~X10377 (Mô đun #4)	64	51C0~51FF	20928~20991	
		X10400~X10477 (Mô đun #5)	64	5200~523F	20992~21055	
		X10500~X10577 (Mô đun #6)	64	5240~527F	21056~21119	
		X10600~X10677 (Mô đun #7)	64	5280~52BF	21120~21183	
		X10700~X10777 (Mô đun #8)	64	52C0~52FF	21184~21247	
		X11000~X11077 (Mô đun #9)	64	5300~533F	21248~21311	
		X11100~X11177 (Mô đun #10)	64	5340~537F	21312~21375	
		X11200~X11277 (Mô đun #11)	64	5380~53BF	21376~21439	
		X11300~X11377 (Mô đun #12)	64	53C0~53FF	21440~21503	
		X11400~X11477 (Mô đun #13)	64	5400~543F	21504~21567	
		X11500~X11577 (Mô đun #14)	64	5440~547F	21568~21631	
		X11600~X11677 (Mô đun #15)	64	5480~54BF	21632~21695	
		X11700~X11777 (Mô đun #16)	64	54C0~54FF	21696~21759	
		X20000~X20077 (#1 BD)	64	58D0~590F	22736~22799	
		Y	Y0~77 (thiết bị chính)	64	6000~603F	24576~24639
			Y10000~Y10077 (Mô đun #1)	640	6100~613F	24832~24895
	Y10100~Y10177 (Mô đun #2)		64	6140~617F	24896~24959	
	Y10200~Y10277 (Mô đun #3)		64	6180~61BF	24960~25023	
	Y10300~Y10377 (Mô đun #4)		64	61C0~61FF	25024~25087	

		Y10400~Y10477 (Mô đun #5)	64	6200~623F	25088~25151
		Y10500~Y10577 (Mô đun #6)	64	6240~627F	25152~25215
		Y10600~Y10677 (Mô đun #7)	64	6280~62BF	25216~25279
		Y10700~Y10777 (Mô đun #8)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (Mô đun #9)	64	6300~633F	25344~25407
		Y11100~Y11177 (Mô đun #10)	64	6340~637F	25408~25471
		Y11200~Y11277 (Mô đun #11)	64	6380~63BF	25472~25535
		Y11300~Y11377 (Mô đun #12)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (Mô đun #13)	64	6400~643F	25600~25663
		Y11500~Y11577 (Mô đun #14)	64	6440~647F	25664~25727
		Y11600~Y11677 (Mô đun #15)	64	6480~64BF	25728~25791
		Y11700~Y11777 (Mô đun #16)	64	64C0~64FF	25792~25855
		Y20000~Y20077(#1 BD)	64	68D0~690F	26832~26895
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~45151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407
	HM※ ¹	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS※ ¹	HS0~HS999	1000	D900~DCEF	55552~56551
	HT※ ¹	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC※ ¹	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC※ ¹	HSC0~HSC36	40	E900~E927	59648~59687
Register word (Từ thanh ghi)	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99 (thiết bị chính)	100	5000~5063	20480~20579
		ID10000~ID10099 (Mô đun #1)	100	5100~5163	20736~20835
		ID10100~ID10199 (Mô đun #2)	100	5164~51C7	20836~20935
		ID10200~ID10299 (Mô đun #3)	100	51C8~522B	20936~21035
		ID10300~ID10399 (Mô đun #4)	100	522C~528F	21036~21135
		ID10400~ID10499 (Mô đun #5)	100	5290~52F3	21136~21235
		ID10500~ID10599 (Mô đun #6)	100	52F4~5357	21236~21335
		ID10600~ID10699	100	5358~53BB	21336~21435

	(Mô đun #7)			
	ID10700~ID10799 (Mô đun #8)	100	53BC~541F	21436~21535
	ID10800~ID10899 (Mô đun #9)	100	5420~5483	21536~21635
	ID10900~ID10999 (Mô đun #10)	100	5484~54E7	21636~21735
	ID11000~ID11099 (Mô đun #11)	100	54E8~554B	21736~21835
	ID11100~ID11199 (Mô đun #12)	100	554C~55AF	21836~21935
	ID11200~ID11299 (Mô đun #13)	100	55B0~5613	21936~22035
	ID11300~ID11399 (Mô đun #14)	100	5614~5677	22036~22135
	ID11400~ID11499 (Mô đun #15)	100	5678~56DB	22136~22235
	ID11500~ID11599 (Mô đun #16)	100	56DC~573F	22236~22335
	ID20000~ID20099(#1 BD)	100	58D0~5933	22736~22835
QD	QD0~QD99 (Thiết bị chính)	100	6000~6063	24576~24675
	QD10000~QD10099 (Mô đun #1)	100	6100~6163	24832~24931
	QD10100~QD10199 (Mô đun #2)	100	6164~61C7	24932~25031
	QD10200~QD10299 (Mô đun #3)	100	61C8~622B	25032~25131
	QD10300~QD10399 (Mô đun #4)	100	622C~628F	25132~25231
	QD10400~QD10499 (Mô đun #5)	100	6290~62F3	25232~25331
	QD10500~QD10599 (Mô đun #6)	100	62F4~6357	25332~25431
	QD10600~QD10699 (Mô đun #7)	100	6358~63BB	25432~25531
	QD10700~QD10799 (Mô đun #8)	100	63BC~641F	25532~25631
	QD10800~QD10899 (Mô đun #9)	100	6420~6483	25632~25731
	QD10900~QD10999 (Mô đun #10)	100	6484~64E7	25732~25831
	QD11000~QD11099 (Mô đun #11)	100	64E8~654B	25832~25931
	QD11100~QD11199 (Mô đun #12)	100	654C~65AF	25932~26031
	QD11200~QD11299 (Mô đun #13)	100	65B0~6613	26032~26131
	QD11300~QD11399 (Mô đun #14)	100	6614~6677	26132~26231
	QD11400~QD11499 (Mô đun #15)	100	6678~66DB	26232~26331
	QD11500~QD11599	100	66DC~673F	26332~26431

		(#16 module)			
		QD20000~QD20099(#1 BD)	100	68D0~6933	26832~26931
SD		SD0~SD4095	4096	7000~7FFF	28672~32767
TD		TD0~TD4095	4096	8000~8FFF	32768~36863
CD		CD0~CD4095	4096	9000~9FFF	36864~40959
ETD		ETD0~ETD39	40	A000~A027	40960~40999
HD* ¹		HD0~HD6143	6144	A080~B87F	41088~47231
HSD* ¹		HSD0~HSD1023	1024	B880~BC7F	47232~48255
HTD* ¹		HTD0~HTD1023	1024	BC80~C07F	48256~49279
HCD* ¹		HCD0~HCD1023	1024	C080~C47F	49280~40303
HSCD* ¹		HSCD0~HSCD39	40	C480~C4A7	50304~50343
FD* ²		FD0~FD8191	8192	C4C0~E4BF	50368~58559
SFD* ²		SFD0~SFD5999	6000	E4C0~FC2F	58560~64559
FS* ²		FS0~FS47	48	F4C0~F4EF	62656~62703

3) Địa chỉ Modbus PLC dòng XDH và XLH và bảng phần tử lập trình bên trong:

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)
	M	M0~M20479	20480	0~4FFF	0~20479
Coil bit (Bit cuộn dây)	X	X0~X77 (thiết bị chính)	64	5000~503F	20480~20543
		X10000~X10077(Mô đun #1)	64	5100~513F	20736~20799
		X10100~X10177(Mô đun #2)	64	5140~517F	20800~20863
		X10200~X10277(Mô đun #3)	64	5180~51BF	20864~20927
		X10300~X10377(Mô đun #4)	64	51C0~51F F	20928~20991
		X10400~X10477(Mô đun #5)	64	5200~523F	20992~21055
		X10500~X10577(Mô đun #6)	64	5240~527F	21056~21119
		X10600~X10677(Mô đun #7)	64	5280~52BF	21120~21183
		X10700~X10777(Mô đun #8)	64	52C0~52F F	21184~21247
		X11000~X11077(Mô đun #9)	64	5300~533F	21248~21311
		X11100~X11177(Mô đun #10)	64	5340~537F	21312~21375
		X11200~X11277(Mô đun #11)	64	5380~53BF	21376~21439
		X11300~X11377(Mô đun #12)	64	53C0~53F F	21440~21503
		X11400~X11477(Mô đun #13)	64	5400~543F	21504~21567
		X11500~X11577(Mô đun #14)	64	5440~547F	21568~21631
X11600~X11677(Mô đun #15)	64	5480~54BF	21632~21695		

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)	
		X11700~X11777(Mô đun #16)	64	54C0~54FF	21696~21759	
		X20000~X20077(#1 BD)	64	58D0~590F	22736~22799	
		X20100~X20177(#2 BD)	64	5910~594F	22800~22863	
		X30000~X30077(#1 ED)	64	5BF0~5C2F	23536~23599	
	Y	Y0~77 (thiết bị chính)	64	6000~603F	24576~24639	
		Y10000~Y10077(Mô đun #1)	64	6100~613F	24832~24895	
		Y10100~Y10177(Mô đun #2)	64	6140~617F	24896~24959	
		Y10200~Y10277(Mô đun #3)	64	6180~61BF	24960~25023	
		Y10300~Y10377(Mô đun #4)	64	61C0~61FF	25024~25087	
		Y10400~Y10477(Mô đun #5)	64	6200~623F	25088~25151	
		Y10500~Y10577(Mô đun #6)	64	6240~627F	25152~25215	
		Y10600~Y10677(Mô đun #7)	64	6280~62BF	25216~25279	
		Y10700~Y10777(Mô đun #8)	64	62C0~62FF	25280~25343	
		Y11000~Y11077(Mô đun #9)	64	6300~633F	25344~25407	
		Y11100~Y11177(Mô đun #10)	64	6340~637F	25408~25471	
		Y11200~Y11277(Mô đun #11)	64	6380~63BF	25472~25535	
	Y	Y11300~Y11377(Mô đun #12)	64	63C0~63FF	25536~25599	
		Y11400~Y11477(Mô đun #13)	64	6400~643F	25600~25663	
		Y11500~Y11577(Mô đun #14)	64	6440~647F	25664~25727	
		Y11600~Y11677(Mô đun #15)	64	6480~64BF	25728~25791	
		Y11700~Y11777(Mô đun #16)	64	64C0~64FF	25792~25855	
		Y20000~Y20077(#1 BD)	64	68D0~690F	26832~26895	
		Y20100~Y20177(#2 BD)	64	6910~694F	26896~26959	
		Y30000~Y30077(#1 ED)	64	6BF0~6C2F	27632~27695	
		S	S0~S7999	8000	7000~8F3F	28672~36671
		SM	SM0~SM4095	4096	9000~9FFF	36864~40959
T	T0~T4095	4096	A000~AFF	40960~45055		

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)
				F	
	C	C0~C4095	4096	B000~BFF F	45056~49151
	ET	ET0~ET39	40	C000~C02 7	49152~49191
	SEM	SEM0~SEM127	128	C080~C0F F	49280~49407
	HM※1	HM0~HM6143	6144	C100~D8F F	49408~55551
	HS※1	HS0~HS999	1000	D900~DCE F	55552~56551
	HT※1	HT0~HT1023	1024	E100~E4F F	57600~58623
	HC※1	HC0~HC1023	1024	E500~E8F F	58624~59647
	HSC※1	HSC0~HSC39	40	E900~E927	59648~59687
	D	D0~D20479	20480	0~4FFF	0~20479
Register word (Từ thanh ghi)	ID	ID0~ID99(本体)	100	5000~5063	20480~20579
		ID10000~ID10099(Mô đũn #1)	100	5100~5163	20736~20835
		ID10100~ID10199(Mô đũn #2)	100	5164~51C7	20836~20935
		ID10200~ID10299(Mô đũn #3)	100	51C8~522 B	20936~21035
		ID10300~ID10399(Mô đũn #4)	100	522C~528F	21036~21135
		ID10400~ID10499(Mô đũn #5)	100	5290~52F3	21136~21235
		ID10500~ID10599(Mô đũn #6)	100	52F4~5357	21236~21335
		ID10600~ID10699(Mô đũn #7)	100	5358~53B B	21336~21435
		ID10700~ID10799(Mô đũn #8)	100	53BC~541 F	21436~21535
		ID10800~ID10899(Mô đũn #9)	100	5420~5483	21536~21635
		ID10900~ID10999(Mô đũn #10)	100	5484~54E7	21636~21735
		ID11000~ID11099(Mô đũn #11)	100	54E8~554 B	21736~21835
		ID11100~ID11199(Mô đũn #12)	100	554C~55A F	21836~21935
		ID11200~ID11299(Mô đũn #13)	100	55B0~5613	21936~22035
		ID11300~ID11399(Mô đũn #14)	100	5614~5677	22036~22135
		ID11400~ID11499(Mô đũn #15)	100	5678~56D B	22136~22235
ID11500~ID11599(Mô đũn #16)	100	56DC~573 F	22236~22335		

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)	
		ID20000~ID20099(#1 BD)	100	58D0~5933	22736~22835	
		ID20100~ID20199(#2 BD)	100	5934~5997	22836~22935	
		ID30000~ID30099(#1 ED)	100	5BF0~5C53	23536~23635	
	QD		QD0~QD99 (thiết bị chính)	100	6000~6063	24576~24675
			QD10000~QD10099(Mô đun #1)	100	6100~6163	24832~24931
			QD10100~QD10199(Mô đun #2)	100	6164~61C7	24932~25031
			QD10200~QD10299(Mô đun #3)	100	61C8~622B	25032~25131
			QD10300~QD10399(Mô đun #4)	100	622C~628F	25132~25231
			QD10400~QD10499(Mô đun #5)	100	6290~62F3	25232~25331
			QD10500~QD10599(Mô đun #6)	100	62F4~6357	25332~25431
			QD10600~QD10699(Mô đun #7)	100	6358~63BB	25432~25531
			QD10700~QD10799(Mô đun #8)	100	63BC~641F	25532~25631
			QD		QD10800~QD10899(Mô đun #9)	100
	QD10900~QD10999(Mô đun #10)	100			6484~64E7	25732~25831
	QD11000~QD11099(Mô đun #11)	100			64E8~654B	25832~25931
	QD11100~QD11199(Mô đun #12)	100			654C~65AF	25932~26031
	QD11200~QD11299(Mô đun #13)	100			65B0~6613	26032~26131
	QD11300~QD11399(Mô đun #14)	100			6614~6677	26132~26231
	QD11400~QD11499(Mô đun #15)	100			6678~66DB	26232~26331
	QD11500~QD11599(Mô đun #16)	100			66DC~673F	26332~26431
	QD20000~QD20099(#1 BD)	100			68D0~6933	26832~26931
	QD20100~QD20199(#2 BD)	100			6934~6997	26932~27031
			QD30000~QD30099(#1 ED)	100	6BF0~6C53	27632~27731
	SD	SD0~SD4095	4096	7000~7FFF	28672~32767	
	TD	TD0~TD4095	4096	8000~8FFF	32768~36863	
	CD	CD0~CD4095	4096	9000~9FFF	36864~40959	
	ETD	ETD0~ETD39	40	A000~A027	40960~40999	

Loại	Thành phần	Địa chỉ	Số	Địa chỉ Modbus (Thập lục phân)	Địa chỉ Modbus (Thập phân)
	HD*1	HD0~HD6143	6144	A080~B87 F	41088~47231
	HSD*1	HSD0~HSD1023	1024	B880~BC7 F	47232~48255
	HTD*1	HTD0~HTD1023	1024	BC80~C07 F	48256~49279
	HCD*1	HCD0~HCD1023	1024	C080~C47 F	49280~50303
	HSCD*1	HSCD0~HSCD39	40	C480~C4A 7	50304~50343
	FD*2	FD0~FD8191	8192	C4C0~E4B F	50368~58559
	SFD*2	SFD0~SFD4095	4096	E4C0~FC2 F	58560~64559
	FS*2	FS0~FS47	256	F4C0~F4E F	62656~62911

Lưu ý:

- Khu vực giữ nguồn điện được đánh dấu bằng*1, và vùng flash được đánh dấu bằng*2.
- Địa chỉ trong bảng trên được sử dụng khi PLC là máy tính cấp dưới và giao thức Modbus RTU hoặc MODBUS ASCII được sử dụng để giao tiếp, máy tính cấp trên chung là: SCADA/HMI/PLC.
- Nếu máy tính cấp trên là PLC, việc lập trình tuân theo giao thức Modbus RTU hoặc MODBUS ASCII
- Nếu máy tính cấp trên là hệ SCADA hoặc màn hình giao diện HMI, có 2 trường hợp. Đầu tiên là có driver của Xinje ví dụ như Xinje HMI hoặc Zijinqiao SCADA. Chương trình có thể được viết trực tiếp bằng cách sử dụng các phần tử lập trình bên trong PLC (Y0/M0); đối với loại thứ hai, Modbus RTU hoặc Modbus ASCII được chọn nếu không có trình điều khiển Xinje, sau đó sử dụng các địa chỉ trong bảng trên để xác định các biến dữ liệu.
- Điểm đầu vào và đầu ra thuộc hệ bát phân, vui lòng tính toán địa chỉ MODBUS điểm đầu vào và đầu ra tương ứng theo hệ bát phân, ví dụ: MODBUS tương ứng với Y0, địa chỉ là H6000, địa chỉ Modbus tương ứng với Y10 là H6008 (không phải H6010) và địa chỉ Modbus tương ứng với Y20 là H6010 (không phải H6020).
- Khi địa chỉ Modbus vượt quá 32767, nó cần được thể hiện bằng hệ thập lục phân và phải thêm "0" trước địa chỉ. Ví dụ: MODBUS của HD0 là 41088 ở dạng thập phân (vượt quá 32767) và 41088 không thể ghi vào phần mềm nên cần thể hiện dưới dạng thập lục phân là H0A080.
- Tính địa chỉ Modbus của X và Y, lấy X làm ví dụ, cách tính địa chỉ Modbus của Y cũng tương tự như của X.
X0: 20480 X10: 20480+8 X20: 20480+16 X30: 16384+24...
X10000: 20736 X10010: 20736+8 X10020: 20736+16...
X10200: 20800 X10210: 20800+8 X10220: 20800+16...

6-2-4 Định dạng dữ liệu Modbus

Chế độ truyền Modbus :

Có hai chế độ truyền: RTU và ASCII; Nó xác định việc truyền nối tiếp nội dung bit trong miền thông báo; nó quyết định cách đóng gói và giải mã thông tin; chế độ truyền (và thông số công) của tất cả các thiết bị trong liên kết nối tiếp Modbus phải giống nhau.

❖ Cấu trúc dữ liệu Modbus-RTU

1. Chế độ RTU:

Trong chế độ Modbus RTU (thiết bị đầu cuối từ xa), tín hiệu có hai ký tự thập lục phân 4 bit trong mỗi byte 8 bit. Chế độ này có mật độ dữ liệu rất cao, tốc độ thông lượng cao hơn Modbus ASCII. Mỗi thông báo phải được gửi bằng các ký tự liên tục.

Miền kiểm tra khung chế độ RTU: kiểm tra dự phòng chu kỳ (CRC).

Mô tả khung chế độ RTU:

Trạm Modbus	Mã chức năng	Dữ liệu	CRC	
1 byte	1 byte	0~252 byte	2 byte	
			CRC cao	CRC thấp

Định dạng:

START	Không có tín hiệu đầu vào $\geq 10\text{ms}$
Address(station no.)	Địa chỉ giao tiếp: nhị phân 8 bit
Function	Mã chức năng: nhị phân 8 bit
DATA(n - 1)	Nội dung dữ liệu: Dữ liệu N*8-bit, N<8, tối đa 8 byte
.....	
DATA 0	
CRC CHK Low	Mã kiểm tra CRC
CRC CHK High	Mã kiểm tra CRC 16 bit bao gồm hai mã nhị phân 8 bit
END	Không có tín hiệu đầu vào $\geq 10\text{ms}$

2. Địa chỉ Modbus:

00H: Tất cả các chương trình phát sóng PLC dòng Xinje XC - - các trạm slave không phản hồi.

01H: Giao tiếp với địa chỉ 01H PLC.

0FH: Giao tiếp với địa chỉ 15H PLC.

10H: Giao tiếp với địa chỉ 16H PLC, v.v. Lên tới 254 (FEH).

3. Hàm chức năng và dữ liệu:

Mã chức năng	Chức năng	Lệnh Modbus
01H	Đọc cuộn dây	COLR
02H	Đọc cuộn dây đầu vào	INPR (không hỗ trợ PLC Xinje)
03H	Đọc thanh ghi	REGR
04H	Đọc thanh ghi đầu vào	INRR
05H	Ghi cuộn dây	COLW
06H	Ghi thanh ghi	REGW
10H	Ghi nhiều thanh ghi	MRGW
0FH	Ghi nhiều cuộn dây	MCLW

(1) Lấy mã chức năng 06H làm ví dụ (ghi một thanh ghi) và giới thiệu định dạng dữ liệu.

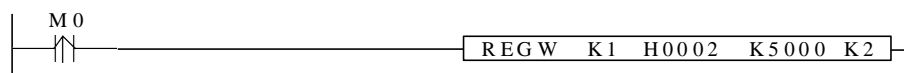
VD: Máy tính cấp trên ghi dữ liệu đến thanh ghi D2 của PLC (địa chỉ Modbus RTU là H0002)
Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
ID	01H	ID	01H
Mã chức năng	06H	Mã chức năng	06H
ID thanh ghi	00H	ID thanh ghi	00H
	02H		02H
Nội dung dữ liệu	13H	Nội dung dữ liệu	13H
	88H		88H
CRC CHECK Cao	25H	CRC CHECK Cao	25H
CRC CHECK Thấp	5CH	CRC CHECK Thấp	5CH

Giải thích:

1. Địa chỉ là số trạm PLC.
2. Mã chức năng là mã đọc/ghi giao thức Modbus-RTU.
3. Địa chỉ thanh ghi là địa chỉ modbus PLC, vui lòng xem chương 6-2-3.
4. Nội dung dữ liệu là giá trị trong D2.
5. CRC CHECK Cao / CRC CHECK Thấp là bit cao và thấp của giá trị kiểm tra CRC.

Nếu 2 PLC dòng Xinje XD3 giao tiếp với phần còn lại, hãy ghi K5000 vào D2.



M0 là điều kiện kích hoạt (Sườn lên). Nếu giao tiếp không thành công, lệnh sẽ thử lại hai lần.
Nếu giao tiếp lần thứ ba không thành công thì giao tiếp sẽ kết thúc.

Mối quan hệ giữa giao thức REGW và Modbus RTU (các lệnh khác giống nhau)

REGW	Mã chức năng 06H
K1	Số trạm
H0002	Địa chỉ Modbus
K5000	Nội dung dữ liệu 1388H
K2	Cổng nối tiếp PLC

Dữ liệu truyền thông hoàn chỉnh là: 01H 06H 00H 02H 13H 88H (hệ thống tự động kiểm tra CRC)

Nếu giám sát dữ liệu cổng nối tiếp 2 bằng công cụ gỡ lỗi cổng nối tiếp, dữ liệu là: 01 06 00 02 13 88 25 5C

Lưu ý: Lệnh không phân biệt thập phân, thập lục phân, nhị phân, bát phân, v.v. Ví dụ:

B10000, K16 và H10 là cùng giá trị nên các lệnh sau đều giống nhau.

REGW K1 B111110100 D1 K2

REGW K1 K500 D1 K2

REGW K1 H1F4 D1 K2

(2) Mã chức năng 01H/02H: đọc cuộn dây/đọc cuộn dây đầu vào

VD. Đọc địa chỉ cuộn dây 6000H (Y0). Lúc này, Y0 và Y1 đang

BẬT. Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
Địa chỉ	01H	Địa chỉ	01H

Mã chức năng	01H/02H	Mã chức năng	01H/02H
Địa chỉ cuộn dây	60H	Số byte	01H
	00H		
Số cuộn dây	00H	Nội dung dữ liệu	03H
	02H		
CRC CHECK Thấp	A3H	CRC CHECK Thấp	11H
CRC CHECK Cao	CBH	CRC CHECK Cao	89H

Khi trạng thái của Y0 và Y1 là BẬT, nội dung dữ liệu là 03H (0000 0011).

(3) Mã chức năng 03H: đọc thanh ghi

VD. Đọc hai thanh ghi bắt đầu từ 03E8H (D1000, D1001).

Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
Địa chỉ	01H	Địa chỉ	01H
Mã chức năng	03H	Mã chức năng	03H
Địa chỉ thanh ghi	03H	Số byte	04H
	E8H		
Số thanh ghi	00H	Nội dung dữ liệu	12H
			2EH
			04H
			E8H
CRC CHECK Thấp	44H	CRC CHECK Thấp	9DH
CRC CHECK Cao	7BH	CRC CHECK Cao	CCH

Lúc này dữ liệu đọc từ D1000 và D1001 là 122EH (4654) và 04E8H (1256).

(4) Mã chức năng 05H: ghi một cuộn dây VD.

Cài đặt trên địa chỉ cuộn dây 6000H (Y0).

Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
Địa chỉ	01H	Địa chỉ	01H
Mã chức năng	05H	Mã chức năng	05H
Địa chỉ cuộn dây	60H	Địa chỉ cuộn dây	60H
	00H		00H
Nội dung dữ liệu (byte thấp đứng trước byte cao)	FFH	Nội dung dữ liệu	FFH
	00H		00H
CRC CHECK Thấp	92H	CRC CHECK Thấp	92H
CRC CHECK Cao	3AH	CRC CHECK Cao	3AH

Lưu ý: khi ghi một cuộn dây, ON là 00FFH, OFF là 0000H; byte thấp nằm trước byte cao cho nội dung dữ liệu.

(5) Mã chức năng 0FH: ghi nhiều cuộn dây

VD. Ghi 16 cuộn dây bắt đầu từ địa chỉ 6000H (Y0).

Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
Địa chỉ	01H	Địa chỉ	01H
Mã chức năng	0FH	Mã chức năng	0FH
Địa chỉ cuộn dây	60H	Địa chỉ cuộn dây	60H
	00H		00H
Số cuộn dây	00H	Số cuộn dây	00H
	10H		10H
Số Byte	02H	-	-
Nội dung dữ liệu (byte thấp đứng trước byte cao)	03H		
	01H		
CRC CHECK Thấp	43H	CRC CHECK Thấp	4AH
CRC CHECK Cao	16H	CRC CHECK Cao	07H

Nội dung dữ liệu là 0103H, định dạng nhị phân là 0000 0001 0000 0011, ghi Y17~Y0 tương ứng, vì vậy Y0, Y1, Y10 được BẬT.

Lưu ý: khi ghi nội dung dữ liệu, byte thấp đứng trước byte cao.

(6) Mã chức năng 10H: ghi nhiều thanh ghi

VD: Ghi 3 thanh ghi bắt đầu từ địa chỉ 0000H (D0).

Chế độ RTU:

Định dạng hỏi		Định dạng phản hồi	
Địa chỉ	01H	Địa chỉ	01H
Mã chức năng	10H	Mã chức năng	10H
Địa chỉ thanh ghi	00H	Địa chỉ thanh ghi	00H
	00H		00H
Số thanh ghi	00H	Số thanh ghi	00H
	03H		03H
Số Byte	06H	-	-
Nội dung dữ liệu	00H		
	01H		
	00H		
	02H		
	00H		
	03H		
CRC CHECK Thấp	3AH	CRC CHECK Thấp	3AH
CRC CHECK Cao	81H	CRC CHECK Cao	81H

Sau khi thực thi, giá trị tại D0, D1, D2 lần lượt là 1, 2, 3.

Lưu ý: số byte = số thanh ghi * 2.

❖ Cấu trúc dữ liệu Modbus-ASCII

1. Chế độ ASCII:

Đối với chế độ Modbus ASCII (Chuẩn mã trao đổi thông tin Hoa Kỳ) trong các liên kết nối tiếp, mỗi byte 8 bit được gửi dưới dạng hai ký tự ASCII. Khi các liên kết và thiết bị giao tiếp không phù hợp với bộ giám sát thời gian ở chế độ RTU, chúng tôi thường sử dụng chế độ ASCII.

Lưu ý: Một byte cần hai ký tự, do đó chế độ ASCII có hiệu quả cao hơn chế độ RTU.

VD: Byte 0X5B sẽ được mã hóa thành hai ký tự: 0x35 và 0x42 (Mã ASCII 0x35 ="5", 0x42 ="B").

Miền kiểm tra khung chế độ ASCII: Kiểm tra dự phòng theo chiều dọc (LRC)

Mô tả khung chế độ ASCII:

Dấu bắt đầu	Số Modbus	Mã chức năng	Dữ liệu	LRC	Dấu kết thúc	
1 ký tự	2 ký tự	2 ký tự	0~252*2 ký tự	2 ký tự	2 ký tự	
0x3A					0x0D	0x0A

Định dạng:

STX (3AH)	Dấu bắt đầu=3AH
Mã địa chỉ bit cao	Vị trí giao tiếp (không): Gồm 2 mã ASCII
Mã địa chỉ bit thấp	
Mã chức năng bit cao	Mã chức năng (lệnh): Gồm 2 mã ASCII
Mã chức năng bit thấp	
ID bắt đầu lệnh	Bit bắt đầu lệnh: Gồm 4 mã ASCII
ID bắt đầu lệnh	
ID bắt đầu lệnh	
ID bắt đầu lệnh	
Độ dài dữ liệu	Độ dài từ đầu đến cuối: Gồm 4 mã ASCII
Độ dài dữ liệu	
Độ dài dữ liệu	
Độ dài dữ liệu	
LRC check bit cao	Mã kiểm tra LRC: Gồm 2 mã ASCII
LRC check bit thấp	
END bit cao	Dấu kết thúc: END Hi=CR(0DH), END Lo=CR(0AH)
END bit thấp	

2. Địa chỉ giao tiếp:

00H: Tắt cả chương trình phát sóng PLC dòng Xinje XC - - trạm slave không phản hồi.

01H: Giao tiếp với địa chỉ 01H PLC.

0FH: Giao tiếp với địa chỉ 15H PLC.

10H: Giao tiếp với địa chỉ 16H PLC.

Và cứ thế, lên tới 254 (FEH).

3. Chức năng và DỮ LIỆU:

Mã chức năng	Chức năng	Modbus tương ứng
01H	Đọc cuộn dây	COLR
02H	Đọc cuộn dây đầu vào	INRR
03H	Đọc thanh ghi	REGR
04H	Đọc thanh ghi đầu vào	INRR
05H	Ghi một cuộn dây	COLW
06H	Ghi một thanh ghi	REGW
10H	Ghi nhiều thanh ghi	MRGW
0FH	Ghi nhiều cuộn dây	MCLW

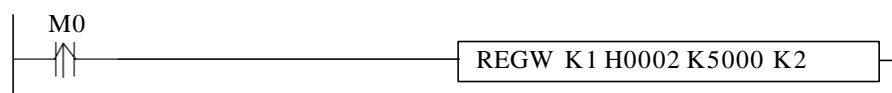
Lấy mã chức năng 06H (ghi một thanh ghi) làm ví dụ và giới thiệu định dạng dữ liệu (tương tự với các chức năng khác):
 VD: **Máy tính cấp trên ghi giá trị 5000 (Hexa là H1388) đến thanh ghi D2 của PLC (địa chỉ Modbus RTU là H0002)**
 Chế độ ASCII:

Dấu bắt đầu	3AH
ID	30H
	31H
Mã chức năng	30H
	36H
ID thanh ghi byte cao	30H
	30H
ID thanh ghi byte thấp	30H
	32H
Nội dung dữ liệu byte cao	31H
	33H
Nội dung dữ liệu byte thấp	38H
	38H
LRC	35H
	43H
Dấu kết thúc	0DH
	0AH

Mô tả:

1. Địa chỉ là số trạm PLC.
2. Mã chức năng là mã đọc/ghi giao thức Modbus-ASCII.
3. ID thanh ghi là ID giao tiếp modbus PLC, vui lòng xem chương 7-2-2.
4. Nội dung dữ liệu là giá trị trong D2.
5. LRC CHECK Thấp / CRC CHECK Cao là bit thấp và cao của giá trị kiểm tra CRC.

Nếu hai PLC Xinje XD3 giao tiếp với nhau, ghi K5000 vào D2.



M0 là điều kiện kích hoạt (sườn lên). Khi Xinje PLC giao tiếp bằng Modbus, nếu giao tiếp không thành công, lệnh sẽ thử hai lần. Nếu giao tiếp lần thứ ba không thành công thì giao tiếp sẽ kết thúc.

Mối quan hệ giữa giao thức REGW và ASCII (các lệnh khác cũng tương tự):

REGW	Mã chức năng 06H
K1	Số trạm
H0002	ID Modbus
K5000	Nội dung dữ liệu là 1388H
K2	Cổng nối tiếp giao tiếp PLC

Chuỗi dữ liệu hoàn chỉnh: 3AH 30H
 31H 30H 36H 30H 30H 30H
 32H 31H 33H 38H 38H 35H 43H

(hệ thống tự động kiểm tra CRC)

Nếu giám sát cổng nối tiếp 2 bằng công cụ gỡ lỗi cổng nối tiếp, dữ liệu là: 3AH 30H 31H 30H
 36H 30H 30H 30H 32H 31H 33H 38H 38H 35H 43H 0DH 0AH

Lưu ý: Dữ liệu không phân biệt thập phân, nhị phân, thập lục phân, v.v. Ví dụ: B10000, K16 và H10 cùng giá trị nên các lệnh sau đều giống nhau.

REGW K1 B111110100 D1 K2
 REGW K1 K500 D1 K2
 REGW K1 H1F4 D1 K2

6-2-5 Lệnh giao tiếp

Các lệnh Modbus bao gồm đọc/ghi cuộn dây, đọc/ghi thanh ghi; dưới đây sẽ giới thiệu chi tiết.

Chi tiết về các lệnh:

Định nghĩa toán hạng trong lệnh:

1. Trạm giao tiếp từ xa và số cổng nối tiếp.

VD: Một PLC kết nối 3 bộ biến tần. PLC cần ghi và đọc các thông số của biến tần. Số trạm biến tần là 1, 2 và 3.

Vậy số giao tiếp từ xa là 1, 2 và 3.

2. Số ID khởi động thanh ghi/cuộn dây từ xa :

Gán số cuộn dây/thanh ghi từ xa: ID cuộn dây/thanh ghi khởi động của PLC đọc và ghi, thường được sử dụng với 'số cuộn dây/thanh ghi được gán'.

VD: PLC đọc tần số đầu ra của biến tần Xinje (H2103), dòng điện đầu ra (H2104), điện áp bus (H2105), sau đó ID khởi động thanh ghi/cuộn dây từ xa là H2103, số cuộn dây được gán là K3.

3. Địa chỉ truyền thông thanh ghi/coil nội bộ : Thanh ghi/coil trong PLC được sử dụng để trao đổi dữ liệu với máy tính cấp dưới

VD: Ghi giá trị bit dữ liệu M0: ghi trạng thái M0 vào địa chỉ được chỉ định ở máy tính cấp dưới

Ghi giá trị thanh ghi D0: ghi giá trị D0 vào địa chỉ được chỉ định

Đọc giá trị bit dữ liệu M1: đọc nội dung trong máy tính cấp dưới được gán địa chỉ cho M1

Đọc giá trị bit thanh ghi D1: đọc nội dung trong máy tính cấp dưới được gán địa chỉ cho D1

4. Điều kiện giao tiếp:

Các điều kiện tiên quyết của giao tiếp Modbus có thể là cuộn dây đóng/mở bình thường và sườn lên/xuống. Khi cuộn dây đóng/mở kích hoạt, các lệnh Modbus sẽ luôn được thực thi. Khi giao tiếp giữa nhiều trạm slave hoặc lưu lượng lớn, độ trễ giao tiếp có thể xảy ra. Cuộn dây dao động có thể được sử dụng làm điều kiện kích hoạt. Khi sườn lên/xuống kích hoạt, các lệnh Modbus sẽ chỉ được thực thi một lần và chỉ khi sườn lên/xuống tiếp theo xuất hiện, các lệnh Modbus mới được thực thi lại.

Coil read [COLR]

1) Tóm tắt

Đọc trạng thái cuộn dây của trạm được chỉ định vào thiết bị cục bộ;

Coil read [COLR]			
Lệnh 16 bit	COLR	Lệnh 32 bit	-
Điều kiện thực thi	Cuộn dây Thường hở/đóng	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

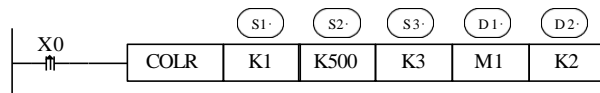
Toán hạng	Chức năng	Loại
S1	Chỉ định số trạm giao tiếp từ xa.	16 bit, BIN
S2	Chỉ định địa chỉ bắt đầu cuộn dây từ xa	16 bit, BIN
S3	Chỉ định số lượng cuộn dây	16 bit, BIN
D1	Chỉ định địa chỉ bắt đầu cuộn cục bộ	Bit
D2	Chỉ định số cổng nối tiếp.	16 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit							
	Hệ								Hãng	Mô đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn
S1	•	•		•	•				•									
S2	•	•		•	•				•									
S3	•	•		•	•				•									
D1												•	•	•	•	•	•	
D2									K									

Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD;
DM bao gồm DM, DHM. M gồm có M, HM, SM; S bao gồm S và HS; T bao gồm T và HT;
C bao gồm C và HC.

Chức năng



- Đọc cuộn dây, mã chức năng Modbus 01H.
- Cổng nối tiếp: K0~K5.K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Toán hạng S3: K1~K2000, số lượng cuộn dây tối đa là 2000.
- Khi X0 BẬT, lệnh COLR được thực thi. Khi lệnh bắt đầu thực thi, cờ đọc và ghi Modbus SM160 (cổng nối tiếp 2) được bật; khi quá trình thực thi hoàn tất, SM160 (cổng nối tiếp 2) được TẮT. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân gây ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 nằm ở SD160.

Đọc coil đầu vào [INPR]

1) Tóm tắt

Đọc trạng thái cuộn dây đầu vào của trạm được chỉ định vào thiết bị cục bộ.

Input coil read [INPR]			
Lệnh 16 bit	INPR	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây ON/OFF, sườn lên	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

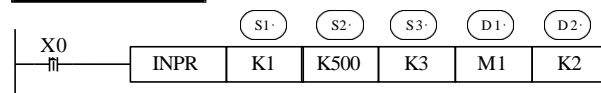
Toán hạng	Chức năng	Loại
S1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
S2	Chỉ định số địa chỉ bắt đầu cuộn dây từ xa	16 bit, BIN
S3	Chỉ định số cuộn dây	16 bit, BIN
D1	Chỉ định số địa chỉ bắt đầu của cuộn dây nhận cục bộ	bit
D2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit							
	Hệ								Hãng	Mô đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•		•	•				•									
S2	•	•		•	•				•									
S3	•	•		•	•				•									
D1												•	•	•	•	•	•	
D2									K									

Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S và HS; T bao gồm T và HT; C bao gồm C và HC.

Chức năng



- Đọc cuộn dây đầu vào, mã chức năng Modbus là 02H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1 (RS232), K2: Port2 (RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Toán hạng S3: K1~K2000, số cuộn dây đầu vào tối đa là 2008.
- Khi X0 BẬT, lệnh INPR được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2) được BẬT, SM160 được đặt TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được tự động thực hiện. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân gây ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

- Lệnh này không thể đọc được cuộn dây đầu vào PLC XINJE.

Single Coil Write [COLW]

1) Tóm tắt

Ghi cuộn dây được chỉ định của thiết bị cục bộ vào cuộn dây của trạm từ xa.

Single Coil write [COLW]			
Lệnh 16 bit	COLW	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phân cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

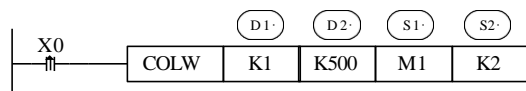
Toán hạng	Chức năng	Loại
D1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
D2	Chỉ định địa chỉ bắt đầu cuộn dây từ xa	16 bit, BIN
S1	Chỉ định địa chỉ bắt đầu của cuộn dây cục bộ	bit
S2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word									Phần tử lập trình Bit								
	Hệ								Hãng	Mô đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn
		D	D	X	Y	M	S		D	D							m	
D1	•	•		•	•			•										
D2	•	•		•	•			•										
S1											•	•	•	•	•	•		
S2								K										

Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S và HS; T bao gồm T và HT; C bao gồm C và HC.

Chức năng



- Ghi một cuộn dây, mã chức năng Modbus là 05H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1 (RS232), K2: Port2 (RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Khi X0 BẬT, lệnh COLW được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2)

được cài đặt BẬT, SM160 được đặt TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc và ghi Modbus của cổng nối tiếp 2 là ở SD160.

Multiple coils write [MCLW]

1) Tóm tắt

Ghi nhiều cuộn dây của thiết bị cục bộ vào cuộn dây của trạm từ xa.

Multiple coils write [MCLW]			
Lệnh 16 bit	MCLW	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

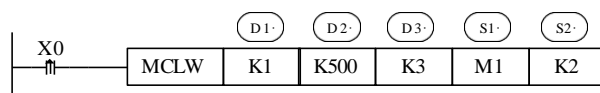
Toán hạng	Chức năng	Loại
D1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
D2	Chỉ định địa chỉ bắt đầu cuộn dây từ xa	16 bit, BIN
D3	Chỉ định số cuộn dây	16 bit, BIN
S1	Chỉ định địa chỉ bắt đầu của cuộn dây cục bộ	Bit
S2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình Word									Phân tử lập trình Bit								
	Hệ								Hàng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•					•									
S2	•	•	•	•					•									
S3	•	•	•	•					•									
D1												•	•	•	•	•	•	
D2									K									

Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm có M, HM, SM; S bao gồm S và HS; T bao gồm T và HT; C bao gồm C và HC.

Chức năng



- Ghi nhiều cuộn dây, mã chức năng Modbus là 0FH.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1 (RS232), K2: Port2 (RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng mở rộng 1), K5: Port5 (Cổng mở rộng 2).
- Toán hạng D3: số cuộn dây tối đa là 1976.
- Khi X0 BẬT, lệnh MCLW được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2) được cài đặt BẬT, SM160 được cài đặt TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

Register read [REGR]

1) Tóm tắt

Đọc thanh ghi của số trạm từ xa vào thiết bị cục bộ.

Register read [REGR]			
Lệnh 16 bit	REGR	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

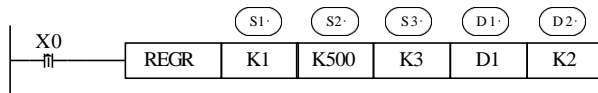
Toán hạng	Chức năng	Loại
S1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
S2	Chỉ định địa chỉ bắt đầu thanh ghi từ xa	16 bit, BIN
S3	Chỉ định số thanh ghi	16 bit, BIN
D1	Chỉ định địa chỉ bắt đầu của thanh ghi cục bộ	16 bit, BIN
D2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word											Phần tử lập trình Bit						
	Hệ								Hàng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•					•									
S2	•		•	•					•									
S3	•	•	•	•					•									
D1	•																	
D2									K									

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



- Đọc thanh ghi, mã chức năng Modbus là 03H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Toán hạng S3: số thanh ghi tối đa là 125.
- Khi X0 BẬT, lệnh REGR được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2) được BẬT, SM160 được đặt TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

Input register read [INRR]

1) Tóm tắt

Đọc thanh ghi đầu vào của trạm từ xa vào thiết bị cục bộ.

Input register read [INRR]			
Lệnh 16 bit	INRR	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Loại
S1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
S2	Chỉ định địa chỉ bắt đầu thanh ghi từ xa	16 bit, BIN
S3	Chỉ định số thanh ghi	16 bit, BIN
D1	Chỉ định địa chỉ bắt đầu của thanh ghi cục bộ	16 bit, BIN
D2	Chỉ định số cổng nối tiếp	16 bit, BIN

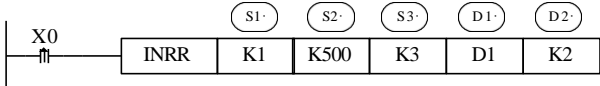
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit							
	Hệ								Hàng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•					•									
S2	•		•	•					•									

S3	•	•	•	•						•								
D1	•																	
D2										K								

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



- Đọc thanh ghi đầu vào, mã chức năng Modbus là 04H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Toán hạng S3: số thanh ghi tối đa là 125.
- Khi X0 BẬT, lệnh INRR được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2) được đặt BẬT, SM160 được đặt TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

Single Register write [REGW]

1) Tóm tắt
Ghi thanh ghi thiết bị cục bộ vào thanh ghi số trạm từ xa được chỉ định.

Register write [REGW]			
Lệnh 16 bit	REGW	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

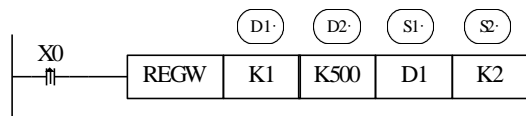
Toán hạng	Chức năng	Loại
D1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
D2	Chỉ định địa chỉ bắt đầu thanh ghi từ xa	16 bit, BIN
S1	Chỉ định địa chỉ bắt đầu của thanh ghi cục bộ	16 bit, BIN
S2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình Word								Phần tử lập trình Bit										
	Hệ								Hãng	Mô đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
D1	•	•	•	•					•										
D2	•	•	•	•					•										
S1	•																		
S2									K										

* Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



- Ghi thanh ghi, mã chức năng Modbus là 06H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1 (RS232), K2: Port2 (RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Khi X0 BẬT, lệnh REGW được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp 2) được BẬT, SM160 được TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

Multiple registers write [MRGW]

1) Tóm tắt

Ghi nhiều thanh ghi của thiết bị cục bộ vào các thanh ghi của trạm từ xa.

Multi-register write [MRGW]			
Lệnh 16 bit	MRGW	Lệnh 32 bit	-
Điều kiện thực thi	Là cuộn dây Thường hở/đóng, kích hoạt sườn	Model phù hợp	XD, XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

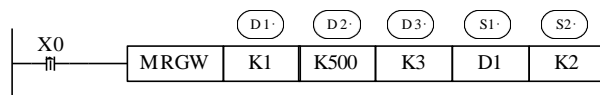
Toán hạng	Chức năng	Loại
D1	Chỉ định số trạm giao tiếp từ xa	16 bit, BIN
D2	Chỉ định địa chỉ bắt đầu giao tiếp từ xa	16 bit, BIN
D3	Chỉ định số thanh ghi	16 bit, BIN
S1	Chỉ định địa chỉ bắt đầu của các thanh ghi cục bộ	16 bit, BIN
S2	Chỉ định số cổng nối tiếp	16 bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình Word								Phân tử lập trình Bit									
	Hệ								Hãng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn
	D	D	D	X	Y	M	S		D	D							m	
D1	•	•	•	•				•										
D2	•	•	•	•				•										
D3	•	•	•	•				•										
S1	•																	
S2								K										

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



- Ghi nhiều thanh ghi, mã chức năng Modbus là 10H.
- Cổng nối tiếp: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2 (RS485), K3: Port3 (cổng mở rộng bên trái), K4: Port4 (Cổng bo mở rộng 1), K5: Port5 (Cổng bo mở rộng 2).
- Toán hạng D3: số thanh ghi tối đa là 123.
- Khi X0 BẬT, lệnh MRGW được thực thi, cờ ghi đọc Modbus SM160 (cổng nối tiếp2) được BẬT, SM160 được TẮT khi quá trình thực thi hoàn tất. Nếu xảy ra lỗi giao tiếp và số lần gửi lại được đặt, việc gửi lại sẽ được thực hiện tự động. Người dùng có thể kiểm tra các thanh ghi liên quan để xác định nguyên nhân xảy ra lỗi. Kết quả thực hiện lệnh đọc ghi Modbus của cổng nối tiếp 2 là SD160.

6 Cấu hình cổng nối tiếp Modbus

Có hai cách để thiết lập các thông số giao tiếp Modbus: 1. cài đặt thông số bằng phần mềm lập trình; 2. cài đặt thông số bằng công cụ XINJEConfig;

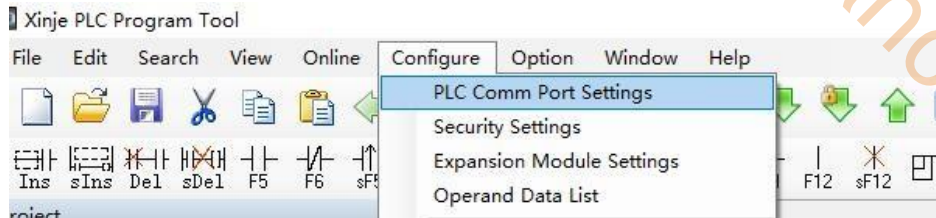
1. Cài đặt thông số bằng phần mềm lập trình

Khi sử dụng phần mềm lập trình để cấu hình các thông số cổng nối tiếp PLC thì phiên bản dưới V3.4 phải sử dụng chế độ giao tiếp XNET, phiên bản trên V3.4 cũng có thể sử dụng chế độ giao tiếp Modbus (cổng RS232).

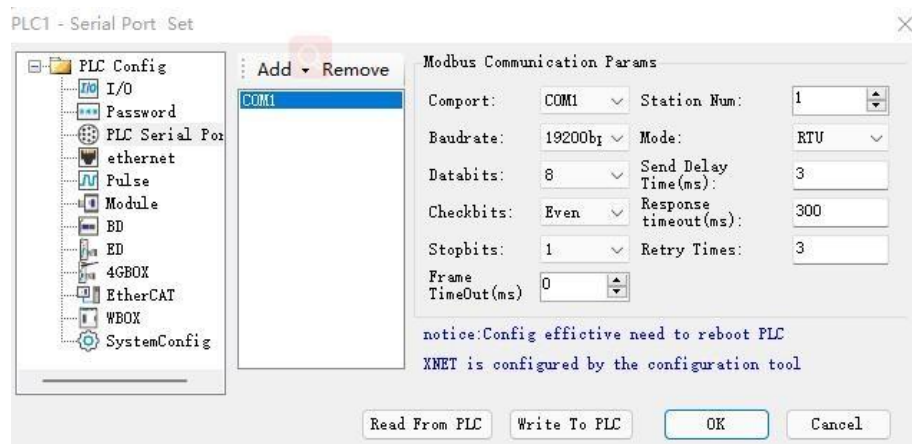
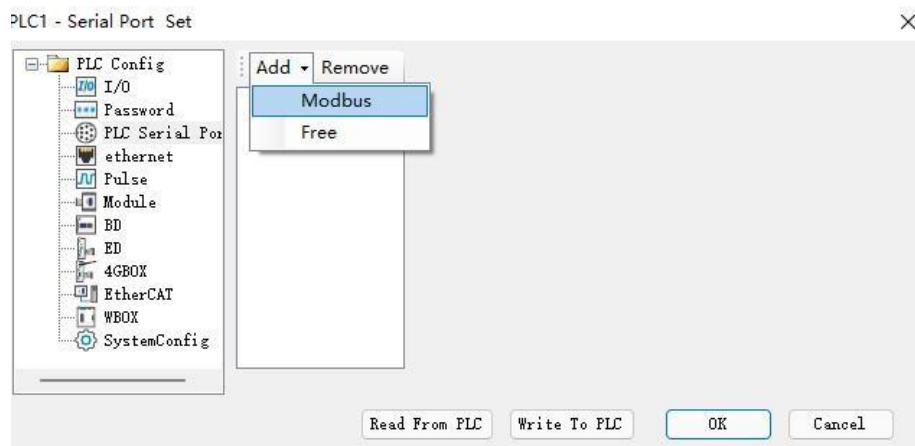
- (1) Sử dụng cáp tải xuống USB để kết nối PLC với máy tính. Cáp USB tải xuống ở đây chính là cáp tải xuống HMI, như hình bên dưới, phần mềm phải chuyển sang chế độ giao tiếp XNet.



- (2) Mở phần mềm lập trình, nhấp vào configure/PLC com port settings (cấu hình/cài đặt cổng com PLC), giao diện dưới đây sẽ xuất hiện:



- (3) Nhấp vào Add sẽ hiển thị hai chế độ, chế độ modbus và chế độ tự do, vui lòng chọn chế độ modbus, giao diện dưới đây sẽ hiển thị.



Port No.: Tức là Cổng PLC, COM0 là Cổng 0 (RS232), COM1 là Cổng 1 (RS232), COM2 là Cổng 2 (RS485) hoặc Cổng 2-RS232 (RS485) hoặc Cổng 2-RS485 (RS485), COM3 là Cổng 3 (cổng ED mở rộng bên trái), COM4 là Cổng 4 (cổng BD mở rộng phía trên 1), COM5 là Cổng 5 (cổng BD 2 mở rộng phía trên).

Baud rate (Tốc độ truyền), data bit, parity bit (bit chẵn lẻ), stop bit phải giống với thiết bị giao tiếp.

Station number: nếu PLC là master thì số trạm được mặc định là 1, nếu PLC là slave thì cần đặt số trạm khác.

Two communication modes (Hai chế độ giao tiếp): RTU, ASCII.

Delay before sending: Thời gian chờ đợi trước khi PLC gửi dữ liệu. Ở PLC dòng XC ban đầu, nếu PLC master giao tiếp với PLC slave, thì PLC master sẽ gửi dữ liệu đến PLC slave. Nếu PLC master gửi dữ liệu đến PLC slave sau lần đầu tiên và PLC slave chưa kịp nhận dữ liệu thì PLC master lại gửi dữ liệu đến PLC slave, điều này dễ dẫn đến lỗi của PLC slave; Ở PLC dòng XD, có độ trễ gửi để giải quyết vấn đề. Nghĩa là, sau khi nhận dữ liệu từ trạm slave, phải có một khoảng thời gian trễ nhất định để nhận dữ liệu giao tiếp tiếp theo, để không gây ra các vấn đề trên.

Reply overtime (ms): thời điểm PLC không thể nhận được phản hồi sau khi gửi yêu cầu và chờ gửi lại.

Retry times: Số lần PLC không thể nhận được phản hồi và mỗi phản hồi cần có thời gian chờ phản hồi.

- (4) Sau khi cài đặt, nhấp vào Write to PLC, sau đó cắt nguồn điện PLC và bật lại để cài đặt có hiệu lực.

Lưu ý: Phiên bản V3.4 trở về trước của dòng XD của dữ liệu cấu hình nối tiếp tải xuống và tải lên PLC phải sử dụng chế độ giao tiếp XNET, tức là sử dụng cổng USB để tải xuống và tải lên dữ liệu cấu hình. Nếu lời nhắc sau xuất hiện, bạn cần kiểm tra xem các thông số cổng nối tiếp bạn đã cấu hình có được tải xuống từ cổng USB về PLC hay không.

Lưu ý: Phiên bản V3.4 trở lên có thể được cấu hình ở chế độ giao tiếp Modbus (cổng RS232); Phiên bản V3.4 trở xuống dòng XD PLC phải sử dụng chế độ giao tiếp X-NET khi tải xuống và tải lên dữ liệu cấu hình nối tiếp, tức là tải xuống và tải lên dữ liệu cấu hình qua cổng USB.

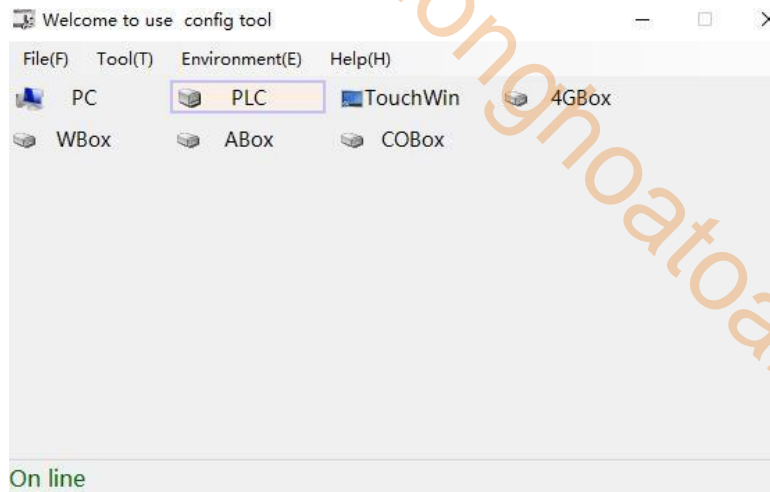
2. Cài đặt thông số bằng công cụ XINJEConfig

Khi sử dụng công cụ XINJEConfig để cấu hình các thông số của cổng nối tiếp PLC, các công cụ XINJEConfig phiên bản V1.6.308 trở xuống phải sử dụng cổng USB. Công cụ XINJEConfig dành cho V1.6.309 trở lên cũng có thể được cấu hình bằng cổng RS232.

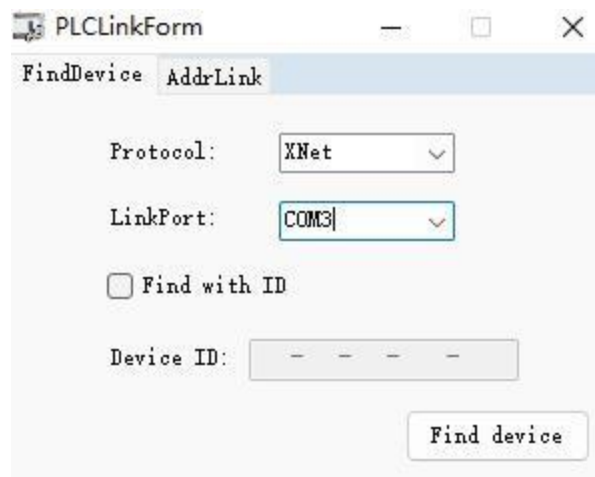
- (1) Sử dụng cáp tải xuống USB để kết nối PLC với máy tính. Cáp USB tải xuống ở đây chính là cáp tải xuống HMI như hình dưới đây.



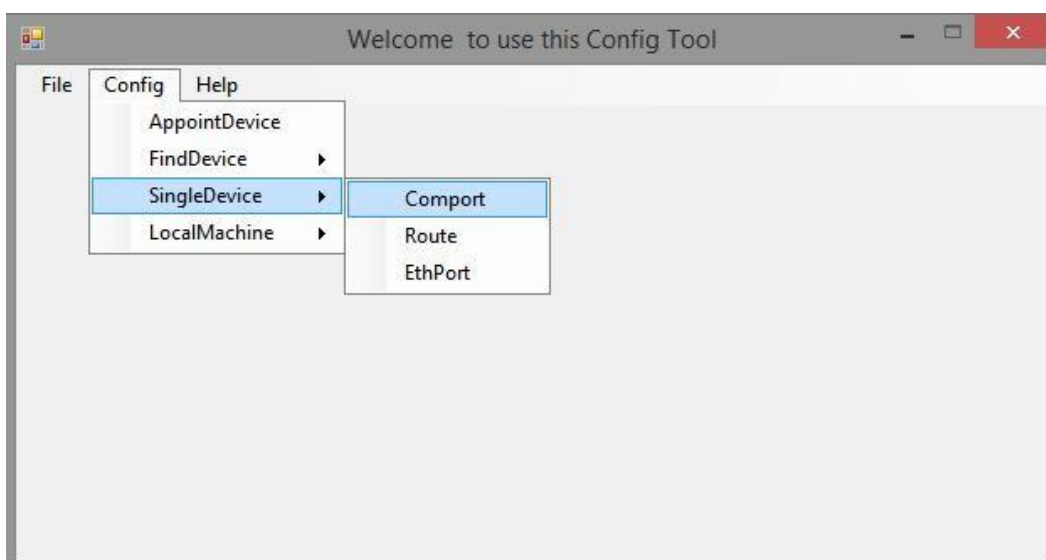
- (2) Mở công cụ XinjeConfig, nhấp PLC



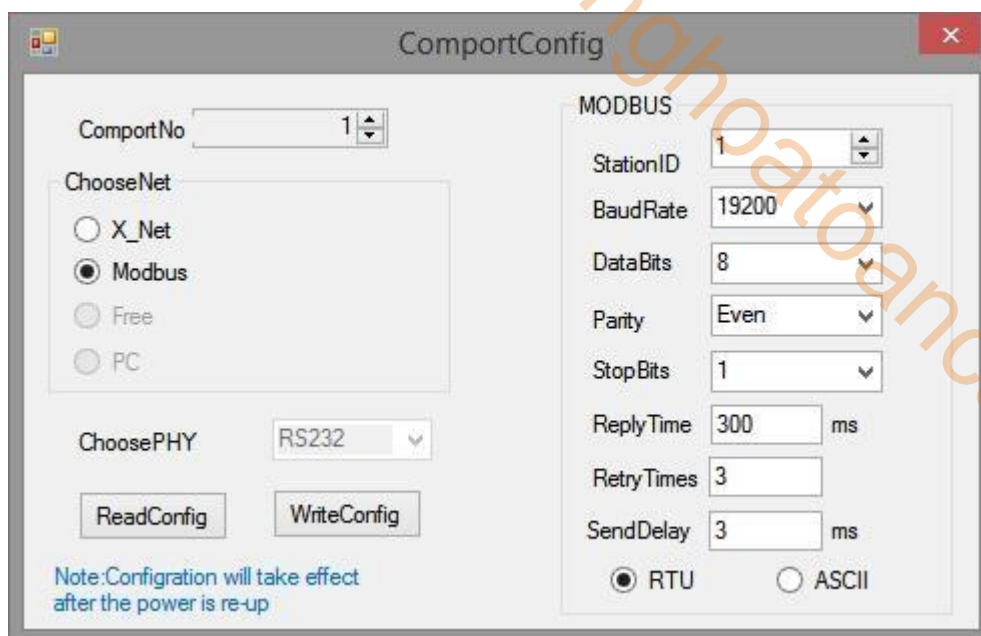
(3) Nhập config/find device:



(4) Chọn cổng com kết nối PC và PLC, nhấp OK. Nhấp vào config/single device/comport.



(5) Cửa sổ dưới đây sẽ xuất hiện.



Cổng nối tiếp: K0 ~ K5. Port0 (RS232), Port1 (RS232), Port2 (RS485) hoặc Port2-RS232 (RS232) hoặc Port2-RS485 (RS485), Port3 (cổng mở rộng bên trái), Port4 (cổng mở rộng phía trên 1), Port5 (cổng mở rộng phía trên 2).

Tại đây chúng ta có thể thiết lập chế độ giao tiếp và thông số của từng cổng giao tiếp.

(6) Khi cài đặt thông số cổng com hoàn tất, hãy nhấp vào writeconfig. Nó sẽ hiển thị thông báo “write configuration success” (ghi cấu hình thành công).



(7) Đóng công cụ XINJEConfig, tắt nguồn PLC và bật lại để cài đặt có hiệu lực.

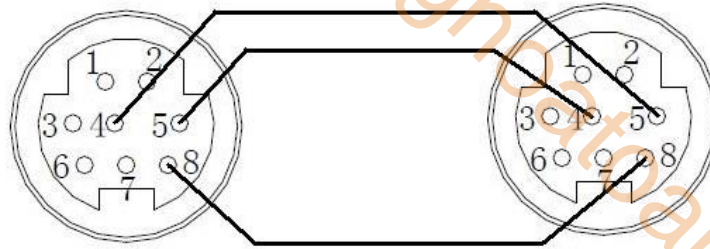
6-2-7 Ứng dụng giao tiếp Modbus

Có hai phương pháp nối dây:

Phương thức 232

COM2*1 diagram

- 4: RxD
- 5: TxD
- 8: GND

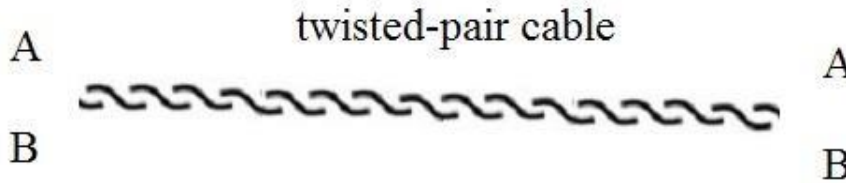


Mini Din 8 Pins port

Lưu ý:

1. COM2 có *1 chỉ hiển thị các chân RS232.
2. PLC dòng XD/XL, RS232 không hỗ trợ chế độ song công toàn phần nên chỉ có thể giao tiếp theo một hướng.
3. Khoảng cách giao tiếp RS232 ngắn (khoảng 13m); RS485 phù hợp với khoảng cách xa hơn.

Phương thức 485

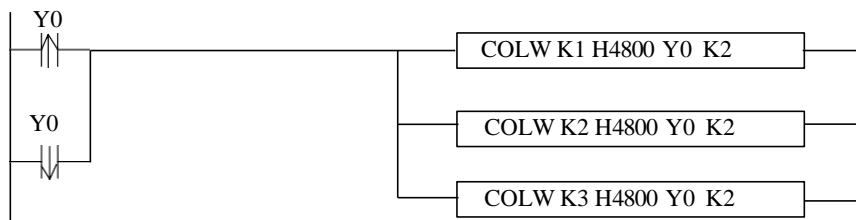


Kết nối tất cả các đầu nối dây A, kết nối tất cả các đầu nối dây B. A là RS485+, B là RS485-.

Ứng dụng: Một PLC dòng xinje XD3 điều khiển 3 PLC dòng XC, PLC phụ tuân theo hành động của PLC chủ. (Nếu Y0 của PLC chủ ở trạng thái ON, sau đó Y0 của PLC phụ ở trạng thái OFF; Nếu Y0 của PLC chính ở trạng thái OFF, sau đó Y0 của PLC phụ ở trạng thái OFF).

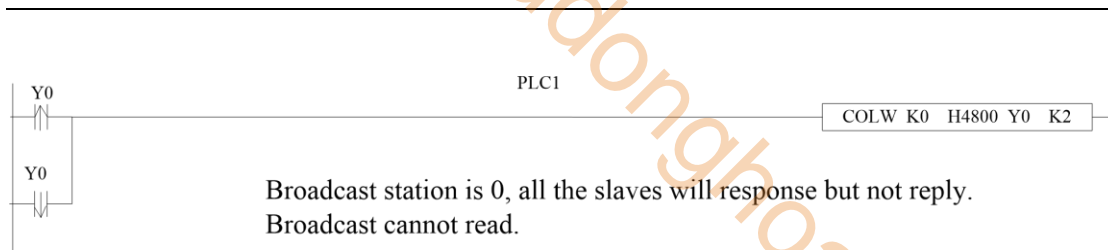
Điều kiện tiên quyết: trạng thái on/off Y0 khiến giao tiếp có đủ thời gian để phản ứng. Ngoài ra, ba PLC phụ có thể không đồng bộ (không đồng bộ hoàn toàn).

Chương trình thông thường của Phương thức 1



Chương trình lấy cổng nối tiếp 2 làm ví dụ, vì vậy cờ giao tiếp tương ứng là của cổng nối tiếp 2. Để tìm hiểu về các cổng nối tiếp khác, vui lòng tham khảo phụ lục 1. Để tìm hiểu về cổng nối tiếp, vui lòng tham khảo phụ lục 1.

Phương pháp thứ 2 sử dụng chức năng bản tin quảng bá (chức năng truyền thông đến tất cả Slave):



Khi trạng thái Y0 chính thay đổi, nó sẽ phát trạng thái này tới tất cả các PLC phụ. Việc đồng bộ hóa ba PLC tốt hơn phương pháp 1.

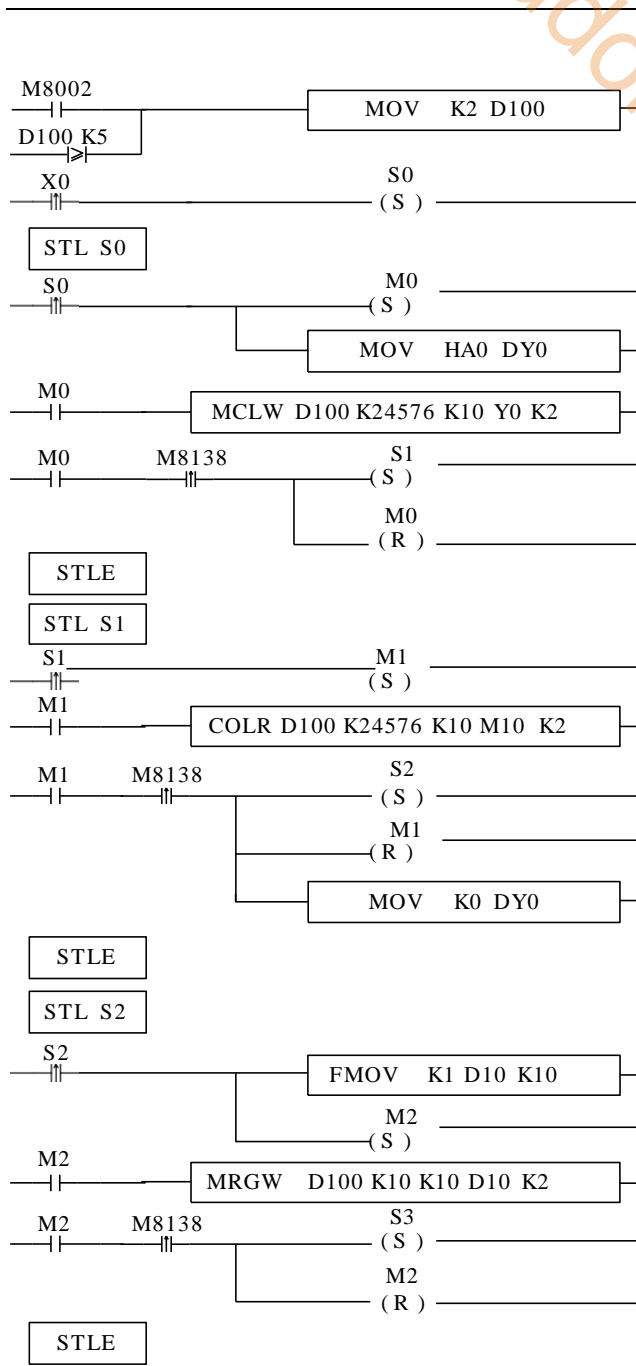
Ví dụ 1:

Sau đây là các chương trình đọc và ghi giao tiếp Modbus giữa 1 trạm chủ và 3 trạm phụ.

Thao tác chương trình:

- (1) Ghi trạng thái Y0 ~Y11 của PLC chủ vào Y0~Y11 của PLC phụ 2
- (2) Đọc trạng thái Y0~Y11 của PLC phụ 2 sang M10~M19 của PLC chủ
- (3) Ghi trạng thái D10~D19 của PLC chủ vào D10~D19 của PLC phụ 2
- (4) Đọc trạng thái D10~D19 của PLC phụ 2 sang D20~D29 của PLC chủ
- (5) Tương tự với PLC phụ 3 và 4

Dưới đây là so sánh các chương trình truyền thông Modbus-RTU dòng XC và XD cho mục đích tham khảo. Các chương trình truyền thông trong dòng XC như sau:

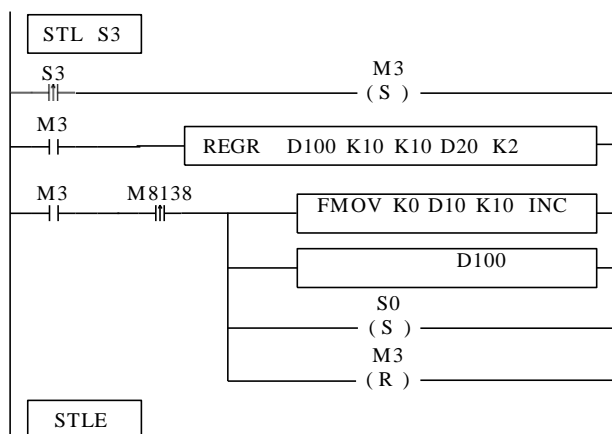


//gửi trạm số 2 tới D100,
thực thi quá trình S0

//đặt Y0~Y11 của trạm chủ ở trạng
thái ON, ghi trạng thái của trạm
chủ vào Y0~Y11 của PLC phụ 2,
3, 4. Nhập quá trình S1 khi giao
tiếp thành công.

//đọc trạng thái Y0~Y11 của PLC
phụ 2, 3, 4 sang M10~M19 của
PLC chủ. Đặt lại PLC chính
Y0~Y11 và nhập quá trình S2 sau
khi giao tiếp thành công.

//Ghi 1 vào D10~D19 của PLC
chủ, ghi trạng thái D10~D19 của
PLC chủ vào D10~D19 của PLC
phụ 2, 3, 4. Nhập quá trình S3 khi
giao tiếp thành công.

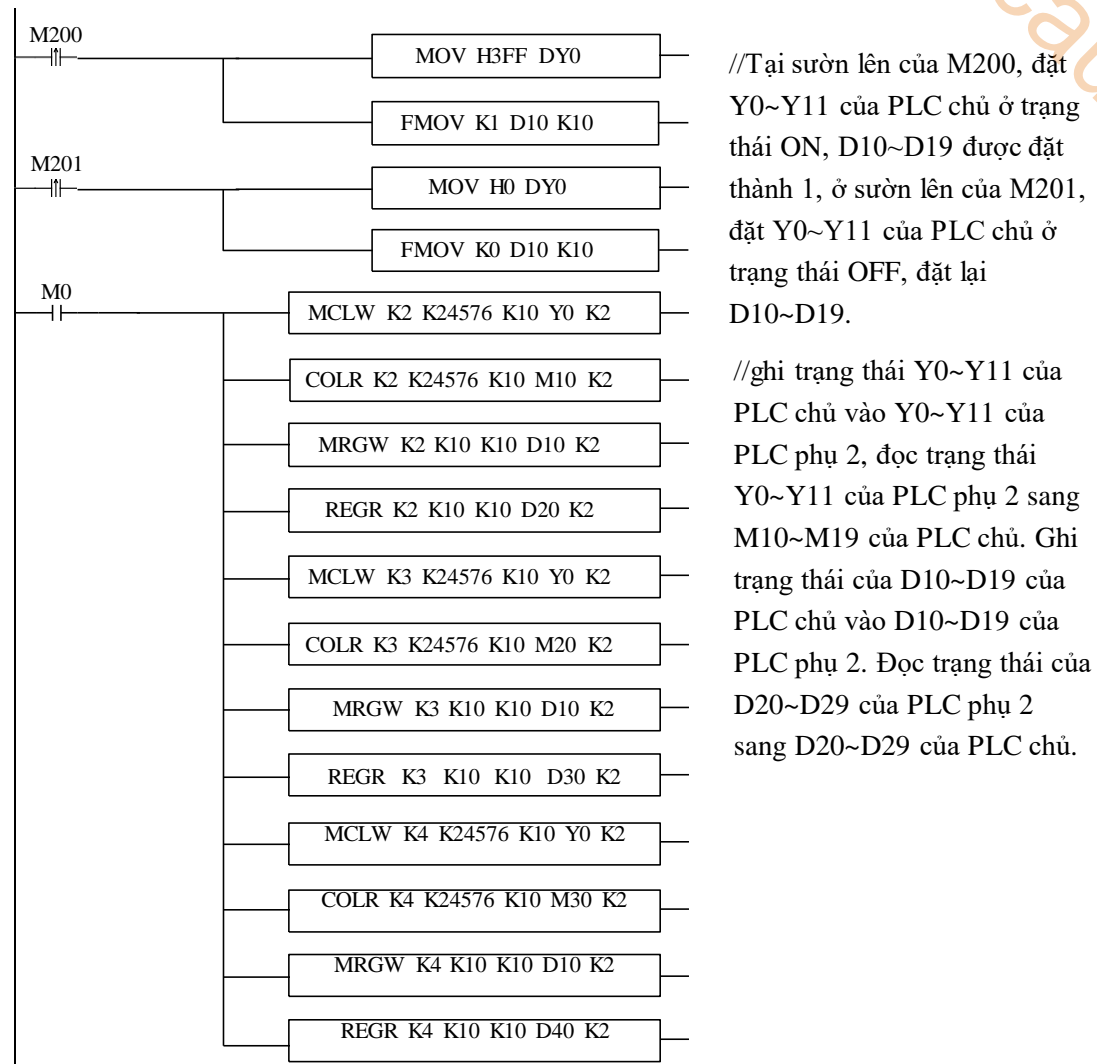


//đọc trạng thái D10~D19 của
PLC phụ 2, 3, 4 sang D20~D29
của PLC chủ, đặt lại D10~D19
sau khi giao tiếp thành công,
sau khi địa chỉ trạm thêm 1 thì chu
trình S0 được thực hiện theo tính
chu kỳ

Chế độ xử lý lệnh Modbus-RTU đã thay đổi. Người dùng có thể viết lệnh Modbus-RTU trực tiếp trong chương trình người dùng. Ngăn xếp giao thức sẽ xếp hàng các yêu cầu giao tiếp Modbus-RTU.

Giao tiếp là một nhiệm vụ khác. Trong chương trình chính, người dùng có thể viết nhiều lệnh giao tiếp Modbus-RTU cùng nhau và kích hoạt chúng cùng lúc thông qua cùng một điều kiện kích hoạt. PLC sẽ kích hoạt các lệnh giao tiếp này. Các lệnh được Modbus-RTU xếp hàng theo trạm giao thức, điều này sẽ không gây ra lỗi giao tiếp khi nhiều lệnh giao tiếp được thực thi cùng lúc với PLC dòng XC ban đầu.

Chương trình PLC dòng XD:



6-3 Giao tiếp tự do (Free Protocol)

6-3-1 Chế độ giao tiếp tự do

Giao tiếp định dạng tự do là truyền dữ liệu dưới dạng khối dữ liệu, bị giới hạn bởi bộ đệm PLC, lượng dữ liệu tối đa được gửi mỗi lần là 256 byte.

Giao tiếp tự do, tức là giao thức giao tiếp tùy chỉnh, hiện nay nhiều thiết bị thông minh trên thị trường hỗ trợ giao tiếp RS232 hoặc RS485, nhưng các giao thức được sử dụng bởi các sản phẩm khác nhau là khác nhau, chẳng hạn như: Xinje PLC sử dụng giao thức Modbus-RTU tiêu chuẩn, một số nhà sản xuất bộ điều khiển nhiệt độ sử dụng các giao thức tùy chỉnh; nếu sử dụng Xinje PLC để giao tiếp với bộ điều khiển nhiệt độ, cần sử dụng giao tiếp tự do để gửi dữ liệu đầy đủ theo giao thức của nhà sản xuất thiết bị để giao tiếp.

Điều kiện tiên quyết cho chế độ giao tiếp tự do:

1. Cổng0(RS232), Cổng1(RS232), Cổng2(RS485) hoặc Cổng2-RS232(RS232) hoặc Cổng2-RS485(RS485), Cổng3(cổng mở rộng bên trái), Cổng4(Cổng bo mở rộng 1), Cổng5(Cổng bo mở rộng 2) đều hỗ trợ giao tiếp tự do. Vì giao tiếp tự do cần thay đổi các tham số giao tiếp nên không nên sử dụng cổng 1.
2. Tốc độ truyền: 300bps~3Mbps, 4.5Mbps~9Mbps (model đặc biệt được hỗ trợ)
3. Định dạng dữ liệu phải giống với cài đặt thiết bị ở cấp thấp hơn. Có một số lựa chọn như sau:

Bit dữ liệu: 5 bit (hỗ trợ model đặc biệt), 6 bit (hỗ trợ model đặc biệt), 7 bit, 8 bit, 9 bit.

Bit chẵn lẻ: **none, odd parity, even parity, empty, mask**

Stop bit: 1 bit, 1,5 bit, 2 bit

4. **Starter: 1 byte, terminator: 1 byte**

Người dùng có thể đặt ký tự bắt đầu/kết thúc. Sau khi thiết lập ký tự bắt đầu/kết thúc, PLC sẽ tự động thêm ký tự bắt đầu/kết thúc khi gửi dữ liệu và tự động xóa ký tự bắt đầu/kết thúc khi nhận dữ liệu.

Trong thực tế, **initiator and terminator (nút khởi tạo và nút kết thúc)** có thể được coi là phần đầu và phần cuối của khung dữ liệu trong giao thức. Do đó, nếu giao tiếp thiết bị cấp thấp hơn có ký tự bắt đầu và kết thúc, nó có thể được đặt trong phần mềm hoặc được ghi trong giao thức.

5. Chế độ giao tiếp: 8 bit, 16 bit

Khi bộ đệm 8 bit được chọn để giao tiếp, các byte cao của thanh ghi không hợp lệ. PLC chỉ sử dụng byte thấp của thanh ghi để gửi và nhận dữ liệu.

Khi bộ đệm 16 bit được chọn để giao tiếp, PLC sẽ gửi tất cả dữ liệu của thanh ghi và gửi dữ liệu byte thấp trước, sau đó đến dữ liệu byte cao.

2.

Khi cần chuyển byte thấp và byte cao của thanh ghi 16 bit này sang thanh ghi 16 bit khác, bộ đệm 16 bit phải được chọn để giao tiếp và số byte giao tiếp là 2.

Khi giá trị được lưu trong thanh ghi 16 bit chỉ chiếm byte thấp, chúng ta có thể chọn bộ đệm 8 bit để giao tiếp. Số byte giao tiếp là 1. Thông thường khi giao tiếp, dữ liệu sẽ không vượt quá byte thấp của một thanh ghi (HFF) nên chúng ta chỉ cần sử dụng bộ đệm 8 bit mặc định trong phần mềm để giao tiếp.

6. Timeout: frame timeout (ms), reply timeout (ms)

Frame: Một chuỗi dữ liệu

Frame timeout: đề cập đến khoảng thời gian giữa hai khung dữ liệu mà PLC nhận được, điều này đảm bảo rằng PLC có thể phân biệt thời gian kết thúc nhận khung. Nó thường được sử dụng để đánh giá xem khung dữ liệu trong PLC đã được nhận hay chưa. Khi khoảng thời gian giữa hai khung dữ liệu dài hơn thời gian **frame timeout**, điều đó có nghĩa là một khung dữ liệu giao tiếp đã kết thúc.

Reply timeout: là thời điểm PLC không thể nhận được phản hồi sau khi gửi yêu cầu, chờ gửi lại. Nếu thời gian phản hồi được đặt vượt quá 300 ms thì khi giao tiếp mặc định, PLC sẽ đợi 300ms để bên kia phản hồi. Nếu không nhận được thời gian phản hồi, yêu cầu sẽ được gửi lại.

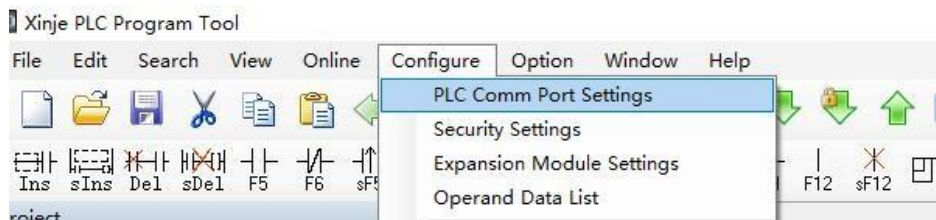
Nếu bạn muốn rút ngắn thời gian giao tiếp, bạn có thể điều chỉnh hai thông số trên theo kích thước của tốc độ truyền.

6-3-2 Cấu hình cổng nối tiếp

(1) Sử dụng cáp download USB để kết nối PLC với máy tính. Ở đây cáp download USB chính là cáp download HMI, như hình bên dưới thì phần mềm phải chuyển sang chế độ giao tiếp XNet.

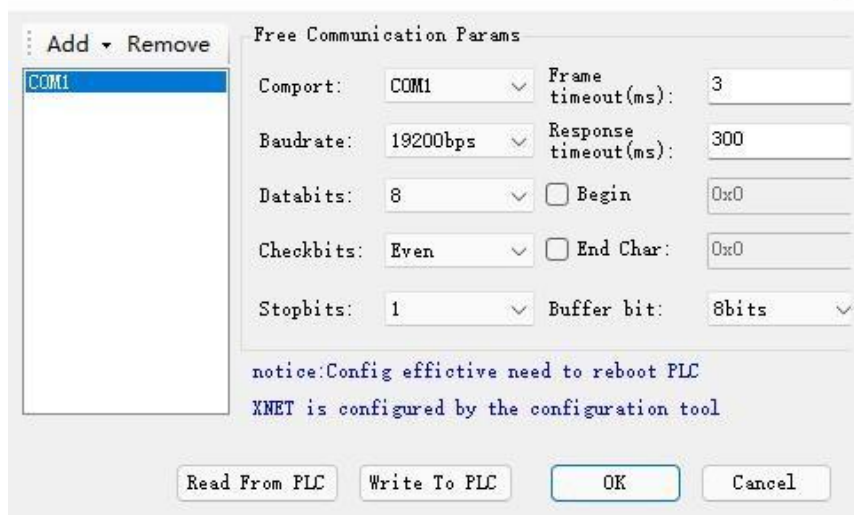
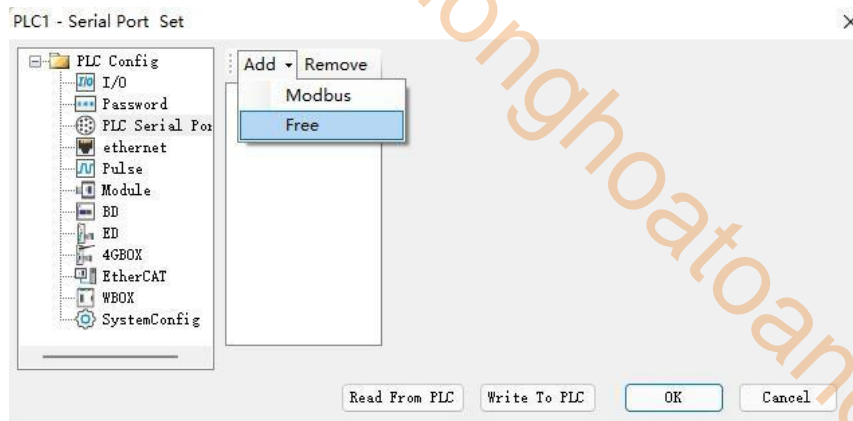


(2) Mở phần mềm lập trình, nhấp vào configure/PLC comm port settings. Nó sẽ hiển thị như hình dưới đây:



(3) Click add, it will show two modes, modbus mode and free mode, please select free mode, it will show below figure.

Nhấp vào add, nó sẽ hiển thị hai chế độ, chế độ modbus và chế độ tự do, vui lòng chọn chế độ tự do, nó sẽ hiển thị như hình bên dưới.



Số cổng/ Địa chỉ cổng: là cổng PLC, COM0 đề cập đến cổng 0 (RS232), COM1 đề cập đến cổng 1 (RS232), COM2 đề cập đến cổng 2 (RS485) hoặc cổng 2-RS232 (RS485) hoặc cổng 2-RS485 (RS485), COM3 đề cập đến Cổng 3 (cổng ED mở rộng bên trái), COM4 đề cập đến Cổng 4 (Cổng bo mở rộng 1), COM5 đề cập đến Cổng 5 (cổng Bo mở rộng 2).

Frame timeout (ms): Nó đề cập đến khoảng thời gian giữa hai khung dữ liệu được gửi bởi PLC, đảm bảo rằng người nhận phân biệt được thời gian kết thúc nhận khung.

Response timeout (ms): đề cập đến thời gian PLC không thể nhận được phản hồi sau khi gửi yêu cầu, chờ gửi lại.

Các cổng nối tiếp khác có thể được đặt theo thông số của thiết bị cấp thấp hơn.

(4) Sau khi cài đặt, nhấp vào “write to PLC”, sau đó cắt nguồn điện PLC và bật lại để cài đặt được áp dụng.

Lưu ý: Phiên bản V3.4 trở lên có thể được cấu hình ở chế độ giao tiếp Modbus (cổng RS232); Phiên bản V3.4 trở xuống dòng XD PLC phải sử dụng chế độ giao tiếp X-NET khi tải xuống và tải lên dữ liệu cấu hình qua cổng nối tiếp, tức là tải xuống và tải lên dữ liệu cấu hình qua cổng USB.

6-3-3 Các trường hợp áp dụng phù hợp

Khi nào cần sử dụng chế độ giao tiếp tự do?

Ví dụ: tình huống được mô tả trong phần trên là XINJE PLC giao tiếp với thiết bị điều khiển nhiệt độ và thiết bị sử dụng giao thức truyền thông riêng, quy định rằng nhiệt độ đọc phải được gửi bốn ký tự: "R", "T", "CR". Mỗi ký tự có ý nghĩa như sau:

Ký tự	Ý nghĩa
:	Data start (bắt đầu dữ liệu)
R	Read (đọc)
T	Temperature (nhiệt độ)
CR	Enter, data end (kết thúc dữ liệu)

PLC cần gửi mã ASCII của các ký tự trên đến thiết bị để đọc giá trị nhiệt độ hiện tại được đo bởi thiết bị. Các giá trị mã ASCII (dạng thập lục phân) của mỗi ký tự có thể thu được bằng cách truy vấn bảng mã ASCII.

Ký tự	Giá trị mã ASCII
:	3A
R	52
T	54
CR	0D

Rõ ràng, theo tình huống được mô tả ở trên, sử dụng lệnh MODBUS sẽ không thể giao tiếp, lúc này bạn cần sử dụng chế độ giao tiếp tự do. Cách sử dụng chi tiết sẽ được dùng làm minh họa để lập trình chương trình mẫu trong các chương sau.

6-3-4 Các lệnh giao tiếp tự do

Lệnh gửi dữ liệu [SEND]

1) Sơ lược

Ghi dữ liệu cục bộ vào địa chỉ trạm từ xa được chỉ định.

Gửi dữ liệu [SEND]			
Lệnh 16 bit	SEND	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, kích hoạt sườn lên	Model phù hợp	XD, XL
Phần cứng	Phiên bản V3.2.3 trở lên	Phần mềm	Phiên bản V3.2.2 trở lên

2) Toán hạng

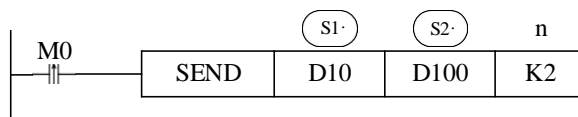
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ bắt đầu dữ liệu cục bộ	16 bits, BIN
S2	Số byte cần gửi	16 bits, BIN
n	Số cổng giao tiếp	16 bits, BIN

3) Phần tử lập trình phù hợp

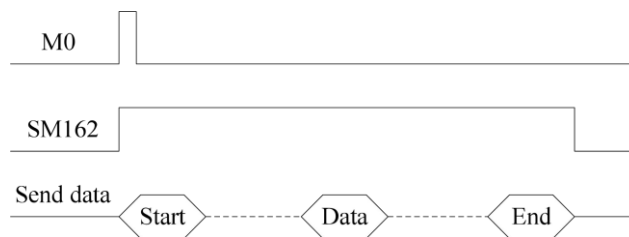
Toán hạng	Phần tử lập trình Word								Phần tử lập trình Bit										
	Hệ								Hãng	Mô đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•	•	•	•															
S2	•	•	•	•					•										
n	•								K										

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



- Lệnh gửi dữ liệu, sườn lên của M0 gửi dữ liệu một lần.
Phạm vi công giao tiếp. : K0 ~ K5. K0: Cổng 0 (RS232), K1: Cổng1(RS232), K2: Cổng2(RS485), K3: Cổng3(cổng mở rộng bên trái), K4: Cổng4(Công bo mở rộng 1), K5: Cổng5 (Công bo mở rộng 2).
- Trong quá trình truyền dữ liệu, cờ "gửi/send" SM162 (cổng giao tiếp 2) được bật



- Khi số bộ đệm là 8 bit, chỉ có dữ liệu byte thấp được gửi, vì vậy D100 = số lượng thanh ghi được gửi, ví dụ để gửi dữ liệu byte thấp trong D10-D17, D100 phải được đặt thành 8.
- Khi số bộ đệm là 16 bit, dữ liệu byte cao và thấp sẽ được gửi, do đó D100 = số lượng thanh ghi được gửi * 2. Ví dụ: khi gửi dữ liệu byte cao và thấp trong D10-D17, D100 phải được đặt thành 16 và khi gửi, byte thấp sẽ được truyền trước byte cao.

Lệnh nhận dữ liệu [RCV]

1) Sơ lược

Write the specified remote station no's data to local device.

Nhận dữ liệu [RCV]			
Lệnh 16 bit	RCV	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, kích hoạt sườn lên	Model phù hợp	XD, XL
Phần cứng	Phiên bản V3.2.3 trở lên	Phần mềm	Phiên bản V3.2.2 trở lên

2) Toán hạng

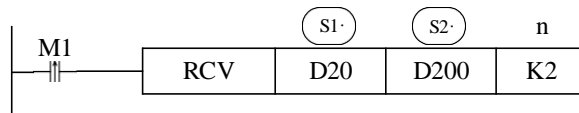
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ bắt đầu dữ liệu cục bộ	16 bits, BIN
S2	Số byte truyền hoặc địa chỉ phần tử lập trình	16 bits, BIN
n	Số cổng giao tiếp	16 bits, BIN

3) Phần tử lập trình phù hợp

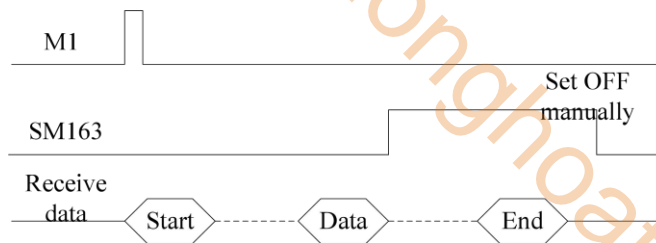
Toán hạng	Phần tử lập trình Word									Phần tử lập trình Bit									
	Hệ								Hãng số	Mô đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.	
S1	•	•	•	•															
S2	•	•	•	•					•										
n	•								K										

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



- Lệnh nhận dữ liệu, sườn lên của M1 nhận dữ liệu một lần.
- Phạm vi cổng giao tiếp: K0 ~ K5. K0: Cổng0 (RS232), K1: Cổng1 (RS232), K2: Cổng2 (RS485), K3: Cổng3 (cổng mở rộng bên trái), K4: Cổng4 (Cổng 1 mở rộng phía trên), K5: Port5 (cổng 2 mở rộng phía trên).
- After receiving the data, the "received" flag SM163 (communication port 2) is set on.
- Sau khi nhận được dữ liệu, cờ "đã nhận/ received" SM163 (cổng giao tiếp 2) được bật.



- Khi số bộ đệm là 8 bit thì dữ liệu nhận được chỉ được lưu ở byte thấp nên $D200 = \text{số byte cần nhận} * 2$, ví dụ để nhận 8 byte dữ liệu, lần lượt được lưu ở byte thấp của 8 thanh ghi D20-D27, lúc này nên đặt D200 là 16.
- Khi số bộ đệm là 16 bit thì dữ liệu nhận được sẽ được lưu trong một thanh ghi đầy đủ, do đó $D200 = \text{số byte cần nhận}$, ví dụ để nhận 8 byte dữ liệu, lần lượt được lưu trong 4 thanh ghi D20-D23, tại thời điểm này, D200 phải được đặt thành 8. Và khi nhận, byte thấp nằm trước byte cao.

Lệnh Giải phóng cổng nối tiếp [RCVST]

1) Sơ lược

Giải phóng cổng nối tiếp được chỉ định.

Release serial port[RCVST]			
Lệnh 16 bit	RCVST	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, kích hoạt sườn lên	Model phù hợp	XD, XL
Phần cứng	Phiên bản V3.2.3 trở lên	Phần mềm	Phiên bản V3.2.2 trở lên

2) Toán hạng

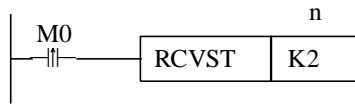
Toán hạng	Chức năng	Kiểu dữ liệu
n	Số cổng giao tiếp	16 bits, BIN

3) Phần tử lập trình phù hợp

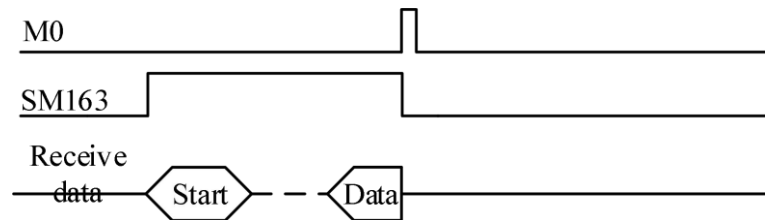
Toán hạng	Phần tử lập trình Word									Phần tử lập trình Bit								
	Hệ								Hằng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn.
n	•							K										m

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



- Lệnh cổng COM, sườn lên của M0 thực thi một lần.
- Phạm vi cổng giao tiếp: K0 ~ K5. K0: Cổng0 (RS232), K1: Cổng1(RS232), K2: Cổng2(RS485), K3: Cổng3(cổng mở rộng bên trái), K4: Cổng4(Cổng 1 mở rộng phía trên), K5: Port5 (cổng 2 mở rộng phía trên).
- Khi giải phóng cổng nối tiếp, cờ "đã nhận" SM163 (cổng giao tiếp 2) ở trạng thái OFF.
- Để sử dụng chế độ giao tiếp tự do, nếu không có thời gian chờ (timeout) hoặc thời gian chờ được đặt quá dài, tài nguyên cổng nối tiếp bị chiếm dụng có thể được giải phóng ngay lập tức thông qua các lệnh RCVST cho các thao tác giao tiếp khác.

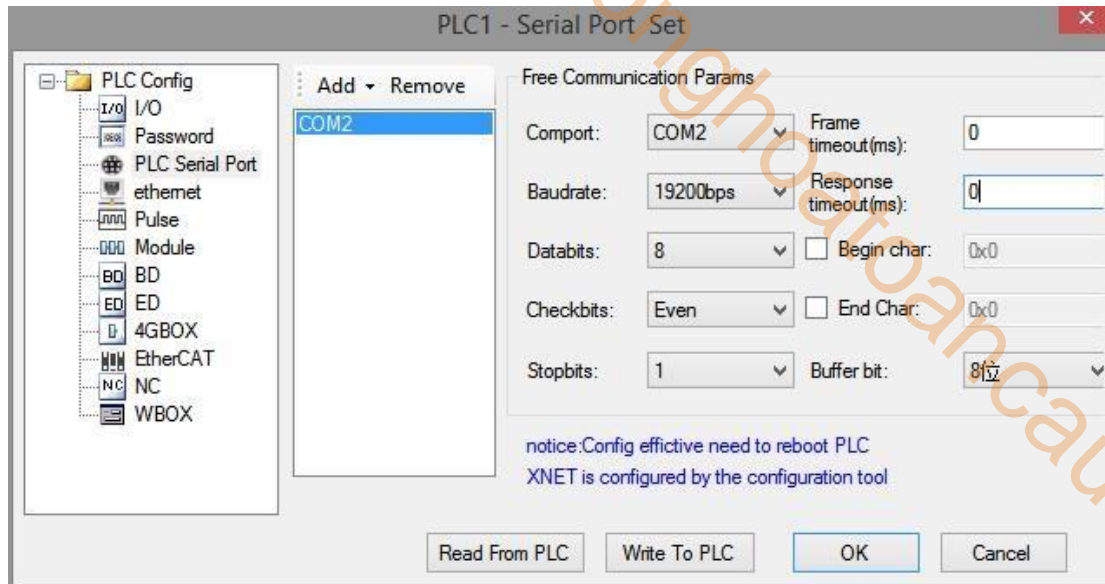


6-3-5 Ví dụ về giao tiếp tự do

Ví dụ 1: Trong chương 6-3-3, chúng tôi đưa ra ví dụ về giao tiếp giữa Xinje PLC và thiết bị điều khiển nhiệt độ khi giải thích lý do sử dụng chế độ giao tiếp tự do. Dưới đây là một ví dụ.

Các bước thao tác:

1. Kết nối phần cứng đầu tiên. Ở đây chúng ta sử dụng cổng nối tiếp số 2 của PLC để giao tiếp, tức là 485+ trên thiết bị được kết nối với A của cổng đầu ra của PLC, và 485- trên thiết bị được kết nối với B của cổng đầu ra của PLC.
2. Đặt các thông số cổng nối tiếp của PLC theo các thông số truyền thông của thiết bị điều khiển nhiệt độ. Các thông số được thiết lập như sau. Sau khi cài đặt các thông số, nguồn điện có thể được khởi động lại.



3. Thực hiện chương trình theo mô tả trong chương 6-3-3.

Đọc nhiệt độ: “ : ”“R”“T”“CR”

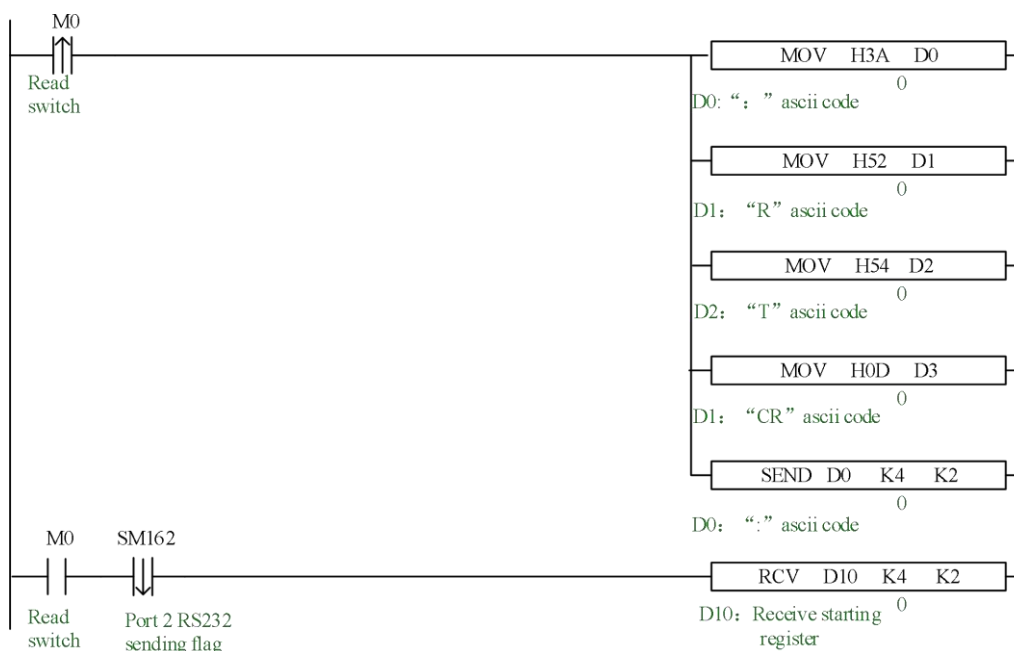
“ : ” ----- data start

“R” ----- read

“T” ----- temperature

“CR” ----- enter, data end

Chương trình



Khi cố gắng giao tiếp giữa PLC và các thiết bị thông minh khác, bạn nên sử dụng công cụ gỡ lỗi nối tiếp để xác định định dạng dữ liệu của giao tiếp, tức là giao thức. Ưu điểm của phương thức này là: công cụ gỡ lỗi nối tiếp dễ sửa đổi và sử dụng linh hoạt; sau khi công cụ gỡ lỗi nối

tiếp xác định rằng giao tiếp có thể thành công, chương trình PLC được viết theo định dạng dữ liệu thu được, thường mang lại kết quả gấp đôi với ít công sức hơn.

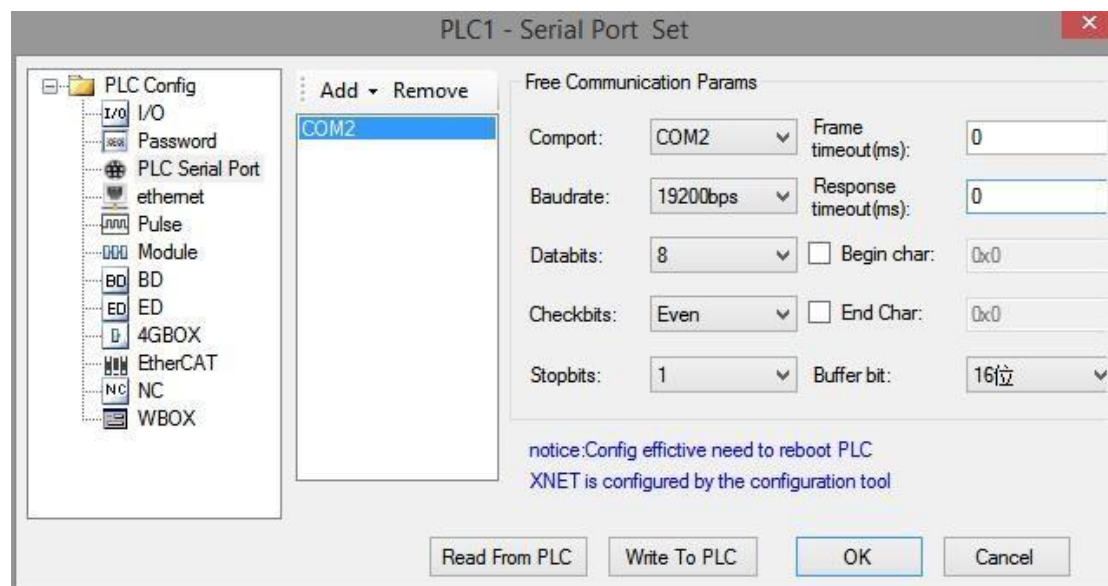
Trên thực tế, giao thức Modbus-RTU có thể được coi là một loại giao thức tự do đặc biệt. Mọi quan hệ giữa chúng tương tự như hình elip và hình tròn. Chúng ta có thể thử sử dụng định dạng tự do để nhận ra chức năng của lệnh Modbus.

Ví dụ 2: Các giá trị của năm thanh ghi của PLC XD3 được gửi đến D1-D5 của PLC XDM khác.

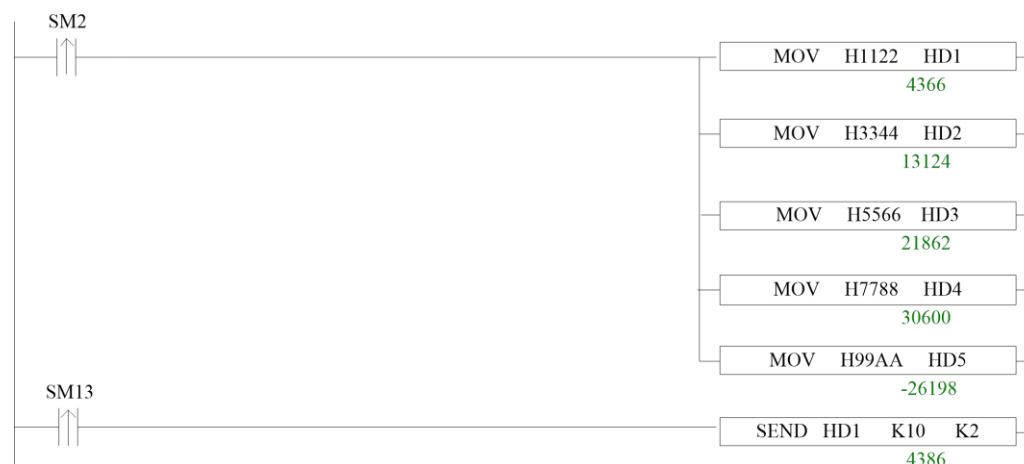
Nếu người dùng nắm rõ giao tiếp Modbus, họ có thể sử dụng chế độ giao tiếp Modbus-RTU để làm như vậy, miễn là họ viết ""write multiple register instructions (MRGW)" trong máy chủ. Ở đây chúng tôi thực hiện điều đó ở chế độ giao tiếp tự do.

Các bước thao tác:

1. Kết nối phần cứng đầu tiên. Ở đây chúng ta sử dụng cổng nối tiếp 2 của PLC để giao tiếp, tức là kết nối A của hai PLC và kết nối B của hai PLC.
2. Thiết lập các tham số cổng nối tiếp giống nhau của hai PLC. Các thông số được thiết lập như sau. Sau khi cài đặt các thông số, nguồn điện có thể được khởi động lại.



3. Chương trình cho XD3:



Chương trình dòng XDM:



Đôi khi dữ liệu giao tiếp của người dùng được lưu trữ trong nhiều thanh ghi dưới dạng mã ASCII. Người dùng cần lấy giá trị này ra, lưu vào thanh ghi và hiển thị trên HMI. Khách hàng thường cần nhắc sử dụng lệnh HEX (ASCII sang thập lục phân) để đạt được điều đó. Nhưng lệnh HEX rất khó sử dụng và khó hiểu. Thông thường, chúng tôi sẽ không sử dụng lệnh này để hoàn thành nó. Mọi quan hệ giữa các giá trị có thể được tìm thấy bằng bảng so sánh mã ASCII. Bảng mã ASCII:

Giá trị ASCII	Ký tự điều khiển	Giá trị ASCII	Ký tự điều khiển	Giá trị ASCII	Ký tự điều khiển	Giá trị ASCII	Ký tự điều khiển
0	NUT	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	”	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	o	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	—	127	DEL

Ví dụ 3: Bộ điều khiển áp suất giao tiếp với PLC ở chế độ giao tiếp tự do để thực hiện thu thập dữ liệu. Giá trị hiển thị trên bộ điều khiển áp suất là -0,7814 MPa. Giá trị được PLC thu thập

được lưu trữ từ D0 và bảy thanh ghi lần lượt được lưu trữ. Tuy nhiên, giá trị của tổ hợp bảy thanh ghi cần được lấy ra và lưu trữ trong D46 dưới dạng thập phân. Thông qua giám sát dữ liệu của PLC, mã ASCII trong thanh ghi D0~D6 có thể được giám sát như sau:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	-	0	.	7	8	1	4			
D10										
D20										
D30										
D40										

Chuyển sang định dạng thập phân và hiển thị như bên dưới:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	45	48	46	55	56	49	52	0	0	0
D10	0	0	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0	0	0

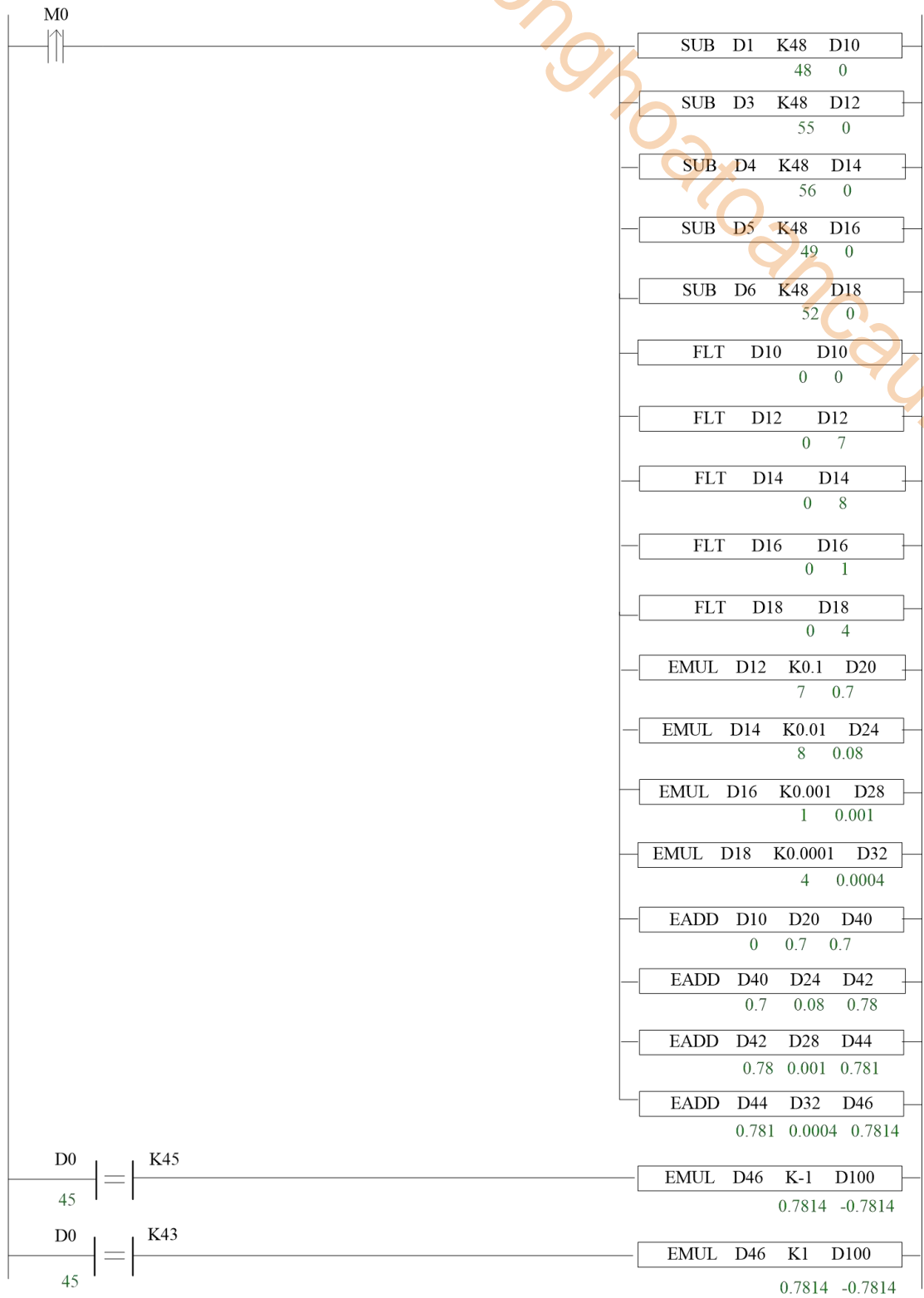
Bằng cách so sánh mối quan hệ giữa mã ASCII và giá trị thập phân, chúng ta có thể tìm ra quy luật có 48 điểm khác biệt giữa mã ASCII ở các giá trị D1, D3, D4, D5, D6 và giá trị thập phân. Các giá trị thập phân cuối cùng thu được bằng cách trừ các giá trị trong thanh ghi cho K48 và nhân với 10. Công thức như sau:

$$D46 = (D1 - 48) * 1 + (D3 - 48) * 0.1 + (D4 - 48) * 0.01 + (D5 - 48) * 0.001 + (D6 - 48) * 0.0001$$

D0 là một bit dấu. Tra cứu bảng, chúng ta thấy khi D0 = K45 thì biểu thị giá trị âm;

khi D0 = K43, nó đại diện cho giá trị dương.

Sơ đồ bậc thang như sau:



6-4 Cờ và thanh ghi truyền thông

Cờ báo truyền thông

Công nối tiếp	Địa chỉ thanh ghi	Chức năng	Giải thích
Công 0	SM140	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM142	Cờ gửi qua chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM143	Cờ nhận được dữ liệu qua chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF
Công 1	SM150	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM152	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM153	Cờ đã nhận dữ liệu bằng chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF
Công 2	SM160	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM162	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM163	Cờ đã nhận dữ liệu bằng chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF
Công 3	SM170	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM172	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM173	Cờ đã nhận dữ liệu bằng chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF

Cổng 4	SM180	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM182	Cờ gửi qua chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM183	Cờ nhận được dữ liệu qua chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF
Cổng 5	SM190	Cờ thực thi lệnh đọc-ghi Modbus	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM192	Cờ gửi qua chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, cờ ở trạng thái ON Khi thực hiện xong, cờ ở trạng thái OFF
	SM193	Cờ nhận được dữ liệu qua chế độ giao tiếp tự do	Khi nhận khung dữ liệu hoặc nhận dữ liệu hết thời gian chờ (data timeout), cờ sẽ ở trạng thái OFF. Yêu cầu đặt chương trình người dùng ở trạng thái OFF

Thanh ghi truyền thông

	Địa chỉ	Chức năng	Giải thích
Cổng 0	SD140	Kết quả thực thi lệnh đọc và ghi Modbus	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)
	SD141	Kết quả giao tiếp X-Net	0: correct 1: communication timeout 2: memory error 3: receive CRC error
	SD142	Kết quả gửi dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: free communication buffer overflow
	SD143	Kết quả nhận dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol

			416: no end symbol
	SD144	Địa chỉ nhận dữ liệu bằng chế độ giao tiếp tự do	Tính theo byte, không bao gồm ký tự bắt đầu và ký tự kết thúc
		
	SD149		
Cổng 1	SD150	Kết quả thực thi lệnh đọc và ghi Modbus	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)
	SD151	Kết quả giao tiếp X-Net	0: correct 1: communication timeout 2: memory error 3: receive CRC error
	SD152	Kết quả gửi dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: free communication buffer overflow
	SD153	Kết quả nhận dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol 416: no end symbol
	SD154	Số dữ liệu nhận bằng chế độ giao tiếp tự do	Count as byte, not include start symbol and end symbol
		
	SD159		
Cổng 2	SD160	Kết quả thực thi lệnh đọc và ghi bằng chế độ Modbus	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)
	SD161	Kết quả giao tiếp X-Net	0: correct 1: communication timeout

			2: memory error 3: receive CRC error
	SD162	Kết quả gửi dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: free communication buffer overflow
	SD163	Kết quả nhận dữ liệu bằng chế độ giao tiếp tự do	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol 416: no end symbol
	SD164	Số dữ liệu nhận bằng chế độ giao tiếp tự do	Count as byte, not include start symbol and end symbol
		
	SD169		
Port 3	SD170~SD179		
Port 4	SD180~SD189		
Port 5	SD190~SD199		

6-5 Lệnh đọc/ghi thông số công nối tiếp

Ngoài việc sửa đổi các tham số giao tiếp thông qua bảng cấu hình công nối tiếp, nó cũng có thể được thực hiện bằng lệnh đọc [CFGCR] các tham số nối tiếp và lệnh ghi [CFGCW] các tham số nối tiếp.

6-5-1 Lệnh đọc thông số công nối tiếp [CFGCR]

1) Sơ lược

Đọc các tham số công nối tiếp vào các thanh ghi được chỉ định cục bộ.

Đọc thông số công nối tiếp [CFGCR]			
Lệnh 16 bit	CFGCR	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, kích hoạt sườn lên	Model phù hợp	XD, XL
Phần cứng	-	Phần mềm	Phiên bản V3.4 trở lên

2) Toán hạng

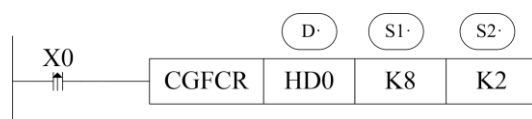
Toán hạng	Chức năng	Kiểu dữ liệu
D	Địa chỉ bắt đầu thanh ghi cục bộ	16-bit, BIN
S1	Số lượng tham số đọc từ công COM	16-bit, BIN
S2	Địa chỉ công nối tiếp	16-bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word										Phần tử lập trình bit							
	Hệ								Hãng	Mô đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D	•																	
S1	•	•						•										
S2	•							K										

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



- Toán hạng S1: Số lượng thanh ghi được sử dụng để đọc tham số công nối tiếp thường là 8 (dòng XD5E/XDME là 9).
- Toán hạng S2: Phạm vi công nối tiếp: K0 ~ K5. K0: Cổng0, K1: Cổng1, K2: Cổng2 hoặc Cổng2- RS232 hoặc Cổng2-RS485, K3: Cổng3, K4: Cổng4, K5: Cổng5.

- Đọc 8 thông số của cổng nối tiếp 2 vào HD0~HD7. Tham khảo phần 6-5-3 để biết tên và định nghĩa của các thông số cụ thể.

6-5-2 Lệnh ghi thông số cổng nối tiếp [CFGCW]

1) Sơ lược

Ghi giá trị của thanh ghi được chỉ định cục bộ vào cổng nối tiếp cụ thể.

Ghi thông số cổng nối tiếp [CFGCW]			
Lệnh 16 bit	CFGCW	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, kích hoạt sườn lên	Model phù hợp	XD, XL
Phần cứng	-	Phần mềm	Phiên bản V3.4 trở lên

2) Toán hạng

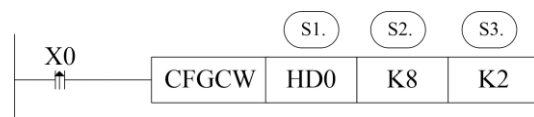
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ bắt đầu thanh ghi cục bộ	16-bit, BIN
S2	Số lượng tham số ghi từ cổng COM	16-bit, BIN
S3	Địa chỉ cổng nối tiếp	16-bit, BIN

3) Phân tử lập trình phù hợp

Toán hạng	Phân tử lập trình word									Phân tử lập trình bit								
	Hệ								Hãng	Mô đun	Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•																	
S2	•	•							•									
S3	•								K									

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



- Toán hạng S2: Số lượng thanh ghi được sử dụng để ghi tham số cổng nối tiếp thường là 8 (dòng XD5E/XDME là 9).
- Toán hạng S3: Phạm vi cổng nối tiếp: K0 ~ K5. K0: Cổng0, K1: Cổng1, K2: Cổng2 hoặc Cổng2- RS232 hoặc Cổng2-RS485, K3: Cổng3, K4: Cổng4, K5: Cổng5.
- Ghi các tham số HD0~HD7 vào cổng nối tiếp 2. Tham khảo phần 6-5-3 để biết tên và định nghĩa của các tham số cụ thể.

6-5-3 Tên các tham số cổng nối tiếp và cài đặt

Giả sử HD0-HD14 tương ứng với các tham số cổng nối tiếp, tên tham số và cài đặt được biểu thị bằng các thanh ghi được hiển thị trong bảng bên dưới.

Địa chỉ tham số	Parameter name and settings				
	Giao tiếp MODBUS (HD 0=1)	Giao tiếp tự do (HD0=2)	Giao tiếp X-NET		Giao tiếp Ethernet (HD0=3)
			OMMS (HD0=3)	TBN (HD0=3)	
HD0	Kiểu mạng 1: MODBUS; 2: free ; 3: X-NET; 4: MODBU-TCP				
HD1	Trạm MODBUS số 1~254	Tốc độ truyền, tham khảo bảng 1	Net ID 0~32767	Net ID 0~32767	Net ID Địa chỉ IP 2 byte cao
HD2	Chế độ truyền 0: RTU 128: ASCII	Định dạng khung, tham khảo bảng 2	Trạm số 0~100	Trạm số 0~100	Địa chỉ trạm Địa chỉ IP 2 byte thấp
HD3	Tốc độ truyền, tham khảo bảng 1	Free properties bit7: 1: with start character 0: no start character bit6: 1: with end character 0: no end character	Kiểu Physical layer type 1: PHY_RS485 2: PHY_SOF(Unidirectional Fiber Ring Network) 3: PHY_OFPP(Optical Fiber Point Network) 4: PHY_RS232 5: PHY_RS422 6: PHY_TTL (TTLvoltage network)		
HD4	Định dạng khung, tham khảo bảng 2	Kí tự bắt đầu	Kiểu Link Layer 0: TBN 1: HDN 2: CCN 3: PPFD 4: PPU 5: Ethernet		
HD5	retry count/ đếm số lần thử lại 0~5	Kí tự kết thúc	Thuộc tính OMMS 128: Hỗ trợ định kỳ giao tiếp, nếu không thì không hỗ trợ	Tốc độ truyền, tham khảo bảng 1	Mật nạ mạng con 2-byte cao
HD6	Thời gian chờ phản hồi 0~65535	Frame timeout 0~255Z	Tốc độ truyền OMMS tham khảo bảng 1	Thời gian chu kì Token 1~60000(ms)	Mật nạ mạng con 2-byte thấp
HD7	Tri hoãn trước khi gửi 0~255	Thời gian chờ phản hồi 0~65535 (0 là Thời gian chờ vô hạn)	Danh sách trạm phụ OMMS Mỗi bit của mỗi byte trong mảng cho biết liệu trạm phụ có thể truy cập được	Số trạm tối đa 1~100	Địa chỉ cổng 2 byte cao

			hay không (trạm chính hợp lệ, tức là số trạm là 1).		
HD8	-	-	-	-	Gateway address low 2-byte

Địa chỉ của "buffer digits" được hiển thị trong Phụ lục 3.

Lưu ý: Bảng không chứa "buffer digits" ở chế độ giao tiếp tự do, do đó "buffer digits" không thể được đọc và ghi thông qua lệnh CFGCR và CFGCW, nhưng có thể được đọc và ghi bằng lệnh MOV. Địa chỉ của "buffer digits" được hiển thị trong Phụ lục 3.

Bảng 1:

Gía trị	Tốc độ truyền	Gía trị	Tốc độ truyền	Gía trị	Tốc độ truyền	Gía trị	Tốc độ truyền
1	300 bps	7	19200 bps	13	256000 bps	19	1000000 bps
2	600 bps	8	28800 bps	14	288000 bps	20	1200000 bps
3	1200 bps	9	38400 bps	15	384000 bps	21	1500000 bps
4	2400 bps	10	57600 bps	16	512000 bps	22	2400000 bps
5	4800 bps	11	115200 bps	17	576000 bps	23	3000000 bps
6	9600 bps	12	192000 bps	18	768000 bps		

Bảng 2: Định dạng khung

Stop bit		Parity bit			Độ dài bit dữ liệu		
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00: 1		000: no			000: 5		
01: 1.5		001: odd			001: 6		
10: 2		010: even			010: 7		
		011: empty			011: 8		
		100: Mask			100: 9		

Chương 7 Chức năng điều khiển PID

Trong chương này chúng tôi chủ yếu giới thiệu các ứng dụng của lệnh PID cho dòng XD, XL, bao gồm: gọi lệnh, thiết lập các tham số, các mục cần chú ý, chương trình mẫu, v.v.

7-1 Giới thiệu về PID

Lệnh PID và chức năng tự động điều chỉnh được thêm vào các bộ PLC cơ bản dòng XD/XL. Thông qua phương pháp tự động điều chỉnh, người dùng có thể biết được thời gian lấy mẫu và thông số PID tốt nhất, đồng thời cải thiện độ chính xác của điều khiển.

Lệnh PID đã mang lại nhiều lợi ích cho người sử dụng.

Đầu ra có thể là dạng dữ liệu D, HD và số lượng trạng thái on-off của Y, người dùng có thể lựa chọn khi lập trình.

Thông qua tính năng tự động điều chỉnh, người dùng có thể có được thời gian lấy mẫu và thông số PID tốt nhất, đồng thời cải thiện độ chính xác của điều khiển.

Người dùng có thể chọn điều khiển theo sai lệch dương hoặc sai lệch âm thông qua cài đặt phần mềm. Sai lệch dương được sử dụng cho điều khiển gia nhiệt. Sai lệch âm được sử dụng cho điều khiển làm mát.

Điều khiển PID tách các thiết bị cơ bản với các phần mở rộng, giúp cải thiện tính linh hoạt của chức năng này.

PLC dòng XD/XL có hai phương pháp tự động điều chỉnh, phương pháp phản hồi bước và phương pháp dao động tới hạn.

Đối với đối tượng điều khiển nhiệt độ:

Phương pháp phản hồi bước: tự động điều chỉnh PID sẽ bắt đầu khi nhiệt độ hiện tại của đối tượng được điều khiển bằng nhiệt độ môi trường.

Phương pháp dao động tới hạn: tự động điều chỉnh PID có thể bắt đầu ở bất kỳ nhiệt độ nào.

7-2 Định dạng lệnh

1) Sơ lược

Thực thi các lệnh điều khiển PID với dữ liệu trong các thanh ghi được chỉ định.

PID control [PID]			
Lệnh 16 bit	PID	Lệnh 32 bit	-
Điều kiện thực thi	Thường mở/đóng, sườn coil	Model phù hợp	XD/XL
Yêu cầu phần cứng	-	Yêu cầu phần mềm	Phiên bản V3.2 trở lên

2) Toán hạng

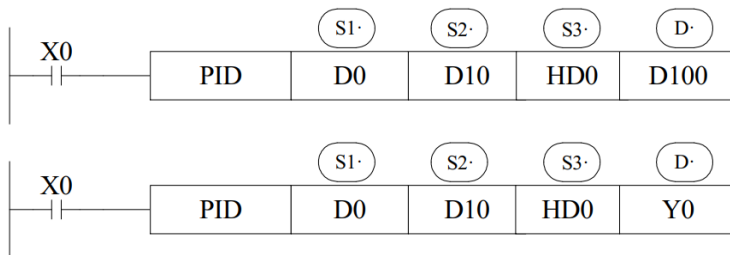
Toán hạng	Chức năng	Kiểu dữ liệu
S1	đặt địa chỉ của giá trị đích (SV)	16bits, BIN
S2	đặt địa chỉ của giá trị phản hồi (PV)	16 bits, BIN
S3	đặt địa chỉ bắt đầu của các tham số điều khiển	16 bits, BIN
D	địa chỉ của kết quả thực thi (MV) hoặc công đầu ra	16 its, BIN; bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word								Phần tử lập trình bit										
	Hệ								Hàng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1	•	•						•											
S2	•	•																	
S3	•	•																	
D	•	•										•	•	•	•	•			

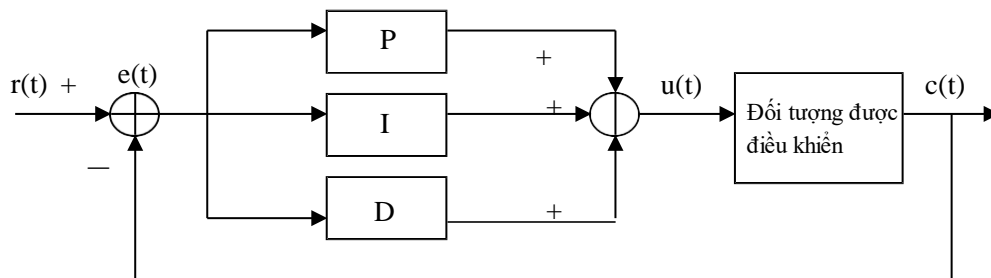
* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hoạt động



- S3~ S3+ 69 sẽ bị lệnh này chiếm dụng, vì vậy vui lòng không sử dụng chúng làm thanh ghi dữ liệu thông thường.
- Lệnh này được thực thi khi mỗi khoảng thời gian lấy mẫu đến.
- Đối với kết quả thực thi, các thanh ghi dữ liệu được sử dụng để lưu trữ các giá trị đầu ra PID; các điểm đầu ra được sử dụng để xuất tỉ lệ phát xung điều chế ở trạng thái ON/OFF.
- Quy tắc điều khiển PID như sau:

P: tỷ lệ, I: tích phân, D: vi phân



Hệ thống điều khiển PID analog

$$e(t) = r(t) - c(t) \quad (1-1)$$

$$u(t) = K_p [e(t) + 1/T_i \int e(t)dt + TD de(t)/dt] \quad (1-2)$$

Trong đó, $e(t)$ là giá trị offset, $r(t)$ là giá trị đặt, $c(t)$ là giá trị đầu ra thực tế và $u(t)$ là giá trị điều khiển;

Trong hàm (1-2), K_p là điều chỉnh tỷ lệ với sai lệch theo thời gian lấy mẫu, T_i là tích phân của sai lệch theo thời gian lấy mẫu và T_D là vi phân của sai lệch theo thời gian lấy mẫu.

Kết quả thực thi:

1. Đầu ra analog: dạng kỹ thuật số của $MV = u(t)$, phạm vi mặc định là 0~4095.
2. Đầu ra kỹ thuật số: $Y = T * [\text{Giới hạn trên đầu ra MV / PID}]$. Y là thời gian kích hoạt đầu ra trong chu kỳ điều khiển. T là chu kỳ điều khiển, bằng thời gian lấy mẫu. Giá trị mặc định giới hạn trên của đầu ra PID là 4095.

7-3 Cài đặt tham số

Người dùng có thể gọi trực tiếp PID trong phần mềm XDP Pro và cài đặt các thông số trong cửa sổ (xem biểu đồ bên dưới), để tìm hiểu chi tiết vui lòng tham khảo hướng dẫn sử dụng XDP Pro. Người dùng cũng có thể ghi các tham số vào các thanh ghi được chỉ định bằng lệnh MOV trước khi vận hành PID.

PID Instruction Parameter Config

Target Value (SV) D0 Measure Value (PV) D10 Parameter: HD0 Output: Y0

Parameter Config

Manual Auto

Sampling Time : 0 ms

Proportion Gain (KP): 0 %

Integration Time(TI): 0 *100ms

Differential Time(TD): 0 *10ms

PID Computation Scope: 0

PID Control Death Band: 0

Self Study Periodic Value: 0

Self Study Method: Step Response

Self Study PID Control Mode: PID Control

Overshoot Config

Enable Overshoot Disable Overshoot

Each time adjust the increase: 100 %

Current target value resident Count: 15

Suggestion value

Mode Config

Common Mode Advanced Mode

Input Filter Constant (a): 0 %

Differential Increase (KD): 50 %

Output Upper Limit Value: 4095

Output Lower Limit Value: 0

Direction Config

Negative Movement Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Parameter Range: HD0 - HD69

Read From PLC Write To PLC OK Cancel

Chế độ tự động điều chỉnh:

tudonghoatruaicau.com

PID Instruction Parameter Config
✕

Target Value (SV)

Measure Value(PV)

Parameter:

Output:

Parameter Config

Manual Auto

Sampling Time : ms

Proportion Gain (KP): %

Integration Time(TI): *100ms

Differential Time(TD): *10ms

PID Computation Scope:

PID Control Death Band:

Self Study Periodic Value:

Self Study Method:

Self Study PID Control Mode:

Mode Config

Common Mode Advanced Mode

Input Filter Constant (a): %

Differential Increase (KD): %

Output Upper Limit Value:

Output Lower Limit Value:

Direction Config

Negative Movement Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Overshoot Config

Enable Overshoot Disable Overshoot

Each time adjust the increase: %

Current target value resident Count:

Suggestion value

Read From PLC
Write To PLC
OK
Cancel

Phần mềm phiên bản V3.2 trở lên có thể chọn chế độ tự động điều chỉnh: phân hồi từng bước hoặc dao động tới hạn.

7-3-1 Thanh ghi và chức năng

ID tham số tương đối của lệnh điều khiển PID, vui lòng tham khảo bảng dưới đây:

ID	Chức năng	Mô tả	Ghi chú
S3	Thời gian lấy mẫu	Dù là chế độ thủ công hay tự động đều phải thiết lập tham số	32bit không dấu, Đơn vị: ms
S3+2	Cài đặt chế độ	bit0: 0: sai lệch âm 1: sai lệch dương bit1~bit6 không thể sử dụng bit7: 0: PID thủ công; 1: PID tự động điều chỉnh bit8: 1: cờ tự động điều chỉnh thành công bit9~bit10: tự động điều chỉnh 00: phân hồi bước 01: dao động quá hạn bit11~bit12: không hữu dụng bit13~bit14 chế độ PID tự động điều chỉnh (hợp lệ trong phương pháp dao động quá hạn)	

		mode) 00: PID control 01: PI control 10: P control bit15: 0: regular mode; 1: advanced mode;	
S3+3	Độ lợi tỉ lệ (Kp)	0~32767[%]	
S3+4	Tích phân của sai lệch theo thời gian lấy mẫu (TI)	0~32767[unit: 100ms]	0 được coi là không có tích phân.
S3+5	Vi phân của sai lệch theo thời gian lấy mẫu (TD)	0~32767[unit: 10ms]	0 được coi là không có vi phân.
S3+6	Dải biên vận hành PID	0~32767	Giá trị độ rộng dải điều chỉnh PID
S3+7	Điều khiển vùng chết	0~32767	Giá trị đầu ra PID sẽ không thay đổi trong vùng chết
S3+8	Hệ số lọc nhiệt độ lấy mẫu	0~100[%]	Lọc nhiệt độ lấy mẫu đầu vào ở chế độ nâng cao, 0 là không có bộ lọc đầu vào
S3+9	Độ lợi vi sai (KD)	0~100[%]	Chỉ dành cho chế độ nâng cao (giá trị mặc định của chế độ bình thường là 50%), 0 không có độ lợi vi sai
S3+10	Giá trị giới hạn trên của đầu ra	0~32767	
S3+11	Giá trị giới hạn dưới của đầu ra	0~32767	
S3+12	Change of Unit Temperature Corresponds to Change of AD Value	Giá trị AD full-scale*(0,3~1%) giá trị mặc định là 10	16-bit không có dấu, chỉ dành cho bước PID
S3+13	PID auto tune overshoot Tự động điều chỉnh PID vọt lố	0: cho phép vọt lố 1: không vọt lố (giảm độ vọt lố)	Chỉ dành cho step PID
S3+14	Dải % giá trị đặt điều chỉnh trong chế độ Auto tuning	Không thể chỉnh sửa	16-bit không có dấu, chỉ dành cho step PID
S3+15	Số lần quá lượng đặt mục tiêu trong auto tuning khi đạt giới hạn quá dải điều chỉnh		Chỉ dành cho step PID , giá trị mặc định là 15
S3+16	Loại và trạng thái PID	Bit0~bit1: 00: chế độ thủ công 01: chế độ step mode 10: Chế độ dao động tới hạn Bit8: 0: trạng thái điều khiển thủ công 1: tự động điều chỉnh kết thúc, đi đến trạng thái điều khiển thủ công	Các thông số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát

S3+17	Đầu ra tối đa PID	0~32767	Các thông số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát
S3+18	Đầu ra tối thiểu PID	0~32767	Các thông số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát
S3+19	Thời gian lấy mẫu lần cuối	0~thời gian lấy mẫu(đơn vị: ms)	16-bit không có dấu, Các tham số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát
S3+20	Khoảng thời gian lấy mẫu thực tế	Giá trị xung quanh thời gian lấy mẫu	32 bit không dấu, Các tham số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát
S3+22	Lần cuối cùng người dùng đặt nhiệt độ mục tiêu	Giá trị trước khi thay đổi nhiệt độ mục tiêu	Các tham số sử dụng nội bộ của hệ thống chỉ nhằm mục đích giám sát
S3+23	-	-	Tham số được bảo lưu

Sau đây là địa chỉ chung (được chia thành cài đặt bước, cài đặt dao động tới hạn và điều khiển thủ công)

Phần bước phản hồi/ Step part (thông số chỉ đọc, chỉ để theo dõi)

S3+24	Actual sampling space	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+26	Phân đoạn vận hành của PID tự động điều chỉnh	0: Giai đoạn chuẩn bị 1~2: thu thập tham số tự động điều chỉnh 3: tính toán các thông số PID	Thông số sử dụng nội bộ của hệ thống
S3+28	Thời gian tự động điều chỉnh các thông số vận hành PID	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+30	Tích lũy thời gian thực của hai điểm uốn	Xóa và tính toán lại thời gian đạt điểm uốn 0~4294967296(đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+32	Lấy mẫu biến thể của điểm uốn	Lấy mẫu chênh lệch giữa hai điểm uốn- 2147483648~2147483647	Thông số sử dụng nội bộ của hệ thống
S3+34	Khoảng thời gian lấy mẫu của điểm uốn EK	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+36	Thời gian từ khi tự động điều chỉnh PID đến điểm uốn	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống

S3+38	Nhiệt độ lấy mẫu lần cuối	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+39	Thời gian từ khi vận hành PID tự động điều chỉnh đến điểm uốn	-32767~32767 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+40	Giá trị lấy mẫu bắt đầu của vận hành PID tự động điều chỉnh	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+41	Số lần tại điểm uốn trong quá trình tự động điều chỉnh	0~65535	Thông số sử dụng nội bộ của hệ thống
S3+42	Thời gian vô ích	0~4294967296 (unit: ms)	Thông số sử dụng nội bộ của hệ thống
S3+44	Nhiệt độ kết thúc	Nhiệt độ khi kết thúc quá trình tự động điều chỉnh Phạm vi:-32767~32767	Thông số sử dụng nội bộ của hệ thống
Phần dao động tới hạn (các thông số chỉ đọc, chỉ để theo dõi)			
S3+24	Chế độ điều khiển PID	0: Điều khiển PID 1: Điều khiển PI 2: Điều khiển P	16-bit không có dấu hiệu, Thông số sử dụng nội bộ của hệ thống
S3+25	Đoạn tự động điều chỉnh hiện tại	0: Chuẩn bị 1: bắt đầu điều chỉnh tự động 2~3: thu thập tham số tự động điều chỉnh 4: tính toán thông số PID	16-bit không có dấu hiệu, Thông số sử dụng nội bộ của hệ thống
S3+26	Nhiệt độ tự động điều chỉnh nằm ở đỉnh số	0: đỉnh 1 1: đỉnh 2	16-bit không có dấu hiệu, Thông số sử dụng nội bộ của hệ thống
S3+27	Nhiệt độ lấy mẫu thấp nhất	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+28	Nhiệt độ lấy mẫu cao nhất	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+30	thời gian lấy mẫu của nhiệt độ lấy mẫu thấp nhất	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+32	thời gian lấy mẫu của nhiệt độ lấy mẫu cao nhất	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+34	auto-tuning time cumulative	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
Phần điều khiển thủ công(thông số chỉ đọc, chỉ để theo dõi)			
S3+24	nhiệt độ mục tiêu hiện tại	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+25	Cần cập nhật nhiệt độ mục tiêu	0: không cần 1: cần	16-bit không có dấu hiệu,

			Thông số sử dụng nội bộ của hệ thống
S3+26	Số lần đạt được nhiệt độ mục tiêu	0~65535	Thông số sử dụng nội bộ của hệ thống
S3+27	Giới hạn trên của phạm vi hoạt động PID	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+28	Giới hạn dưới của phạm vi hoạt động PID	-32767~32767	Thông số sử dụng nội bộ của hệ thống
S3+30	Thời gian điện áp cao khi PID sử dụng Y đề xuất	0~4294967296 (đơn vị: ms)	Thông số sử dụng nội bộ của hệ thống
S3+32	Nhiệt độ lấy mẫu sau lần lọc cuối cùng	Nhiệt độ đã lọc thu được trong lần lấy mẫu cuối cùng (hàng số bộ lọc đầu vào ở chế độ nâng cao cần được đặt trước)	Số thực dấu phẩy động, thông số sử dụng nội bộ của hệ thống
S3+34	Độ lệch nhiệt độ cuối cùng		Số thực dấu phẩy động, thông số sử dụng nội bộ của hệ thống
S3+36	Giá trị của số hạng tích phân cuối cùng	giá trị số tương ứng với Ui của lần lấy mẫu cuối cùng	Số thực dấu phẩy động, thông số sử dụng nội bộ của hệ thống
S3+38	Giá trị của số hạng vi phân cuối cùng	giá trị số tương ứng với Ud của lần lấy mẫu cuối cùng	Số thực dấu phẩy động, thông số sử dụng nội bộ của hệ thống
S3+40	Đầu ra PID cuối cùng		Số thực dấu phẩy động, thông số sử dụng nội bộ của hệ thống

Lưu ý: Khi chế độ điều khiển tự động được đặt thành điều khiển thủ công, giá trị trong địa chỉ ban đầu của S3+24~S3+40 sẽ bị ghi đè bởi giá trị trong chế độ điều khiển thủ công.

7-3-2 Mô tả các thông số

Hướng tác động điều khiển:

Hướng điều khiển tác động dương: giá trị đầu ra MV sẽ tăng khi giá trị đo được PV tăng, thường được sử dụng để điều khiển làm mát.

Hướng điều khiển tác động âm: giá trị đầu ra MV sẽ giảm khi tăng giá trị đo được PV, thường được sử dụng để điều khiển nhiệt độ.

Cài đặt chế độ

Chế độ bình thường

Phạm vi thanh ghi tham số: S3~S3+69 và S3~S3+7 cần được người dùng đặt;

S3+8~S3+69 bị hệ thống chiếm dụng, người dùng không thể sử dụng được.

Chế độ nâng cao

Phạm vi thanh ghi tham số: S3~S3+69, trong đó S3~S3+7 và S3+8~S3+11 cần được người dùng đặt; S3+16~S3+69 bị hệ thống chiếm dụng, người dùng không thể sử dụng được.

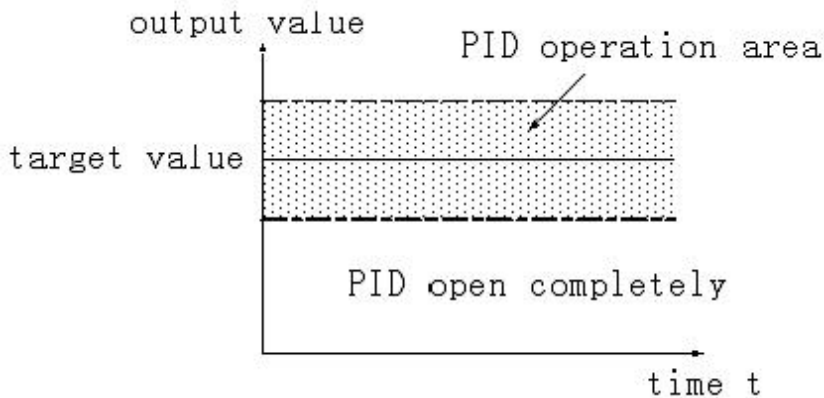
Thời gian lấy mẫu [S3]

Hệ thống lấy mẫu các giá trị hiện tại theo một khoảng thời gian nhất định và so sánh chúng với giá trị đầu ra. Khoảng thời gian này là thời gian lấy mẫu **T**. Không có yêu cầu về **T** trong quá trình xuất **DA**; **T** phải lớn hơn một chu kỳ quét PLC trong quá trình xuất công. Giá trị **T** nên được chọn trong khoảng 100 ~ 1000 lần chu kỳ quét PLC.

Dải biên vận hành PID [S3+6]

Điều khiển PID mở hoàn toàn ở đầu và gần giá trị đích với tốc độ cao nhất (giá trị mặc định là 4095), khi đưa vào dải tính toán PID, các thông số K_p , T_I , T_D sẽ có hiệu lực.

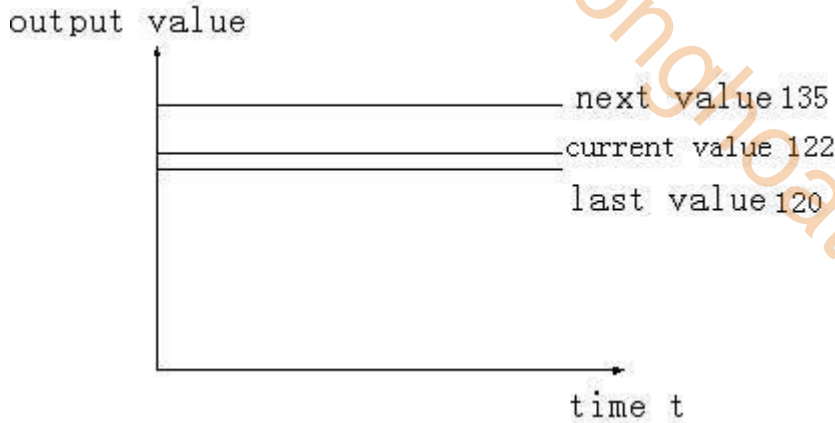
Xem biểu đồ dưới đây:



Nếu giá trị mục tiêu là 100, vùng vận hành của PID là 10, thì vùng vận hành thực của PID là từ 90~110.

Vùng bỏ qua sai lệch (vùng chết của điều khiển) [S3+7]

Nếu giá trị đo được thay đổi một chút trong một thời gian dài và điều khiển PID vẫn ở chế độ làm việc thì nó thuộc về điều khiển chết (vô dụng). Thông qua việc thiết lập **Vùng chết của điều khiển**, chúng ta có thể khắc phục được tình trạng này. Xem biểu đồ dưới đây:



Giả sử: chúng ta thấy giá trị death region là 10. Khi đó trong biểu đồ trên, sai số chỉ là 2 khi so sánh giá trị hiện tại với giá trị cuối cùng. Nó sẽ không thực hiện điều khiển PID; sai số là 13 (nhiều hơn vùng death region 10) so sánh giá trị hiện tại với giá trị tiếp theo, giá trị sai số này lớn hơn giá trị vùng death region, nó sẽ thực hiện điều khiển PID với 135.

7-4 Chế độ điều chỉnh tự động

Nếu người dùng không biết cách cài đặt các tham số PID, họ có thể chọn chế độ tự động điều chỉnh để có thể tự động tìm các tham số điều khiển tốt nhất (thời gian lấy mẫu, độ lợi tỷ lệ K_p , tích phân của sai lệch theo thời gian lấy mẫu T_i , vi phân của sai lệch theo thời gian lấy mẫu T_D).

Chế độ Auto tune phù hợp với các đối tượng được điều khiển: nhiệt độ, áp suất; không phù hợp với mức chất lỏng và dòng chảy

Auto-tuning là quá trình trích xuất các thông số PID. Đôi khi việc tự động điều chỉnh không thể tìm thấy các thông số tốt nhất cùng một lúc. Nó cần tự động điều chỉnh nhiều lần. Việc xảy ra những thay đổi trong quá trình này là điều bình thường. Sau khi tìm thấy các thông số tối ưu ở cuối quá trình tự động điều chỉnh, vui lòng chuyển sang chế độ PID thủ công. Nếu đối tượng điều khiển không ổn định trong quá trình PID thủ công thì không thể điều khiển được ở giá trị mục tiêu không đổi, nguyên nhân có thể là do việc điều chỉnh các tham số không đạt yêu cầu. Cần phải điều chỉnh lại các thông số của PID để đạt được điều khiển ổn định.

Đối với phương pháp phản hồi bước: Người dùng có thể đặt chu kỳ lấy mẫu về 0 khi bắt đầu quá trình tự động điều chỉnh, sau đó sửa đổi giá trị theo cách thủ công theo nhu cầu thực tế sau khi quá trình tự động điều chỉnh hoàn tất.

Đối với phương pháp phản hồi theo bước: Trước khi thực hiện tự động điều chỉnh, hệ thống phải ở trạng thái ổn định không điều khiển. Lấy nhiệt độ làm ví dụ: nhiệt độ đo được phải bằng nhiệt độ môi trường.

Đối với phương pháp phản hồi bước: Trước khi thực hiện tự động điều chỉnh, hệ thống phải ở trạng thái ổn định không điều khiển. Lấy nhiệt độ làm ví dụ: nhiệt độ đo được phải bằng nhiệt độ môi trường.

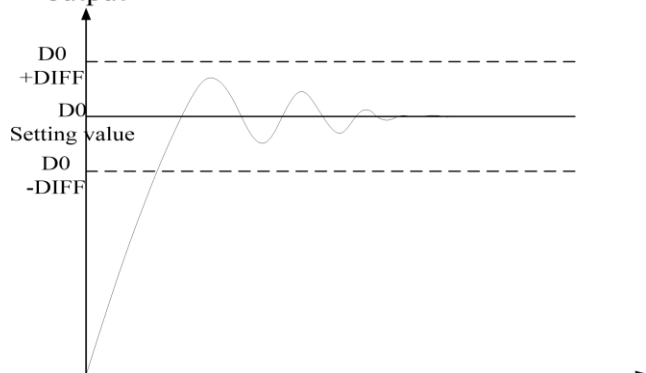
Đối với phương pháp dao động tới hạn: người dùng cần đặt thời gian lấy mẫu khi bắt đầu quá trình điều khiển tự động. Đối với hệ thống phản hồi chậm, đặt là 1000ms. Đối với hệ thống phản hồi nhanh, đặt là 10-100ms.

Đối với phương pháp dao động tới hạn: hệ thống có thể bắt đầu điều khiển tự động ở bất kỳ trạng thái nào. Đối với nhiệt độ của đối tượng, nhiệt độ hiện tại không cần phải giống với nhiệt độ môi trường.

Hai phương pháp khác nhau và sơ đồ điều khiển PID:

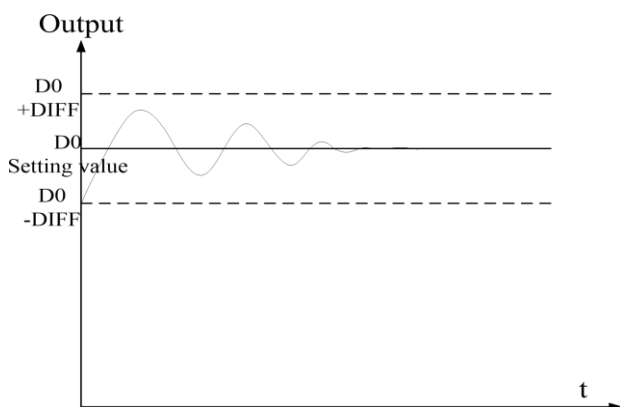
(1) Phương pháp phản hồi bước

Đảm bảo nhiệt độ hiện tại bằng nhiệt độ môi trường
output



(2) Phương pháp dao động tới hạn

Nhiệt độ bắt đầu điều khiển tự động có thể là bất kỳ giá trị nào.



Để sử dụng chế độ tự động điều chỉnh, vui lòng đặt bit7 của (S3+ 2) thành 1 và bật điều kiện vận hành của PID. Nếu bit8 của (S3+ 2) chuyển sang 1, điều đó có nghĩa là việc điều khiển tự động thành công.

Giá trị chu kỳ điều chỉnh tự động PID [S3+12]

Đặt giá trị này trong S3+12 trong khi điều khiển tự động. Giá trị này quyết định hiệu suất tự động điều chỉnh, nói chung, đặt giá trị này là kết quả AD tương ứng với một thiết bị được kiểm tra tiêu chuẩn. Giá trị mặc định là 10. Phạm vi cài đặt được đề xuất: fall-scale AD result \times 0.3~1%.

Người dùng không cần thay đổi giá trị này. Tuy nhiên, nếu hệ bị ảnh hưởng nhiều bởi bên ngoài thì giá trị này cần được tăng lên một cách vừa phải để tránh phán đoán sai về chuyển động dương và âm. Nếu giá trị này quá lớn, khoảng thời gian điều khiển PID (thời gian lấy mẫu) thu được từ quá trình điều khiển tự động sẽ quá dài. Vì vậy, không đặt giá trị này quá lớn.

※1: Nếu người dùng không có kinh nghiệm, vui lòng sử dụng giá trị mặc định 10, đặt thời gian lấy mẫu PID (khoảng thời gian điều khiển) thành 0ms sau đó bắt đầu điều chỉnh tự động.

Cài đặt cho phép vượt quá mức trong điều chỉnh tự động PID [S3+13]

Nếu được đặt bằng 0, thì cho phép quá mức và hệ thống có thể tìm các tham số PID tối ưu mọi lúc. Nhưng trong quá trình tự động điều chỉnh, giá trị được phát hiện có thể thấp hơn hoặc cao hơn giá trị mục tiêu, hệ số an toàn cần được xem xét ở đây.

Nếu đặt 1, việc vượt quá mức không được phép. Đối với những mục tiêu có yêu cầu khắt khe về an toàn như bình chịu áp lực. Đặt [S3+13] thành 1 để tránh giá trị được kiểm tra vượt quá giá trị mục tiêu.

Trong quá trình này, nếu bit8 [S3+2] thay đổi từ 0 thành 1, điều đó có nghĩa là quá trình điều chỉnh tự động thành công và đạt được các tham số tối ưu; nếu [S3+2] bit8 giữ nguyên 0, khi [S3+2] bit7 thay đổi từ 1 thành 0, điều đó có nghĩa là việc tự động điều chỉnh đã hoàn tất, nhưng các tham số không phải là tối ưu nhất và chúng cần được điều chỉnh thủ công.

Dải % giá trị đặt điều chỉnh trong chế độ Auto tuning (Điều chỉnh tự động) [S3+14]

Tham số này chỉ hợp lệ khi [S3+13] bằng 1.

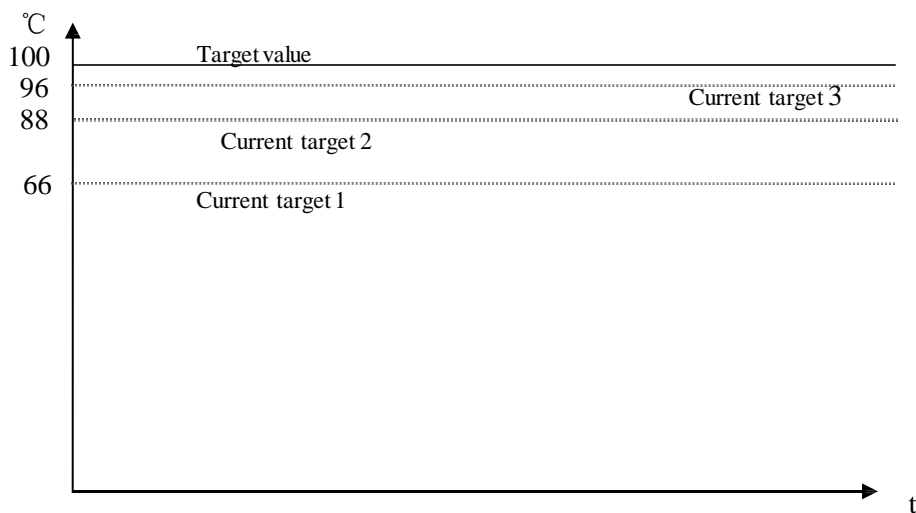
Nếu thực hiện điều khiển PID sau khi tự động điều chỉnh, có thể xảy ra khoảng vượt mức nhỏ. Tốt hơn là giảm tham số này để kiểm soát việc vượt mức. Nhưng độ trễ phản hồi có thể xảy ra nếu giá trị này quá nhỏ. Giá trị mặc định là 100% có nghĩa là tham số không hiệu quả. Phạm vi khuyến nghị là 50~80%.

Giải thích thích:

Phần trăm điều chỉnh giá trị mục tiêu hiện tại là $2/3(S3 + 14 = 67\%)$, nhiệt độ ban đầu của hệ thống là $0\text{ }^{\circ}\text{C}$, nhiệt độ mục tiêu là $100\text{ }^{\circ}\text{C}$ và tình huống điều chỉnh nhiệt độ mục tiêu hiện tại được hiển thị như sau:

Next current target value = current target value + (final target value – current target value) \times 2/3;

Vì vậy trình tự thay đổi của mục tiêu hiện tại là $66\text{ }^{\circ}\text{C}$, $88\text{ }^{\circ}\text{C}$, $96\text{ }^{\circ}\text{C}$, $98\text{ }^{\circ}\text{C}$, $99\text{ }^{\circ}\text{C}$, $100\text{ }^{\circ}\text{C}$.



Giá trị hệ thống hiện tại

Số lần quá lượng đặt mục tiêu trong auto tuning khi đạt giới hạn quá dải điều chỉnh [S3+15]

Tham số này chỉ hợp lệ khi [S3+13] bằng 1;

Nếu nhập trực tiếp vào điều khiển PID sau khi tự động điều chỉnh, có thể xảy ra mức vọt lố nhỏ. Nên tăng thông số này một cách hợp lý để tránh hiện tượng vọt lố. Nhưng nó sẽ gây ra độ trễ phản hồi nếu giá trị này quá lớn. Giá trị mặc định là 15 lần. Phạm vi được đề xuất là từ 5 đến 20.

7-5 Chế độ nâng cao

Người dùng có thể thiết lập một số thông số ở chế độ nâng cao để có được hiệu quả điều khiển PID tốt hơn. Để sử dụng chế độ nâng cao, vui lòng đặt bit [S3+2] 15 thành 1 hoặc đặt nó trong phần mềm XDP Pro.

Hằng số bộ lọc đầu vào khi lấy mẫu [S3+8]

Nó sẽ làm mịn giá trị lấy mẫu. Giá trị mặc định là 0%, nghĩa là không có bộ lọc.

Độ lợi vi phân [S3+9]

Quá trình lọc low-pass sẽ giảm bớt sự thay đổi đột ngột của giá trị đầu ra. Giá trị mặc định là 50%; hiệu quả sẽ rõ ràng hơn nếu tăng giá trị này. Người dùng không cần phải thay đổi nó.

Giá trị giới hạn trên và giới hạn dưới [S3+10], [S3+11]

Người dùng có thể chọn phạm vi đầu ra analog thông qua việc cài đặt giá trị này.

Giá trị mặc định: Giới hạn dưới của đầu ra =0

Giới hạn trên của đầu ra =4095

7-6 Tóm tắt ứng dụng

Trong trường hợp đầu ra liên tục, hệ thống có khả năng tác động sẽ ngừng hoạt động khi thay đổi giá trị phản hồi có thể tự động điều chỉnh, chẳng hạn như nhiệt độ hoặc áp suất. Nó không phù hợp với thông lượng hoặc mức chất lỏng.

Trong điều kiện cho phép vượt mức, hệ thống sẽ nhận được thông số PID tối ưu từ việc tự động điều chỉnh.

Trong điều kiện không cho phép vượt mức, các tham số PID nhận được từ chế độ tự động điều chỉnh đạt giá trị mục tiêu, nghĩa là giá trị mục tiêu khác nhau sẽ tạo ra các tham số PID khác nhau, không phải là tham số tối ưu của hệ thống và chỉ mang tính chất tham khảo.

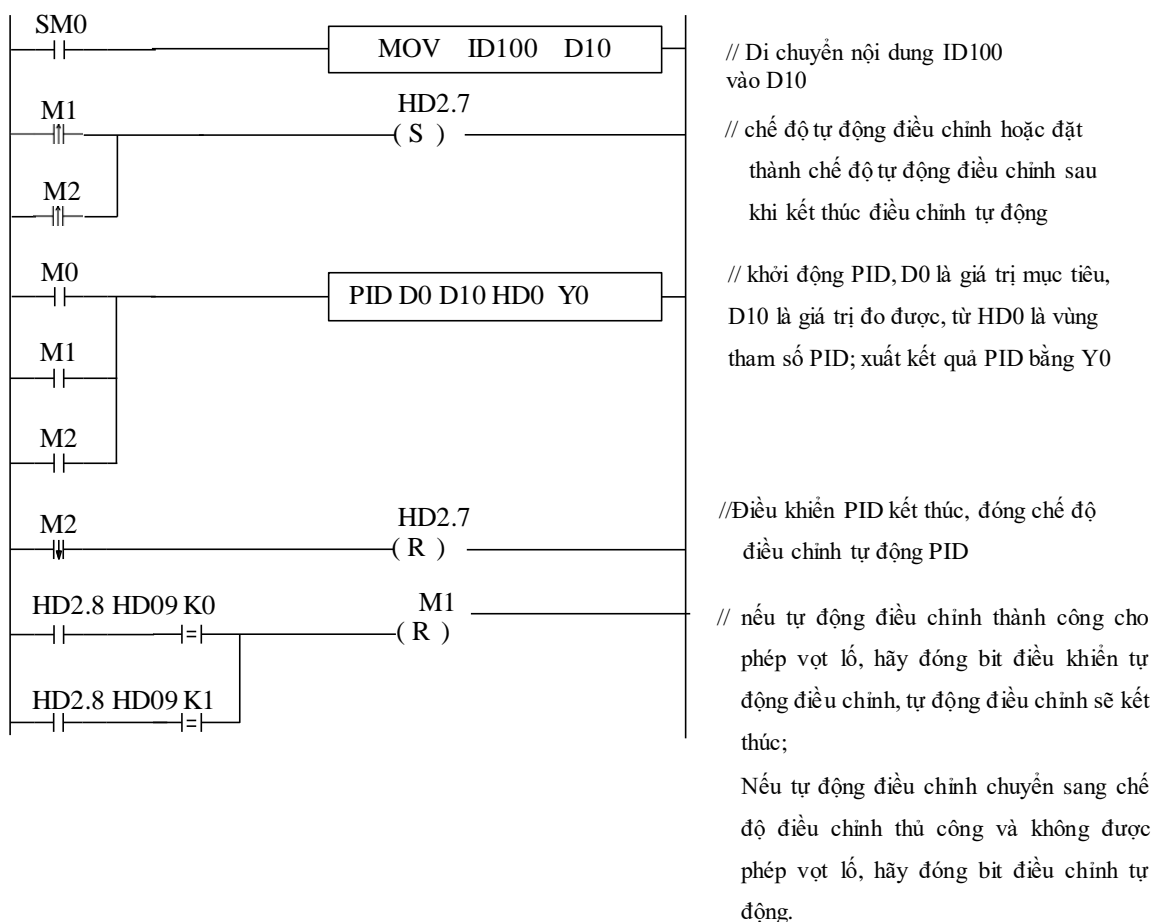
Nếu không có tính năng tự động điều chỉnh, người dùng có thể thiết lập các thông số PID theo thực hành thực tế. Người dùng cần sửa đổi các tham số khi gỡ lỗi. Dưới đây là một số giá trị theo trải nghiệm của hệ thống điều khiển để bạn tham khảo:

- Hệ thống nhiệt độ: P (%) 2000 ~ 6000, I (phút) 3 ~ 10, D (phút) 0,5 ~ 3
- Hệ thống thông lượng: P (%) 4000 ~ 10000, I (phút) 0,1 ~ 1
- Hệ thống áp suất: P (%) 3000 ~ 7000, I (phút) 0,4 ~ 3
- Hệ thống mức chất lỏng: P (%) 2000 ~ 8000, I (phút) 1 ~ 5

7-7 Ví dụ về PID

Ví dụ 1:

Chương trình điều khiển PID được hiển thị dưới đây:



Ghi chú về chức năng của phần tử lập trình:

HD2.7: bit điều chỉnh tự động

HD2.8: Cờ tự động điều chỉnh thành công

M0: Điều khiển PID bình thường

M1: Điều khiển điều chỉnh tự động

M2: Nhập điều khiển PID sau khi tự động điều chỉnh

Các bước thao tác

1. Gửi dữ liệu nhiệt độ thực tế tới thanh ghi thu thập PID
2. Đặt giá trị có thể cho P, I, D, chu kỳ lấy mẫu
3. Đặt bit điều khiển tự động điều chỉnh M1 ở trạng thái ON để khởi động điều chỉnh tự động PID
4. M1 sẽ được đặt lại sau khi quá trình điều chỉnh tự động kết thúc
5. Đặt M0 ở trạng thái ON, sử dụng các tham số PID nhận được từ chế độ tự động điều chỉnh
6. Nếu hiệu quả PID khi sử dụng các tham số PID tự động điều chỉnh không cao, người dùng có thể điều chỉnh các tham số PID để có hiệu quả tốt.

Lưu ý: Chương trình điều khiển PID nhiệt độ của PLC này có thể áp dụng cho hầu hết các dự án điều khiển nhiệt độ.

Ví dụ 2:

Kiểm soát nhiệt độ mục tiêu 60°C ở chế độ phản hồi bước.

Cho phép vượt mức

1. Nhiệt độ mục tiêu 60°C (600)
2. Cài đặt thông số

PID Instruction Parameter Config

Target Value (SV): Measure Value(PV): Parameter: Output:

Parameter Config

Manual Auto

Sampling Time: ms

Proportion Gain (KP): %

Integration Time(TI): *100ms

Differential Time(TD): *10ms

PID Computation Scope:

PID Control Death Band:

Self Study Periodic Value:

Self Study Method:

Self Study PID Control Mode:

Mode Config

Common Mode Advanced Mode

Input Filter Constant (a): %

Differential Increase (KD): %

Output Upper Limit Value:

Output Lower Limit Value:

Direction Config

Negative Movement Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Parameter Range: D4000 - D4069

Overshoot Config

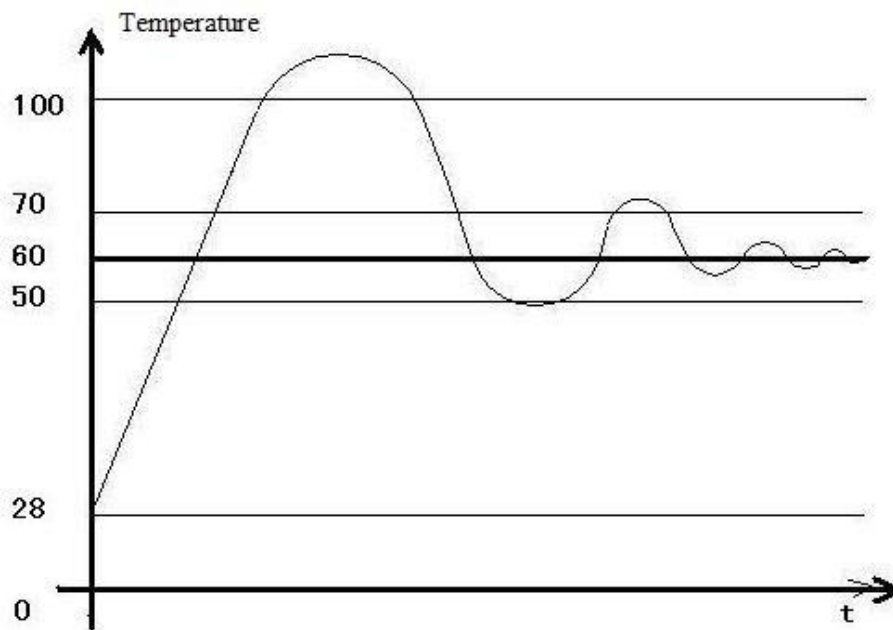
Enable Overshoot Disable Overshoot

Each time adjust the increase: %

Current target value resident Count:

Suggestion value

3. Đường cong kết quả



Giải thích:

Nhiệt độ mục tiêu là 60 độ, phạm vi tính toán PID là 10 độ, vùng dead area của điều khiển PID là 0,2 độ, giá trị thay đổi chu kỳ tự động điều chỉnh là 10. Khi điều khiển PID hoạt động ở nhiệt độ khí quyển bình thường, đầu ra PID sẽ làm nóng nhiệt độ từ 28 đến 100 độ thì đầu ra dừng lại, nhiệt độ tiếp tục tăng lên 110 độ (nhiệt độ tối đa). Sau đó, nhiệt độ tiếp tục giảm xuống 60 độ, đầu ra bắt đầu nóng trở lại đến 70 độ và dừng lại. Nhiệt độ tăng một chút rồi lại giảm. Quá trình này sẽ lặp lại. Cuối cùng, nhiệt độ sẽ dao động, đóng nhiệt độ mục tiêu.

Lưu ý:

1. Khi nhiệt độ đạt đến 100 độ và ngừng gia nhiệt, bit khởi động PID D4002.7 sẽ không được đặt lại ngay lập tức, mà có độ trễ trước khi đặt lại.
2. Khi nhiệt độ đạt tới 100 độ và ngừng gia nhiệt, bit điều chỉnh tự động thành công PID D4002.8 sẽ ở trạng thái ON ngay lập tức.
3. Khi bắt đầu tính toán PID, PLC sẽ tự động đặt thời gian lấy mẫu (khoảng 2500). Thông số này sẽ được thay thế bằng thời gian lấy mẫu tốt nhất của PID sau khi dừng gia nhiệt ở 100 độ.
4. Khi bắt đầu tính toán PID, PLC sẽ tự động thiết lập các thông số PID (P=4454, I=926, D=2317). Các thông số này sẽ được thay thế bằng giá trị PID tốt nhất sau khi dừng gia nhiệt ở 100 độ.
5. Khi nhiệt độ đạt đến 100 độ và ngừng gia nhiệt, bit khởi động PID D4002.7 sẽ không được đặt lại ngay lập tức, nó có độ trễ trước khi đặt lại. Tại thời điểm này, nhiệt độ lấy mẫu cao hơn nhiệt độ mục tiêu. Nếu người dùng đặt chế độ tự động điều chỉnh PID ở trạng thái ON trở lại, PLC sẽ nhận tất cả các tham số PID là 0. Vui lòng đặt trạng PID ở trạng thái ON sau khi nhiệt độ giảm xuống dưới nhiệt độ khí quyển bình thường.
6. Nếu bit khởi động điều chỉnh tự động PID và bit tự động điều chỉnh thành công có khả năng lưu giữ khi tắt nguồn, vui lòng đặt hoặc đặt lại chúng cho phù hợp để tránh lỗi tính toán khi khởi động PLC lần sau.
7. Nhiệt độ gia nhiệt cuối cùng sẽ lên tới 110 độ khi cho phép vọt lố. Nó vượt quá nhiệt độ mục tiêu 50 độ, độ vọt lố quá lớn.
8. Khi PID bắt đầu hoạt động, đầu ra sẽ làm nóng đối tượng từ 28 độ đến 60 độ, sau đó đầu ra buộc phải dừng gia nhiệt để tránh quá nhiệt, nhưng điều này sẽ làm gián đoạn quá trình tự động điều chỉnh PID.
9. Mở rộng phạm vi tính toán PID có thể ngăn chặn sự vọt lố nhiệt độ.

Không được phép vọt lố

1. Nhiệt độ mục tiêu 60°C (600)
2. Cài đặt thông số liên quan:

PID Instruction Parameter Config

Target Value (SV): Measure Value(PV): Parameter: Output:

Parameter Config

Manual Auto

Sampling Time: ms

Proportion Gain (KP): %

Integration Time(TI): *100ms

Differential Time(TD): *10ms

PID Computation Scope:

PID Control Death Band:

Self Study Periodic Value:

Self Study Method:

Self Study PID Control Mode:

Mode Config

Common Mode Advanced Mode

Input Filter Constant (a): %

Differential Increase (KD): %

Output Upper Limit Value:

Output Lower Limit Value:

Direction Config

Negative Movement Positive Movement

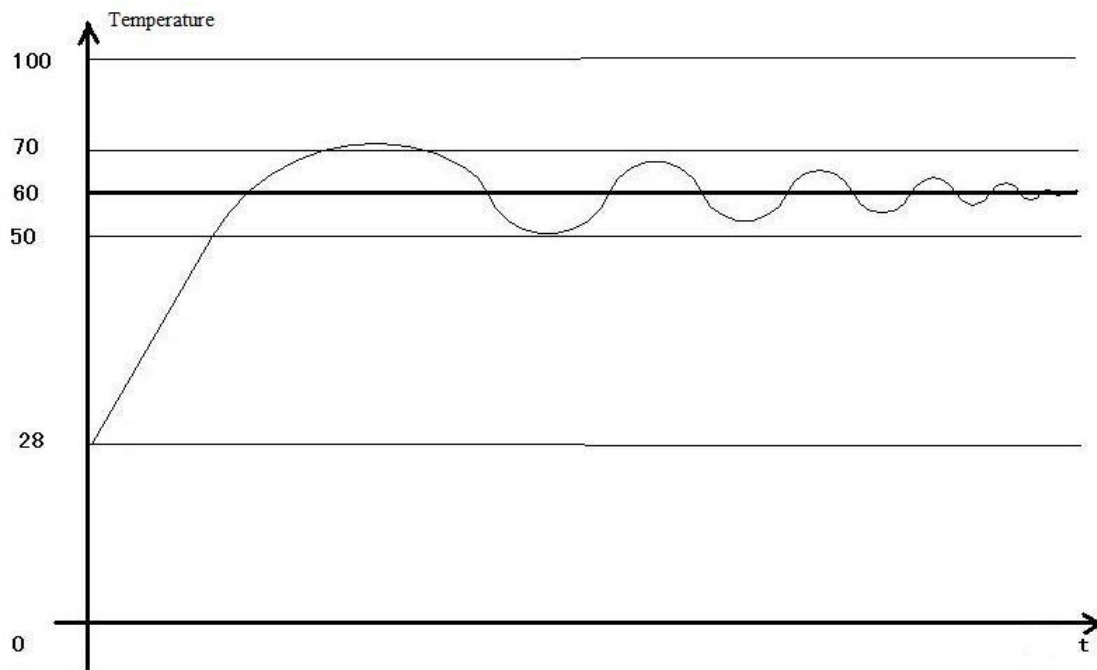
Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Parameter Range: D4000 - D4069

Suggestion value

3. Đường cong kết quả



Giải thích:

Nhiệt độ mục tiêu là 60 độ, phạm vi tính toán PID là 10 độ, vùng chết điều khiển PID là 0,2 độ, giá trị thay đổi chu kỳ tự động điều chỉnh là 10. Khi điều khiển PID hoạt động ở nhiệt độ khí quyển bình thường, đầu ra PID sẽ làm nóng nhiệt độ từ 28 đến 48 độ, sau đó đầu ra dừng lại, nhiệt độ tiếp tục tăng lên 70 độ (nhiệt độ tối đa). Sau đó, nhiệt độ tiếp tục giảm xuống 60 độ, đầu ra bắt đầu nóng trở lại đến 62 độ và dừng lại. Nhiệt độ tăng lên một chút (khoảng 64 độ) rồi lại giảm. Quá trình này sẽ lặp lại. Cuối cùng, nhiệt độ sẽ dao động, đóng nhiệt độ mục tiêu. Độ chính xác là $\pm 0,25$ độ.

Lưu ý:

1. Khi nhiệt độ đạt tới 48 độ và ngừng gia nhiệt, bit khởi động PID D4002.7 sẽ không được đặt lại ngay lập tức, nó có độ trễ trước khi đặt lại.
2. Khi nhiệt độ đạt đến 48 độ và ngừng gia nhiệt, bit tự động điều chỉnh thành công PID D4002.8 sẽ không ở trạng thái ON ngay lập tức. Nó không ở trạng thái ON ngay cả khi điều chỉnh tự động thành công.
3. Khi bắt đầu tính toán PID, PLC sẽ tự động đặt thời gian lấy mẫu (khoảng 2500). Thông số này sẽ được thay thế bằng thời gian lấy mẫu tối ưu nhất của PID sau khi dừng gia nhiệt ở 48 độ.
4. When it starts PID calculation, the PLC will auto set the PID parameters (P=4454, I=926, D=2317). These parameters will be replaced by the best PID value after stoping heating at 48 degree.
5. Khi bắt đầu tính toán PID, PLC sẽ tự động thiết lập các thông số PID (P=4454, I=926, D=2317). Các thông số này sẽ được thay thế bằng giá trị PID tối ưu nhất sau khi ngừng gia nhiệt ở 48 độ.
6. Khi nhiệt độ đạt tới 48 độ và ngừng gia nhiệt, bit khởi động PID D4002.7 sẽ không được đặt lại ngay lập tức, nó có độ trễ trước khi đặt lại. Tại thời điểm này, nhiệt độ lấy mẫu cao hơn nhiệt độ mục tiêu. Nếu người dùng đặt tự động điều chỉnh PID ở trạng thái ON trở lại, PLC sẽ nhận tất cả các tham số PID là 0. Vui lòng đặt PID ở trạng thái ON sau khi nhiệt độ giảm xuống dưới mức nhiệt độ khí quyển bình thường.
7. Nếu bit khởi động điều chỉnh tự động PID và bit tự động điều chỉnh thành công có thể lưu giữ khi tắt nguồn, vui lòng đặt hoặc đặt lại chúng cho phù hợp để tránh lỗi tính toán khi khởi động PLC lần sau.
8. Nhiệt độ gia nhiệt cuối cùng sẽ lên tới 70 độ khi vọt lố được cho phép. Nó vượt quá nhiệt độ mục tiêu 10 độ, độ vọt lố khá nhỏ.
9. Mở rộng phạm vi tính toán PID có thể ngăn chặn vọt lố nhiệt độ.

Chương 8 : Khối hàm (Function Block) trong ngôn ngữ lập trình C

Trong chương này, chúng tôi tập trung vào các thông số kỹ thuật, phiên bản, cách gọi lệnh, điểm ứng dụng, v.v. của khối hàm trong ngôn ngữ C. Chúng tôi cũng đính kèm danh sách các hàm phổ biến.

8-1 Sơ lược

Dòng XD/XL hỗ trợ viết các khối hàm bằng ngôn ngữ C trong phần mềm Xinje PLC và gọi chúng khi cần. Nó hỗ trợ hầu hết tất cả các hàm ngôn ngữ C (so với dòng XC, dòng XD/XL cũng hỗ trợ các biến toàn cục), giúp tăng cường tính bảo mật của chương trình. Đồng thời, nó có thể gọi ở nhiều địa chỉ và các tập tin khác nhau, giúp cải thiện đáng kể hiệu quả của các lập trình viên.

8-2 Định dạng lệnh

1) Sơ lược

Gọi khối hàm ngôn ngữ C tại địa chỉ được chỉ định.

Gọi khối hàm ngôn ngữ C [NAME_C]			
Lệnh 16 bit	NAME_C	Lệnh 32 bit	-
Điều kiện thực thi	Thường hở, thường đóng, Kích hoạt sườn lên, sườn xuống	Model phù hợp	XD, XL
Phần cứng		Phần mềm	

2) Toán hạng

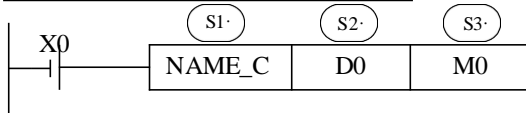
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Tên Khối hàm C, do người dùng xác định	String
S2	ID bắt đầu tương ứng của word W trong hàm ngôn ngữ C	16 bits, BIN
S3	ID bắt đầu tương ứng của bit B trong hàm ngôn ngữ C	bit, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word									Phần tử lập trình bit									
	Hệ								Hàng số	Mô-đun		Hệ							
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m	
S1																			
S2	•																		
S3														•					

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng và hành động



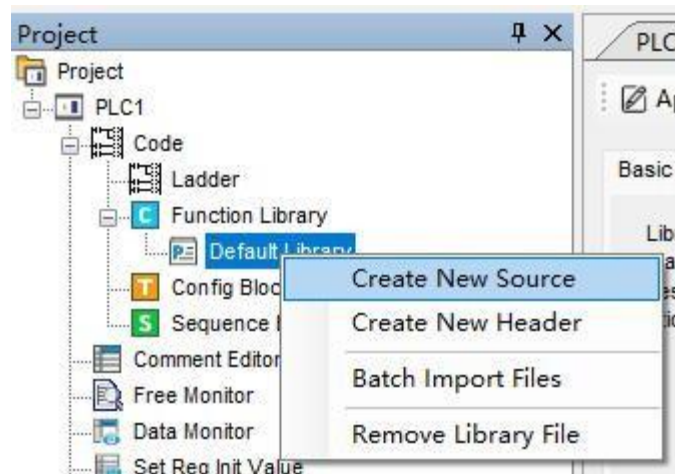
S1 là tên hàm. Nó bao gồm các số, chữ cái và gạch chân. Ký tự đầu tiên không được là số và độ dài phải ≤ 9 ký tự ASCII.

Tên có thể giống với các lệnh tự điều khiển của PLC như LD, ADD, SUB, PLSR, v.v.

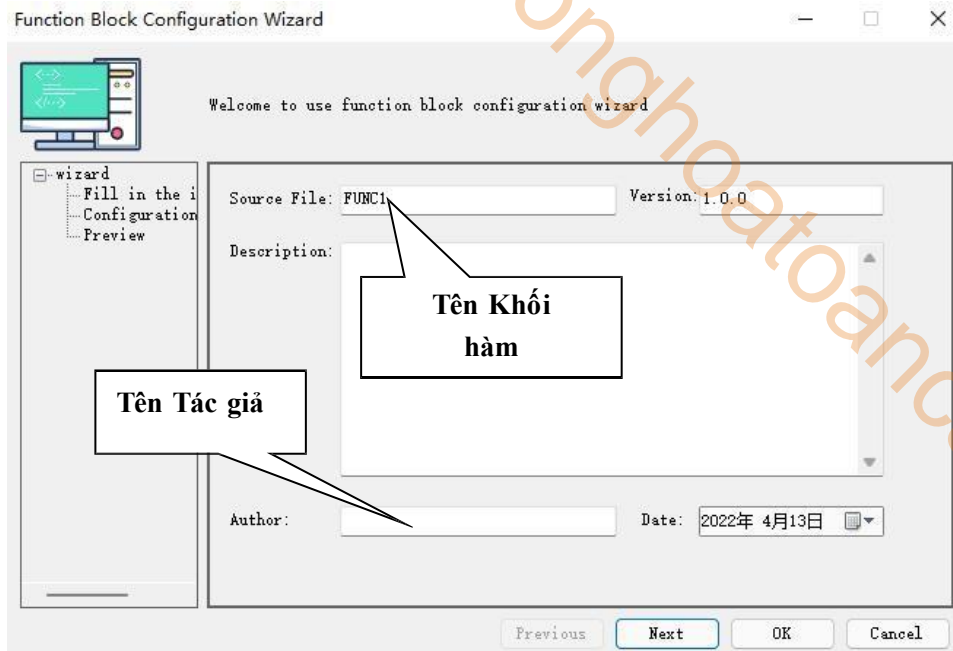
Tên không được trùng với các khối hàm có trong PLC hiện tại;

8-3 Các bước thao tác

1. Mở thanh công cụ chỉnh sửa PLC, trên thanh công cụ “Project” bên trái, chọn “Func Block”, nhấp chuột phải vào nó và chọn “Add New Func Block”.



2. Xem biểu đồ bên dưới, điền thông tin hàm của bạn;



Tên Khối hàm (Function Block) là tên chúng ta dùng để gọi BLOCK. Ví dụ: sơ đồ của FUNC1 được viết như sau:



3. Sau khi tạo Khối hàm mới, bạn có thể thấy giao diện chỉnh sửa như hình dưới đây:

- Cách truyền tham số: nếu gọi **Function Block** trong sơ đồ bậc thang thì D (HD) và M (HM) được truyền là ID bắt đầu của W và B. Lấy biểu đồ trên làm ví dụ, bắt đầu bằng D0 và M0, sau đó là W[0] là D0, W[10] là D10, B[0] là M0, B[10] là M10; nếu các

tham số trong sơ đồ bậc thang là HD0, HM0 thì W[0]=HD0, B[0]=HM0; nếu các tham số trong sơ đồ bậc thang là D100, HM100 thì W[0]=D100, B[0]=HM100. Vì vậy, địa chỉ bắt đầu của các thành phần word và bit được người dùng xác định trong chương trình PLC.

Lưu ý: Biến cục bộ được xác định bên trong hàm ngôn ngữ C không được dài hơn 100 từ.

- Tham số **W**: đại diện cho thành phần phần mềm **Word**, sử dụng dưới dạng nhóm dữ liệu. Ví dụ: W[0]=1; W[1]=W[2]+W[3]; trong chương trình sử dụng các phần tử lập trình theo quy tắc chuẩn của ngôn ngữ C.

Tham số **B**: biểu thị phần tử lập trình **Bit**, sử dụng dưới dạng nhóm dữ liệu. Hỗ trợ **SET** và **RESET**. Ví dụ: B[0]=1; B[1]=0; And assignment (Phép gán AND), ví dụ: B[0]=B[1].

- Phép toán Double-word: thêm **D** vào trước **W**. Ví dụ, DW[10]=100000, nó có nghĩa là gán cho double-word W[10]W[11]. Phép toán Double-word: Hỗ trợ xác định biến số thực dấu phẩy động trong hàm và thực hiện phép số thực dấu phẩy động; (Ví dụ: thanh ghi float D0 (double-word) có nghĩa là FW[0], FW[0]=123.456)
- Xác định phần tử lập trình khác trong ngôn ngữ C:

Khi một khối hàm được tạo, #define SysRegAddr_HD_D_HM_M được xác định mặc định trong hàm main. Nếu cần sử dụng đầu vào (X) và đầu ra (Y), bạn cần thêm X, Y vào định nghĩa Macro mặc định "#define SysRegAddrHD_D_HM_M", sẽ là "#define SysRegAddrHD_D_HM_M_X_Y". Ví dụ: đặt trạng thái X0 thành cuộn M0, B[0]=X[0]; đặt trạng thái Y0 cho cuộn M10, B[10]= Y[0]. (Lưu ý: X và Y tương ứng được biểu thị bằng hệ thập phân thay vì hệ bát phân trong ngôn ngữ C).

Tương tự, các ứng dụng trong C giống nhau đối với thanh ghi khả nhớ khi tắt nguồn S, bộ đếm C, bộ định thời T, thanh ghi bộ đếm CD, thanh ghi bộ định thời TD, thanh ghi D (HD) và cuộn dây M (HM), v.v. Định nghĩa Macro "#define SysRegAddr_S_C_T_CD_TD_D_M". Nếu chúng tắt nguồn bộ nhớ xử lý HS, bộ đếm HC, bộ định thời HT, thanh ghi bộ đếm HCD, thanh ghi bộ định thời HTD, v.v., định nghĩa macro "#define SysRegAddr_HS_HC_HT_HCD_HTD". Ví dụ: W[0]=CD[0]; W[1]=TD[0]; B[1]=C[0]; B[2]=T[0];

Lưu ý: Các loại phần tử lập trình được hỗ trợ ngoại trừ SEM.

- Khi khối hàm được tạo, định nghĩa mặc định #define SysRegAddr_HD_D_HM_M trong hàm main.

```

7
8 *****
9 void FUNC1( WORD W , BIT B )
10 {
11     #define SysRegAddr_HD_D_HM_M
12
13
14 }
15

```

Bạn nên sử dụng nó làm định nghĩa macro cục bộ, nghĩa là bên trong thân hàm.

- Function Library: Function Library: Khối hàm người dùng có thể sử dụng trực tiếp các hàm và hằng được xác định trong thư viện hàm. Tham khảo chương 8-10 để biết các hàm và hằng số có trong thư viện hàm.
- Kiểu dữ liệu khác được hỗ trợ:

```

BOOL;           //BOOL Quantity
INT8U;          //8 bits unsigned integer
INT8S;          //8 bits signed integer
INT16U          //16 bits unsigned integer
INT16S         //16 bits signed integer
INT32U          //32 bits unsigned integer
INT32S         //32 bits signed integer
FP32;          // single precision floating
FP64;          //double precision floating
Ví dụ: #define DHD*(INT32S*)&HD //DHD means double word HD
       #define FFW*(FP64*)&D //FFW means double precision floating numbers
       #define DDW*(long long*)&D //DDW means four words register

```

Giải thích: DHD là số nguyên có dấu 32 bit. DHD[0] đại diện cho một thanh ghi khả nhớ khi tắt nguồn dạng số nguyên có dấu 32 bit bao gồm HD0 và HD1.

Các Macro được định nghĩa trước:

```

#define True 1
#define False 0
#define TRUE 1
#define FALSE 0

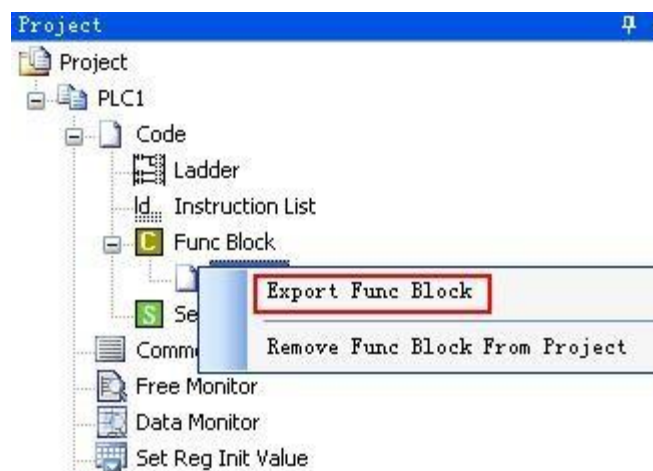
```

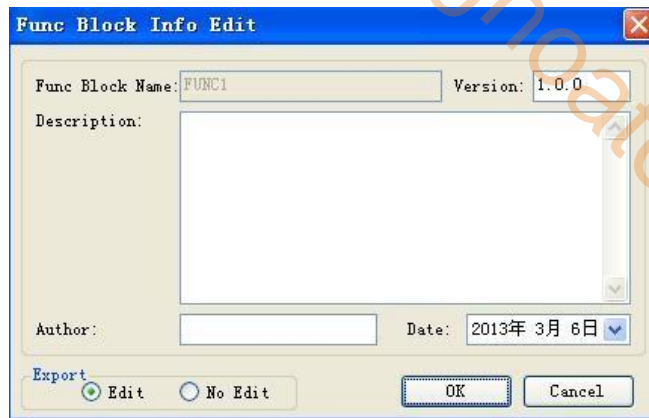
- Không có tùy chọn không thể chỉnh sửa để xuất tệp tiêu đề (header files), các tùy chọn khác giống như tệp nguồn
- Trong ngôn ngữ C, có hai quy tắc để tham chiếu các tệp tiêu đề, #include "xx.h" và #include <xxx.h>. khi sử dụng tệp tiêu đề trong dự án PLC, nó cần sử dụng #include "xxx.h" trong tệp nguồn.
- Không sử dụng định nghĩa Marco #define SysRegAddr trong tệp tiêu đề, định nghĩa Marco này không hiệu quả trong tệp tiêu đề, chỉ có thể được sử dụng trong tệp nguồn

8-4 Nhập và xuất các hàm

1. Xuất

(1) Chức năng: Xuất hàm dưới dạng file, sau đó chương trình PLC khác có thể nhập vào để sử dụng;



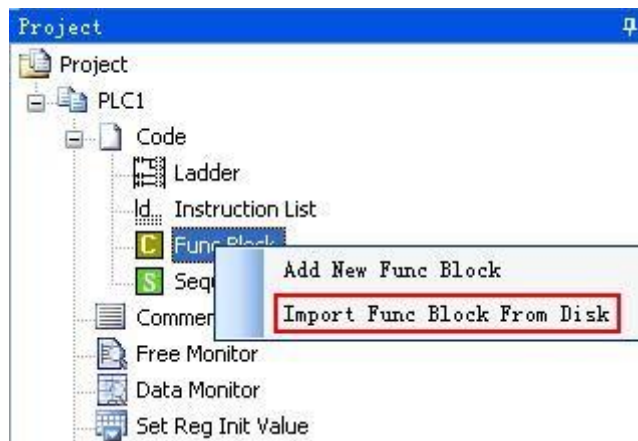


(2) Định dạng Xuất

- a) Edit: Xuất mã nguồn ra và lưu dưới dạng tệp. Nếu nhập lại, tệp có thể chỉnh sửa được;
- b) No edit: Không xuất mã nguồn, nếu nhập tệp thì không thể chỉnh sửa được. Các model Ethernet và các model không phải Ethernet không thể được sử dụng chung. Bạn chỉ cần sửa đổi model trước khi xuất nó.

2. Nhập

Chức năng: Nhập **Func Blockfile** hiện có để sử dụng trong chương trình PLC



Chọn **Func Block**, nhấp chuột phải vào 'Import Func Block from Disk', chọn đúng tệp và sau đó nhấp vào OK.

8-5 Chỉnh sửa Func Blocks

Ví dụ: Thêm D0 và D1 vào thanh ghi của PLC, sau đó gán giá trị cho D2;

- (1) Trong thanh công cụ 'Project', tạo mới **Func Block**, ở đây chúng tôi đặt tên **Func Block** là ADD_2, sau đó chỉnh sửa chương trình ngôn ngữ C;
- (2) Nhấp 'compile' sau khi chỉnh sửa.

```

PLC1 - Ladder FuncBlock-ADD_2
Information Export Compile
7      W [2] =W [0] +W [1]
8      *****
9      void ADD_2( WORD W , BIT B )
10     { W [2] =W [0] +W [1]
11     }
12     }
13

```

Information(1)

Error List Output

1. ...\\.\tmp\PrjFuncB\ADD_2.c: In function 'ADD_2':
 ..\\.\tmp\PrjFuncB\ADD_2.c:6:1: error: expected ';' before 'asm'

Danh sách thông tin

Theo thông tin hiển thị ở vùng trống đầu ra, chúng ta có thể tìm kiếm và sửa đổi lỗi ngữ pháp trong chương trình ngôn ngữ C. Ở đây chúng ta có thể thấy rằng trong chương trình không có dấu “;”. Biên dịch lại chương trình sau khi sửa đổi chương trình. Trong danh sách thông tin, chúng tôi có thể xác nhận rằng chương trình không có lỗi ngữ pháp.

```

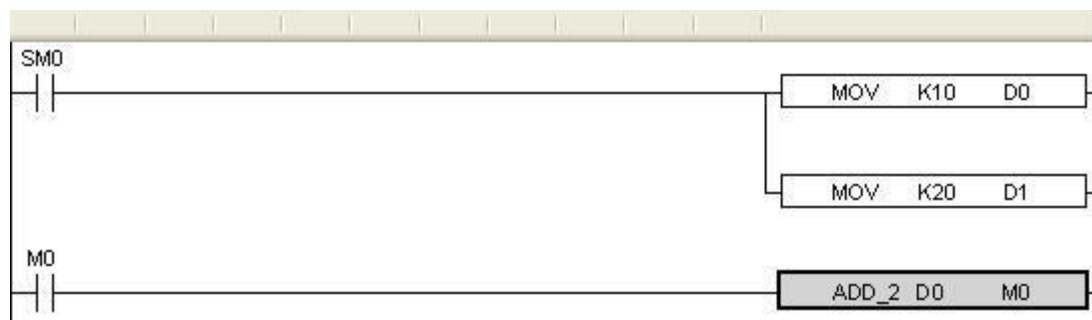
Information Export Compile
6      Comment:
7      W [2] =W [0] +W [1]
8      *****
9      void ADD_1( WORD W , BIT B )
10     { W [2] =W [0] +W [1];
11     }
12     }

```

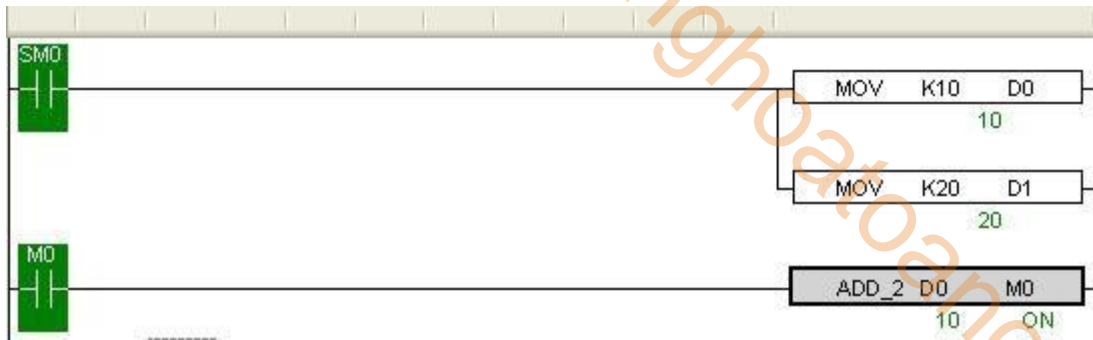
Information

Error List Output

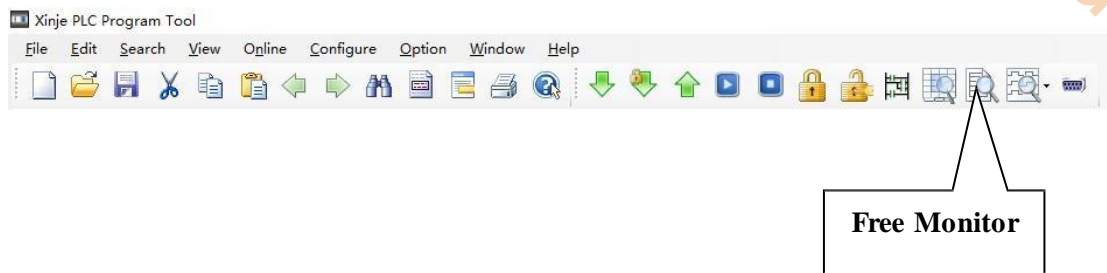
(3) Viết chương trình PLC, gán riêng giá trị 10 và 20 vào các thanh ghi D0, D1, sau đó gọi Func Block ADD_2, xem biểu đồ bên dưới:



(4) Tải chương trình vào PLC, chạy PLC và đặt M0.



(5) Từ Free Monitor trên thanh công cụ, chúng ta có thể thấy D2 thay đổi thành 30, nghĩa là phép gán đã thành công;



8-6 Ví dụ về chương trình

Nếu PLC cần thực hiện các phép tính phức tạp (bao gồm cả phép tính cộng và trừ), phép tính sẽ được sử dụng nhiều lần, các hàm trong ngôn ngữ C rất dễ sử dụng.

Ví dụ 1:

Phép tính $a = b/c + b*c + (c-3)*d$

Phương pháp 1: sử dụng sơ đồ bậc thang

Nhận kết quả của $c-3$

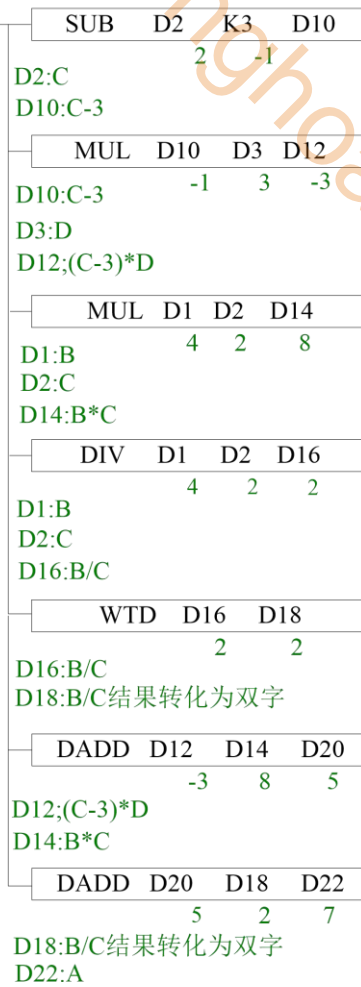
Lấy kết quả của ba phương trình nhân

Lấy kết quả tổng

Sơ đồ bậc thang chỉ hỗ trợ hai toán hạng gốc, cần nhiều bước mới có được kết quả.



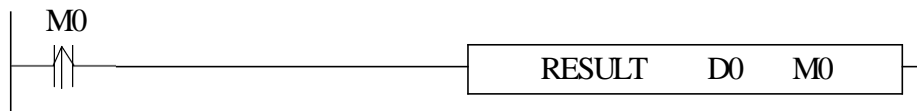
运行常
ON线圈



Lưu ý:

1. Kết quả của MUL (Phép nhân) là Dword, kết quả được lưu trong D14~D15.
2. Kết quả của DIV (Phép chia) có thương D16 và số dư D17. Nếu D17 là một giá trị khác 0 thì độ chính xác của phép tính sẽ giảm. Vui lòng sử dụng định dạng số thực dấu phẩy động để đảm bảo độ chính xác.
3. Thương số D16 là giá trị word, trong phép tính cộng, tất cả dữ liệu sẽ được thay đổi thành Dword. Kết quả cuối cùng được lưu trữ trong D22~D23.

Phương pháp 2: Sử dụng ngôn ngữ C:



Kết quả	Tên hàm
D0	In the function, W [0] =D0, W [1] =D1... If D0=D32, then W [0] =D32, W [1] =D33... If S2=HD32, then W [0] =HD32, W [1] =HD33...
M0	In the function, B [0] = M0, B [1] =M1... If S2=M32, then B [0] = M32, B [1] =M33... If S2=HM32, then B [0] = HM32, B [1] =HM33...

Chương trình C

```
9 void RESULT( WORD W , BIT B )
10 {
11     long int a,b,c,d;;
12     b=W[1];
13     c=W[2];
14     d=W[3];
15     a=b/c+b*c+(c-3)*d;
16     DW[4]=a;
17 }
```

Phương pháp 2 có thể đơn giản hóa chương trình.

Hàm ngôn ngữ C ở trên tương tự như sơ đồ bậc thang của phương pháp 1, độ chính xác không cao. Nếu cần độ chính xác cao, vui lòng sử dụng phép tính dạng số thực dấu phẩy động.

Ví dụ 2: Tính giá trị chẵn lẻ CRC thông qua Func Block

Nguyên tắc tính CRC:

- (1)Đặt thanh ghi 16 bit (thanh ghi CRC) = FFFF H
- (2)XOR (Exclusive OR) thông báo byte 8 bit đầu tiên và thanh ghi CRC 16 bit thấp.
- (3)Dịch phải 1 bit của thanh ghi CRC, điền số 0 vào bit cao nhất.
- (4)Kiểm tra giá trị dịch chuyển phải, nếu bằng 0, lưu giá trị mới từ bước 3 vào thanh ghi CRC; nếu nó khác 0, XOR giá trị thanh ghi CRC với A001 H và sau đó lưu kết quả vào thanh ghi CRC.
- (5)Lặp lại bước 3 & 4 cho đến khi tất cả 8 bit được tính toán.
- (6)Lặp lại bước (2)~(5), sau đó tính toán thông báo 8 bit tiếp theo. Cho đến khi tất cả các thông báo được tính toán xong, kết quả sẽ là mã chẵn lẻ CRC trong thanh ghi CRC.

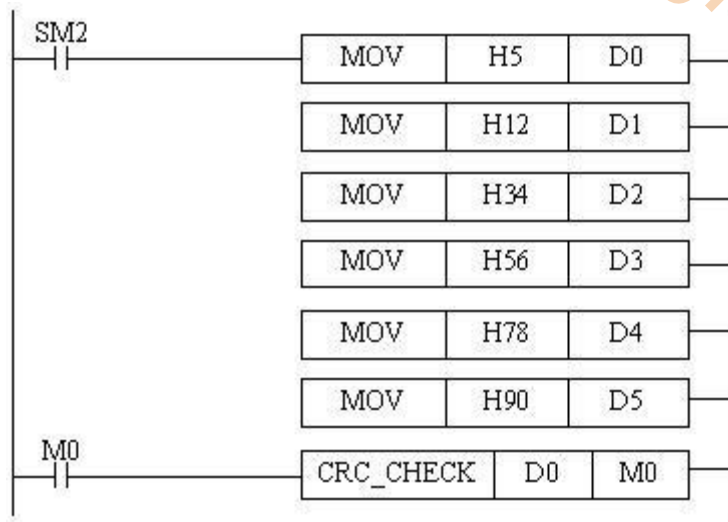
Chỉnh sửa chương trình Function Block ngôn ngữ C, xem biểu đồ bên dưới:

```
9 void CRC_CHECK( WORD W , BIT B )
10 {
11     int i,j,m,n;
12     unsigned int reg_crc=0xffff,k;
13
14     for( i = 0 ; i < W[0] ; i++ )
15     {
16         reg_crc^=W[i+1];
17         for(j=0;j<8;j++)
18         {
19             if(reg_crc&0x01)
20                 reg_crc=(reg_crc>>1)^0xa001;
21             else
22                 reg_crc=reg_crc>>1;
23         }
24     }
25
26     m=W[0]+1;
27     n=W[0]+2;
28     k=reg_crc&0xff00;
29     W[n] = k>>8;
30     W[m]=reg_crc&0xff;
31 }
```


Chỉnh sửa chương trình bậc thang PLC,

D0: Kiểm tra số byte dữ liệu,

D1~D5: Kiểm tra nội dung dữ liệu. Xem biểu đồ dưới đây:

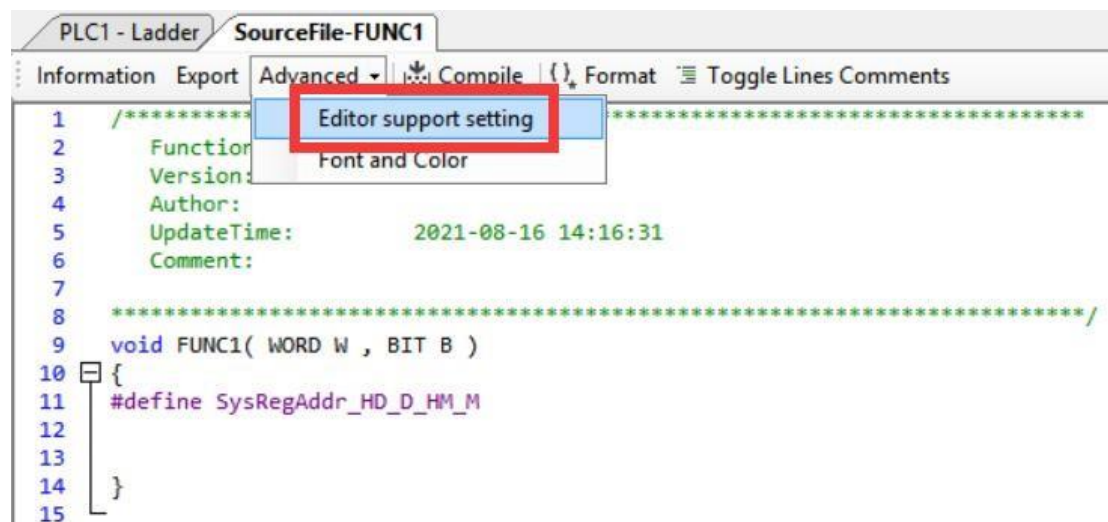


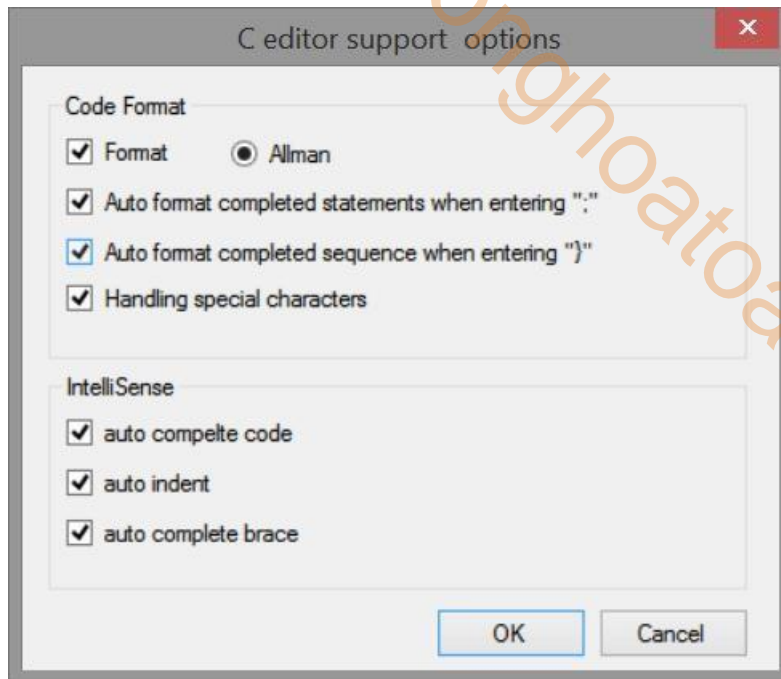
Tải xuống PLC, sau đó CHẠY PLC, đặt M0, thông qua Free Monitor, chúng ta có thể thấy rằng các giá trị trong D6 và D7 là bit cao nhất và thấp nhất của giá trị chẵn lẻ CRC;

8-7 Các hàm mới

(1) Định dạng

Nhấp vào menu cài đặt hỗ trợ chỉnh sửa/nâng cao để mở cửa sổ tùy chọn hỗ trợ trình chỉnh sửa C.





(2) Định dạng tự động mã cục bộ

- Tự động định dạng các câu lệnh đã hoàn thành khi nhập ";"

Khi người dùng nhập ký tự ";" định dạng câu lệnh của hàng hiện tại.

- Tự động định dạng chuỗi đã hoàn thành khi nhập "{"

Khi người dùng nhập "}", hãy định dạng nội dung trong dấu "{}".

(3) Xử lý các ký tự đặc biệt

Các ký tự full-width được người dùng nhập vào trình chỉnh sửa cần phải được chuyển đổi thành các ký tự half-width vì chúng không được trình biên dịch nhận dạng.

(4) Tự động biên dịch mã code

Khi người dùng nhập ký tự, chức năng nhắc nhở sẽ đưa ra một số lời nhắc nhất định để giúp người dùng nhập và hoàn thành mã.

- Gửi

Khi người dùng nhấn Enter hoặc ";", mã hiện đang được chỉnh sửa sẽ được gửi đến bộ phân tích để phân tích và danh sách code tips sẽ được tạo ra.

Khi người dùng nhập ký tự, điều khiển nhắc mã sẽ tự động bật lên để khớp với dữ liệu nhập của người dùng và đưa ra lời nhắc.

```

void FUNC1( WORD W, BIT B )
{
#define SysRegAddr_HD_D_HM_M
}
if
int
INT8U
INT8S
INT16U
INT16S
INT32U
INT32S
if() { }
if() { } else { }

```

➤ Các lời khuyên truy cập biến thành viên

Khi người dùng nhập "." "hoặc" -> ", chức năng nhắc mã sẽ giúp người dùng nhắc nhở các biến thành viên trong cấu trúc hoặc kiểu tập hợp của biến được xác định, như minh họa trong hình sau.

```

struct TestStruct
{
int a;
int b;
}
void FUNC1( WORD W, BIT B )
{
#define SysRegAddr_HD_D_HM_M

TestStruct test;
test.
a
b

```

➤ Tự động thực lờ

Chức năng thực lờ tự động của trình soạn thảo được tối ưu hóa, phù hợp hơn với thói quen của người dùng.

➤ Tự động đóng các dấu ngoặc

Khi người dùng nhập "(" [{"", nó sẽ tự động giúp người dùng tạo dấu ngoặc tương ứng ")" "}".

(5) Diễn giải/ bỏ diễn giải code

Chọn/ bỏ chọn Comment cho chức năng nhận xét của hàng.

Phím tắt là Ctrl +/.

(6) Function library/ Thư viện chức năng

Vui lòng tham khảo chương 8-8.

8-8 Thư viện chức năng

Nó cung cấp các chức năng mã hóa, đóng gói (encapsulation), xuất và nhập các khối hàm C.

8-8-1 Chức năng mới

+ Phân loại thư viện

Thư viện chức năng được chia thành thư viện dự án (project library) và thư viện chung (global library).

Project library/ Thư viện dự án: các chức năng trong thư viện dự án của người dùng được lưu trong dự án và có thể được sử dụng trực tiếp.

Global Library/ Thư viện chung: các chức năng trong thư viện chung của người dùng được lưu trong thư mục cục bộ để thuận tiện cho người dùng.

8-8-2 Các chức năng cơ bản

+ Mở và lưu tệp

Khởi động phần mềm XDPro, chạy một dự án trống hoặc mở bất kỳ dự án hiện có nào để xem thư viện chức năng.

Lưu ý:

Thư viện chức năng được chia thành thư viện dự án và thư viện chung. Một thư viện mặc định (tức là thư viện dự án) được thêm vào dự án trống theo mặc định;

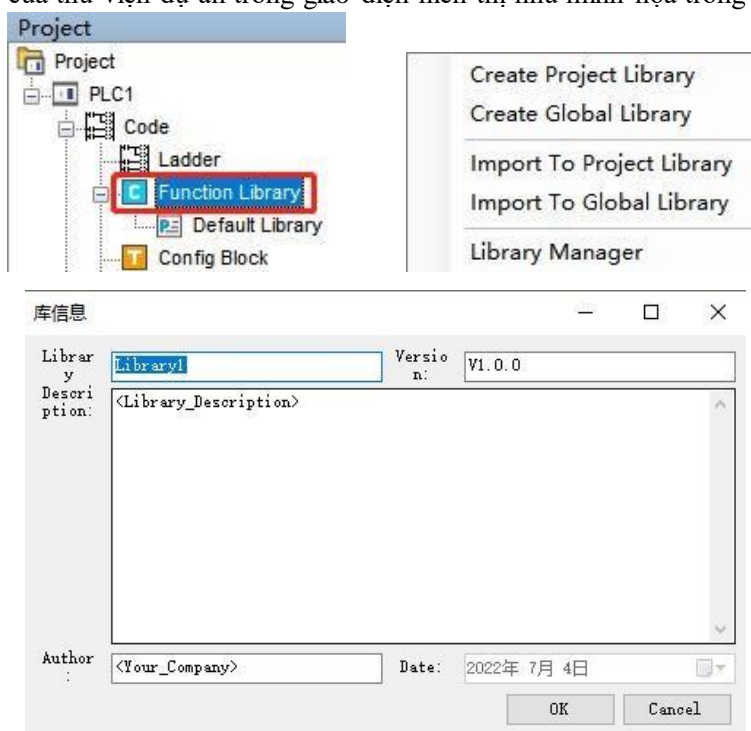
Nếu dự án thuộc phiên bản cũ được mở bằng phiên bản mới, hàm chức năng của nó sẽ được thêm vào thư viện mặc định;

Nếu dự án trong phiên bản mới được mở bằng phiên bản cũ, các hàm chức năng trong thư viện mặc định sẽ được giữ lại và phần còn lại không thể phân tích cú pháp được.

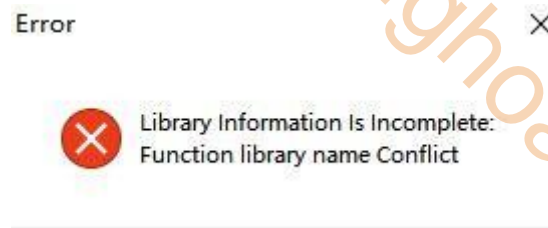
8-8-3 Tạo mới

+ Tạo thư viện dự án

Chọn "Function library" trong thanh công cụ "Project" ở bên trái, nhấp chuột phải và chọn "Create Project Library" và bạn có thể chỉnh sửa tên, phiên bản, mô tả, tác giả và các thông tin khác của thư viện dự án trong giao diện hiển thị như minh họa trong hình sau:

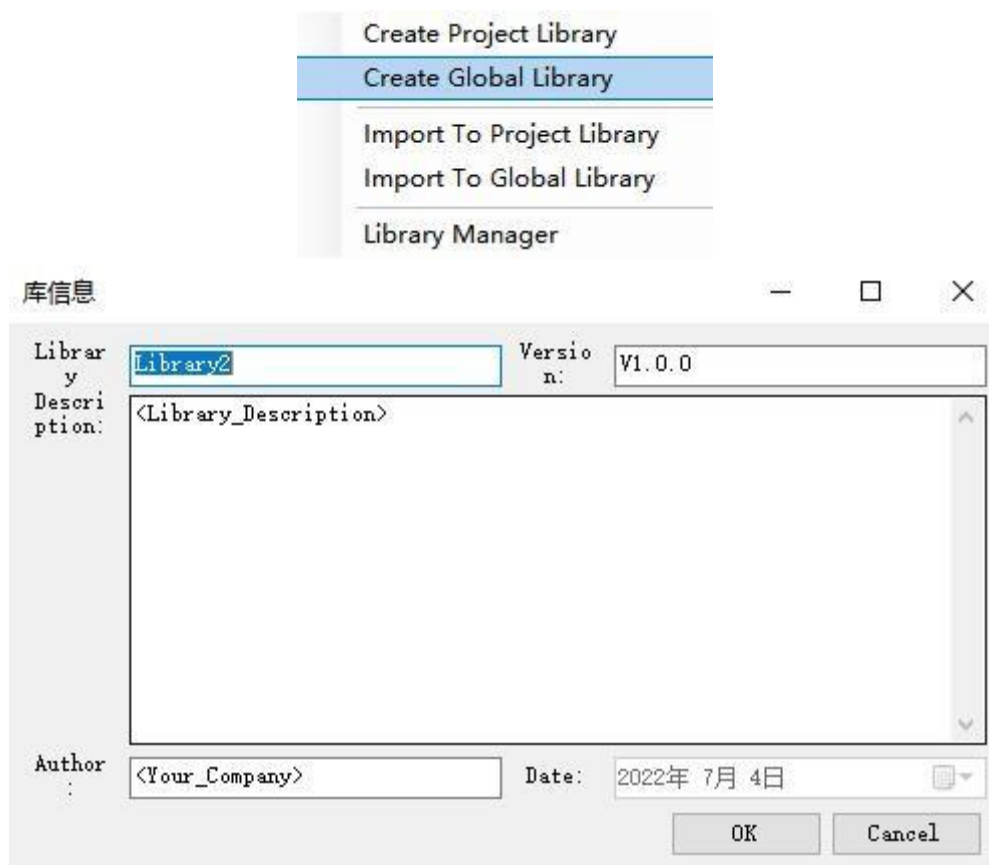


Lưu ý: nếu tên thư viện giống với bất kỳ tên thư viện nào trong thư viện hiện tại, cửa hiển thị sẽ xuất hiện:

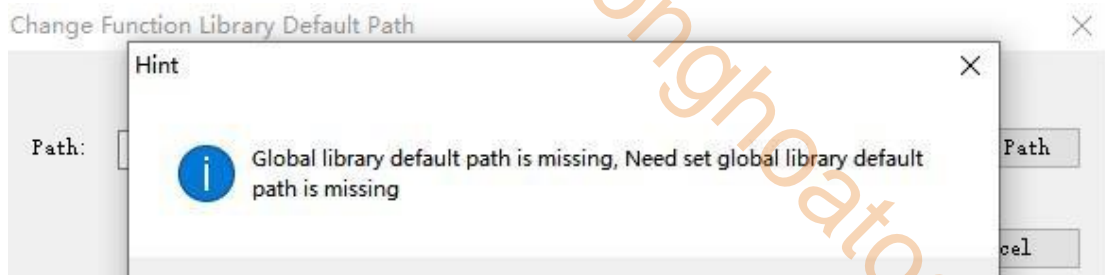


+ Tạo thư viện chung

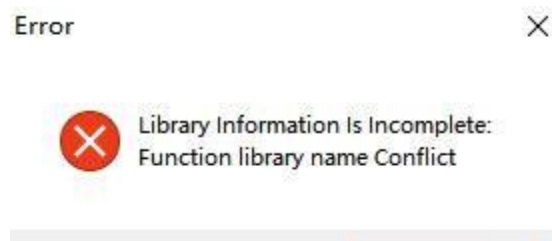
Chọn "Function library" trên thanh công cụ "Project" bên trái, nhấp chuột phải và chọn "Create Global Library", và bạn có thể chỉnh sửa tên, phiên bản, mô tả, tác giả và các thông tin khác của thư viện chung trong giao diện hiển thị như minh họa trong hình sau:



Lưu ý: nếu thư mục thư viện chung không được thiết lập, thông báo nhắc nhở trong hình sau sẽ xuất hiện và giao diện cài đặt Thư mục thư viện chung sẽ hiển thị:

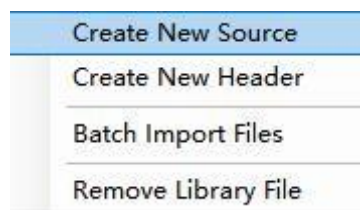


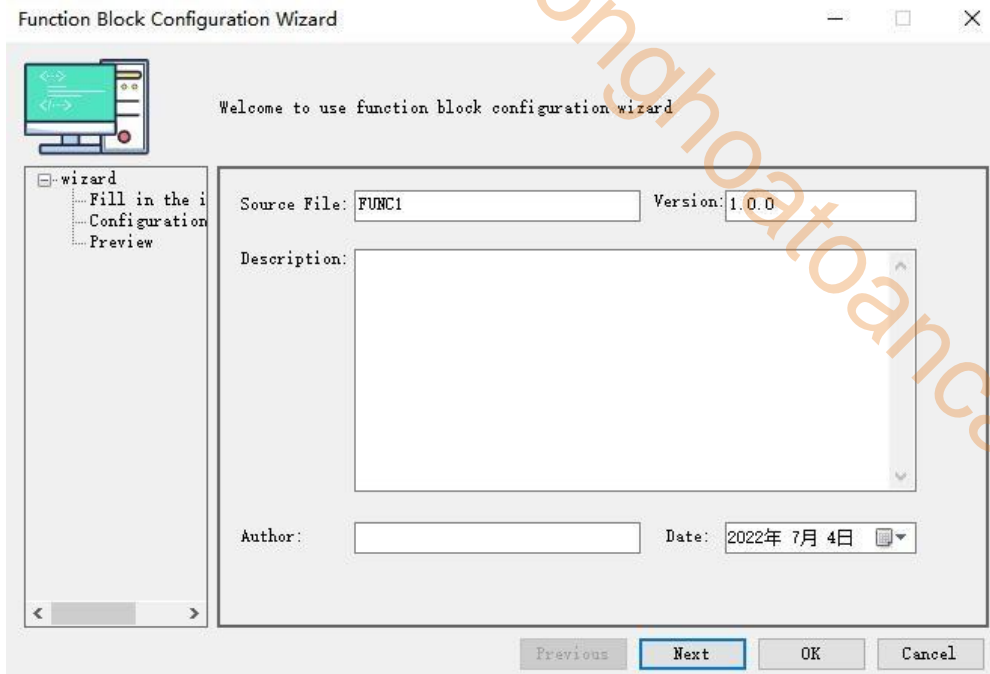
Sau khi thiết lập đường dẫn, cửa sổ tệp thư viện mới sẽ được hiển thị và thông tin thư viện (tên, phiên bản, mô tả, tác giả) được điền vào. Nếu tên thư viện giống với bất kỳ tên thư viện nào trong thư viện hiện tại, cửa sổ hiển thị sẽ xuất hiện:



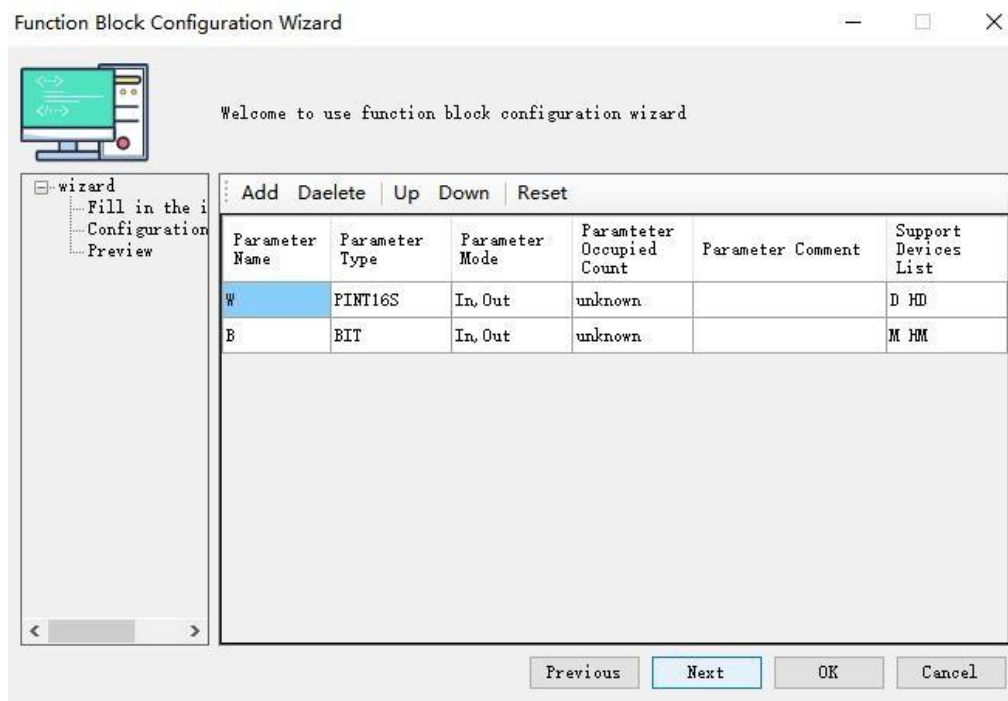
+ Tạo tệp nguồn mới

Trong thanh công cụ "project" ở bên trái, chọn thư viện dự án hoặc thư viện chung mà tệp nguồn cần được thêm vào trong "function library", nhấp chuột phải và chọn "new source file" để chỉnh sửa tên, phiên bản, mô tả, tác giả và các thông tin khác của tệp nguồn trong giao diện hiển thị, như trong hình sau:

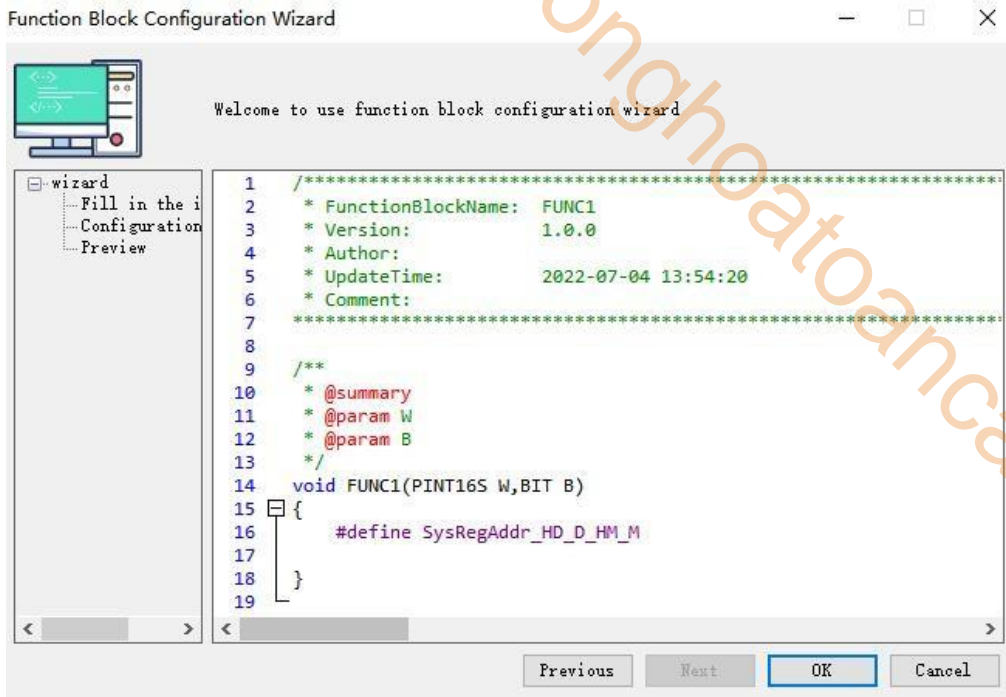




Nhấp vào “next” sau khi điền để định cấu hình thông tin tham số:

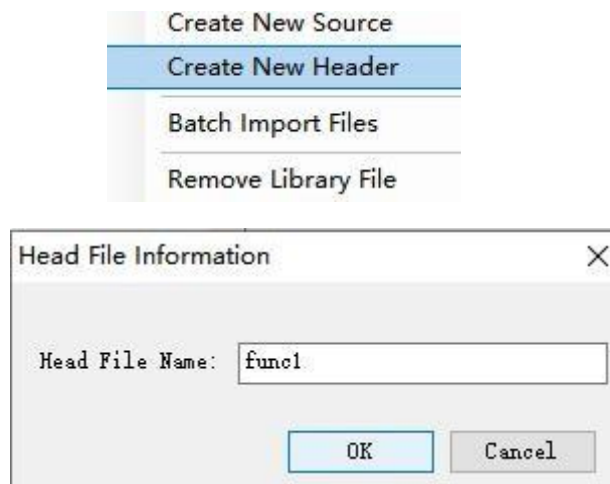


Sau khi hoàn tất cấu hình tham số, nhấn “next” để hiển thị giao diện xem trước của tệp nguồn. Nếu có vấn đề, nhấp vào “previous” để thiết lập lại các thông số. Nếu không có vấn đề gì, hãy nhấp vào “OK” để hoàn tất việc thêm tệp nguồn.



+ Tạo tệp tiêu đề mới

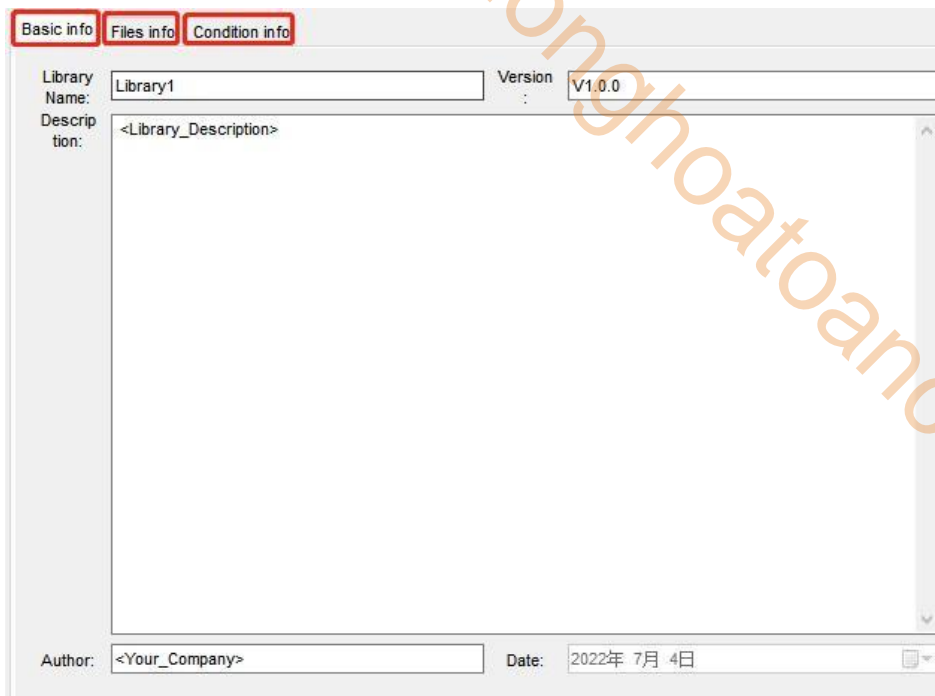
Trên thanh công cụ "Project" ở bên trái, chọn thư viện dự án hoặc thư viện chung mà tệp nguồn cần được thêm vào trong "Function Library", nhấp chuột phải và chọn "New header file" để chỉnh sửa tên của tệp tiêu đề trong giao diện hiển thị, như thể hiện trong hình sau:



8-8-4 Chỉnh sửa

+ Chỉnh sửa thông tin thư viện

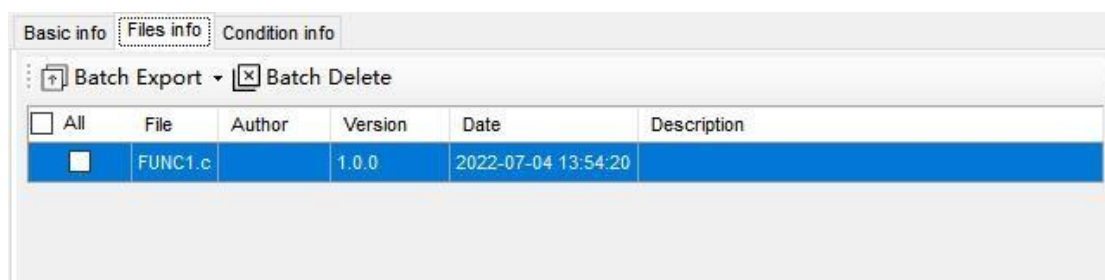
Nhấp vào "thư viện dự án" hoặc "thư viện chung" trong thanh dự án bên trái để chỉnh sửa thông tin và bạn có thể xem và chỉnh sửa thông tin cơ bản/thông tin tệp/thông tin hạn chế của thư viện trong giao diện thông tin thư viện được hiển thị:



+ Thông tin cơ bản

Tên thư viện: chỉ cho phép chữ cái và số khi đặt tên thư viện.

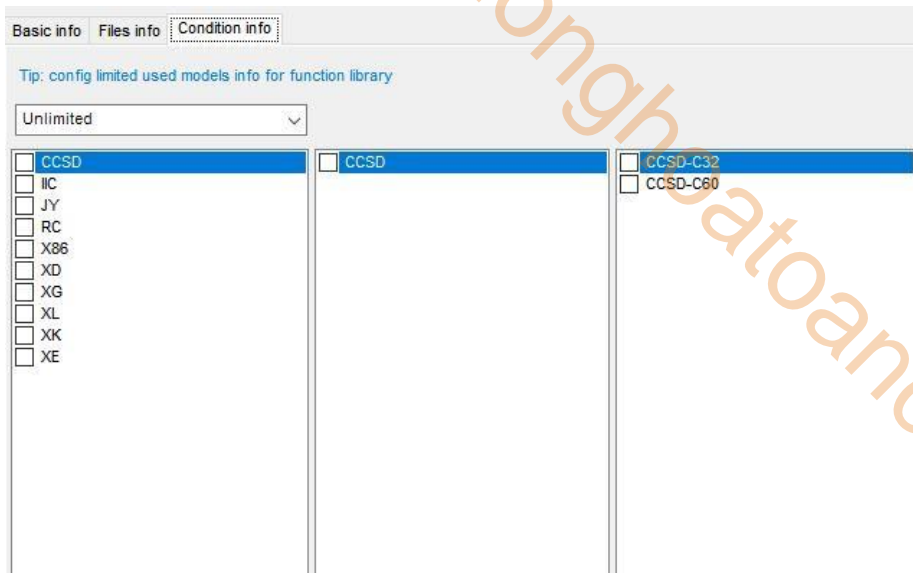
Thông tin tệp



- Thêm tệp nguồn/tệp tiêu đề theo thư viện chức năng đã chọn, giao diện thông tin tệp hiển thị các thông tin cơ bản của tệp.
- Tệp đã nhập sẽ xác định liệu người dùng có thể chỉnh sửa nó hay không.
- Các tệp được xuất hàng loạt có thể được chỉnh sửa hoặc không.
- After deleting the application in batch, remove the reference of the library file in the PLC project.
- Sau khi xóa ứng dụng hàng loạt, hãy xóa tham chiếu của tệp thư viện trong dự án PLC.

+ Thông tin điều kiện

Không thể sử dụng các model trong danh sách đen và chỉ có thể sử dụng các model trong danh sách trắng.



+ Thông tin tệp nguồn

Nhấp vào tệp nguồn để chỉnh sửa thông tin trên thanh dự án:



Trong giao diện tệp nguồn được hiển thị, nhấp vào Information để sửa đổi thông tin tệp nguồn, đặc tính chức năng được sửa đổi và mã được sửa đổi tương ứng.

Information Export Search Advance Compile Format Code Switch Comment Code

```

1  /*****
2  * FunctionBlockName: FUNC1
3  * Version: 1.0.0
4  * Author:
5  * UpdateTime: 2022-07-04 13:54:20
6  * Comment:
7  *****/
8
9  /**
10 * @summary
11 * @param W
12 * @param B
13 */
14 void FUNC1(PINT16S W,BIT B)
15 {
16     #define SysRegAddr_HD_D_HM_M
17 }
18
19

```

Function Block Information Edit Form

Name: FUNC1 Version: 1.0.0
 Author: Date: 2022年 7月 4日

Parameter Name	Parameter Type	Parameter Mode	Parameter Occupied Count	Parameter Comment	Support Devices List
W	PINT16S	In, Out	unknown		D HD
B	BIT	In, Out	unknown		M HM

Function Description:

OK Cancel

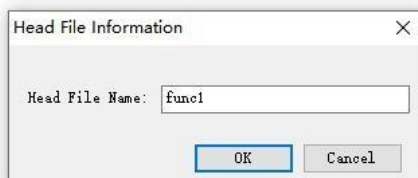
+ Thông tin tệp tiêu đề

Nhấp vào tệp tiêu đề để chỉnh sửa thông tin trên thanh dự án:



Trong giao diện tệp tiêu đề được hiển thị, nhấp vào Information để sửa đổi thông tin tệp đầu, header function signature được sửa đổi và mã được sửa đổi tương ứng.

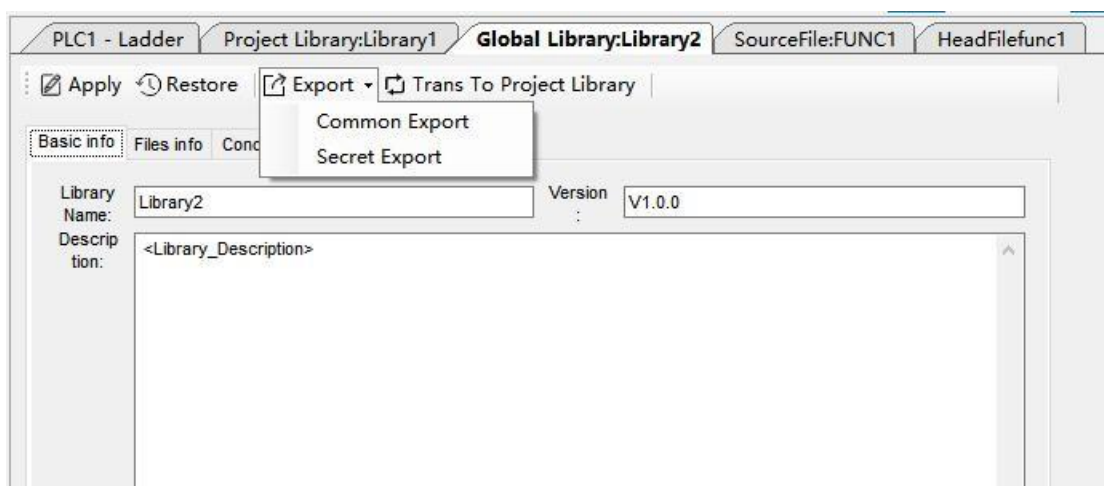
```
Information Export Search Advance Compile Format Code Switch Comment Code
1 #ifndef _FUNC1_H
2 #define _FUNC1_H
3
4 #endif
```



8-8-5 Xuất thư viện

+ Xuất thư viện chức năng

Nhấp vào "Project library" hoặc "Global library" trên thanh dự án bên trái để chỉnh sửa thông tin, và nhấp vào "Export" trên giao diện thông tin thư viện được hiển thị:

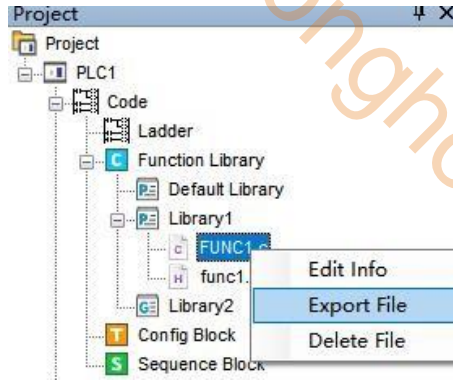


Xuất bình thường/ Normal export: nếu tệp thư viện là thư viện có thể chỉnh sửa, hãy xuất tệp đó dưới dạng thư viện có thể chỉnh sửa; Nếu tệp thư viện là thư viện không thể chỉnh sửa, hãy xuất nó dưới dạng thư viện không thể chỉnh sửa.

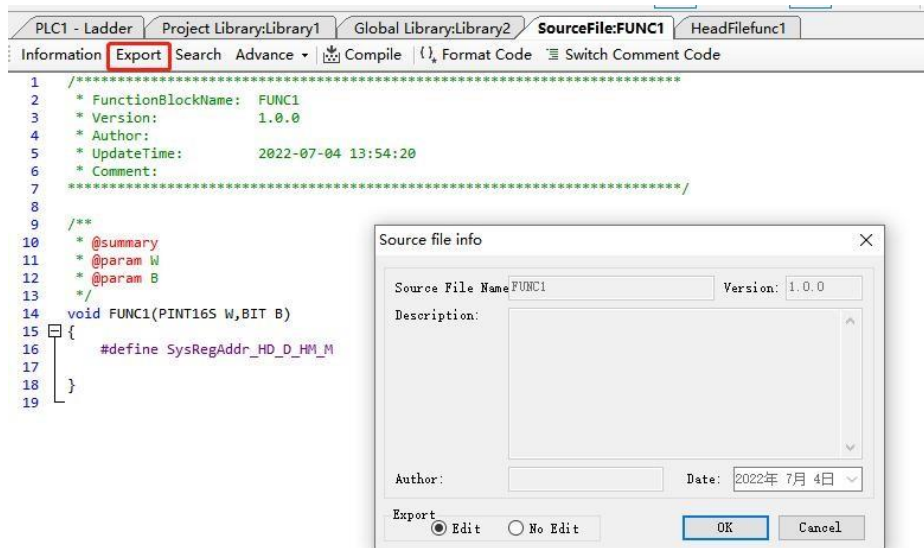
Xuất được mã hóa/ Encrypted export: nếu tệp thư viện có thể chỉnh sửa được, tệp nguồn trong tệp thư viện sẽ được biên dịch và xuất dưới dạng thư viện không thể chỉnh sửa; Nếu tệp thư viện không thể chỉnh sửa được, hãy lưu tệp thư viện trực tiếp.

+ Xuất tệp nguồn/ tệp tiêu đề

Nhấp chuột phải vào tệp nguồn/tệp tiêu đề cần được xuất trong thanh dự án --> Xuất tệp:



Hoặc nhấp vào file nguồn/tệp tiêu đề cần xuất ở cột dự án bên trái và nhấp vào “export” trong giao diện chỉnh sửa bên phải:



Chọn chế độ xuất (có thể chỉnh sửa hoặc không thể chỉnh sửa) trong thông tin tệp được hiển thị

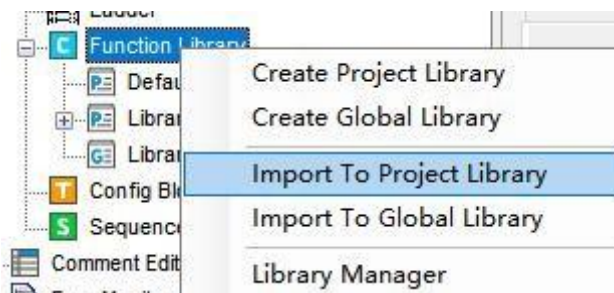
Nhấn OK sau khi thiết lập và chọn đường dẫn lưu tệp.

Sau khi chọn đường dẫn, nhấp vào OK để hoàn tất quá trình xuất.

8-8-6 Nhập thư viện

+ Nhập thư viện chức năng

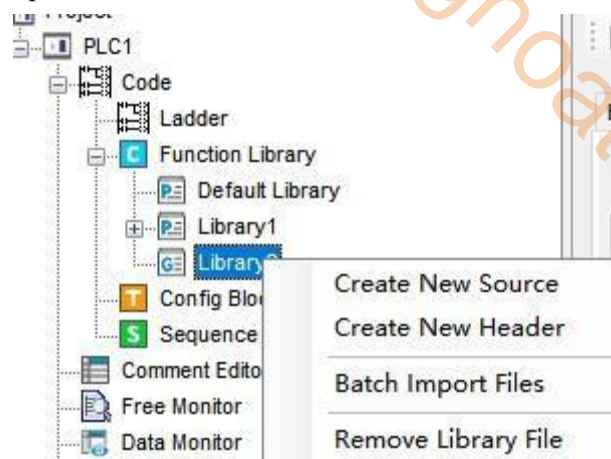
Chọn "Function Library" trên thanh công cụ "Project" ở bên trái, nhấp chuột phải và chọn "Import to Project Library" hoặc "Import to Global Library":



Trong giao diện "chọn tệp thư viện chức năng" được hiển thị, hãy chọn một tệp và nhấp vào “open” để hoàn tất quá trình nhập.

+ Nhập tệp chức năng

Nhấp phải vào "Project Library" hoặc "Global Library" trên thanh dự án bên trái để nhập tệp, và chọn "Batch import files":

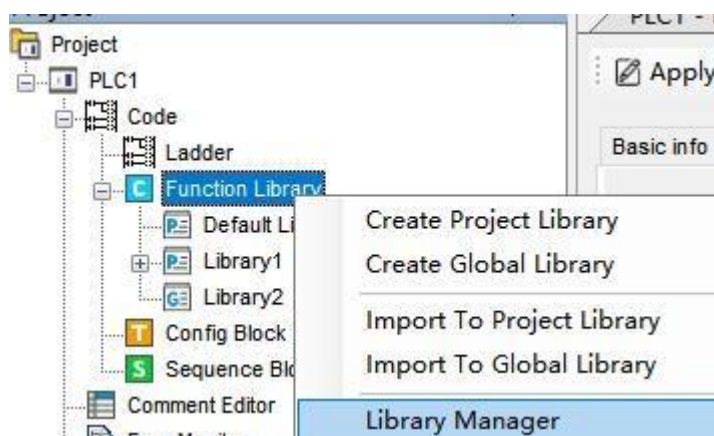


Chọn tệp chức năng cần nhập trong giao diện "chọn tệp/ select files" và nhấp vào "open" để hoàn tất quá trình nhập tệp.

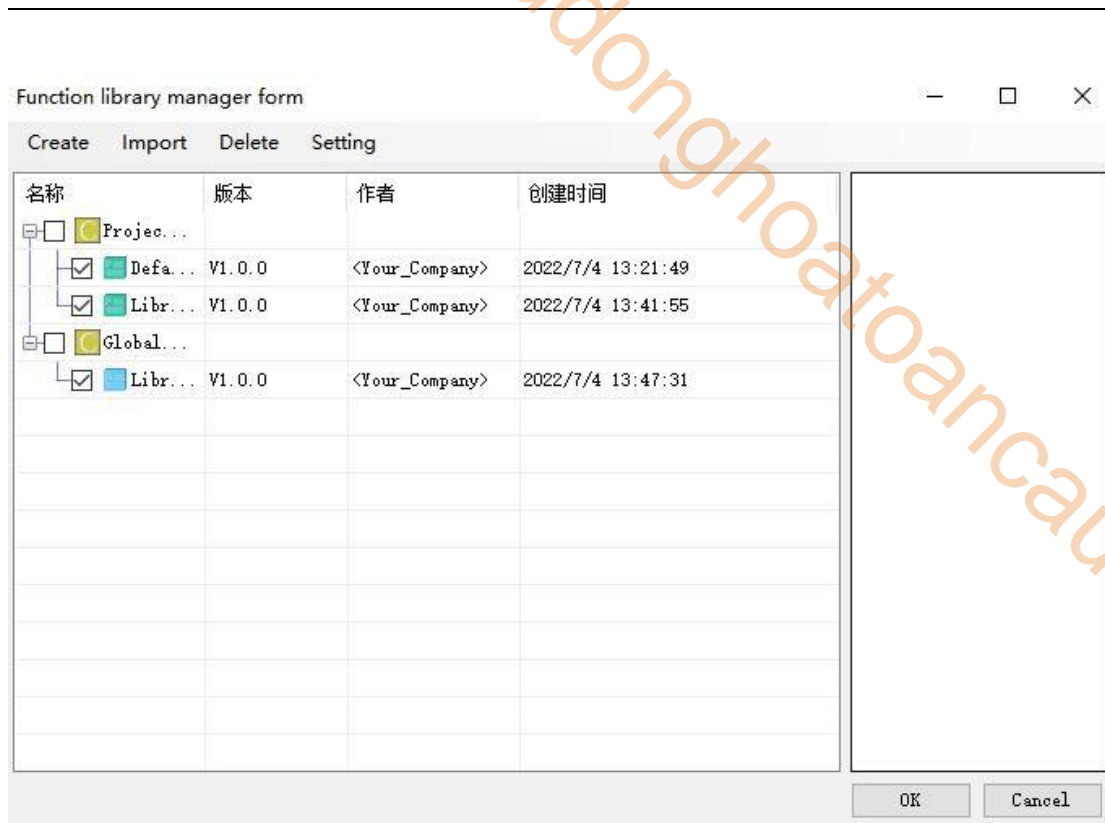
8-8-7 Các chức năng khác

+ Trình quản lý thư viện

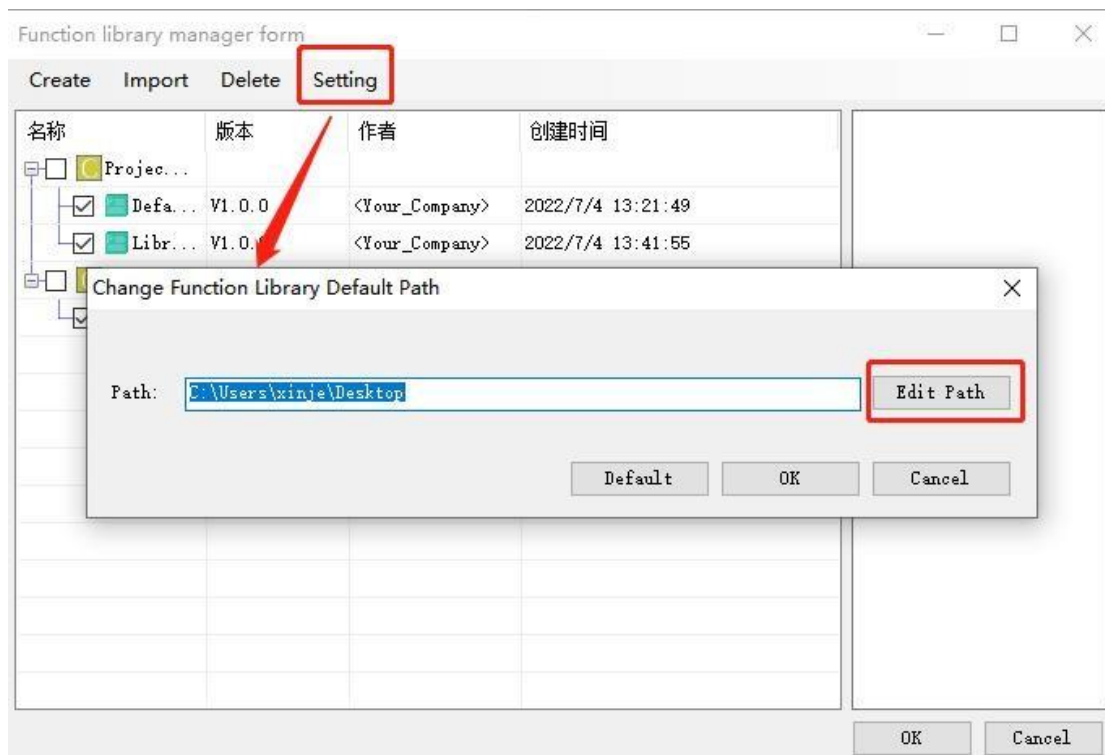
Chọn "Function Library" trên thanh công cụ "Project" ở bên trái và nhấp chuột phải để chọn "library management":



Trong "Cửa sổ quản lý Thư viện Chức năng" được hiển thị, bạn có thể hoàn tất việc tạo, nhập, xóa (và xóa các tệp thư viện được tham chiếu trong dự án) và cài đặt thư viện chức năng. Bằng cách chọn thư viện chức năng trong phần quản lý, bạn có thể tạo và sau đó gọi nó trong dự án.



Nhấp vào cài đặt để thay đổi Thư mục Thư viện Chung:



+ Xóa tệp thư viện

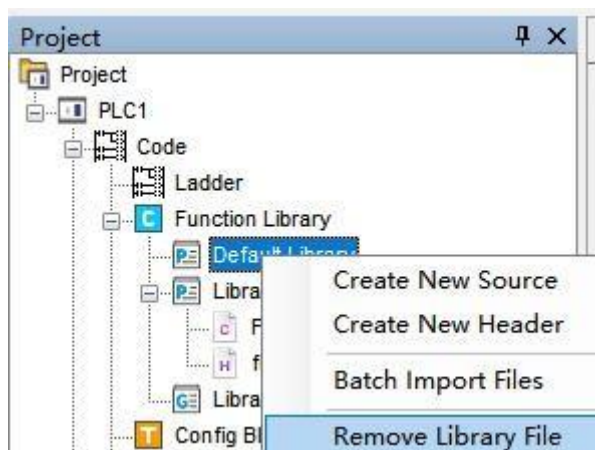
Trong "cửa sổ quản lý thư viện chức năng" của chương trước, kiểm tra tệp thư viện tương ứng và nhấp vào Delete để xóa tệp thư viện trong dự án hiện tại.

Function library manager form

名称	版本	作者	创建时间
Project Library			
<input checked="" type="checkbox"/> Default Li...	V1.0.0	<Your_Company>	2022/7/4 13:21:49
<input checked="" type="checkbox"/> Library1	V1.0.0	<Your_Company>	2022/7/4 13:41:55
Global Library			
<input checked="" type="checkbox"/> Library2	V1.0.0	<Your_Company>	2022/7/4 13:47:31

+ Xóa tệp thư viện

Nhấp phải vào "Project Library" hoặc "Global Library" trên thanh dự án bên trái để nhập tệp và chọn "Remove Library file":

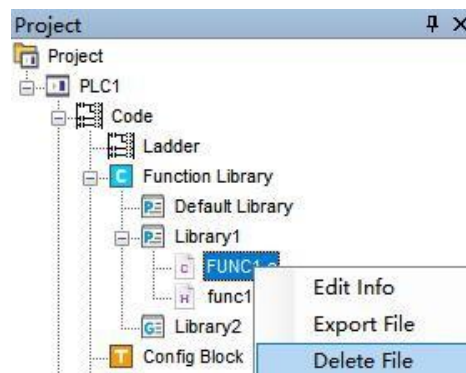


Lưu ý: Xóa tệp thư viện có nghĩa là hủy ứng dụng của tệp khỏi dự án hiện tại mà không xóa ứng dụng.

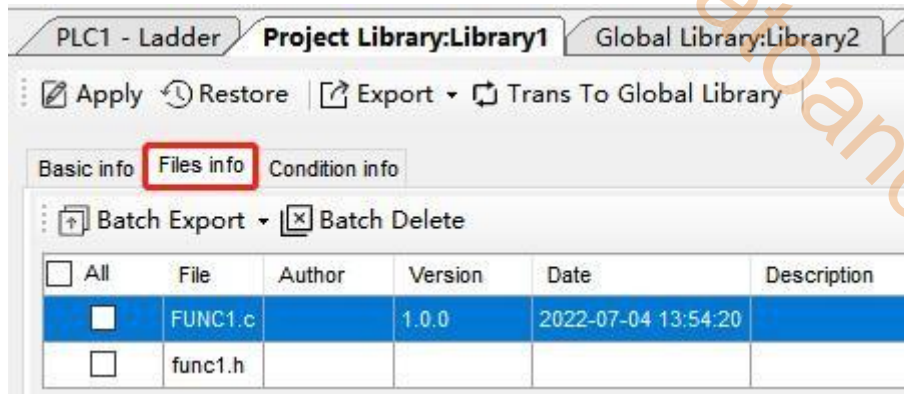
+ Xóa tệp nguồn/ tệp tiêu đề

Có hai cách để xóa tệp nguồn/tệp tiêu đề:

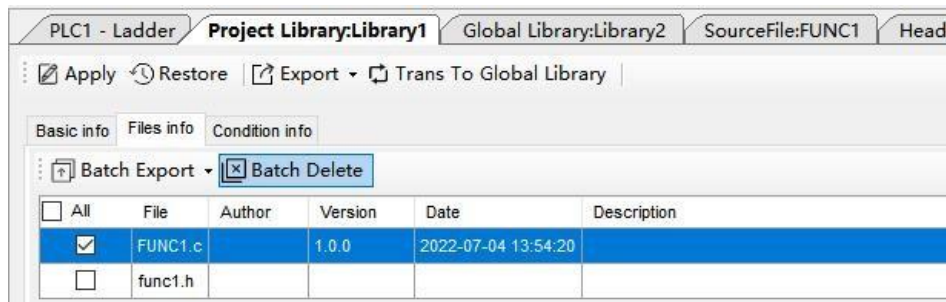
Cách 1: nhấp chuột phải vào tệp nguồn/tệp tiêu đề sẽ được xuất trong thanh dự án -> xóa tệp (Delete file):



Cách 2: nhấp vào thư viện chức năng để xóa tệp trong thanh dự án bên trái:



Chọn file cần xóa và nhấp vào "batch delete":



Nhấp vào "Apply" và thông báo "successfully applied" sẽ xuất hiện. Tập tin đã bị xóa.

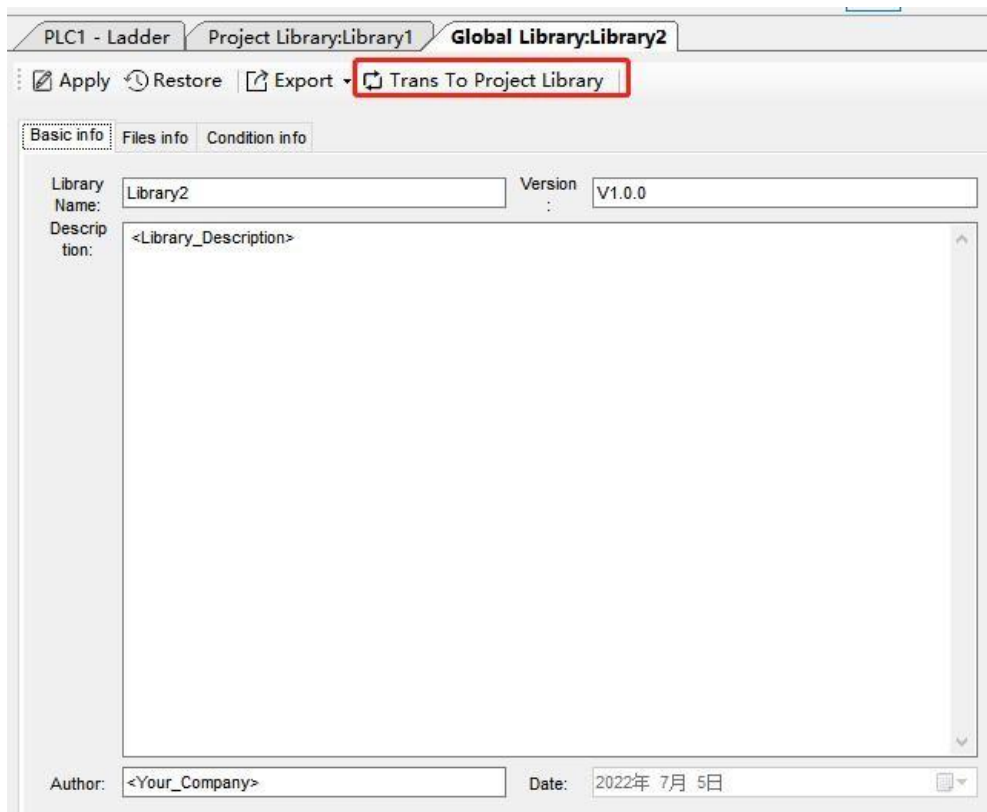
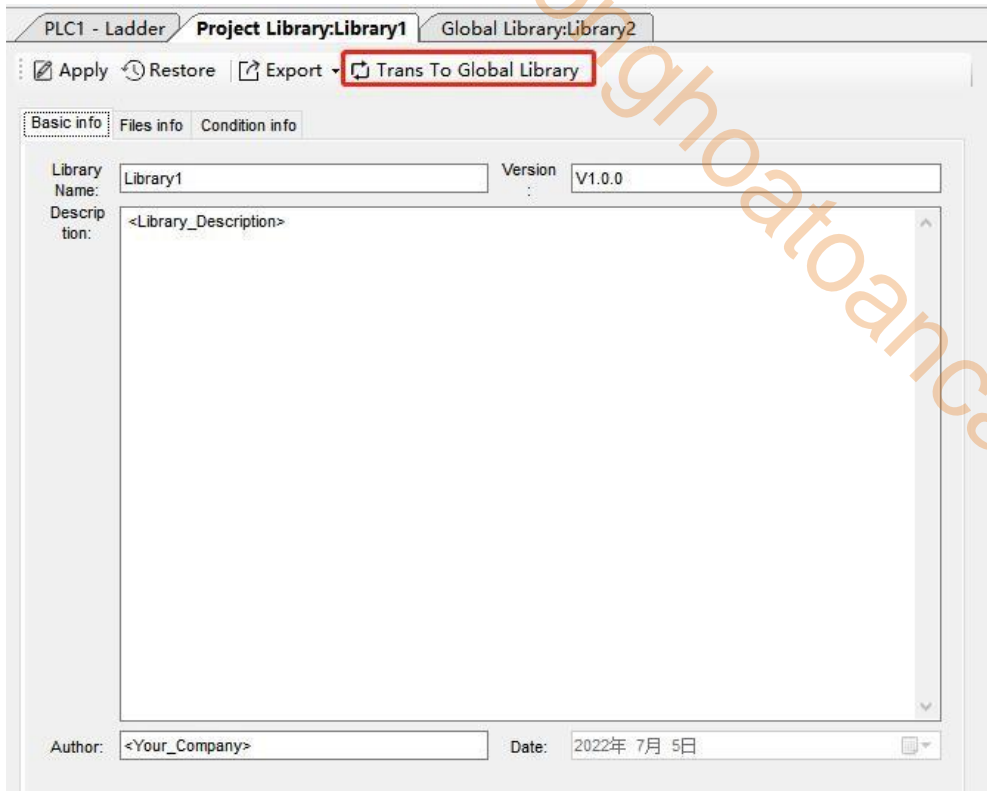
Hint



Apply Success

+ Chuyển đổi

"Thư viện chung" và "thư viện dự án" có thể được chuyển đổi lẫn nhau và có thể mở giao diện chỉnh sửa của thư viện chức năng (để tìm hiểu các bước cụ thể, hãy tham khảo chương 8-8-7-4, cách 2).



+ Biên dịch

Nhấp vào tệp nguồn trong thanh dự án ở bên trái và nhấp vào "compile" trong giao diện chỉnh sửa ở bên phải.


```

1  /*****
2  * FunctionBlockName:  FUNC1
3  * Version:           1.0.0
4  * Author:
5  * UpdateTime:       2022-07-05 08:35:17
6  * Comment:
7  *****/
8
9
10 /**
11  * @summary
12  * @param W
13  * @param B
14  */
15 void FUNC1(PINT16S W,BIT B)
16 {
17     #define SysRegAddr_HD_D_HM_M
18 }
19

```

8-8-7-7 Đặt Thư Mục Thư Viện Chung

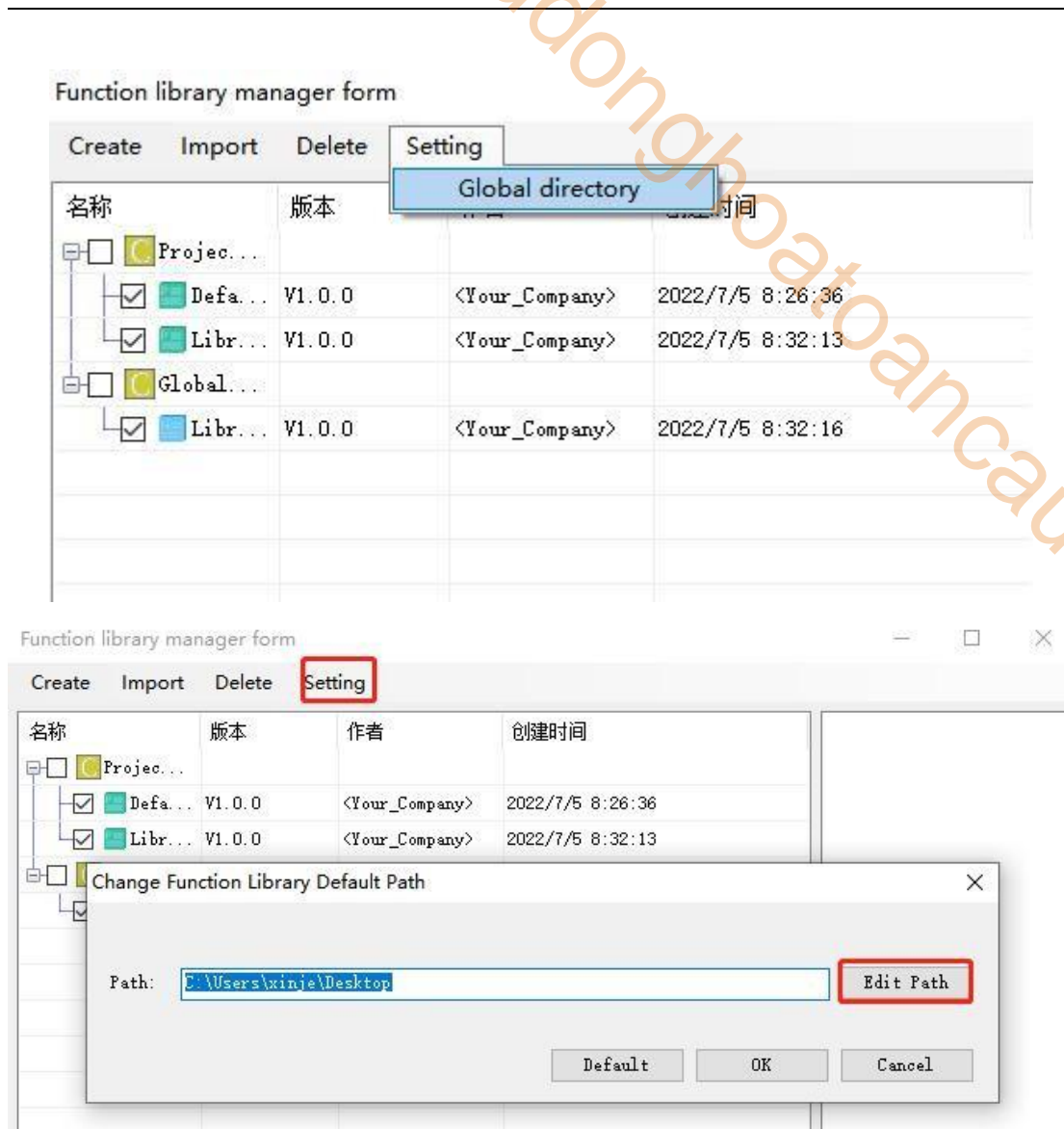
Có ba cách để thiết lập thư viện chung:

Cách 1: Mở giao diện quản lý thư viện (vui lòng tham khảo [8-8-7-1. Library manager](#) để biết các bước cụ thể). Nếu thư mục thư viện chung chưa được đặt, lời nhắc đặt thư mục thư viện chung sẽ xuất hiện.



Cách 2: Trong quá trình tạo thư viện chung mới, nếu thư mục thư viện chung chưa được đặt, lời nhắc tương tự như trong cách 1 cũng sẽ xuất hiện. Bạn có thể đặt đường dẫn trong cửa sổ "Thay đổi đường dẫn mặc định của thư viện chức năng/ Change Function Library Default Path" được hiển thị.

Cách 3: Mở giao diện "Library manger", vui lòng tham khảo [8-8-7-1. Library manager/ Trình quản lý thư viện](#) và nhấp vào "Settings" -> "Global Library Directory" như sau:



8-9 Các lưu ý về ứng dụng

- Trong một tệp Func Block, bạn có thể viết nhiều hàm và chúng có thể được gọi lẫn nhau.
- Mỗi tệp Func Block là độc lập, hàm trong khối hàm khác không thể gọi được.
- Các tệp Func Block có thể gọi các hàm trong thư viện ngôn ngữ C dưới dạng số thực dấu phẩy động, số học như sin, cos, tan.
- PLC dòng XC chỉ hỗ trợ biến cục bộ, trong khi PLC dòng XD/XL hỗ trợ cả biến cục bộ và biến toàn cục. Điều này làm cho Khối hàm ngôn ngữ C trở nên linh hoạt và thuận tiện hơn.
- Khuyến nghị sử dụng các biến toàn cục:
- ① Sử dụng vùng phân tử lập trình thay vì bộ nhớ thông thường để lưu trữ dữ liệu của các biến toàn cục.
- Không gian phân tử lập trình của PLC có thể được sử dụng làm không gian biến toàn cục và tính bảo mật được đảm bảo.
- ② Ví dụ về cách sử dụng

Lấy kiểu FP64 làm ví dụ:

```

1  /*
2  FunctionBlockName: FUNC1
3  Version:          1.0.0
4  Author:
5  UpdateTime:      2020/1/3 10:30:47
6  Comment:
7
8  */
9  void Test():
10 FP64 * GlobalV;      declaration
11
12 void FUNC1( WORD W , BIT B )
13 {
14 #define SysRegAddr_HD_D_MM_M
15
16 GlobalV = (FP64*)&W[0];  initialization
17
18 Test();
19 }
20 void Test()
21 {
22 #define SysRegAddr_HD_D_MM_M
23
24 FP64 value = GlobalV[0];  using
25 *(FP64*)&HD[0] = value;
26 }

```

Như thể hiện trong hình trên, biến con trỏ toàn cục GlobalV được khai báo bên ngoài hàm và sau đó được khởi tạo trong hàm chính để trỏ đến không gian của thành phần phần mềm. Địa chỉ đầu tiên của không gian là địa chỉ nơi W[0] được đặt. Cuối cùng, giá trị của biến có thể thu được thông qua thao tác con trỏ trong các hàm khác.

Lấy kiểu cấu trúc (struct) làm ví dụ:

```

1  #ifndef _STRUCT_H
2  #define _STRUCT_H
3
4  typedef struct
5  {
6      INT16U V;
7      FP64 S;
8  }ExStruct;
9
10 #endif

```

Khai báo cấu trúc

```

1  /*
2  FunctionBlockName: STRUCT
3  Version:          1.0.0
4  Author:
5  UpdateTime:      2020/1/3 10:58:49
6  Comment:
7
8  */
9  #include "struct.h"  the header file contained declaration
10 void Test():
11 ExStruct* ST;
12 void STRUCT( WORD W , BIT B )
13 {
14
15 #define SysRegAddr_HD_D_MM_M
16
17 ST = ((ExStruct*)&W[0]);  initialization
18
19 ST->V = 10;
20 ST->S = 100.001;
21
22 Test(ST);
23 }
24 void Test(ExStruct* ex)
25 {
26 #define SysRegAddr_HD_D_MM_M
27
28 *(INT16*)&HD[0] = ex->V;  using
29 *(FP64*)&HD[2] = ex->S;
30 }

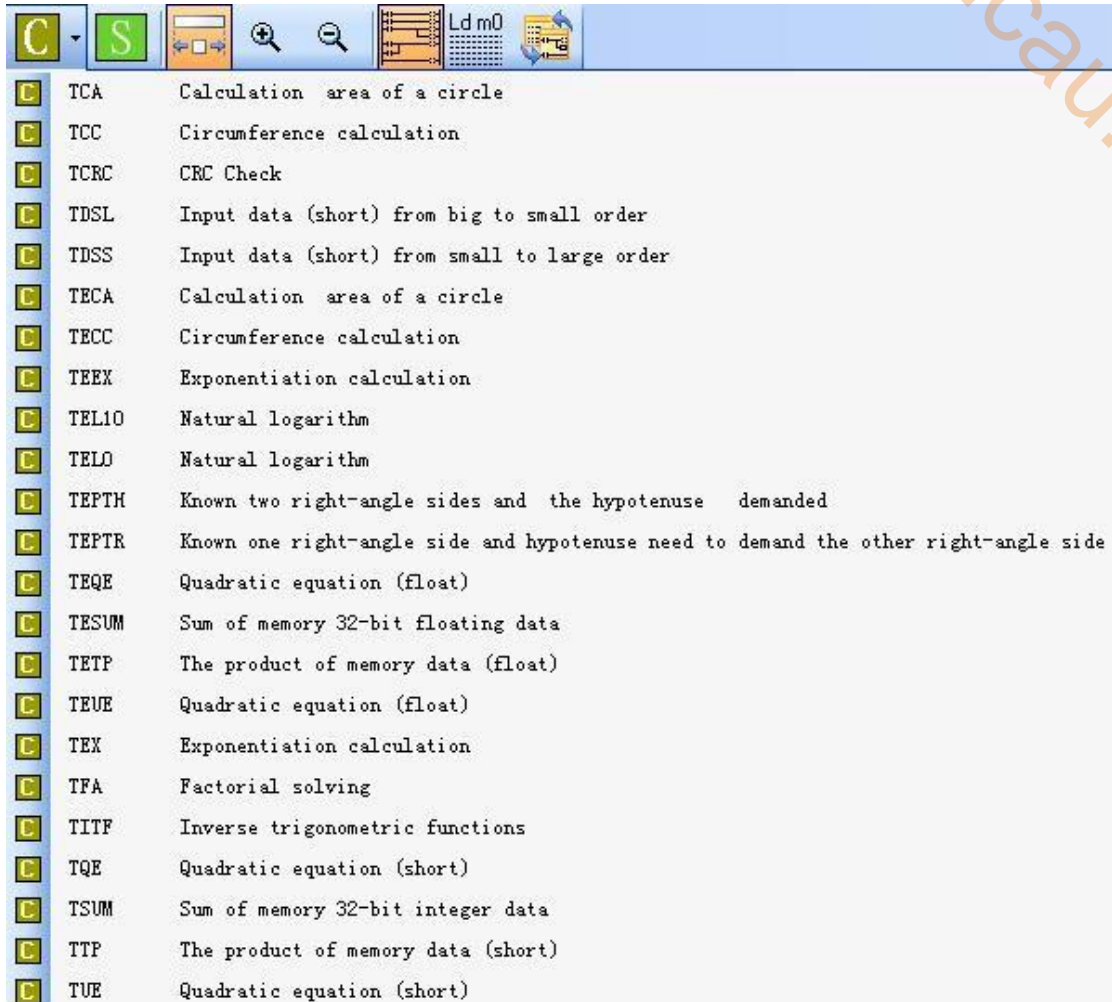
```

Ví dụ về cách sử dụng biến toàn cục kiểu cấu trúc

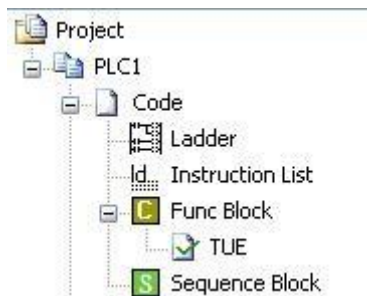
- Phần mềm XDPro v3.3 và phiên bản mới hơn lưu thư viện hàm C:



Trong khối hàm này, người dùng có thể gọi trực tiếp hàm C:



Ví dụ: nhấp vào TUE, tên hàm sẽ hiển thị trên thanh dự án:



Người dùng có thể gọi nó trong cửa sổ chỉnh sửa sơ đồ bậc thang bất cứ lúc nào.

8-10 Hỏi & Đáp về Ngôn ngữ C

(1) định nghĩa macro thứ hai cho cuộn dây

Some users have further extended the software component type after defining it, as shown in the following code:

Một số người dùng đã mở rộng hơn nữa kiểu thành phần phần mềm sau khi xác định nó, như được hiển thị trong đoạn mã sau:

```
#define SysRegAddr_HD_D_HM_M_X_Y
#define OUT Y[1]
OUT = 100;
```

Định nghĩa macro thứ hai của các cuộn dây ví dụ như Y là không được phép vì việc đọc và ghi dữ liệu cuộn dây không chỉ đơn giản là một con trỏ mà thông qua một hàm. Trong trường hợp này, trình biên dịch không thể xử lý được, dẫn đến lỗi.

(2) Sử dụng giá trị của cuộn dây làm điều kiện đánh giá

Người dùng sử dụng giá trị của cuộn dây làm điều kiện đánh giá của câu lệnh if, như trong đoạn mã sau:

```
if(X[0])D[0] =10;
```

Cách viết này sẽ báo lỗi trong quá trình biên dịch vì trình biên dịch của chúng ta đã xảy ra lỗi trong quá trình xử lý nội bộ. Bạn nên thay đổi dòng như sau: if(X[0])

```
D[0] =10;
```

(3) Sử dụng DM

DM[0] hiện không được hỗ trợ. Chỉ hỗ trợ các lệnh kiểu double word DW và FW.

(4) Đã xảy ra lỗi trong quá trình biên dịch và màu định nghĩa macro chuyển sang màu đen

Hiện tượng này xảy ra do các ký tự full-angle trong mã. Các ký tự full-angle có thể được xóa bằng cách sử dụng định dạng.

(5) Hàm trong ngôn ngữ C trong tệp tiêu đề không có chức năng biên dịch.



The screenshot shows a software interface for PLC programming. On the left, a tree view displays the project structure, including a folder named '头文件' (Header Files) containing 'HAND_A'. The main window shows the source code for 'HAND_A'. The code defines several macros and a function:

```
1 #ifndef _HAND_A_H
2 #define _HAND_A_H
3 #endif
4
5 #define Home_speed 10000
6 #define AxisEnable 1
7
8
9 INT32S ADDINGS(int x , int y) {
10
11     int max;
12     max=fabs(x)*5+fabs(y)*8;
13
14 }
15
16
17
```


Không có chức năng biên dịch trong tệp tiêu đề. Chỉ có thể biên dịch tệp nguồn. Tệp tiêu đề không thể được biên dịch riêng lẻ.

(6) Khi hai tệp nguồn gọi tệp tiêu đề, bạn chỉ cần ghi khai báo trong một tệp nguồn. Ghi vào cả 2 tệp nguồn và biên dịch đúng nhưng chương trình tải về bị sai.

Việc sử dụng `#include "xxx.h"` ngoài hàm có thể hiểu là include tệp tiêu đề này một lần nữa. Không có lỗi gì khi biên dịch một tệp nguồn riêng biệt.

Hàm của tệp tiêu đề có thể hiểu là: trình biên dịch thay thế `#include "xxx.h"` bằng các biến và hàm được khai báo trong tệp khai báo trong quá trình tiền xử lý code.

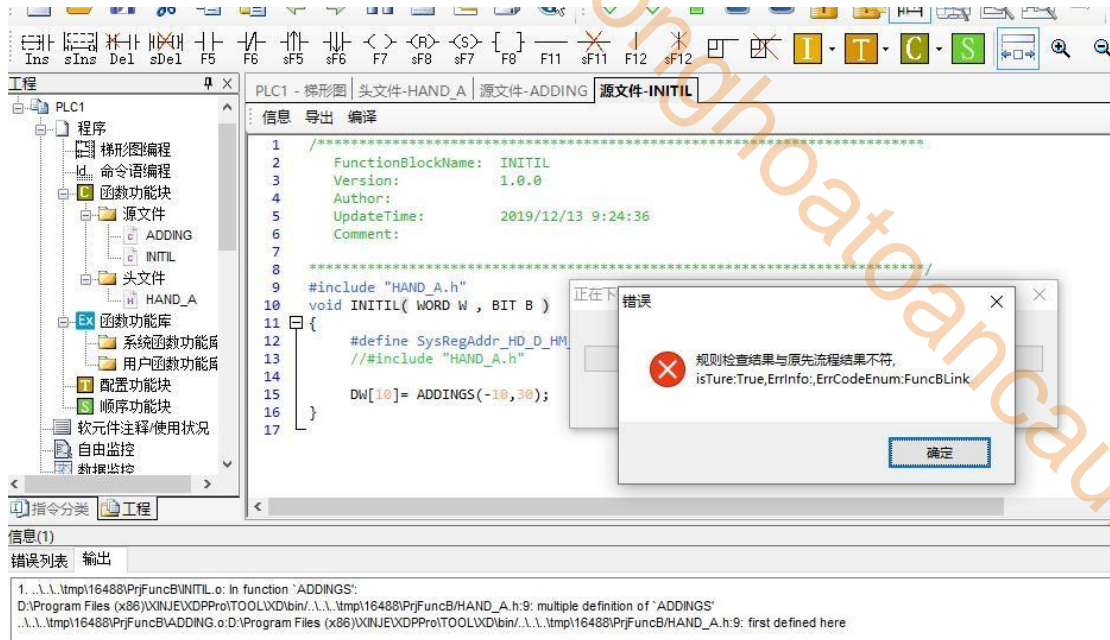
Tuy nhiên, trong quá trình tải xuống, nhiều tệp nguồn được biên dịch và liên kết. Sau khi tiền xử lý, cả hai tệp nguồn đều có khai báo biến và hàm trong tệp tiêu đề. Lỗi khai báo lặp đi lặp lại sẽ xảy ra trong quá trình liên kết và XDPpro được hiển thị dưới dạng lỗi liên kết.

Gợi ý:

Hãy include một cách chính xác tệp tiêu đề nơi nội dung của tệp tiêu đề cần được sử dụng, thay vì include tệp tiêu đề một cách tùy tiện ngay bên ngoài hàm.

```
2     FunctionsBlockName:  ADDING
3     Version:             1.0.0
4     Author:
5     UpdateTime:         2019/12/13 9:24:59
6     Comment:
7
8     *****
9     #include "HAND_A.h"
10
11 void ADDING( WORD W , BIT B )
12 {
13     #define SysRegAddr_HD_D_HM_M
14     // #include "HAND_A.h"
15
16     INT32S MUL_A(INT32S r);
17
18     DW[10]= ADDINGS(-10,30);
19     DW[14]= MUL_A(1);
20
21 }
22
23
24 INT32S MUL_A(INT32S r) {
25     INT32S mas ;
26
27     mas=ADDINGS(-10,30)+r;
28
29 }
30
```

```
3     Version:             1.0.0
4     Author:
5     UpdateTime:         2019/12/13 9:24:36
6     Comment:
7
8     *****
9     #include "HAND_A.h"
10 void INITIL( WORD W , BIT B )
11 {
12     #define SysRegAddr_HD_D_HM_M
13     // #include "HAND_A.h"
14
15     DW[10]= ADDINGS(-10,30);
16 }
17
```



8-11 Bảng các hàm

Các hàm thư viện mặc định

Hằng số	Dữ liệu	Mô tả
_LOG2	(double)0.693147180559945309417232121458	Logarit của 2
_LOG10	(double)2.3025850929940459010936137929093	Logarit của 10
_SQRT2	(double)1.41421356237309504880168872421	Căn bậc 2
_PI	(double)3.1415926535897932384626433832795	PI
_PIP2	(double)1.57079632679489661923132169163975	PI/2
_PIP2x3	(double)4.71238898038468985769396507491925	PI*3/2

Hàm xử lý chuỗi	Mô tả
void * memchr(const void *s, int c, size_t n);	Trả về vị trí c đầu tiên trong n ký tự trước vị trí s
int memcmp(const void *s1, const void *s2, size_t n);	So sánh n ký tự đầu tiên của vị trí s1 và s2
void * memcpy(void *s1, const void *s2, size_t n);	Sao chép n ký tự từ vị trí s2 sang s1 và trả về s1
void * memset(void *s, int c, size_t n);	Thay n ký tự bắt đầu từ vị trí s bằng ký tự c và trả về vị trí s
char * strcat(char *s1, const char *s2);	Nối chuỗi ct đằng sau chuỗi s
char * strchr(const char *s, int c);	Trả về vị trí ký tự c đầu tiên trong chuỗi s
int strcmp(const char *s1, const char *s2);	So sánh chuỗi s1 và s2
char * strcpy(char *s1, const char *s2);	Sao chép chuỗi s1 sang chuỗi s2

Hàm toán học Double-precision	Hàm toán học single-precision	Mô tả
double acos(double x);	float acosf(float x);	Hàm cosin nghịch đảo
double asin(double x);	float asinf(float x);	Hàm sine nghịch đảo
double atan(double x);	float atanf(float x);	Hàm tan nghịch đảo
double atan2(double y, double x);	float atan2f(float y, float x);	Hàm tan nghịch đảo của tọa độ x và tọa độ y đã xác định
double ceil(double x);	float ceilf(float x);	Trả về giá trị số nguyên nhỏ nhất lớn hơn hoặc bằng tham số x
double cos(double x);	float cosf(float x);	Hàm cosine
double cosh(double x);	float coshf(float x);	Hàm cosin hyperbol, $\cosh(x)=(e^x+e^{-x})/2$
double exp(double x);	float expf(float x);	Số mũ (e^x) của dữ liệu tự nhiên
double fabs(double x);	float fabsf(float x);	Giá trị tuyệt đối của tham số x
double floor(double x);	float floorf(float x);	Trả về số nguyên lớn nhất nhỏ hơn hoặc bằng x
double fmod(double x, double y);	float fmodf(float x, float y);	Nếu y khác 0, trả về phần dư của phép chia x/y
double frexp(double val, int *_far *exp);	float frexpf(float val, int *_far *exp);	Chia nhỏ dữ liệu nổi x thành phần định trị và số mũ $x = m \cdot 2^{\text{exp}}$, trả về phần định trị của m, lưu logarit thành exp. (mũ)
double ldexp(double x, int exp);	float ldexpf(float x, int exp);	X nhân với (2^{exp}), tức là $x \cdot 2^{\text{exp}}$.
double log(double x);	float logf(float x);	Logic logarit tự nhiên
double log10(double x);	float log10f(float x);	logarit ($\log_{10}x$)
double modf(double val, double *_pd);	float modff(float val, float *_pd);	Chia dữ liệu số thực dấu phẩy động X thành phần nguyên và phần thập phân, trả về phần thập phân, lưu phần số nguyên vào tham số ip.
double pow(double x, double y);	float powf(float x, float y);	Giá trị lũy thừa của tham số y (x^y)
double sin(double x);	float sinf(float x);	Hàm sine
double sinh(double x);	float sinhf(float x);	Hàm sin hyperbol, $\sinh(x)=(e^x-e^{-x})/2$
double sqrt(double x);	float sqrtf(float x);	Căn bậc hai của tham số X
double tan(double x);	float tanf(float x);	Hàm tan
double tanh(double x);	float tanhf(float x);	hàm tan hyperbol $\tanh(x)=(e^x-e^{-x})/(e^2+e^{-x})$

Cách sử dụng các hàm trong bảng:

float asinf(float x);

float asinf: float có nghĩa là giá trị trả về có định dạng float (định dạng số thực dấu phẩy động);

float x: float có nghĩa là tham số hình thức của hàm có định dạng float. Trong thực tế sử dụng không cần ghi float. Xem dòng 14 trong ví dụ sau:


```

9 void ZHENGXIAN( WORD W , BIT B )
10 {
11 int a;
12 float x,y,z;
13 x=FW[0];
14 y=asinf(x);
15 z=180*y/3.14159;
16 a=(int)z;
17 W[2]=a;
18 }

```

Thư viện chức năng đặc biệt cho các lệnh thanh ghi flash

Thư viện chức năng đặc biệt cho các lệnh thanh ghi flash	Giải thích
flash_copy (void *dst, void *src, size_t len);	Hàm sao chép dữ liệu vào một thanh ghi flash DST: địa chỉ bắt đầu của thanh ghi đích được sao chép vào; SRC: địa chỉ dữ liệu nguồn; Len: số byte được sao chép;
flash_move (void *dst, void *src, size_t len);	các byte sao chép của thanh ghi flash, nếu vùng đích và vùng nguồn trùng nhau, flash_Move có thể đảm bảo rằng các byte của vùng trùng lặp được sao chép vào vùng đích trước khi chuỗi nguồn bị ghi đè, nhưng nội dung nguồn sẽ bị thay đổi sau sao chép. Tuy nhiên, khi vùng mục tiêu không trùng với vùng nguồn, thì nó tương tự như hàm flash_copy DST: địa chỉ bắt đầu của thanh ghi đích được sao chép vào ; SRC: địa chỉ dữ liệu nguồn; Len: số byte được sao chép;
flash_set_int8 (void* dst, int8 data);	Thực hiện một kiểu gán cho thanh ghi flash. DST: địa chỉ bắt đầu của thanh ghi đích; Dữ liệu: các kiểu dữ liệu khác nhau;
flash_set_int16 (void* dst, int16 data);	
flash_set_int32 (void* dst, int32 data);	
flash_set_int64 (void* dst, int64 data);	
flash_set_float32(void* dst, float32 data);	
flash_set_float64(void* dst, float64 data);	

Lấy việc sao chép dữ liệu và gán của thanh ghi flash làm ví dụ minh họa cách sử dụng các hàm trong bảng hàm:

Ví dụ 1: Sao chép dữ liệu sang Thanh ghi Flash FD100

```
flash_copy ( void *dst, void *src, size_t len );
```

Void trong hàm flash_copy đại diện cho loại tham số. Trong sử dụng thực tế không cần phải viết void. Xem dòng 13 trong ví dụ sau:

```

9 void FUNC1( WORD W , BIT B )
10 {
11 #define SysRegAddr_HD_D_HM_M_FD_SFD
12 char a[8] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
13 flash_copy ( &FD[100], &a, sizeof(a) );//使用sizeof(a)计算a的长度;
14
15 }
16

```

Ví dụ 2: đặt giá trị trong thanh ghi Flash

flash_set_int16 (void* dst, int16 data);

Ưu điểm của offlash_set_int16 so với flash_copy:

Nếu sử dụng flash_copy để đặt giá trị thanh ghi flash. Nó rất bất tiện khi sử dụng. int temp_val = 1000;

flash_copy(&FD[1000], &temp_val, sizeof(temp_val));

Nếu sử dụng flash_set:flash_set_int32(&FD[1000], 1000);

Xem dòng13~18 trong ví dụ dưới đây:

```

9 void FUNC1( WORD W , BIT B )
10 {
11 #define SysRegAddr_HD_D_HM_M_FD_SFD
12 //flash_set系列函数的使用示例
13 flash_set_int8 ( &FD[104], 8 );
14 flash_set_int16 ( &FD[106], 16 );
15 flash_set_int32 ( &FD[108], 32 );
16 flash_set_int64 ( &FD[112], 64 );
17 flash_set_float32 ( &FD[120], 32.32 );
18 flash_set_float64 ( &FD[122], 64.64 );
19
20 }
21

```



Lưu ý:

(1)Hàm flash_ move yêu cầu sự hỗ trợ của phiên bản firmware PLC của máy tính thấp hơn (phiên bản firmware: v3.7.2 ngày : 20210528).

(2)Thanh ghi flash có thể được ghi khoảng 1000000 lần và mỗi lần ghi là xóa toàn bộ thanh ghi flash, việc này rất tốn thời gian. Việc ghi thường xuyên sẽ gây tổn hại vĩnh viễn cho thanh ghi flash. Vì vậy, người dùng không nên ghi thường xuyên. Cần thận sử dụng nguồn điện ở trạng thái bật bình thường và cuộn dây dao động (ví dụ: SM0, SM11) làm điều kiện vận hành.

Chương 9: Sequence BLOCK/ KHỐI Tuần tự

Chương này chủ yếu giới thiệu lệnh khối tuần tự và ứng dụng.
Lệnh Khối Tuần tự/ Sequence Block

Tên gọi nhớ	Hàm	Sơ đồ bậc thang	Chương
Sequence Block			
SBSTOP	Tạm dừng Khối		9-6-1
SBGOON	Thực thi Khối		9-6-1

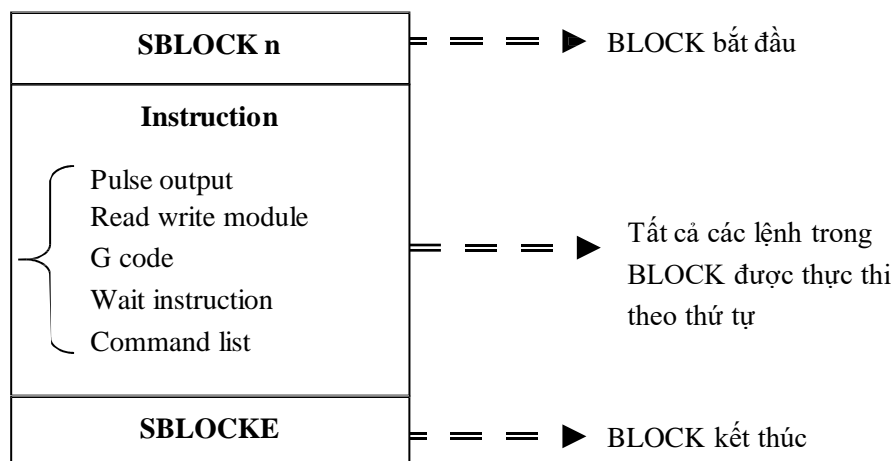
9-1 Khái niệm BLOCK/KHỐI

Khối tuần tự có tên viết tắt là BLOCK/ KHỐI là khối chương trình để thực hiện một số hàm. Là một luồng đặc biệt, tất cả các lệnh trong khối đều được thực thi theo thứ tự, đây là điểm khác biệt lớn nhất so với các quy trình chung.

BLOCK starts from SBLOCK and ends with SBLOCKE, and programmers can write BLOCK bắt đầu bằng SBLOCK và kết thúc bằng SBLOCKE, người lập trình có thể viết lệnh trong BLOCK. Nếu một BLOCK chứa nhiều lệnh đầu ra xung (hoặc các lệnh khác), thì các lệnh đầu ra xung sẽ thực thi theo thứ tự đáp ứng các điều kiện; Và trong khi đó, lệnh đầu ra xung tiếp theo sẽ không được thực thi cho đến khi lệnh hiện tại kết thúc.

PLC dòng XD3, XDM hỗ trợ đa BLOCK ※¹.

Cấu trúc BLOCK hoàn chỉnh được hiển thị như bên dưới



※1: Phiên bản firmware thấp hơn V3.4.5: PLC dòng XD cho phép tối đa tám KHỐI. Phiên bản firmware V3.4.5 trở lên: PLC dòng XD/XL có thể ghi tối đa 100 KHỐI, nhưng chỉ có thể chạy 8 đồng thời.

※2: Khi điều kiện kích hoạt của BLOCK được kích hoạt bằng việc đóng cuộn dây thường mở, nó sẽ lần lượt được thực thi từ trên BLOCK xuống dưới. Khi lệnh cuối cùng được thực thi, việc thực thi BLOCK sẽ được khởi động lại ngay lập tức từ trên xuống dưới. Khi điều kiện kích hoạt bị ngắt kết nối, BLOCK sẽ không dừng ngay lập tức mà sẽ hoàn thành lần quét cuối cùng và dừng sau khi thực thi chương trình chưa được thực thi.

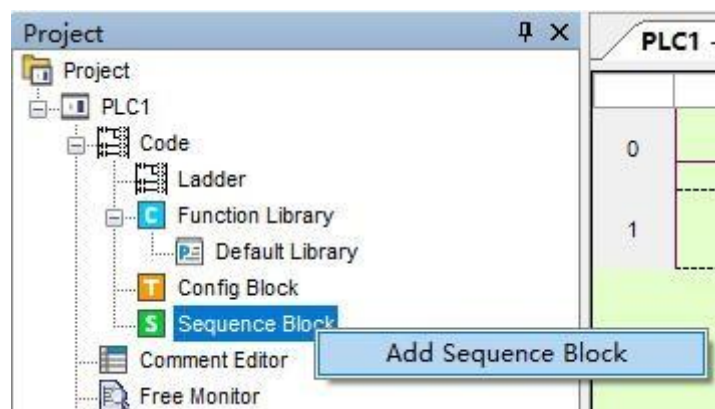
※3: Khi điều kiện kích hoạt BLOCK được kích hoạt bởi sườn lên của cuộn dây, BLOCK sẽ được thực thi một lần từ trên xuống dưới và sẽ không được thực thi theo vòng tròn.

9-2 Gọi BLOCK

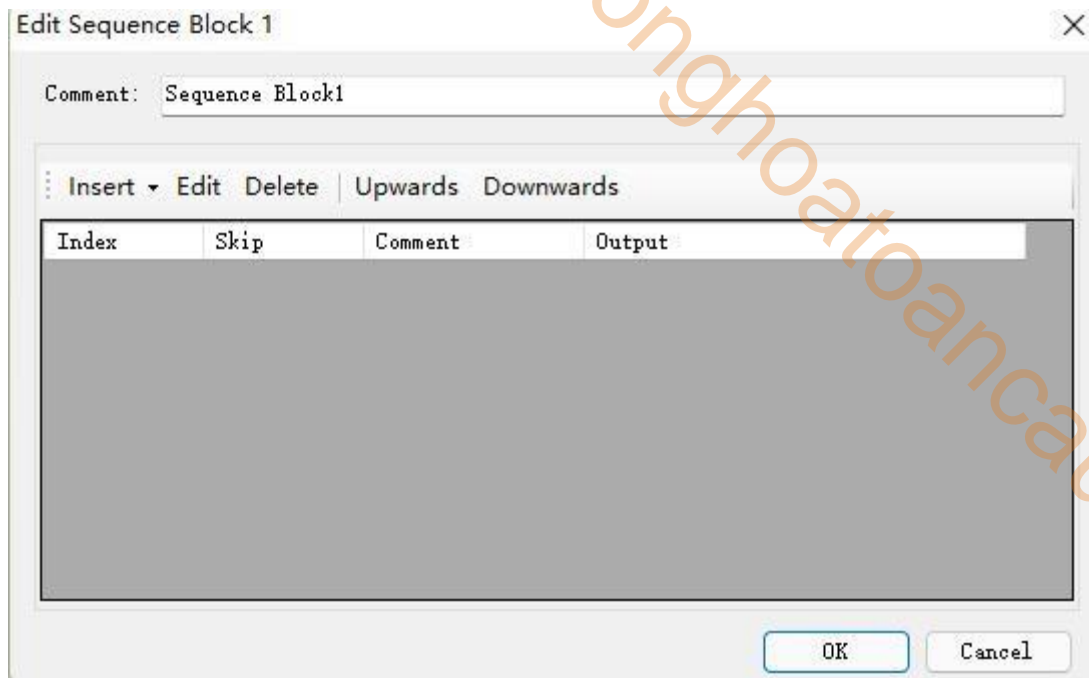
Trong một tệp chương trình, có thể gọi nhiều BLOCK; Sau đây là cách thêm BLOCK vào chương trình.

9-2-1 Thêm BLOCK

Mở phần mềm XDPro, nhấp chuột phải vào sequence block trên thanh dự án:

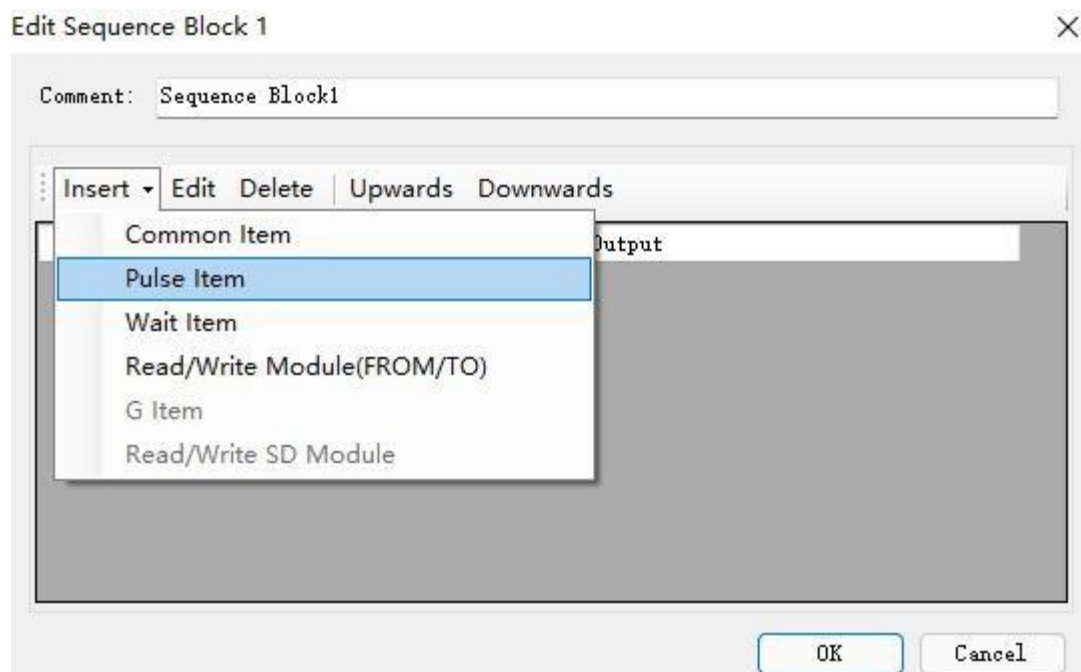


Nhấp vào lệnh 'add sequence block', cửa sổ sau sẽ xuất hiện:

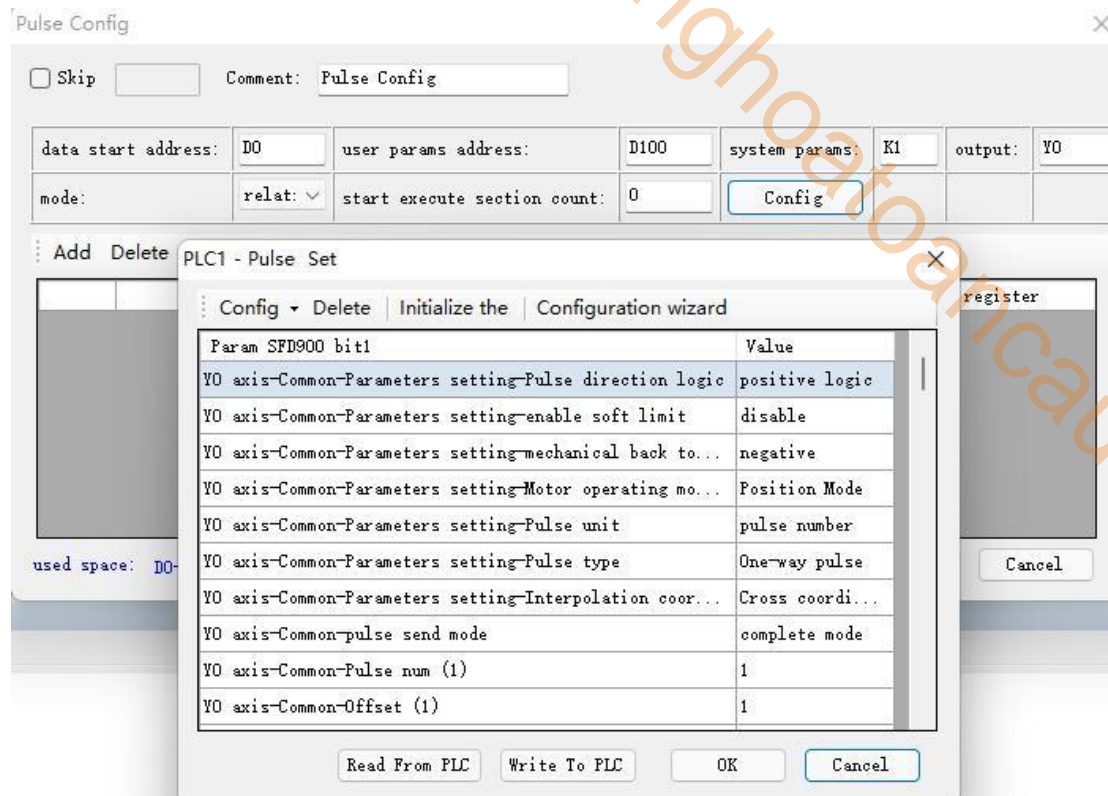


Bạn có thể chỉnh sửa BLOCK trong cửa sổ, Upward/Downward (Lên/ Xuống) được sử dụng để thay đổi vị trí các lệnh trong block.

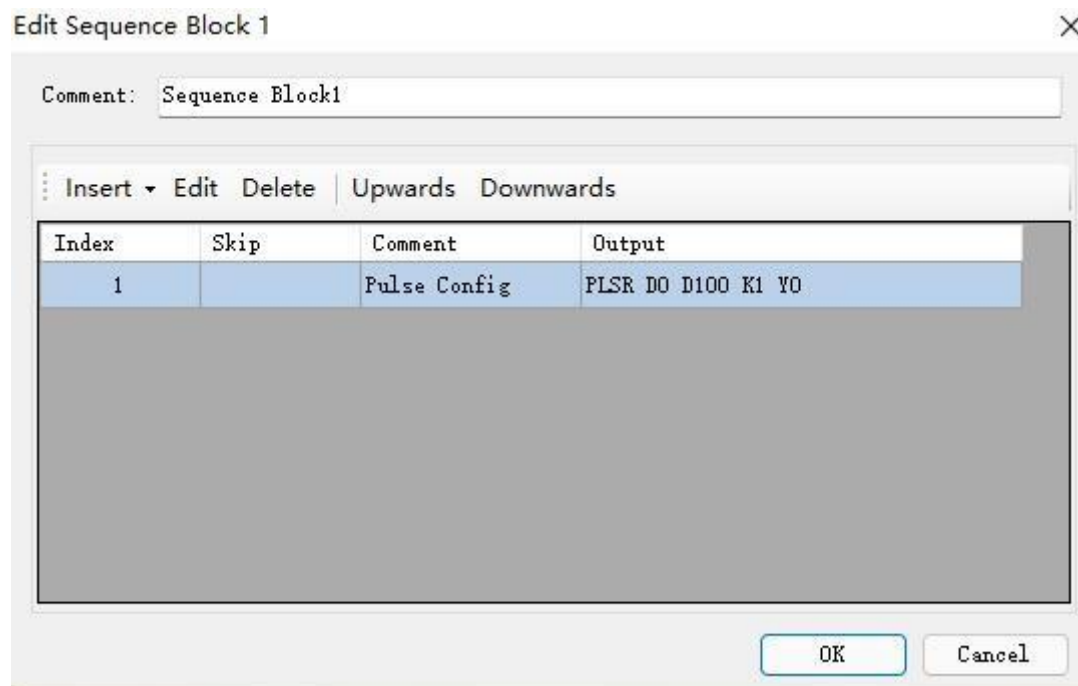
Nhấp vào 'insert', một số danh sách lệnh trong menu:



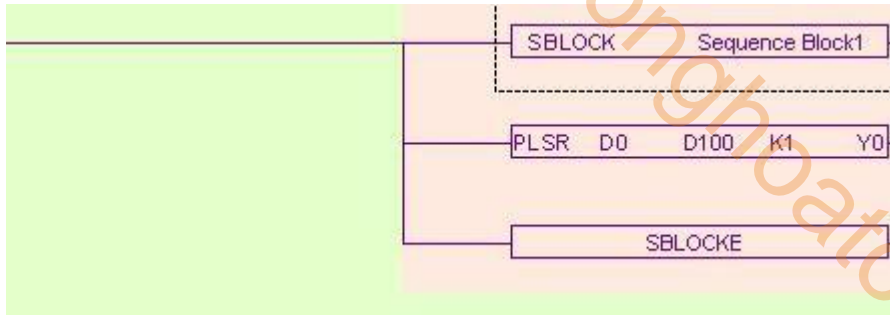
Lấy 'Pulse Item' làm ví dụ:



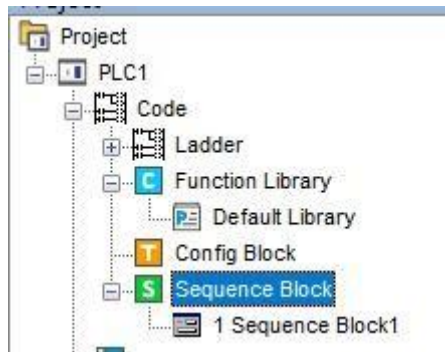
Sau khi nhấp vào 'OK', bạn sẽ tìm thấy thông tin trong cấu hình



Nhấp vào 'OK', các lệnh sau sẽ được thêm vào sơ đồ bậc thang:

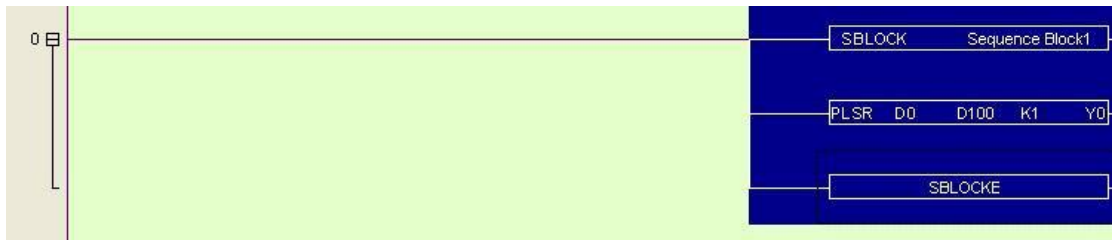


Trong khi đó, một sequence block mới được thêm vào bên trái thanh dự án:

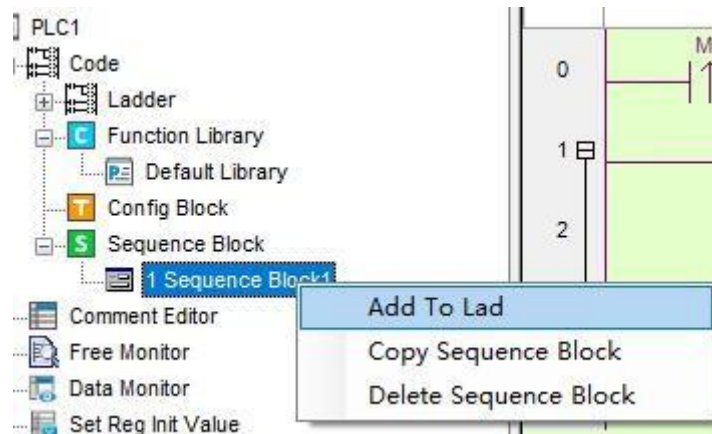


9-2-2 Di chuyển BLOCK

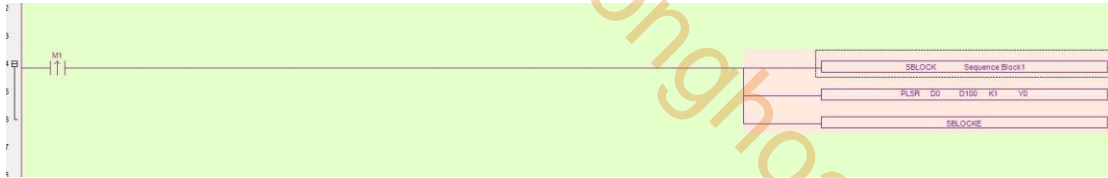
Nếu bạn muốn di chuyển BLOCK đến chỗ khác, bạn phải chọn BLOCK ban đầu và xóa nó (chọn tất cả, sau đó xóa):



Di chuyển con trỏ đến vị trí mới, sau đó nhấp chuột phải vào KHỐI và chọn “add to lad”:



Bây giờ BLOCK được chuyển đến vị trí mới:



9-2-3 Xóa BLOCK

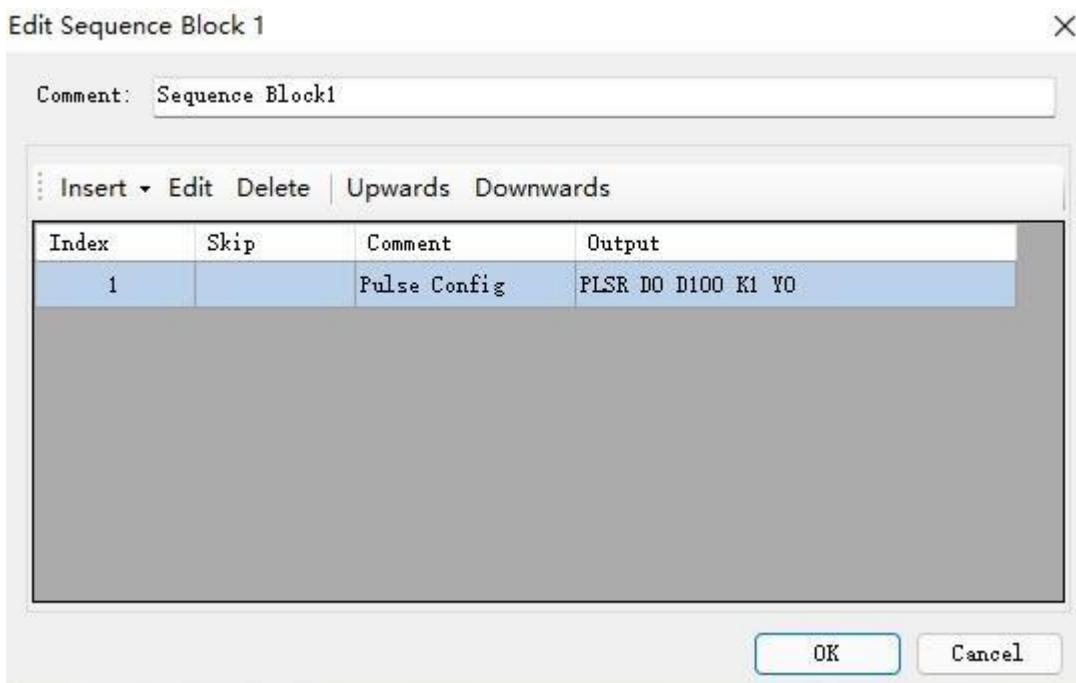
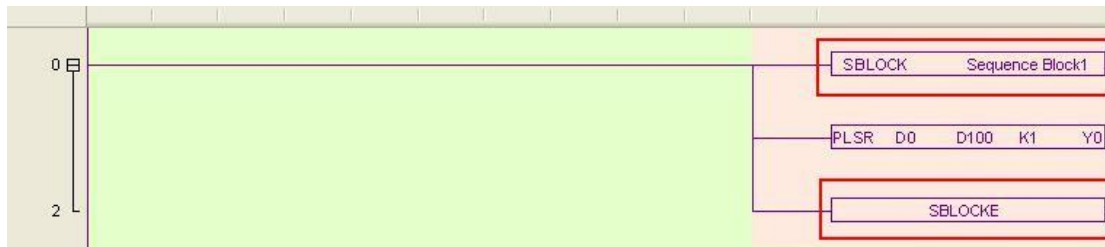
Bạn có thể chọn BLOCK được gọi và xóa nó. Nếu bạn muốn xóa hoàn toàn BLOCK, hãy nhấp chuột phải vào khối chức năng và chọn delete sequence block'. Sau khi thực hiện thao tác này, bạn không thể gọi BLOCK này nữa:



9-2-4 Sửa đổi BLOCK

Có hai cách để sửa đổi BLOCK.

(A) Nhấp đúp vào đoạn bắt đầu/kết thúc để sửa đổi BLOCK thông thường:



(B) Nhấp đúp chuột vào phần giữa để sửa đổi



Pulse Config

Skip Comment:

data start address:	<input type="text" value="D0"/>	user params address:	<input type="text" value="D100"/>	system params:	<input type="text" value="K1"/>	output:	<input type="text" value="Y0"/>
mode:	relat: <input type="text" value="v"/>	start execute section count:	<input type="text" value="0"/>	<input type="button" value="Config"/>			

... Add Delete Upwards Downwards

	frequency	pulse count	jump register

used space: D0-D9, D100-D103

9.3 Chỉnh sửa hàm của BLOCK

9-3-1 Command item

Sử dụng 'command item' để chỉnh sửa chương trình:

Edit Sequence Block 1

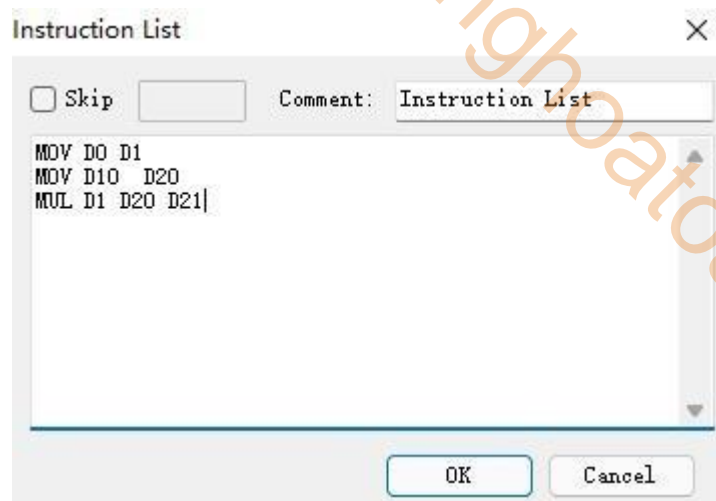
Comment:

... Insert Edit Delete Upwards Downwards

- Common Item
- Pulse Item
- Wait Item
- Read/Write Module(FROM/TO)
- G Item
- Read/Write SD Module

Output
PLSR D0 D100 K1 Y0

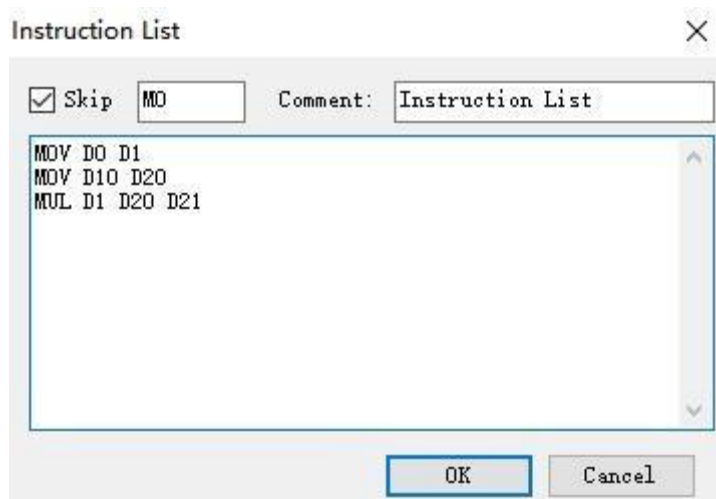
Một “danh sách lệnh/instruction list” sẽ xuất hiện sau khi nhấp vào ‘command item’:



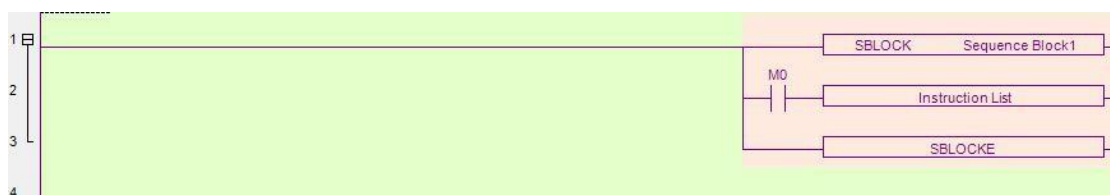
Người dùng có thể thêm lệnh vào khung.

Skip: để kiểm soát việc dừng và chạy các lệnh. Nếu bạn chọn bỏ qua và nhập cuộn điều khiển vào khung thì khi cuộn điều khiển ở trạng thái ON, lệnh sẽ không được thực thi. Nếu không chọn, hành động mặc định là thực thi.

Comment: để sửa đổi ghi chú cho lệnh.

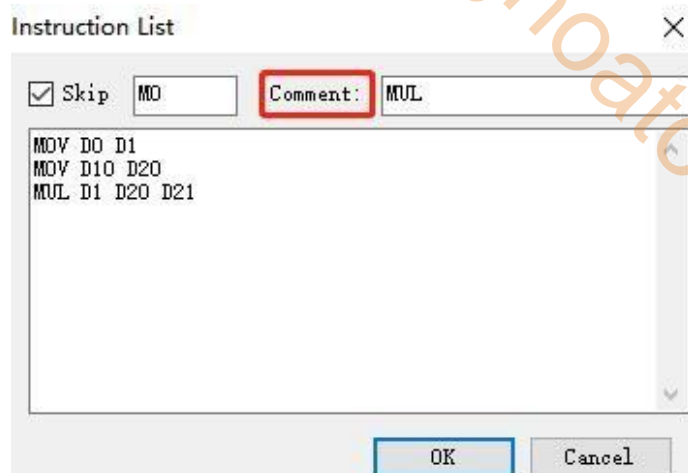


Nhấp vào ‘OK’, chương trình bậc thang sẽ thay đổi như sau:

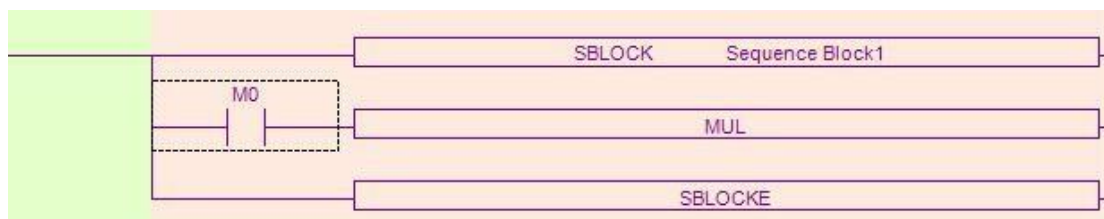


Lưu ý: Chúng ta có thể thêm nhiều lệnh trong một BLOCK và sử dụng “Skip” làm điều kiện thực hiện của mỗi lệnh.

Trong hình trên, đoạn lệnh không được mở rộng trong sơ đồ bậc thang, nhưng chú thích của nó có thể được sửa đổi theo lệnh của phân đoạn, như thể hiện trong hình sau:

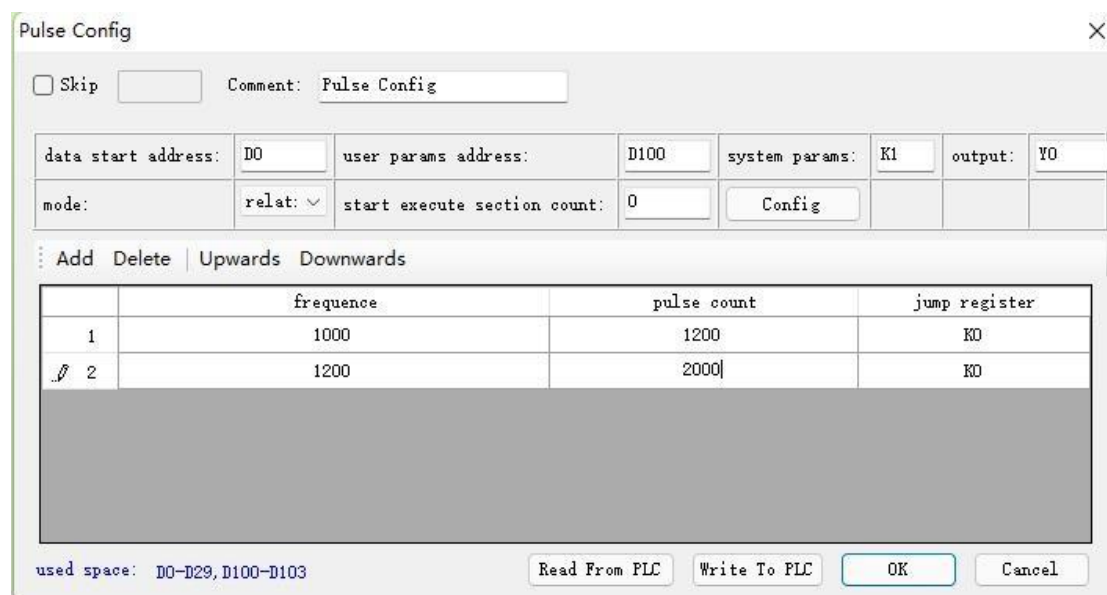


Đoạn block được sửa đổi cũng thay đổi tương ứng, như thể hiện trong hình sau:

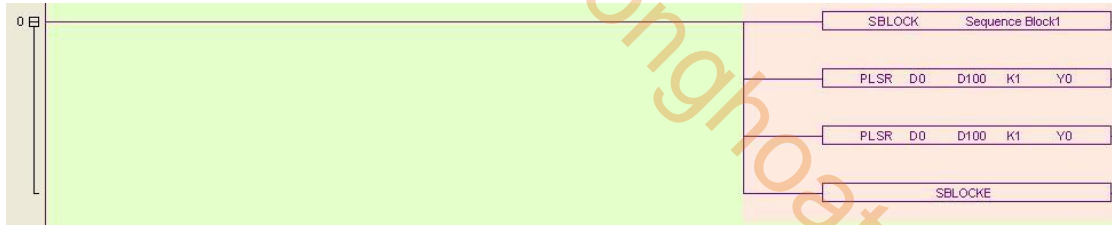


9-3-2 Pulse Item

Mô “pulse item” theo cách tương tự:



Trong BLOCK sau, chúng tôi thêm hai lệnh xung:



9-3-3 Wait Item

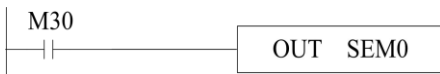
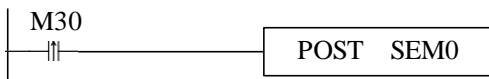
"Wait Item": để chờ cờ cuộn dây (coil flag) hoặc bit bộ định thời (timer wait)

Mở 'Wait Item' theo cách tương tự. Có hai chế độ chờ: flag bit và timer wait.

(A) Flag bit



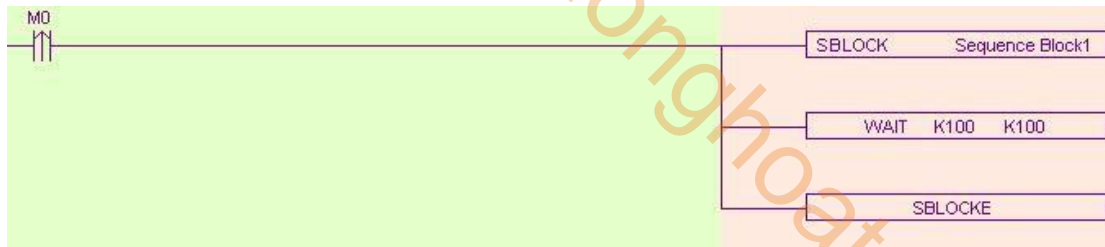
Sơ đồ bậc thang tương ứng của SEM như sau:



(B) Timer wait



(C) Sơ đồ bậc thang tương ứng:

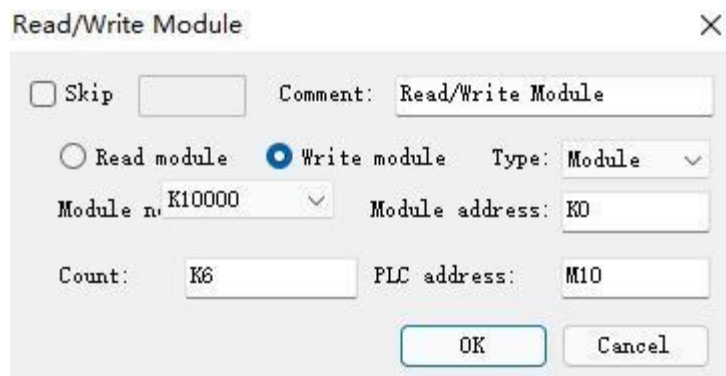


Lưu ý: Không thêm cuộn dây thông thường sau lệnh WAIT trong BLOCK tuần tự của PLC dòng XD/XL và thêm bit SEM tín hiệu đặc biệt của PLC dòng XD, XL (SEM0 ~ SEM31); SEM không thể được điều khiển bằng cách đặt hoặc đặt lại. Nó chỉ có thể được đặt bằng lệnh POST và được đặt lại bằng lệnh WAIT SEM. Hoặc xuất ra thông qua lệnh OUT. Sự khác biệt giữa chúng là lệnh POST cần được kích hoạt bởi sườn xung để giữ trạng thái SEM; lệnh OUT cần được kích hoạt bởi cuộn dây thường mở và SEM được đặt lại khi điều kiện kích hoạt bị ngắt kết nối.

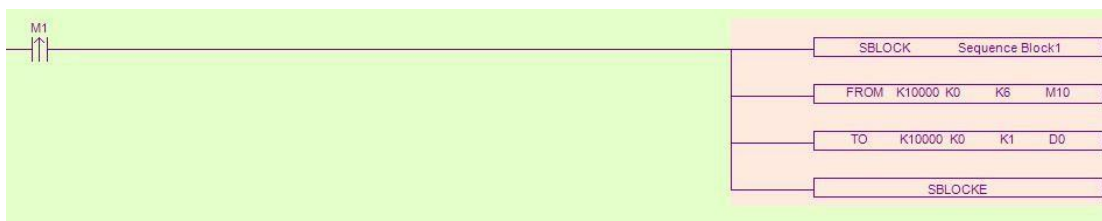
9-3-4 Lệnh đọc và ghi (FROM-TO) mô-đun

Lệnh này được sử dụng để đọc và ghi dữ liệu giữa PLC và các mô-đun và bảng điều khiển vận hành như sau:

1#read



Lệnh FROM\TO có thể được chọn từ danh sách thả xuống

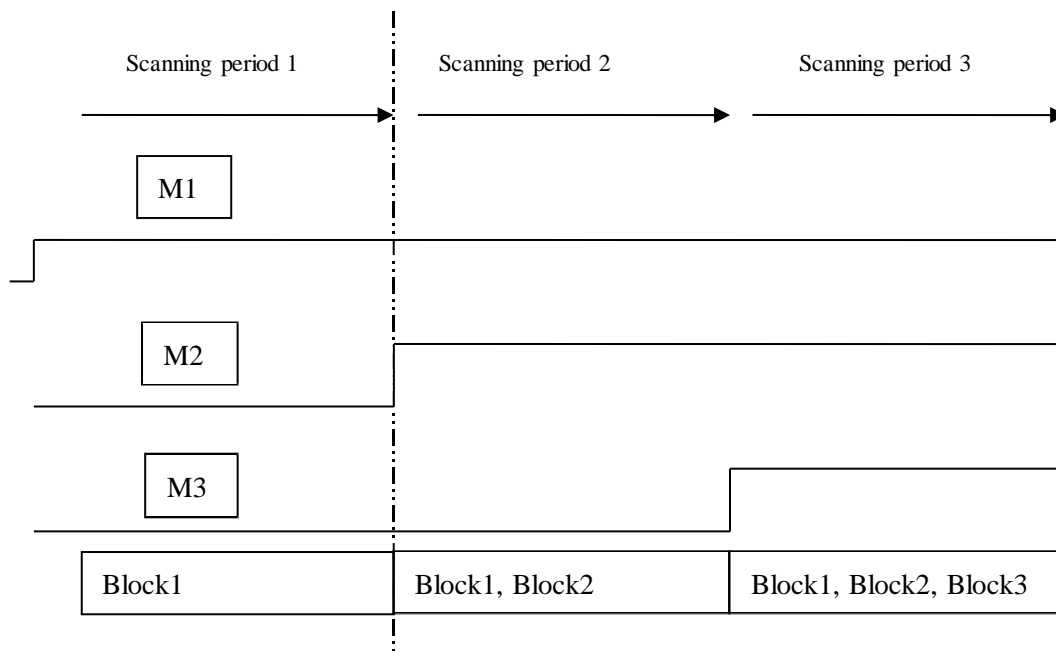
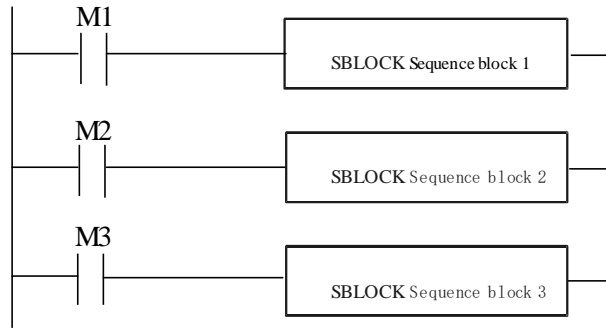


Lưu ý: Như minh họa trong hình trên, trong phần mềm phiên bản V3.4 trở lên, khi số mô-đun được đặt thành K0~K15, sơ đồ bậc thang tương ứng sẽ được hiển thị là K10000~K10015.

9-4 Phương thức chạy BLOCK

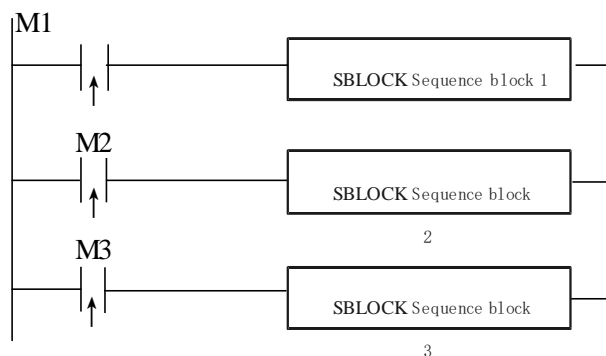
1. Nếu có nhiều khối thì chúng chạy như chương trình bình thường. Khối đang chạy khi điều kiện là ON.

(A) Điều kiện là cuộn coil thường hở, thường đóng



Lưu ý: Khi chương trình trong BLOCK không được thực thi và điều kiện kích hoạt M bị ngắt kết nối, BLOCK sẽ không dừng ngay lập tức mà sẽ hoàn thành lần quét cuối cùng và sẽ dừng sau khi phần còn lại của chương trình được thực thi.

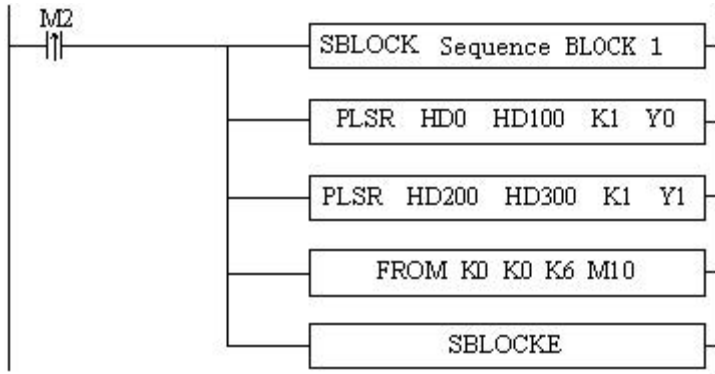
(B) Điều kiện kích hoạt là xung sườn lên hoặc xung sườn xuống



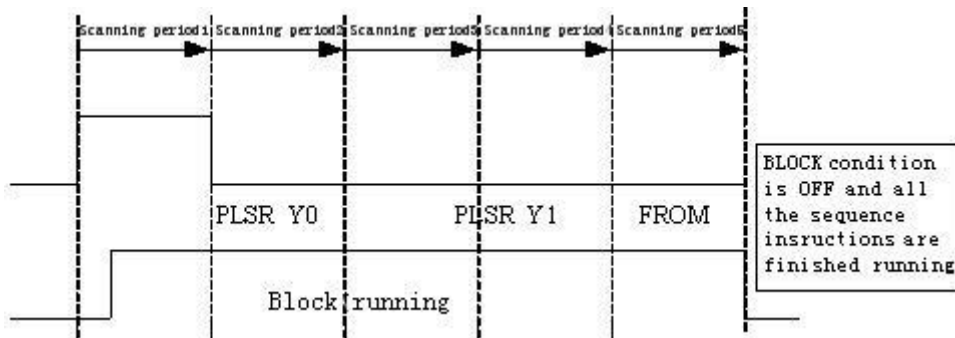
Khi M1, M2, M3 chuyển từ ON sang OFF, tất cả các khối này sẽ chạy một lần.

2. Các lệnh trong khối chạy tuần tự/ trình tự theo thời gian quét. Chúng chạy lần lượt khi điều kiện ở trạng thái ON

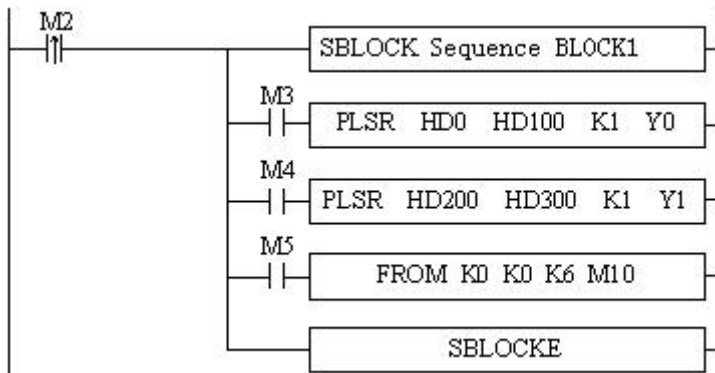
(A) Without SKIP condition



Trình tự chạy các lệnh trong khối 1 được hiển thị như sau:



(B) Với điều kiện SKIP



Giải thích:

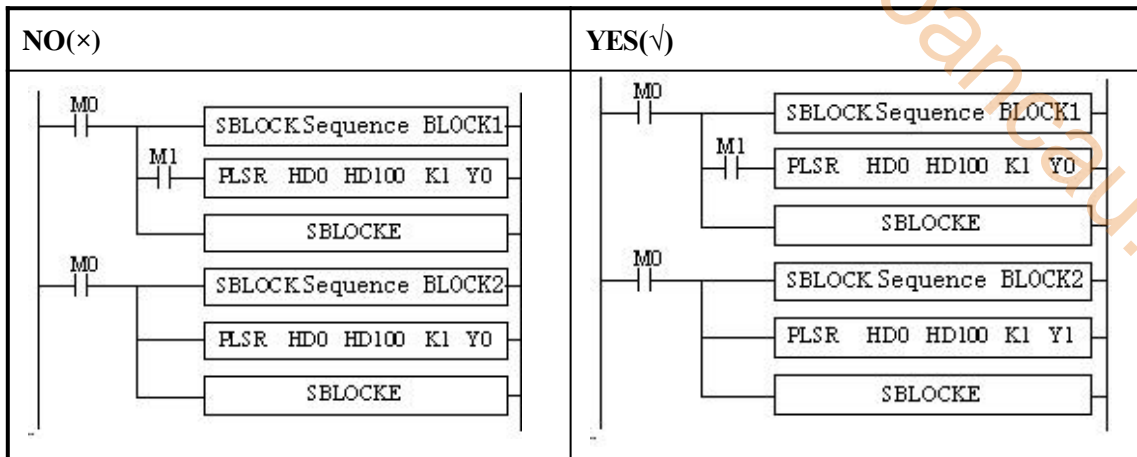
- A) Khi M2 ở trạng thái ON, khối 1 đang chạy.
- B) Tất cả các lệnh chạy theo trình tự trong khối.
- C) M3, M4, M5 are the sign of SKIP/ là dấu hiệu của điều kiện SKIP, khi ON thì lệnh này sẽ không chạy.
- D) Khi M3 ở trạng thái OFF, nếu không có lệnh nào khác sử dụng xung Y0 này, PLSR HD0 HD100 K1 Y0 sẽ chạy; nếu không, PLSR HD0 HD100 K1 Y0 sẽ chạy sau khi được giải phóng bởi các lệnh khác.

E) Sau khi hoàn tất việc phát xung Y0, hãy kiểm tra M4. Nếu M4 ở trạng thái ON, hãy kiểm tra khỏi Y1, nếu M4 ở trạng thái OFF, hãy kiểm tra M5. Nếu M5 ở trạng thái OFF, giao tiếp mô-đun sẽ chạy.

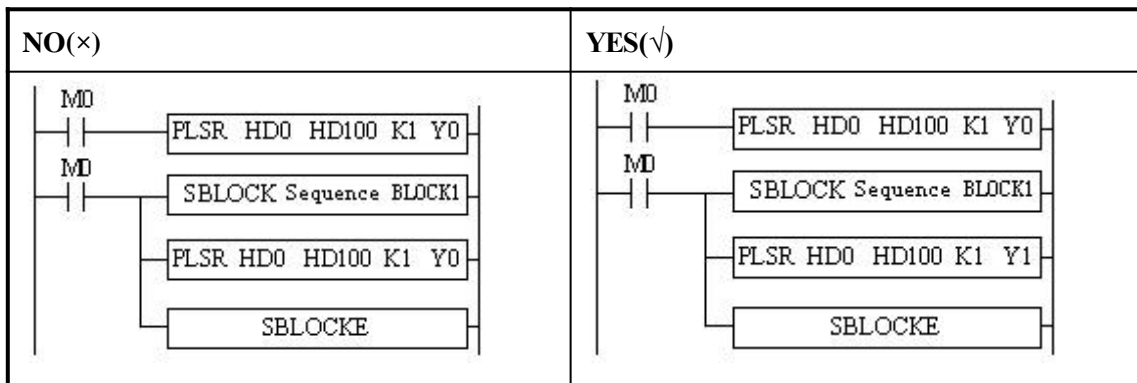
9-5 Quy tắc chỉnh sửa lệnh BLOCK

Trong BLOCK, việc chỉnh sửa lệnh phải tuân theo một số tiêu chuẩn.

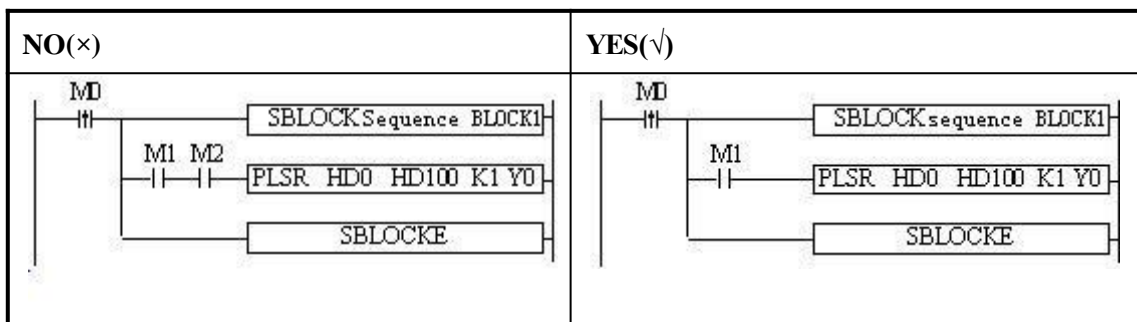
(1) Không sử dụng cùng một ngõ đầu ra xung ở các BLOCK khác nhau.



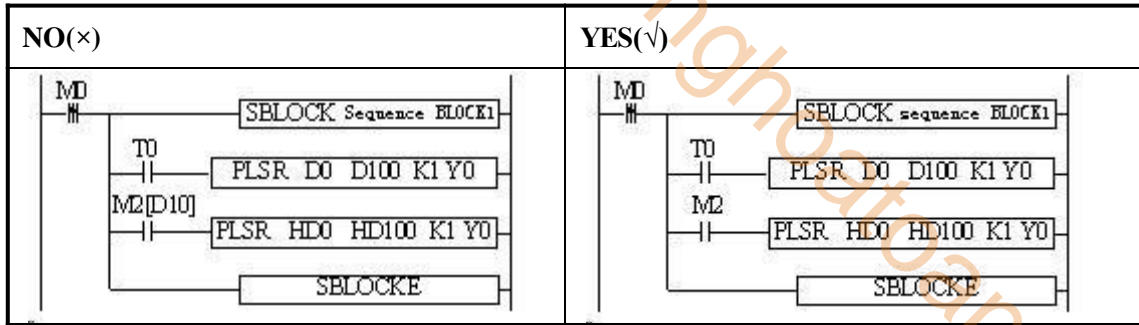
(2) Không sử dụng cùng một ngõ đầu ra xung trong BLOCK và chương trình chính.



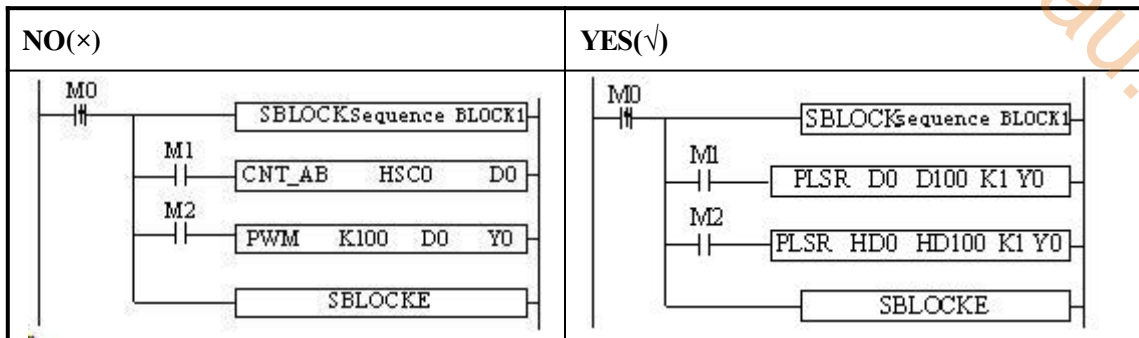
(3) Chỉ có thể có một điều kiện SKIP cho một lệnh BLOCK.



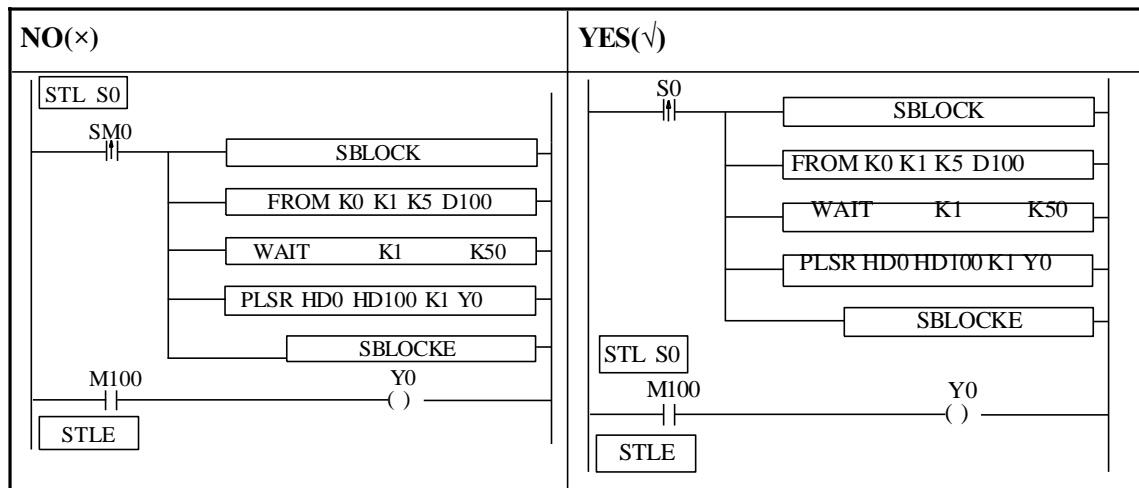
(4) Điều kiện SKIP chỉ dùng được M, X, không dùng được cuộn dây hoặc thanh ghi khác.



(5) Lệnh đầu ra không thể là CNT_AB(CNT),PWM.



(6) Không nên đặt BLOCK vào STL, vì nếu một STL kết thúc, trong khi BLOCK không kết thúc thì trục trục lớn sẽ xảy ra.



(7) Kiểu Label Kind không thể được sử dụng trong khối Ký hiệu P, tôi không thể được sử dụng trong khối. Thậm chí chúng có thể được thêm vào theo khối, nhưng trên thực tế chúng không hoạt động.

9-6 Các lệnh liên quan đến BLOCK

9-6-1 Giải thích lệnh

Lệnh ngừng chạy BLOCK [SBSTOP]

1) Sơ lược

Dừng các lệnh đang chạy trong khối

[SBSTOP]			
16 bits	SBSTOP	32 bits	-
Điều kiện thực thi	NO,NC coil and pulse edge	Model phù hợp	XD, XL
Phản cứng		Phần mềm	V3.2

2) Toán hạng

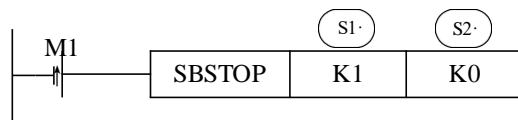
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ/ Số BLOCK	16bits, BIN
S2	Chế độ để dừng BLOCK	16bits, BIN

3) Phân tử lập trình phù hợp

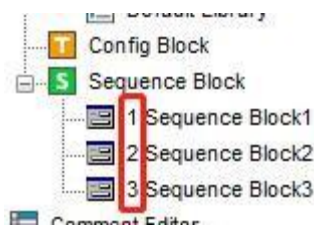
Toán hạng	Phân tử lập trình word											Phân tử lập trình bit						
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•								•									
S2									•									

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



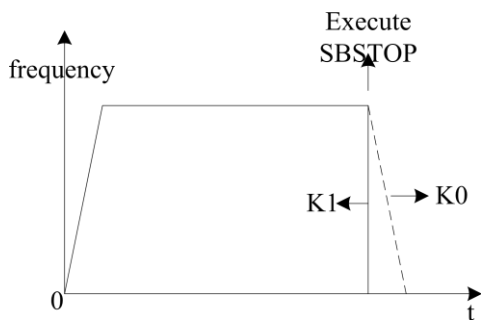
S1 là địa chỉ khối của khối tuần tự. Địa chỉ khối là duy nhất và không thể thay đổi. Nó có thể được xem trong thanh công cụ bên trái như sau.



S2 là chế độ dừng BLOCK, toán hạng: K0, K1, K2

K0: dừng BLOCK từ từ, nếu xung xuất ra, BLOCK sẽ dừng sau khi hoàn tất xuất xung.

K1: dừng BLOCK ngay lập tức; dừng tất cả các lệnh đang chạy trong BLOCK.



K2: Destructive Slow Stop BLOCK, tức là khi xung đang được phát và duy trì điều kiện SBSTOP thì xung sẽ giảm dần theo độ dốc, không sử dụng với lệnh SBGOON nên các lệnh còn lại sẽ không được thực thi. Sau khi thực hiện lệnh này, BLOCK có thể được khởi động lại. (Lưu ý: Chế độ K2 chỉ được hỗ trợ bởi PLC V3.4.2 trở lên)

Tiếp tục chạy BLOCK[SBGOON]

1) Sơ lược

Lệnh này trái ngược với SBSTOP. Dừng để tiếp tục chạy BLOCK.

[SBGOON]			
16 bits	SBGOON	32 bits	-
Điều kiện thực thi	Pulse edge	Model phù hợp	XD, XL
Phần cứng	-	Phần mềm	V3.2

2) Toán hạng

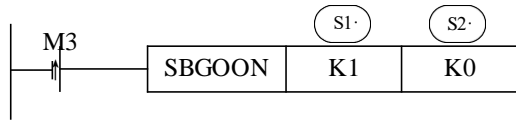
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Địa chỉ của BLOCK	16 bits, BIN
S2	Chế độ tiếp tục chạy BLOCK	16 bits, BIN

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word										Phần tử lập trình bit							
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn
S1	•							•										
S2								•										

*Lưu ý: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Chức năng



S2 là chế độ tiếp tục chạy BLOCK. Toán hạng: K0, K1.

K0: tiếp tục chạy lệnh trong BLOCK.

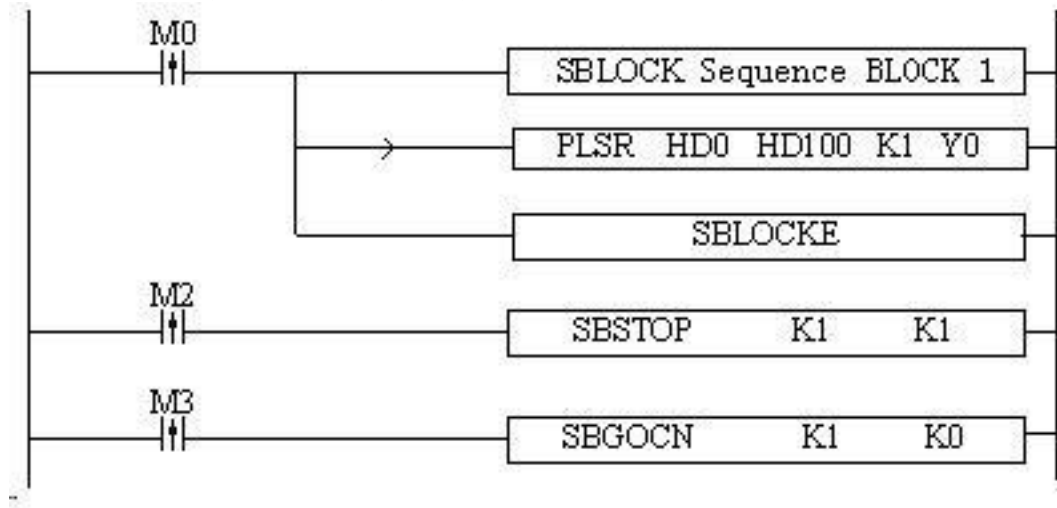
Ví dụ: nếu việc xuất xung đã dừng lần trước, SBGOON sẽ tiếp tục xuất xung còn lại;

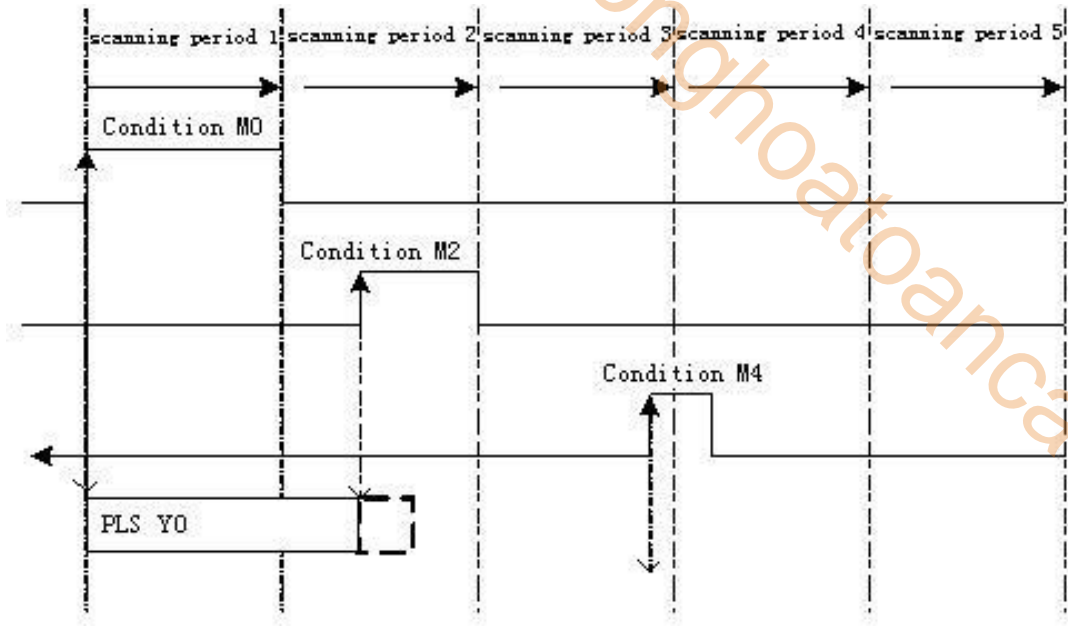
K1: tiếp tục chạy BLOCK nhưng bỏ lệnh chưa hoàn thành lần trước. Chẳng hạn như lệnh xuất xung, nếu xung chưa kết thúc ở lần trước, SBGOON sẽ không tiếp tục xuất xung này mà chuyển sang lệnh tiếp theo trong BLOCK.

Lệnh này chỉ áp dụng cho lệnh PLSR trong BLOCK và chỉ có thể phát các xung còn lại cho các lệnh nội suy không thể bỏ qua.

9-6-2 Thứ tự thời gian của các lệnh

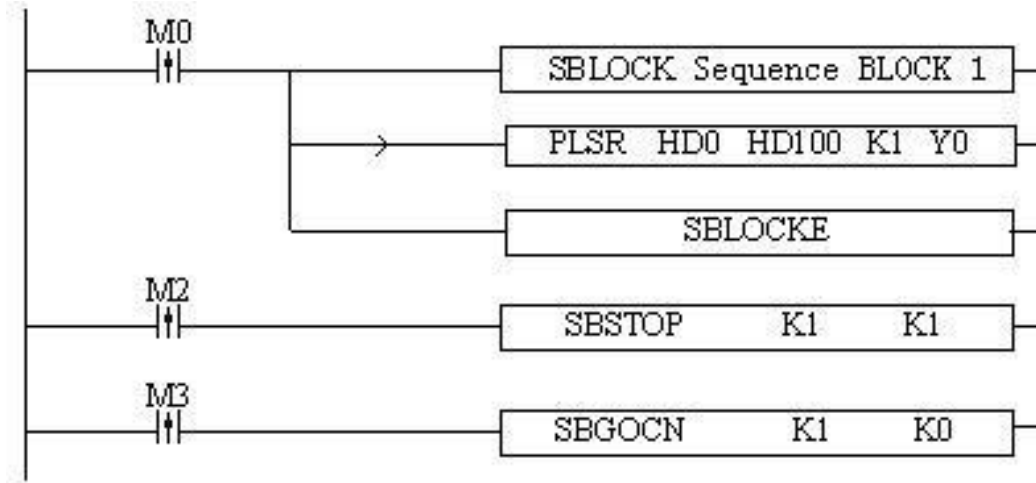
SBSTOP (K1 K1) + SBGOON (K1 K1)

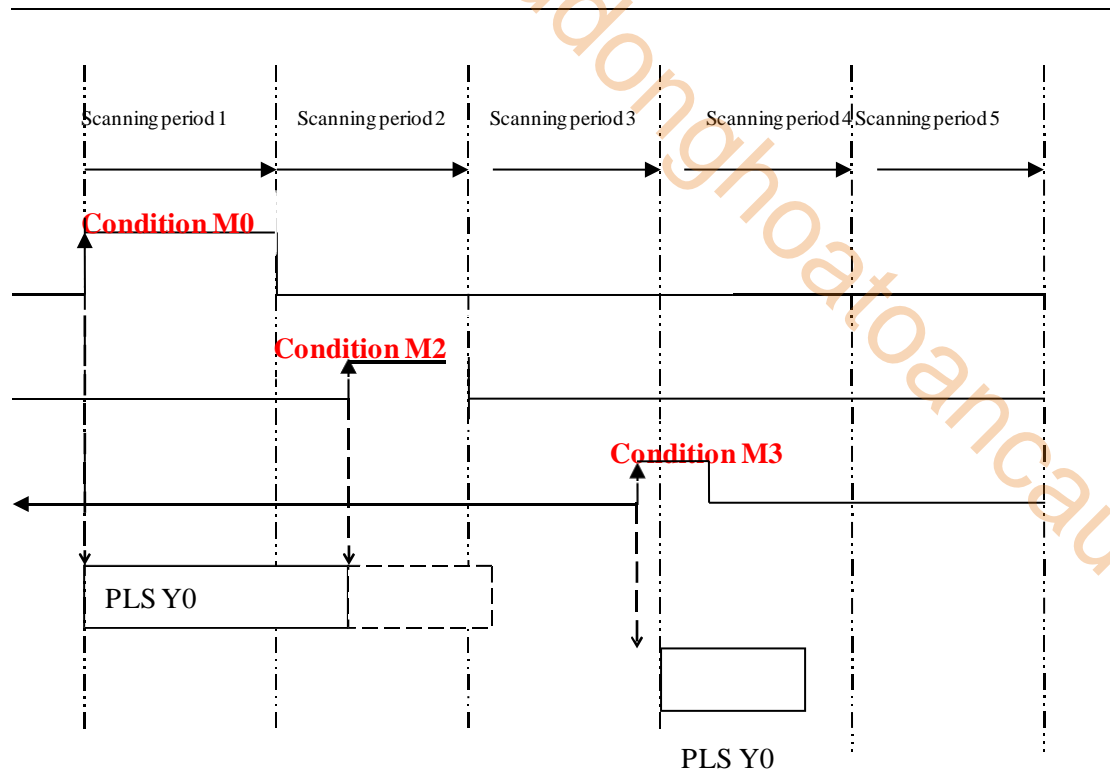




Khi M0 từ OFF→ON, hãy chạy “PLSR HD0 HD100 K1 Y0” trong BLOCK để xuất xung;
 Khi M2 ở trạng thái OFF→ON, BLOCK ngừng chạy ngay lập tức;
 Khi M4 ở trạng thái OFF→ON, hãy bỏ qua xung còn lại.

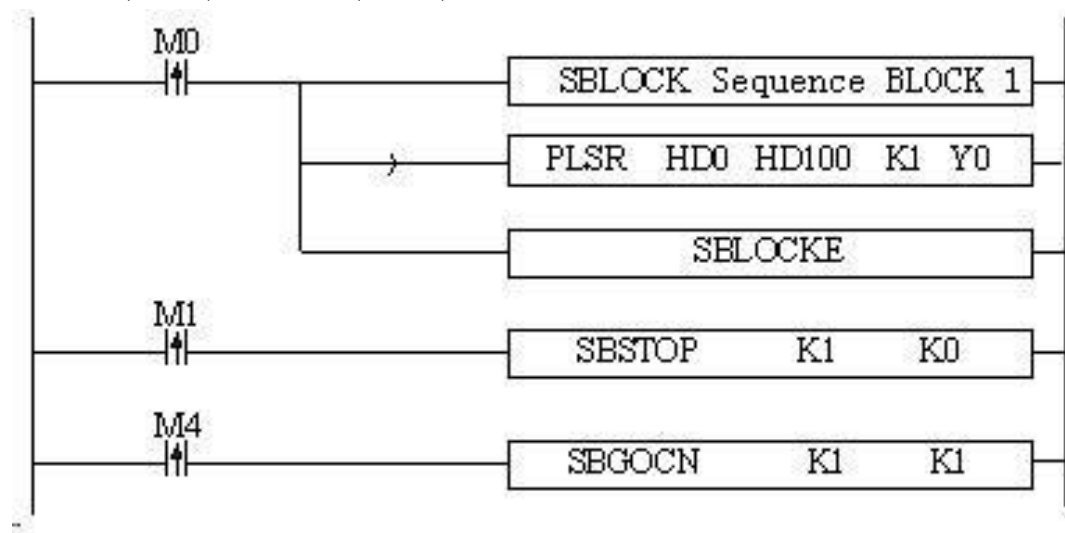
SBSTOP(K1 K1)+SBGOON(K1 K0)

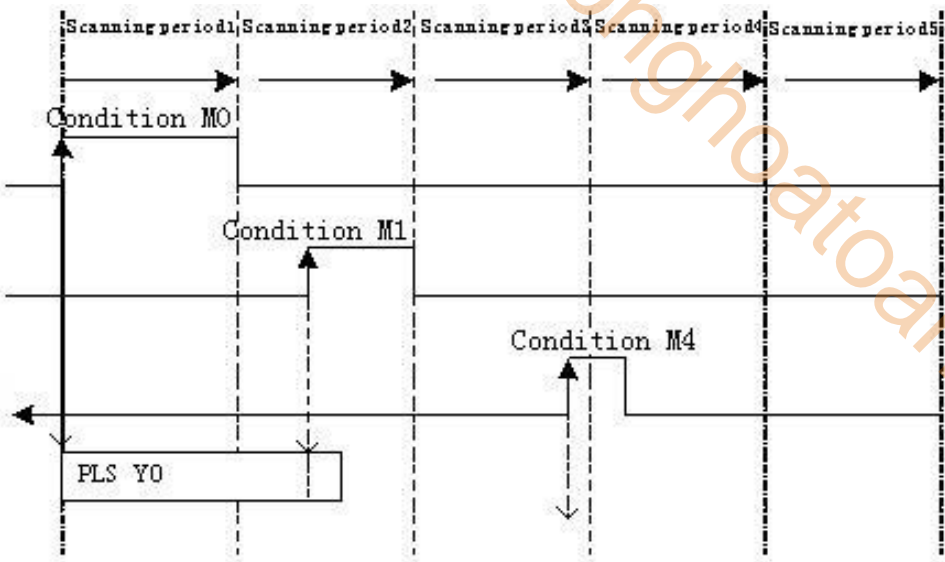




Khi M0 từ OFF→ON, hãy chạy “PLSR HD0 HD100 K1 Y0” trong BLOCK để xuất xung;
 Khi M2 ở trạng thái OFF→ON, BLOCK ngừng chạy, xuất xung dừng ngay lập tức;
 Khi M3 ở trạng thái OFF→ON, xuất xung còn lại.

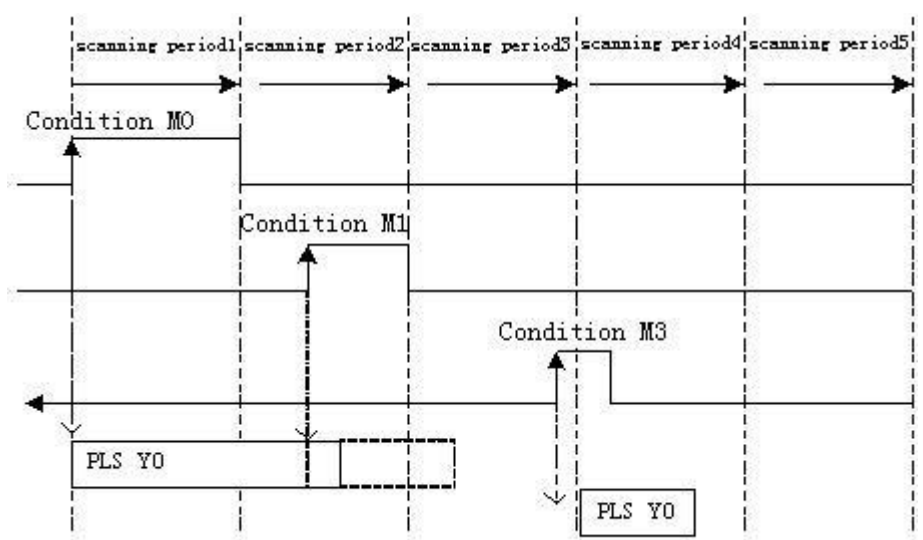
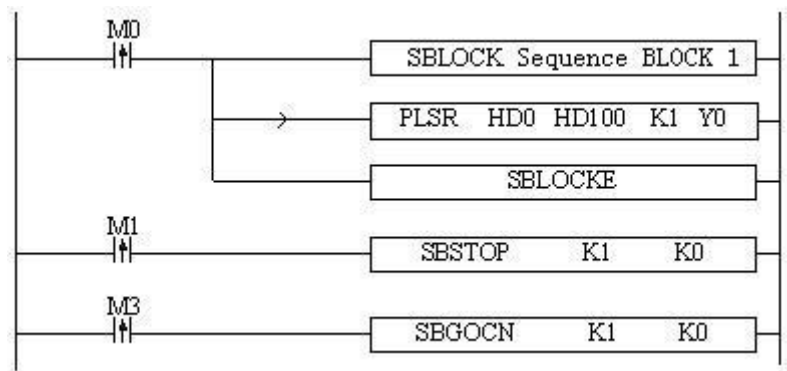
SBSTOP(K1 K0)+SBGOON(K1 K1)





Khi M0 từ OFF→ON, chạy 'PLSR HD0 HD100 K1 Y0' trong BLOCK để xuất xung;
 Khi M1 từ OFF→ON, ngừng chạy BLOCK, xung sẽ dừng chậm dần theo độ dốc;
 Khi M4 ở trạng thái OFF→ON, bỏ qua các xung còn lại.

SBSTOP(K1 K0)+SBGOON(K1 K0)



Khi M0 từ OFF→ON, hãy chạy 'PLSR HD0 HD100 K1 Y0' trong BLOCK để xuất xung;
 Khi M1 từ OFF→ON, tạm dừng chạy BLOCK, xung sẽ dừng chậm theo độ dốc;
 Khi M3 ở trạng thái OFF→ON, xuất ra các xung còn lại.

Xin lưu ý rằng lệnh SBSTOP dừng xung theo độ dốc, có thể vẫn còn một số xung; trong trường hợp này, nếu chạy lại SBGOON K1 K0, nó sẽ xuất ra các xung còn lại.

9-7 Bit cờ trạng thái và thanh ghi với BLOCK

1. Bit cờ BLOCK:

Địa chỉ	Chức năng	Giải thích
SM300	Cờ chạy của BLOCK1	1: chạy 0: dừng chạy
SM301	Cờ chạy của BLOCK2	
SM302	Cờ chạy của BLOCK3	
.....	
.....	
SM399	Cờ chạy của BLOCK100	

2. Thanh ghi BLOCK:

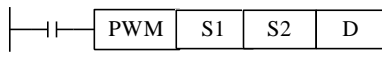
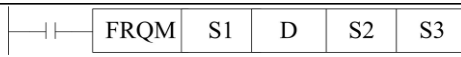
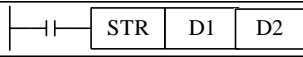



Địa chỉ	Chức năng	Giải thích
SD300	Lệnh chạy BLOCK1	BLOCK sử dụng giá trị này khi điều khiển
SD301	Lệnh chạy BLOCK2	
SD302	Lệnh chạy BLOCK3	
.....	
.....	
SD399	Lệnh chạy BLOCK100	

Nếu sử dụng GBLOCK, nó sẽ chiếm SM399 và SD399

Chương 10: Lệnh chức năng đặc biệt

Chương này chủ yếu giới thiệu về PWM (điều chế độ rộng xung), FRQM, thời gian chính xác, lệnh ngắt, v.v.

Danh sách lệnh chức năng đặc biệt:

Viết tắt	Chức năng	Mạch và phân tử lập trình	Chương
Điều chế độ rộng xung, phát hiện tần số			
PWM	Xung đầu ra với chu kỳ nhiệm vụ và tần số được chỉ định		10-1
FRQM	Đo tần số xung cố định		10-2
Thời gian			
STR	Thời gian chính xác		10-3
Lệnh ngắt			
EI	Kích hoạt lệnh ngắt		10-4-1
DI	Vô hiệu hóa lệnh ngắt		10-4-1
IRET	Kết thúc lệnh ngắt		10-4-1

10-1 Điều chế độ rộng xung [PWM]

1) Tóm tắt

Lệnh thực hiện điều chế độ rộng xung PWM

Điều chế độ rộng xung PWM [PWM]			
Lệnh 16 bit	-	Lệnh 32 bit	PWM
Điều kiện thực thi	Cuộn dây Thường hở/đóng	Model phù hợp	XD/XL (trừ XD1/XL1/XDH/XLH)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

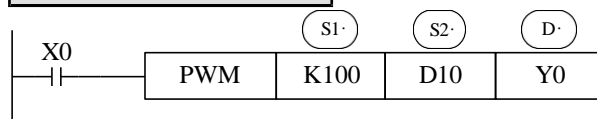
Toán hạng	Chức năng	Loại
S1	Chỉ định giá trị chu kỳ nhiệm vụ hoặc số ID của phân tử lập trình	32 bit, BIN
S2	Chỉ định tần số đầu ra hoặc số ID của phân tử lập trình	32 bit, BIN
D	Chỉ định công đầu ra xung	bit

3) Phần tử lập trình phủ hợp

Toán hạng	Phần tử lập trình Word										Phần tử lập trình Bit							
	Hệ								Hằng số	Mô đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•					•									
S2	•	•	•	•					•									
D													•					

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC.

Lệnh và Hành động



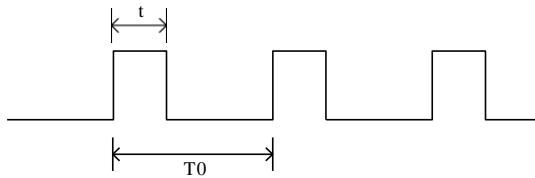
Chu kỳ nhiệm vụ n: 1~65535.

Xung đầu ra f: 1~100KHz

Đầu ra PWM của PLC dòng XD cần ngõ ra transistor:

Model PLC	Ngõ ra PWM
XD2-16T/RT -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD3-16T/RT -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD5-16T/RT -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD5-24T4 -32T4 -48T6 -60T6-60T10	Y0、Y1、Y2、Y3
XD5-24D2T2	Y0、Y2、Y4、Y6
XD5-48D4T4	Y14、Y16
XDM-24T4 -32T4 -60T4 -60T10	Y0、Y1、Y2、Y3
XDC-24T -32T -48T -60T	Y0、Y1
XD5E-30T -60T	Y0、Y1
XD5E-30T4 -60T4 -60T6 -60T10	Y0、Y1、Y2、Y3
XDME-30T4 -60T4 -60T10	Y0、Y1、Y2、Y3
XL3-16T/32T	Y0、Y1
XL5-16T/32T	Y0、Y1
XL5E-16T/32T	Y0、Y1
XL5-32T4, XL5E-32T4/64T6, XLME-32T4	Y0、Y1、Y2、Y3
XL5-64T10, XL5E-64T10, XLME-64T10	Y0、Y1、Y2、Y3

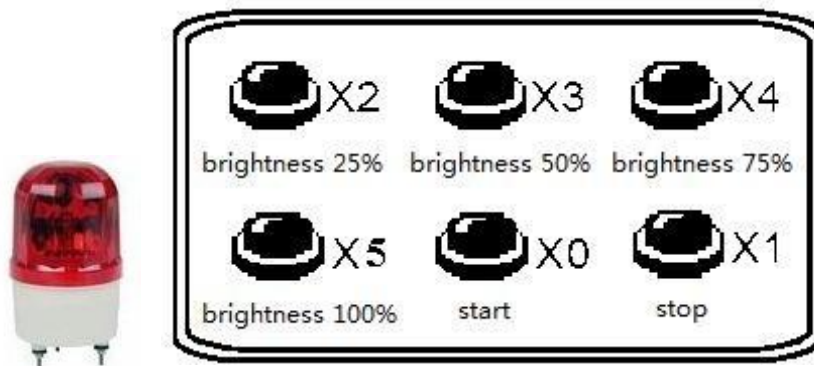
- Chu kỳ hoạt động của đầu ra **PWM** = $n / 65535 \times 100\%$
- PWM sử dụng đơn vị là 0,1Hz nên khi đặt tần số S2 thì giá trị đặt bằng 10 lần tần số thực tế (10f). Ví dụ: đặt tần số là 72 KHz, sau đó đặt giá trị trong S2 là 720000.
- Khi X0 BẬT, xuất ra sóng PWM; Khi X0 TẮT, dừng đầu ra. Đầu ra xung không có tích lũy xung.



Trong biểu đồ bên trái:
 $T0 = 1/f$
 $t/T0 = n/65535$

Lưu ý: cần kết nối điện trở khuếch đại 1K ohm giữa ngõ ra đầu ra và ngõ ra chung khi sử dụng lệnh PLC.

Ví dụ



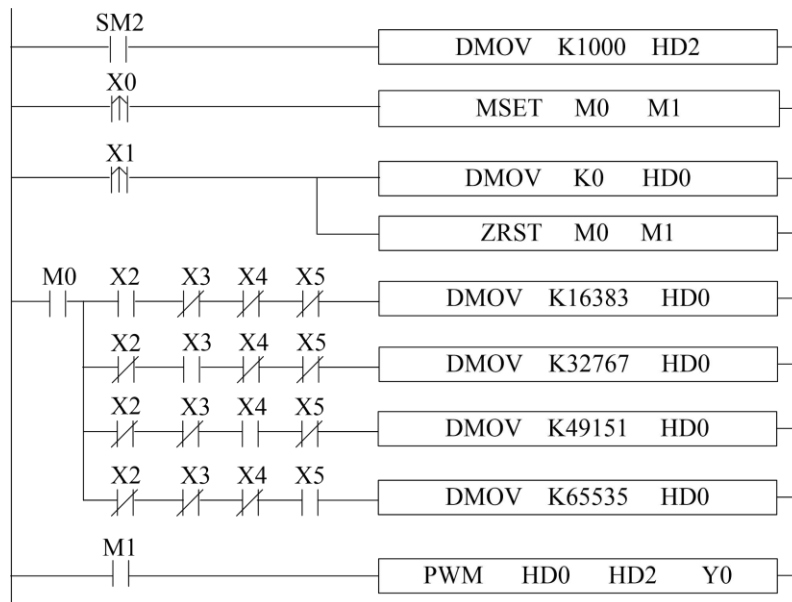
Có một đèn LED sử dụng nguồn DC24V. Cần phải kiểm soát độ sáng của đèn LED. Để giảm tổn thất điện năng của bộ thu sóng, hãy BẬT công tắc tại thời điểm nó TẮT, sau đó TẮT. Quá trình này sẽ diễn ra theo chu kỳ. Kết nối một transistor giữa nguồn điện và đèn LED. Tín hiệu xung sẽ được đưa vào từ ngõ ra transistor. Dòng điện giữa đế và bộ phát là xung. Điện áp đầu vào LED tỷ lệ thuận với tỷ lệ nhiệm vụ. Điện áp đầu vào LED sẽ được thay đổi bằng cách thay đổi tỷ lệ nhiệm vụ. Có nhiều phương pháp để thay đổi giá trị. Cách thông thường là điều chế độ rộng xung (PWM), nghĩa là chỉ thay đổi thời gian giữ BẬT nhưng không thay đổi tần số BẬT. Ví dụ này áp dụng công nghệ PWM để điều chỉnh độ sáng của đèn LED. Bộ điều khiển có thể chấp nhận tín hiệu điều khiển 24VPWM. Phạm vi độ sáng bao gồm 25%, 50%, 75%, 100%. Độ sáng được điều khiển bởi tỷ lệ nhiệm vụ của PWM.

Giải thích thành phần:

Thành phần PLC	Giải thích	Ghi chú
X0	Nút khởi động, X0 BẬT khi nhấn.	
X1	Nút dừng, X1 BẬT khi nhấn.	
X2	Nút độ sáng 25%, X2 BẬT khi nhấn	

X3	Nút độ sáng 50%, X3 BẬT khi nhấn.	
X4	Nút độ sáng 75%, X4 BẬT khi nhấn.	
X5	Nút độ sáng 100%, X5 BẬT khi nhấn.	
HD0	Thanh ghi tỷ lệ nhiệm vụ của PLC	
HD2	Thanh ghi tần số PWM	Mặc định 100Hz

Chương trình:



Giải thích chương trình:

1. HD0 sẽ điều khiển điện áp LED. Điện áp = $24 \cdot \text{HD0} / 65535$, tần số xung đầu ra là 100Hz.
2. Nhấn nút khởi động, X0 BẬT, M0, M1 BẬT, quá trình điều chỉnh độ sáng LED bắt đầu.
3. X2 đang BẬT, $\text{HD0} = 16383$, $\text{HD0} / 65535 = 0,25$, độ sáng LED là 25%.
4. X3 đang BẬT, $\text{HD0} = 32767$, $\text{HD0} / 65535 = 0,5$, độ sáng LED là 50%.
5. X4 đang BẬT, $\text{HD0} = 49151$, $\text{HD0} / 65535 = 0,75$, độ sáng LED là 75%.
6. X5 đang BẬT, $\text{HD0} = 65535$, $\text{HD0} / 65535 = 1$, độ sáng LED là 100%.
7. Nhấn nút tắt, X1 BẬT, HD0 được đặt lại, tắt điều kiện kích hoạt PWM, điện áp LED là 0V.

10-2 Lệnh đo tần suất [FRQM]

1) Sơ lược

Đo tần suất

Đo tần suất [FRQM]			
Lệnh 16 bit	-	Lệnh 32 bit	FRQM
Điều kiện thực thi	Cuộn dây Thường hồ/đóng	Model phù hợp	XD/XL (ngoại trừ XD1/XL1/XDH/XLH)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

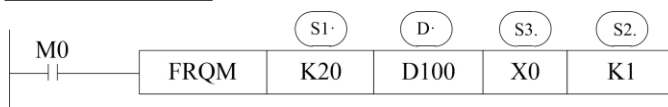
Toán hạng	Chức năng	Kiểu dữ liệu
S1	Số xung lấy mẫu	16 bits, BIN
S2	Độ chính xác hiển thị	16bits, BIN
D	Kết quả đo	32 bits, BIN
S3	Ngõ đầu vào xung	bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word									Phần tử lập trình bit								
	Hệ								Hãng số	Mô-đun		System						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
S1	•	•	•	•					•									
S2	•	•	•	•					•									
D													•					

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC

Chức năng và Hành động



- Số xung lấy mẫu có thể được điều chỉnh theo tần số, tần số càng cao thì số xung lấy mẫu càng lớn.
- Kết quả đo, đơn vị là Hz.
- Độ phân giải màn hình: chỉ có thể đặt thành 1, 10, 100, 1000, 10000.
- Khi M0 ở trạng thái ON, FRQM thu thập 20 xung từ X0 và ghi lại thời gian lấy mẫu. Kết quả số lần lấy mẫu chia cho thời gian lấy mẫu sẽ được lưu trong D100. Quá trình đo sẽ lặp lại. Nếu tần số đo nhỏ hơn phạm vi đo thì kết quả là 0.

•Độ chính xác của phép đo là 0,001%.

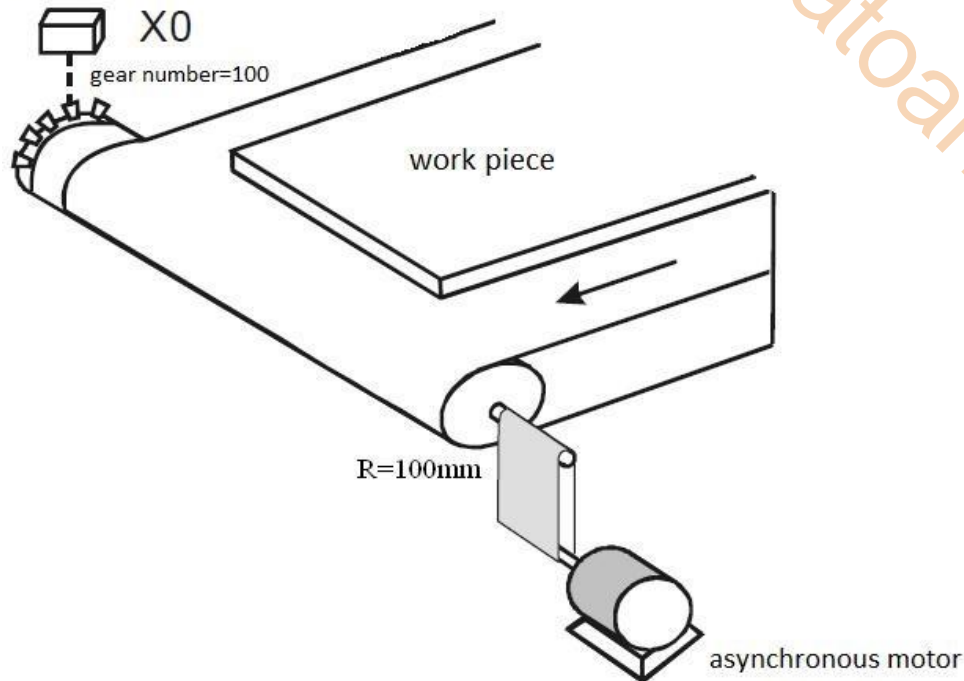
Ngõ đầu vào xung cho lệnh FRQM:

Model		Ngõ X	Tần suất tối đa (Hz)
Dòng XD2	16 I/O	X0, X3, X6	10K
	24/32/48/60 I/O	X0, X3	80K
		X6	10K
Dòng XD3	16/24/32 I/O	X0	80K
		X3, X6	10K
	48/60 I/O	X0, X3	80K
		X6	10K
Dòng XD5	16/24/32/48/60 I/O	X0, X3	80K
		X6	10K
	24T4/32T4/48T4/48D4T4/60T4	X0, X3, X6, X11	80K
	24D2T2/48T6/60T6/60T10	X0, X3, X6, X11	80K
Dòng XDM	24T4/32T4/60T4	X0, X3, X6, X11	80K
	60T10	X0, X3, X6, X11	80K
Dòng XDC	24/32/48/60 I/O	X0, X3, X6, X11	80K
Dòng XD5E	24/30/60 I/O	X0, X3	80K
		X6	10K
	30T4/60T4/60T6/60T10	X0, X3, X6, X11	80K
Dòng XDME	30T4/60T4/60T10	X0, X3, X6, X11	80K
Dòng XL3	16/32 I/O	X0	80K
		X3, X6	10K
Dòng XL5	16/32 I/O	X0, X3	80K
		X6	10K
	32T4	X0, X3, X6, X11	80K
Dòng XL5E	16/32 I/O	X0, X3	80K
		X6	10K
	32T4/64T6/64T10	X0, X3, X6, X11	80K
Dòng XLME	32T4	X0, X3, X6, X11	80K
	64T10	X0, X3, X6, X11	150K

Ví dụ

Động cơ không đồng bộ dẫn động băng tải để chuyển phôi. Nó cần hiển thị thời gian thực tốc độ di chuyển của phần công việc. Đường kính trục truyền động là 100mm, số bánh răng trên trục truyền động là 100, đơn vị tốc độ là m/phút.

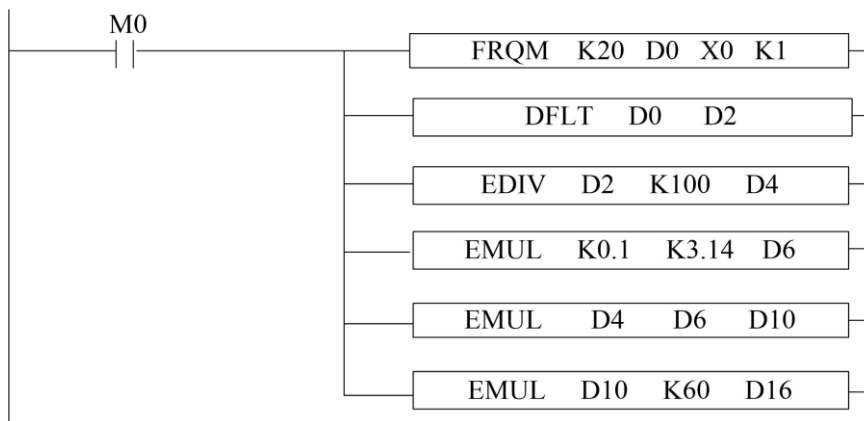
proximity switch



Giải thích thành phần

Thành phần PLC	Chức năng	Ghi chú
X0	Cảm biến tiệm cận, để đếm số bánh răng	
M0	Tín hiệu bắt đầu	
D16	Thanh ghi tốc độ (số thực dấu phẩy động)	

Chương trình:



Giải thích chương trình:

1. Đặt trạng thái ON cho tín hiệu khởi động M0 để chạy chương trình đo tần số
2. Chuyển đổi tần số thành số thực dấu phẩy động, sau đó chia cho 100 (số bánh răng)

- trên mỗi vòng quay), kết quả là số vòng quay trục mỗi giây (số thực dấu phẩy động)
- Tính đường kính của trục truyền và lưu vào thanh ghi D6 (số thực dấu phẩy động), sau đó tính khoảng cách truyền mỗi giây và lưu vào D10 (số thực dấu phẩy động).
 - khoảng cách truyền mỗi giây nhân với 60 là tốc độ (m/phút).

10-3 Lệnh thời gian chính xác[STR]

1) Sơ lược

Đọc và dừng thời gian chính xác khi thực hiện thời gian chính xác

Thời gian chính xác [STR]			
Lệnh 16 bit	-	Lệnh 32 bit	STR
Điều kiện thực thi	Kích hoạt sườn	Model phù hợp	XD/XL (except XDH,XLH)
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

Toán hạng	Chức năng	Kiểu dữ liệu
D1	Địa chỉ bộ định thời	bit
D2	Chỉ định giá trị của bộ định thời hoặc số ID của phần tử lập trình	32 bits, BIN

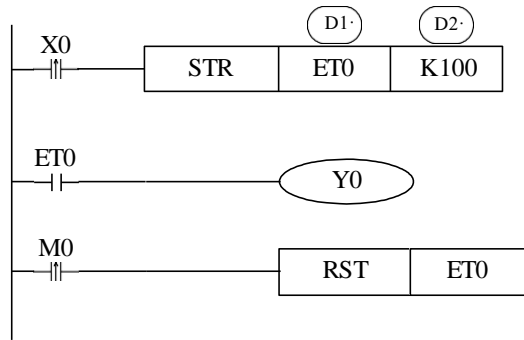
3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word									Phần tử lập trình bit								
	Hệ								Hãng số	Mô-đun		Hệ						
	D	F	T	C	D	D	D	D	K/H	I	Q	X	Y	M	S	T	C	Dn. m
D																		•
D1																		•
D2	•	•	•	•					•									

* Ghi chú: D bao gồm D, HD; TD bao gồm TD, HTD; CD bao gồm CD, HCD, HSCD, HSD; DM bao gồm DM, DHM; DS bao gồm DS, DHS. M gồm M, HM, SM; S gồm S, HS; T gồm T, HT; C gồm C, HC

Chức năng và Hành động

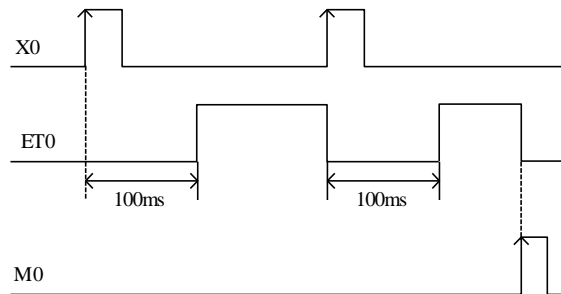
<Precise timing>, <Precise timing reset>



(D1): Địa chỉ của bộ định thời. Phạm vi: ET0~ET30 (ET0, ET2, ET4.....all number should be even)

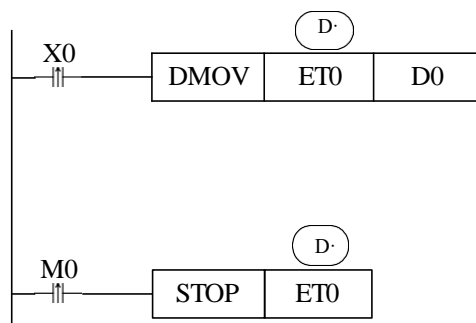
(D2): Giá trị thời gian

- Bộ định thời chính xác hoạt động theo đơn vị 1ms.
- Bộ định thời chính xác 32 bit, phạm vi đếm là 0 ~ + 2.147.483.647.
- Khi thực thi lệnh STR, bộ định thời sẽ được đặt lại trước khi bắt đầu tính thời gian.
- Khi X0 chuyển từ OFF sang ON, ET0 bắt đầu tính thời gian. ET0 sẽ được đặt lại và giữ giá trị 100 khi thời gian tích lũy đạt 100ms; Nếu X0 lại chuyển từ OFF sang ON, bộ định thời T600 chuyển từ ON sang OFF, khởi động lại theo thời gian, khi thời gian tích lũy đạt 100ms, T600 sẽ được đặt lại. Xem biểu đồ dưới đây:



Khi điều kiện trước của STR là cuộn dây thường mở/đóng, bộ định thời chính xác sẽ ON ngay lập tức khi đến **thời gian đếm (timing time)** và đặt lại thời gian, đồng thời quay vòng qua lại.

<read the precise timing>, <stop precise time>



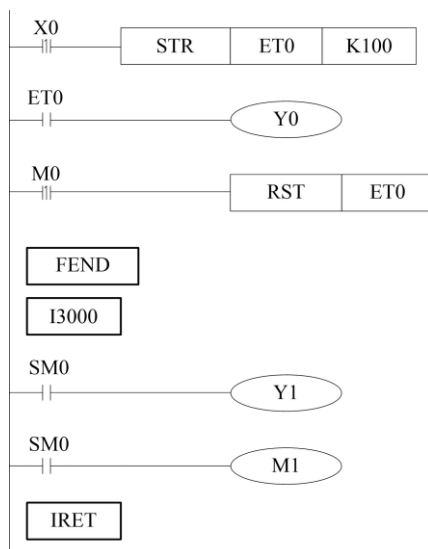
- Khi X0 thay đổi từ OFF sang ON, hãy chuyển ngay giá trị thời gian chính xác hiện tại sang D0, nó sẽ không bị ảnh hưởng bởi chu kỳ quét;
- Khi M0 thay đổi từ OFF sang ON, thực hiện lệnh STOP ngay lập tức, dừng thời gian chính xác và làm mới giá trị đếm trong ETD0. Nó sẽ không bị ảnh hưởng bởi chu kỳ quét;

Ngắt thời gian chính xác

- Khi thời gian chính xác đạt đến giá trị đếm, nó sẽ tạo ra một thẻ ngắt, chương trình con ngắt sẽ được thực thi.
- Có thể bắt đầu tính thời gian chính xác khi bị ngắt thời gian chính xác;
- Mỗi bộ định thời chính xác đều có thẻ ngắt riêng, như hiển thị bên dưới:

Thẻ ngắt tương ứng với Bộ định thời:

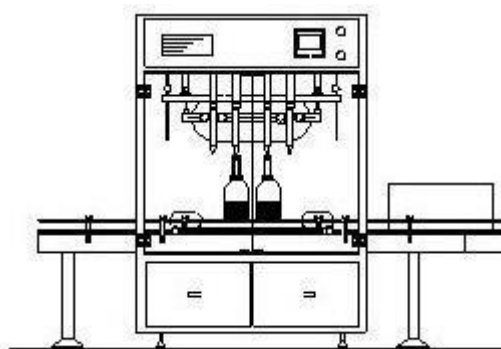
Địa chỉ bộ định thời	Thẻ Ngắt	Địa chỉ bộ định thời	Thẻ Ngắt
ET0	I3000	ET10	I3005
ET2	I3001	ET12	I3006
ET4	I3002
ET6	I3003	ET22	I3011
ET8	I3004	ET24	I3012



Khi X0 thay đổi từ OFF sang ON, ET0 sẽ bắt đầu tính thời gian. Và đặt lại ET0 khi thời gian tích lũy lên tới 100ms; Trong khi đó tạo ra lệnh ngắt, chương trình sẽ nhảy tới thẻ ngắt I3000 và thực thi chương trình con.

Ví dụ 1

Máy chiết rót điều khiển công suất chiết rót bằng cách kiểm soát thời gian mở van chất lỏng (trong ứng dụng này là 3000ms). Để cải thiện độ chính xác của công suất chiết rót, thời gian mở van chất lỏng có thể được kiểm soát bằng thời gian chính xác.

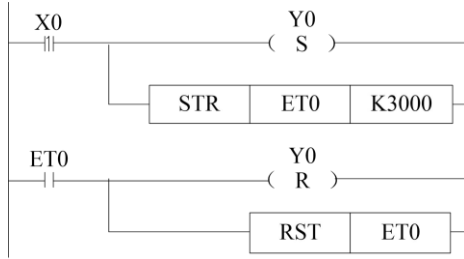


Máy chiết rót

Giải thích thành phần:

Thành phần PLC	Giải thích thành phần	Ghi chú
X0	Nút khởi động, X0 ở trạng thái ON khi nhấn nút	
ET0	Bộ định thời chính xác	
Y0	Điều khiển van chất lỏng, Y0 ở trạng thái ON khi mở van, Y0 ở trạng thái OFF khi đóng van	

Chương trình:

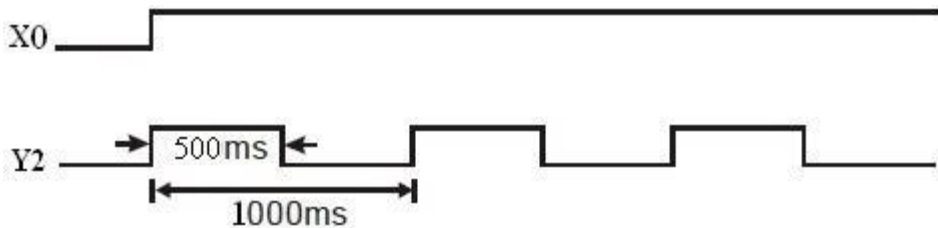


Giải thích chương trình:

1. Khi X0 ở trạng thái ON, van chất lỏng Y0 và bộ định thời chính xác ET0 mở cùng lúc.
2. Tắt van chất lỏng Y0 và bộ định thời chính xác ET0 khi đến thời điểm.

Ví dụ 2

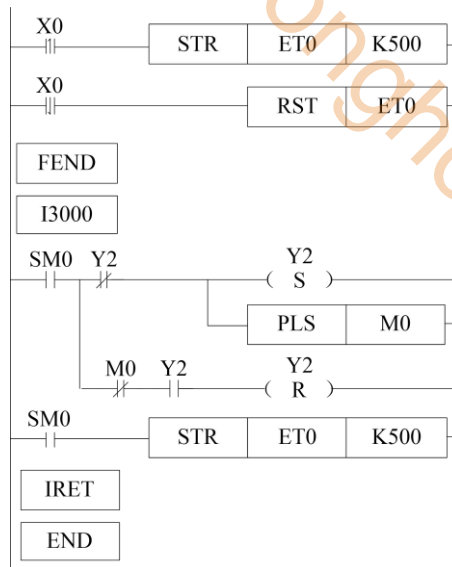
Việc ngắt bộ định thời chính xác có thể tạo ra sóng xung sau. Thời gian ON của Y2 là 500ms, chu kỳ xung là 1000ms.



Giải thích thành phần:

Thành phần PLC	Giải thích thành phần	Ghi chú
X0	Nút khởi động, X0 ở trạng thái ON khi nhấn nút	
Y2	Ngõ ra xung	
M0	Cuộn dây phụ bên trong	
ET0	Bộ định thời chính xác	

Chương trình:

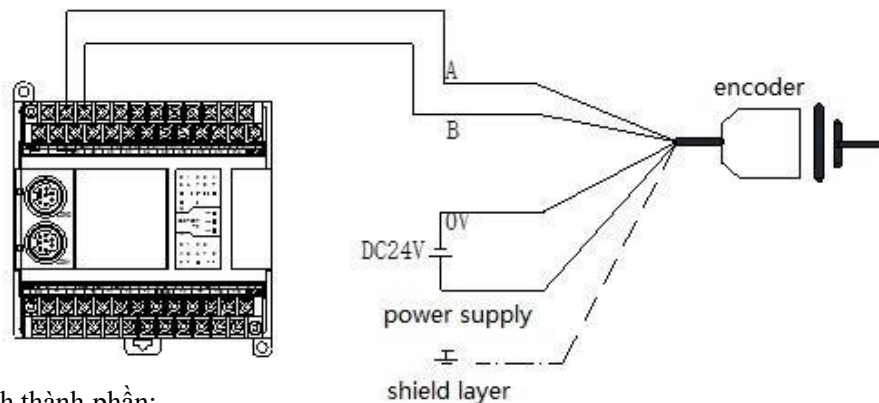


Giải thích chương trình:

1. Khi X0 ở trạng thái ON, bộ định thời ngắt chính xác sẽ hoạt động, Y2 sẽ phát sóng xung.
2. Khi X0 ở trạng thái OFF, tắt lệnh ngắt bộ định thời chính xác, Y2 dừng xuất.

Ví dụ 3

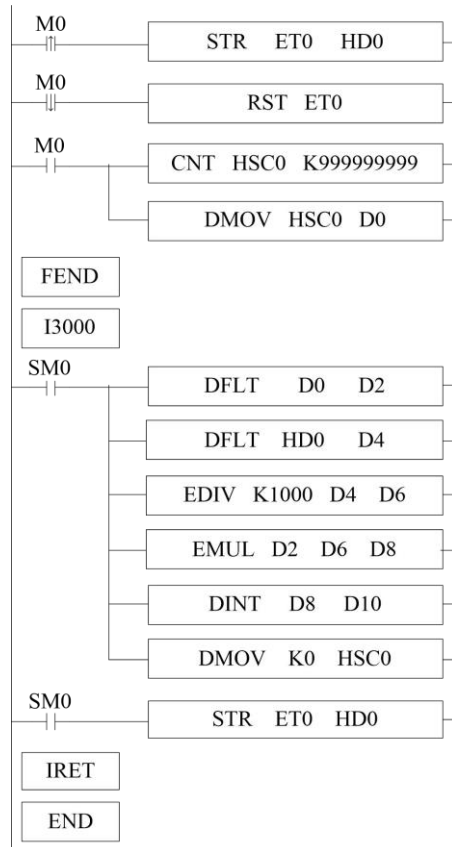
Khi lệnh FRQM tính thời gian cho số xung cố định, chúng ta sẽ thay đổi cách tính số xung theo thời gian cố định.



Giải thích thành phần:

Thành phần PLC	Giải thích thành phần	Ghi chú
M0	Nút khởi động, X0 ở trạng thái ON khi nhấn	
ET0	Bộ định thời chính xác	
HD0	Giá trị cài đặt bộ định thời chính xác (đơn vị: ms)	
HSC0	Bộ đếm tốc độ cao	
D10	Tần số đo được (đơn vị: s)	

Chương trình:



Giải thích chương trình:

1. Đặt thanh ghi chu kỳ lấy mẫu của bộ đếm tốc độ cao HD0, đơn vị là ms.
2. Set ON M0 to start the precise timer interruption and high speed counter, calculate the frequency
3. Đặt trạng thái ON cho M0 để bắt đầu ngắt bộ hẹn giờ chính xác và bộ đếm tốc độ cao, tính toán tần số
4. Dải tần số là 0-80KHz, độ chính xác là 0,005%.

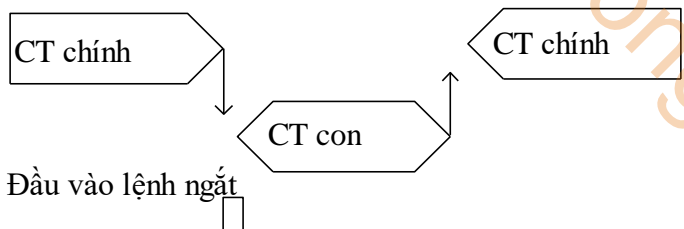
10-4 Lệnh ngắt [EI], [DI], [IRET]

PLC dòng XD/XL có chức năng ngắt, bao gồm ngắt ngoài và ngắt bộ định thời. Bằng hàm ngắt, chúng ta có thể xử lý một số chương trình đặc biệt. Hàm này không bị ảnh hưởng bởi chu kỳ quét

10-4-1 Lệnh ngắt ngoài EI

Các ngõ đầu vào X có thể được sử dụng để nhập lệnh ngắt ngoài. Mỗi đầu vào tương ứng với một lệnh ngắt ngoài. Sự lên/xuống của đầu vào có thể kích hoạt lệnh ngắt. Chương trình con ngắt được viết phía sau chương trình chính (phía sau lệnh FEND).

Sau khi tạo lệnh ngắt, chương trình chính sẽ ngừng chạy ngay, chuyển sang chạy chương trình con tương ứng. Sau khi chương trình con kết thúc, tiếp tục thực hiện chương trình chính.



Lưu ý: Không thể kích hoạt lệnh ngắt ngoài của PLC dòng XC bằng sườn lên và sườn xuống cùng một lúc; nhưng PLC dòng XD/XL hỗ trợ kích hoạt sườn lên và sườn xuống cùng lúc.

Xác định Cổng của lệnh ngắt ngoài

Dòng XD/XL 16 I/O

Ngõ vào	Địa chỉ con trở		Vô hiệu hóa lệnh ngắt
	Lệnh ngắt được kích hoạt bởi sườn lên	Lệnh ngắt được kích hoạt bởi sườn xuống	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052
X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055

Dòng XD/XL 24~64 I/O

Ngõ vào	Địa chỉ con trở		Vô hiệu hóa lệnh ngắt
	Lệnh ngắt được kích hoạt bởi sườn lên	Lệnh ngắt được kích hoạt bởi sườn xuống	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052
X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055
X10	I0600	I0601	SM056
X11	I0700	I0701	SM057
X12	I0800	I0801	SM058
X13	I0900	I0901	SM059

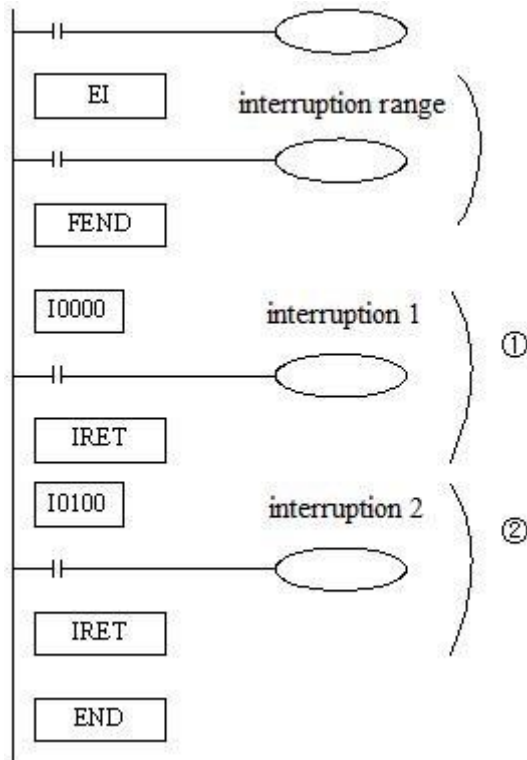
Dòng XD 10 I/O

Ngõ vào	Địa chỉ con trở		Vô hiệu hóa lệnh ngắt
	Lệnh ngắt được kích hoạt bởi sườn lên	Lệnh ngắt được kích hoạt bởi sườn xuống	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052

Lưu ý: khi cuộn cảm lệnh ngắt ở trạng thái ON, lệnh ngắt ngoài sẽ không thực thi.

Lệnh ngắt

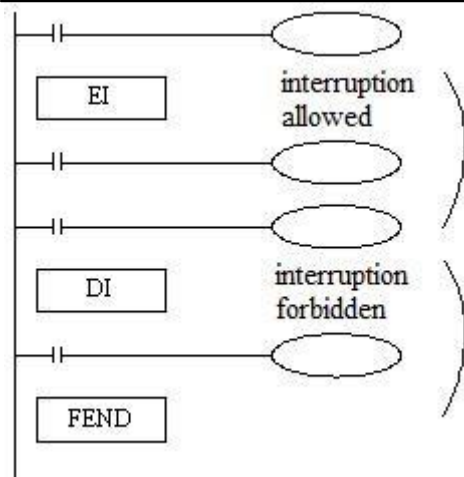
Kích hoạt ngắt[EI], Vô hiệu hóa ngắt[DI], Trở về từ lệnh ngắt[IRET]



- Nếu sử dụng lệnh EI để cho phép ngắt thì khi quét chương trình, nếu đầu vào ngắt thay đổi từ ON sang OFF thì thực hiện chương trình con ①、②. Quay trở lại chương trình chính ban đầu.
- Con trở ngắt (I****) phải ở sau lệnh FEND;
- PLC thường ở trạng thái cho phép ngắt.

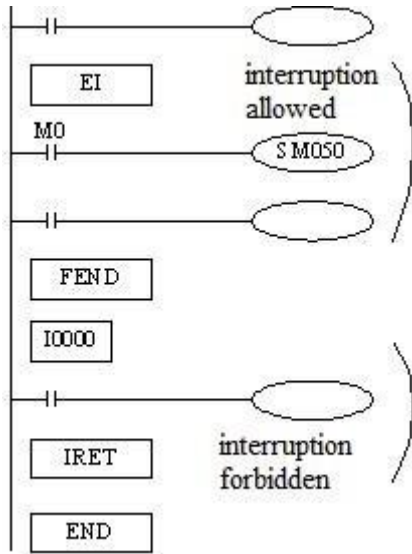
Lưu ý: Trong chương trình con ngắt, chỉ có thể ghi các lệnh đơn giản như đặt, đặt lại, truyền và chạy, các lệnh này có thể được thực thi trong một chu trình quét. Các lệnh khác như phát xung, định thời gian (ngoại trừ bộ định thời chính xác), giao tiếp và các lệnh khác cần được thực thi liên tục đều không được hỗ trợ.

Giới hạn phạm vi ngắt



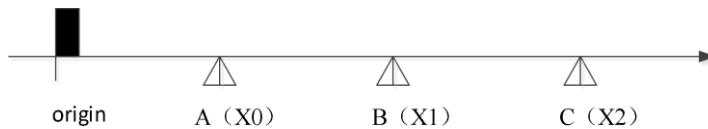
- Bằng cách lập trình lệnh DI, có thể đặt vùng vô hiệu hóa lệnh ngắt;
- Cho phép đầu vào lệnh ngắt giữa EI~DI
- Nếu không cần phải ngừng ngắt, vui lòng chỉ lập trình với lệnh EI và không cần lập trình lệnh DI.

Vô hiệu hóa lệnh ngắt



- Mỗi đầu vào lệnh ngắt đều được trang bị các rơ-le đặc biệt (SM50~SM69) để vô hiệu hóa lệnh ngắt.
- Trong chương trình bên trái, nếu sử dụng M0 để đặt SM50 ở trạng thái “ON”, thì vô hiệu hóa ngắt 0.

Ví dụ 1



Vị trí của A, B, C là các biến. Tốc độ của ba đoạn là khác nhau. Ứng dụng có thể được thực hiện bằng lệnh PLSF và lệnh ngắt ngoài. Chúng ta có thể đặt 3 cảm biến tiệm cận ở vị trí A, B, C và kết nối tín hiệu với đầu vào PLC X0, X1, X2. (giả sử X0, X1, X2 là ngõ ngắt ngoài, ID ngắt sườn lên liên quan là I0000, I0100, I0200. Ngõ ngắt ngoài PLC, vui lòng tham khảo “xác định ngõ ngắt ngoài”). Ngõ xung là Y0, ngõ chiều hướng là Y2. Để cải thiện độ chính xác của việc thay đổi tốc độ, thời gian tăng tốc và giảm tốc là 0. Tốc độ sẽ thay đổi do lệnh ngắt ngoài.

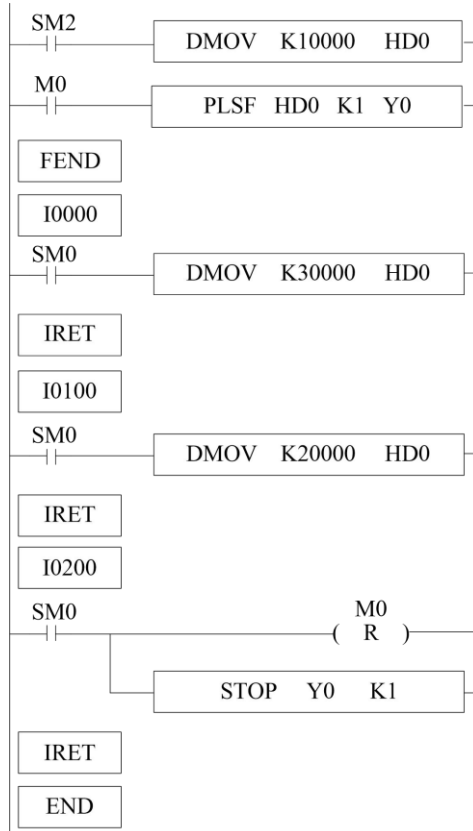
Đoạn	Giá cài đặt tần số(Hz)	Số xung
Origin ---- A	10000	999999999
A---- B	30000	999999999
B ----- C	20000	999999999
Thời gian tăng tốc và giảm tốc	0	

Lưu ý: vì không xác định được số xung của từng đoạn nên số xung phải được đặt đủ lớn để đảm bảo đối tượng có thể di chuyển đến cảm biến tiệm cận. Lệnh STOP sẽ được thực thi bởi lệnh ngắt ngoài khi đối tượng đến vị trí C.

Giải thích thành phần:

Thành phần PLC	Giải thích thành phần	Ghi chú
M0	Nút khởi động, lệnh PLSF sẽ phát xung khi nhấn nút	
HD0	Nút khởi động, lệnh PLSF sẽ phát xung khi nhấn nút	

Chương trình



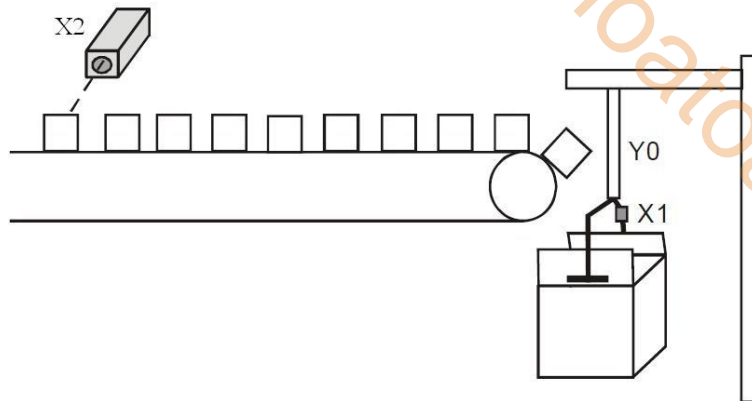
Giải thích chương trình

1. SM2 ở trạng thái ON, đặt HD0 thành 10000, đặt M0 ở trạng thái ON, lệnh PLSF sẽ phát xung 10000Hz, đối tượng sẽ di chuyển từ điểm gốc đến A.
2. Khi đối tượng đến A, X0 sẽ ON ngay lập tức, lệnh ngắt ngoài I0000 sẽ hoạt động, HD0 được đặt thành 30000, đối tượng di chuyển từ A đến B với tốc độ 30000Hz.
3. Khi đối tượng đến B, X1 sẽ ON ngay lập tức, lệnh ngắt ngoài I0100 sẽ hoạt động, HD0 được đặt thành 20000, đối tượng sẽ di chuyển từ B đến C với tốc độ 20000Hz.
4. Khi đối tượng đến C, X2 sẽ ON ngay lập tức, lệnh ngắt bên ngoài I0200 sẽ hoạt động, M0 được đặt OFF, quá trình phát xung sẽ dừng ngay lập tức.

Ví dụ 2

Đây là sơ đồ của máy đóng gói sản phẩm. Robot sẽ đóng gói sản phẩm khi phát hiện được 30 sản phẩm, robot và bộ đếm sẽ được reset sau khi đóng gói xong.

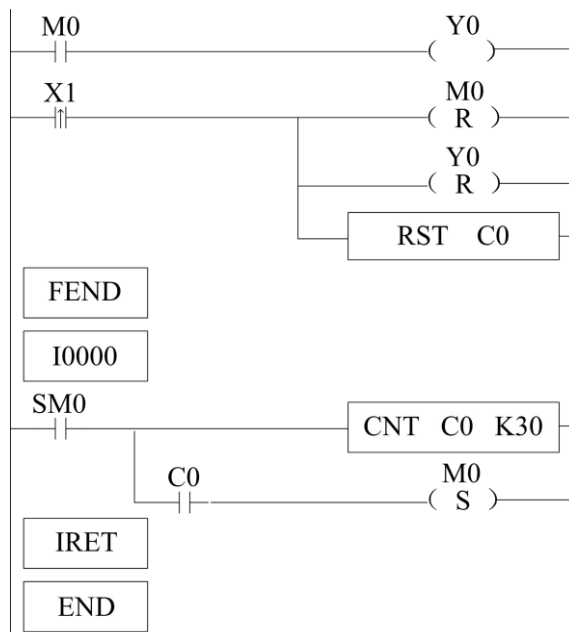
Để nâng cao hiệu quả làm việc, tốc độ gửi sản phẩm rất nhanh, cảm biến X2 phát hiện sản phẩm có thời gian 8ms, thời gian lọc đầu vào PLC là 10ms, bộ đếm thông thường không thể phát hiện sản phẩm. Chúng ta có thể sử dụng lệnh ngắt ngoài để đếm sản phẩm.



Giải thích thành phần

Thành phần PLC	Giải thích chức năng	Ghi chú
X2	Cảm biến quang điện đếm sản phẩm, X2 ở trạng thái ON khi phát hiện sản phẩm	
X1	Cảm biến hoàn thành hành động của robot, X1 ở trạng thái ON khi hành động được hoàn thành	
C0	Bộ đếm 16-bit	
Y0	Robot	

Chương trình:



Giải thích chương trình:

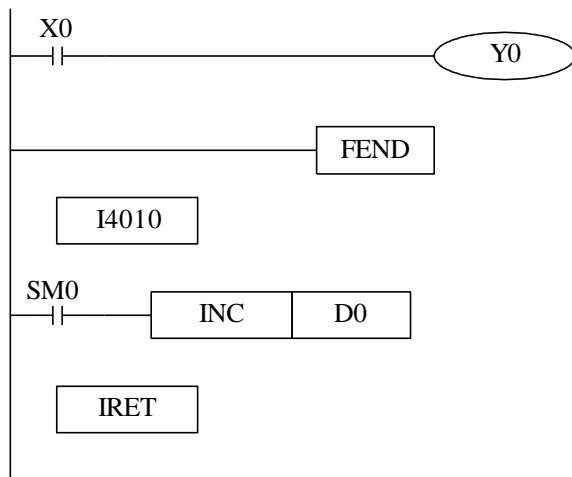
1. Trong chương trình ngắt bên ngoài, đếm đầu vào X2, khi X2 là 30, đặt M0 ở trạng thái ON
2. Trong chương trình chính, nó điều khiển Y0 theo trạng thái M0.
3. Khi hành động của robot hoàn tất, X1 chuyển từ OFF sang ON một lần, lệnh RST hoạt động

hoạt động, Y0 và C0 được đặt lại, M0 OFF, đợi quá trình đóng gói tiếp theo.

10-4-2 Ngắt bộ định thời

Chức năng và Hành động

Trong trường hợp chu kỳ thực hiện chương trình chính rất dài, khi bạn phải xử lý chương trình đặc biệt hoặc thỉnh thoảng thực hiện chương trình cụ thể khi chương trình đang quét trong điều khiển tuần tự, việc ngắt bộ định thời sẽ rất hữu ích. Nó không bị ảnh hưởng bởi chu kỳ quét PLC và thực hiện chương trình con ngắt thời gian mỗi N ms.



- Ngắt bộ định thời là trạng thái mở mặc định, giống như các chương trình con ngắt khác, nó phải được viết phía sau chương trình chính, bắt đầu bằng I40xx, kết thúc bằng IRET.
- Có 20 cổng lệnh ngắt bộ định thời, biểu thị: I40**~I59**(**' có nghĩa là thời gian ngắt; Đơn vị là ms. Ví dụ: I4010 có nghĩa là thực thi một lần lệnh ngắt bộ định thời đầu tiên trong mỗi 10 mili giây.

Địa chỉ lệnh ngắt

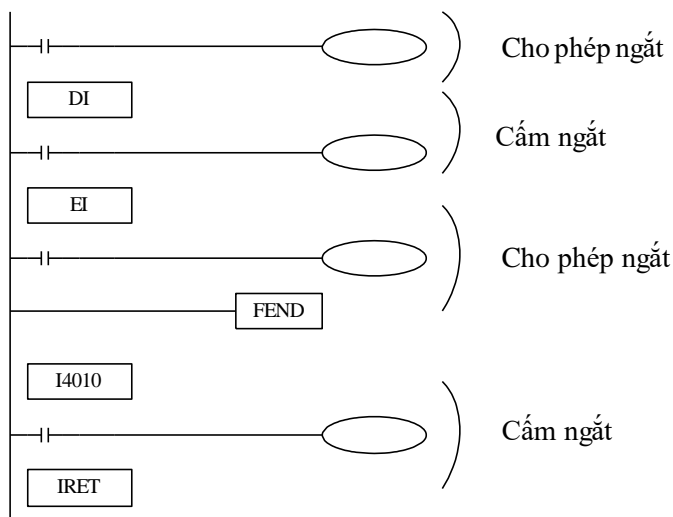
Ngắt bộ định thời dòng XD, XL:

Địa chỉ lệnh ngắt	Lệnh cấm ngắt	Địa chỉ lệnh ngắt	Lệnh cấm ngắt	Giải thích
I40**	SM070	I50**	SM080	** có nghĩa là thời gian ngắt bộ định thời gian, phạm vi là 1 ~ 99, đơn vị là ms. Lưu ý: 1. Chỉ lệnh ngắt bộ định thời I59** mới hỗ trợ 100us. 2. Chức năng lựa chọn theo
I41**	SM071	I51**	SM081	
I42**	SM072	I52**	SM082	
I43**	SM073	I53**	SM083	
I44**	SM074	I54**	SM084	
I45**	SM075	I55**	SM085	
I46**	SM076	I56**	SM086	
I47**	SM077	I57**	SM087	
I48**	SM078	I58**	SM088	

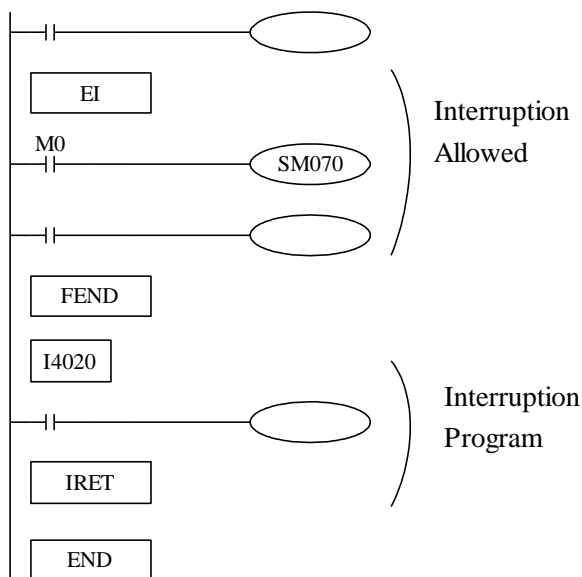
I49**	SM079	I59**	SM089	thời gian chỉ được hỗ trợ bởi phiên bản phần mềm PLC V3.4.6 trở lên.
-------	-------	-------	-------	--

Giới hạn phạm vi ngắt

- Lệnh ngắt bộ định thời thường ở trạng thái ‘cho phép/allow’.
- Có thể đặt vùng cho phép và cấm ngắt bằng lệnh EI,DI. Như được hiển thị trong các hình ảnh bên dưới, tất cả các lệnh ngắt bộ định thời đều bị cấm trong vùng nằm giữa lệnh DI và EI và được phép ngoài vùng DI~EI.



Cấm lệnh ngắt



- The first 3CH timing interruptions are equipped with special relays (SM070~SM079).
- In the left example, if use M0 to set SM070 “ON”, then forbid timing interruption forbidden.

10-5 Đọc và Ghi thẻ SD

XD5 (ngoại trừ XD5-16), thân PLC dòng XDM có thể được mở rộng bằng thẻ SD để lưu trữ và sao lưu dữ liệu. Khe cắm thẻ SD nằm trên bo mạch CPU của PLC. Khi sử dụng, bạn cần nhấn nắp bảo vệ BD lên và lắp thẻ SD vào khe cắm thẻ.



Thẻ SD không được lắp khi PLC rời khỏi nhà máy. Người dùng cần mang theo thẻ nhớ microSD (Thẻ TF) của riêng mình và dung lượng thẻ không được lớn hơn 32GB. Trước khi lắp thẻ SD vào PLC, vui lòng sử dụng đầu đọc thẻ để định dạng thẻ thành định dạng FAT32 trên máy tính.

Lưu ý:

1. PLC của model Ethernet không hỗ trợ thẻ SD.
2. Việc sử dụng thẻ SD xung đột với một số cổng giao tiếp của PLC. XD5-32/60, XD5/XDM-32T4/60T4 xung đột với COM4 và XD5-60T6/T10, XDM-60T10 xung đột với COM4 và COM5.

10-5-1 Nội dung và định dạng tài liệu

Thẻ SD hỗ trợ bốn kiểu dữ liệu, bao gồm Single Word (W), Double Word (DW), Số thực dấu phẩy động (Fm.n) và Ký tự (Sx).

Như thể hiện trong hình dưới đây, hàng đầu tiên trong file excel khai báo kiểu dữ liệu:

	A	B	C	D	E	F	G
1	w	dw	s8	f4.15	dw	w	dw
2	-32765	-32770	hellbaby	1237.20100156164	30000	999	3121
3	454	-91877301	testh	2351.25150102545	-454532088	-15453	124522

Phạm vi dữ liệu và không gian chiếm dụng của từng loại được thể hiện trong bảng sau

Kiểu dữ liệu	W	DW	Fm.n(m<=15,n<=15)	Sx(x<=16)
Phạm vi dữ liệu	- 32768~32767	- 2147483648~2 147483647	- 18446742974197923840 ~1844674297419792384 0	\
Số ký tự chiếm trong thẻ SD	6	11	m+1+n	2*x
Số word	1	2	2	x

Lưu ý:

※ 1: khi độ dài thực tế của dữ liệu nhỏ hơn số ký tự được lưu trong thẻ SD, khoảng trống sẽ được sử dụng để điền vào từ bên trái. Ví dụ: dữ liệu single word là 454, nhỏ hơn 6 ký tự được chiếm bởi kiểu W, vì vậy hãy điền 6-3=3 khoảng trống từ bên trái và thực tế là "L L L 454" (L đại diện cho khoảng trống).

※ 2: khi Fm.n là số âm thì bit dấu "-" cũng chiếm một ký tự. Ví dụ: kiểu số thực dấu phẩy động được xác định là F5.3. Sau khi ghi dữ liệu "-12345.123" vào thẻ SD, bit ít có trọng số thấp nhất sẽ bị xóa và dữ liệu sẽ trở thành "-12345.12".

※ 3: x trong Sx đại diện cho độ dài word chứ không phải độ dài ký tự.

10-5-2 Tên file và vị trí lưu trữ

Thẻ SD hỗ trợ lưu trữ các tệp định dạng ".csv", tệp này phải được lưu trữ trong thư mục gốc của thẻ SD.

Tất cả các tệp ".csv" phải được đặt tên theo "dataxxx.csv" và "xxx" là số chỉ mục của tệp, từ 001 đến 999. Khi xxx nhỏ hơn 100, hãy thêm 0 từ bên trái. Nếu số chỉ mục của tệp là 1 thì tệp đó có tên chính xác là "data001.csv".

10-5-3 Đọc/ ghi thẻ SD

Lệnh đọc và ghi thẻ SD nằm trong khối hàm tuần tự. Mở giao diện khối hàm và chèn "mô-đun đọc và ghi SD". Sau đây là phần giới thiệu về lệnh đọc và ghi thẻ SD.

1) Đọc thẻ SD

Thêm lệnh đọc thẻ SD vào khối hàm tuần tự, như thể hiện trong hình sau:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	1	1	2	0	0	0	0	0	0	0
D10	1	0	0	0	0	0	0	0	0	0
D20	-32765	0	0	0	0	0	0	0	0	0

Trong hình trên, lệnh đọc thẻ SD là: READSD D0 D10 D20 và ý nghĩa chức năng của từng tham số như sau:

D0: số chỉ mục tệp/cột/địa chỉ bắt đầu dòng, D0~D2 lần lượt đề cập đến số chỉ mục tệp, số cột và số dòng.

Như thể hiện trong dữ liệu điều khiển ở trên, D0=1, D1=1, D2=2, cho biết:

Số chỉ mục của tệp là 1: nghĩa là tên tệp là data001.csv.

Cột số 1: cột 1.

Dòng số 2: Dòng 2 (dòng dữ liệu 2).

D10: số word cần đọc dữ liệu.

Như được hiển thị trong phần dữ liệu điều khiển ở trên, D10=1 có nghĩa là đọc 1 word dữ liệu

D20: dữ liệu đọc được lưu trữ ở địa chỉ đầu tiên của phần thân PLC.

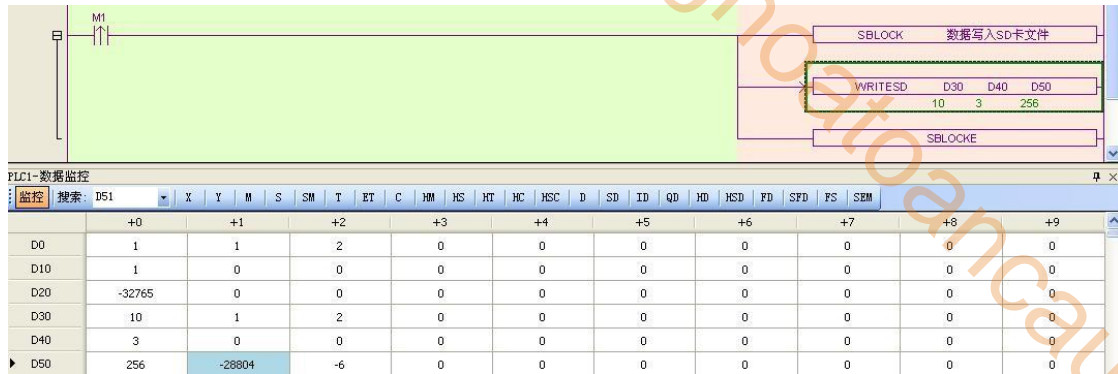
Như thể hiện trong phần dữ liệu điều khiển ở trên, dữ liệu đọc được trong D20 là -32765.

Do đó, ý nghĩa cụ thể của chương trình trên là đọc tên tệp data001 trong thư mục gốc dưới dạng csv của thẻ SD và lưu dữ liệu đã đọc vào thanh ghi D20.

Nếu số cột là 0 thì số word được đọc từ dữ liệu cho biết số hàng được đọc. Ví dụ: khi D0=1, D1=0, D2=1, D10=5, điều đó có nghĩa là dữ liệu của 5 dòng được đọc từ dòng đầu tiên của tệp data001.csv và được lưu trong thanh ghi bắt đầu từ D20.

2) Ghi thẻ SD

Thêm lệnh ghi thẻ SD vào khối hàm tuần tự, như thể hiện trong hình sau:



Lệnh thẻ SD: WRITESD D30 D40 D50 và ý nghĩa chức năng của từng thông số như sau:

D30: số chỉ mục tệp/cột/địa chỉ bắt đầu dòng, D30~D32 lần lượt biểu thị số chỉ mục tệp, số cột và số dòng.

Như thể hiện trong dữ liệu điều khiển ở trên, D30=10, D1=1, D2=2 có nghĩa là:

Số chỉ mục của tệp là 10: tức là tên tệp là data010.csv.

Cột số 1: cột 1.

Dòng số 2: Dòng 2 (dòng dữ liệu 2).

D40: số từ cần đọc dữ liệu.

Như thể hiện trong phần dữ liệu điều khiển ở trên, D40=3 cho biết dữ liệu được ghi thành 3 word.

D50: dữ liệu ghi được lưu trữ ở địa chỉ đầu tiên của phần thân PLC.

Như thể hiện trong phần dữ liệu điều khiển ở trên, hãy ghi dữ liệu thanh ghi bắt đầu từ D50 vào thẻ SD.

Vì vậy, ý nghĩa cụ thể của chương trình trên là: ghi dữ liệu 3 word bắt đầu từ thanh ghi D50 vào thư mục gốc của thẻ SD, tên file là data010.csv tại cột 1 và hàng 2.

Nếu số cột là 0 thì số từ được ghi vào dữ liệu biểu thị số hàng được ghi. Ví dụ: khi D30=2, D31=0, D32=3, D40=5, điều đó có nghĩa là năm hàng dữ liệu bắt đầu từ D50 được ghi vào hàng thứ ba đến thứ bảy của data002.csv.

10-5-4 Lưu ý

1) Chỉ các ký tự hiển thị (chữ cái và số) mới được hỗ trợ trong loại ký tự, nghĩa là các ký tự thuộc phạm vi [32,126] trong bảng mã ASCII.

Ngoại lệ:

- Hỗ trợ ký tự cuối trong các ký tự vô hình
- Không hỗ trợ “,” trong các ký tự hiển thị

2) Những hạn chế của tệp Csv

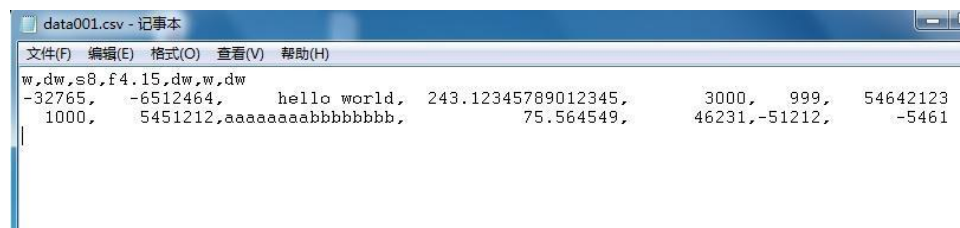
- Số cột trong tệp .csv không được vượt quá 20
- Các ký tự của dòng dữ liệu tệp không được vượt quá 512 ký tự (bao gồm dấu phẩy và ngắt dòng)

3) Giới hạn số lần đọc ghi dữ liệu của word

Dữ liệu không thể chỉ được đọc một phần. Ví dụ: định dạng được xác định là W, DW, S8. Nếu đọc từ cột đầu tiên W và số từ đọc được là 10 thì S8 không thể đọc hết và chương trình sẽ báo lỗi (số từ đúng là 3 hoặc 11). Khi chương trình nhận định rằng các thông số sai, nó sẽ không đọc hoặc ghi vào thẻ SD.

4) Tạo hạn chế cho tệp CSV

- nếu bạn sử dụng notepad do Windows cung cấp để tạo tệp CSV thì cuối dòng cuối cùng phải được ngắt dòng và dữ liệu được tạo theo đúng kiểu được định nghĩa (type definition) (bảng trong phần 10-5-1), như được hiển thị trong hình sau:



- các tệp CSV hiện có chỉ có thể được sử dụng sau khi được chuyển đổi bằng công cụ chuyển đổi định dạng trong phần 10-5-5.

5) Giới hạn đọc và viết

Trong lệnh đọc, nếu số dòng được chỉ định vượt quá số dòng thực tế trong tệp thì chương trình sẽ báo lỗi.

6) Mô tả quá trình ghi

Trong lệnh ghi, nếu số dòng được chỉ định vượt quá số dòng thực tế của tệp thì tệp sẽ được mở rộng. Trong quá trình mở rộng, hãy chèn dấu cách vào các vị trí tương ứng theo kiểu dữ liệu cột, chừa lại một khoảng trống cố định để chèn dữ liệu sau này.

Lưu ý:

Nếu số dòng ghi lớn hơn số dòng thực tế trong file thì file sẽ được mở rộng. Tốc độ mở rộng tệp rất chậm, điều này sẽ ảnh hưởng đến chu kỳ quét PLC và kích hoạt đối tượng điều khiển. Điều này không được khuyến khích.

7) Thông tin trạng thái thẻ SD

Thông tin trạng thái của thẻ SD có thể được xem qua thanh ghi đặc biệt SD453, như trong bảng sau:

SD453	Giải thích	Nguyên nhân
0	Vận hành thành công	
1	Reserve	
2	Reserve	
3	Reserve	
4	Tệp đọc/ghi không tồn tại	Tệp không tồn tại
5	Đọc tệp tin để kết thúc	Đã đến cuối tệp tin

SD453	Giải thích	Nguyên nhân
6	Đọc file	Đang đọc file
7	Ghi file	Đang ghi file
8	Lỗi ghi/đọc	Rút thẻ SD trong khi đọc và ghi
9	Không đủ dung lượng thẻ SD	Không đủ dung lượng thẻ SD
10	Reserve	
11	FAT32 error	Thẻ SD không được lắp đúng cách hoặc thẻ SD không được định dạng là FAT32
12	Reserve	
13	Reserve	
14	Không thể khởi tạo thẻ SD	Thẻ SD không được lắp đúng cách
15	Reserve	
16	Reserve	
17	Reserve	
18	Reserve	
19	Không thể phát hiện thẻ SD	Thẻ SD chưa được lắp vào
20	Lỗi đọc và ghi thông số	Kiểm tra chỉ mục, hàng, dòng, wordent
21	Dữ liệu đọc/ghi không tuân theo định nghĩa định dạng	Chưa xác định kiểu dữ liệu
22	Loại dữ liệu sai của tập tin	Xuất hiện các định nghĩa kiểu dữ liệu khác ngoài W,DW,Sn,Fx.y
23	Kiểu dữ liệu của tệp không được xác định	Dữ liệu dấu phẩy động và ký tự không đáp ứng định nghĩa
24	Tên tập tin bất hợp pháp	Chỉ mục > 999
25	Chỉ mục cột không hợp lệ	Chỉ mục cột lớn hơn số cột của tệp
26	Chỉ mục hàng không hợp lệ	Chỉ số hàng nhỏ hơn 1
27	Đọc và viết word không hợp lệ	Số word đọc và viết ít hơn 1
28	Đọc và viết word không hợp lệ	Số word không thể đọc được hết dữ liệu
29	Số ký tự trong dòng dữ liệu file vượt quá giới hạn	Ký tự dòng dữ liệu vượt quá 512 ký tự
30	Lỗi khác	

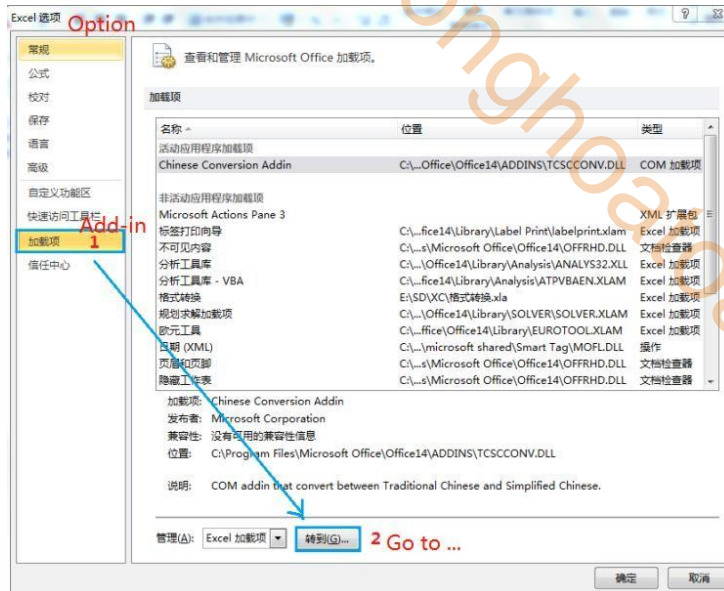
10-5-5 Các công cụ sử dụng để chuyển đổi định dạng

1) Công cụ Microsoft Office 2010 excel

Khi chuyển dữ liệu hiện có sang định dạng CSV và lưu vào thẻ SD, vui lòng sử dụng công cụ chuyển đổi định dạng.

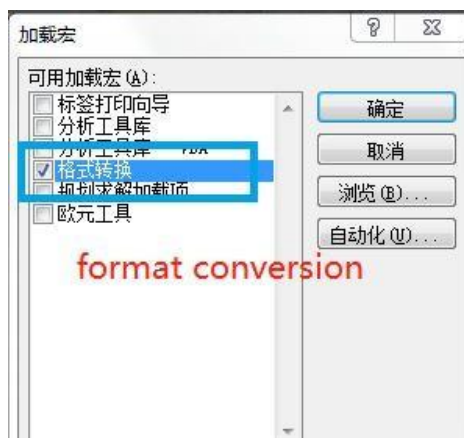
Các bước sử dụng:

- (1) Giải nén [format conversion.rar] vào bất kỳ thư mục nào
- (2) Mở excel và thực thi tệp -> tùy chọn để mở hộp thoại Tùy chọn excel.
- (3) Như thể hiện trong hình dưới đây, hãy mở hộp thoại thêm vào;

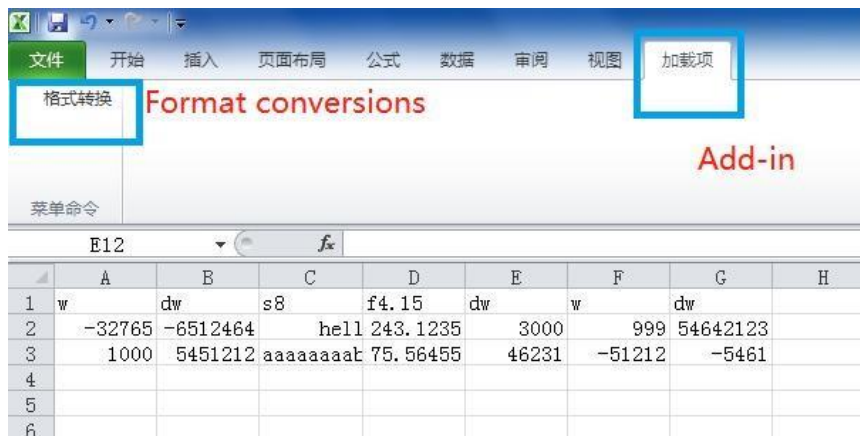


((4) Nhấp vào [Browse] và chọn tệp [Format conversion.xla] được trích xuất trước đó.

(5) 【Format conversion】 được thêm vào các phần bổ sung có sẵn và được đánh dấu ở phía trước.



(6) Sau khi nhấp vào[OK], cột 【add in】 và tùy chọn 【format conversion】 được thêm vào thanh công cụ excel, như được minh họa trong hình sau:



(7) Mở tệp csv hiện có và nhấp vào chuyển đổi định dạng để hoàn tất chuyển đổi.

(8) Khi lưu sẽ hiện ra hộp thoại các bạn chọn Yes và file sẽ có định dạng csv.

10-6 Lệnh điều khiển đa trạm[MSC]

1) Sơ lược

Lấy giá trị bộ mã hóa theo đầu vào kích hoạt, tính toán và lưu giá trị đầu vào và giá trị đầu ra của phôi trong mỗi trạm, so sánh giá trị được lưu trữ của từng phôi trong mỗi trạm với giá trị hiện tại của bộ mã hóa và xuất ra kết quả so sánh.

Điều khiển đa trạm[MSC]			
16 bits	-	32 bits	MSC
Điều kiện thực thi	Thường hở/ đóng	Model phù hợp	XD5, XDM, XDH, XD5E, XDME, XL5E, XLME
Yêu cầu phần cứng	-	Yêu cầu phần mềm	-

2) Toán hạng

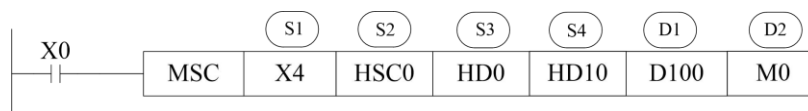
Toán hạng	Chức năng	Model
S1	Chỉ định số địa chỉ thành phần phần mềm của điểm đầu vào kích hoạt lệnh	bit
S2	Chỉ định địa chỉ bộ đếm tốc độ cao	32 bits, BIN
S3	Chỉ định số lượng trạm và phôi và số địa chỉ đầu tiên của thanh ghi thời gian lọc	16 bits, BIN
S4	Chỉ định số địa chỉ đầu tiên của thanh ghi cho giá trị tham chiếu và giá trị độ lệch	32 bits, BIN
D1	Chỉ định số địa chỉ đầu tiên của thanh ghi lưu giá trị chỉ số và giá trị so sánh	16/32 bits, BIN
D2	Chỉ định số địa chỉ thành phần phần mềm của kết quả đầu ra	bit

3) Phần tử lập trình phù hợp

Toán hạng	Phần tử lập trình word											Phần tử lập trình bit							
	Hệ								Hãng số	Mô-đun		Hệ							
	D	FD	TD	CD	DX	DY	D M	DS	KH	ID	QD	X	Y	M	S	T	C	Dnm	
S1												•							
S2	Only HSC																		
S3	•																		
S4	•																		
D1	•																		
D2													•	•					

*Lưu ý: D bao gồm D, HD. TD bao gồm TD, HTD. CD bao gồm CD, HCD, HSCD, HSD. DM bao gồm DM, DHM. DS bao gồm DS, DHS. M bao gồm M, HM, SM. S bao gồm S và HS. T bao gồm T và HT. C bao gồm C và HC.

Chức năng và hành động



- S1: đó là điểm đầu vào kích hoạt lệnh, có thể chọn điểm đầu vào ngắt ngoài hoặc điểm đầu vào thông thường, kích hoạt lệnh tại sườn lên, sườn xuống và lấy giá trị bộ mã hóa.
- S2: là số lượng bộ đếm tốc độ cao được sử dụng cùng nhau, được sử dụng cho đầu vào tín hiệu bộ mã hóa. Chế độ đếm tốc độ cao là chế độ tăng dần một pha.
- S3: ba thanh ghi 16 bit (từ đơn) được sử dụng liên tục để thiết lập số trạm, số lượng phôi và thời gian lọc. Nên sử dụng thanh ghi khả nhớ khi tắt nguồn.

Việc phân bổ thanh ghi cụ thể như sau:

S3: đặt số trạm, ghi là n, phạm vi: 1~32;

S3+1: đặt số lượng phôi tối đa có thể được xử lý, ghi là m, phạm vi: 1~64;

S3+2: đặt thời gian lọc, phạm vi: 0~32767, đơn vị: ms. Thông số này có thể được sử dụng để ngăn chặn các lỗi do jitter cơ học gây ra. Nếu thời gian lọc được đặt thành 0, điều đó có nghĩa là không lọc. Nếu nhỏ hơn 0 thì nó sẽ được coi là 0. Giả sử rằng thời gian lọc được đặt thành t và điểm đầu vào kích hoạt là X4, việc thu tín hiệu đầu vào sẽ áp dụng các phương pháp sau:

Sườn tăng: sau khi trạng thái off X4 được duy trì trong ít nhất t ms, sườn lên được phát hiện đầu tiên là tín hiệu kích hoạt;

Sườn giảm: sau khi trạng thái on X4 được duy trì trong ít nhất t ms, sườn xuống đầu tiên được phát hiện là tín hiệu kích hoạt.

- S4: 3n Các thanh ghi 32-bit (double words) được sử dụng liên tục, được sử dụng để thiết lập giá trị tham chiếu, giá trị độ lệch đầu vào phôi và giá trị độ lệch đầu ra phôi (**workpiece departure deviation**) của mỗi trạm. Mỗi tham số chiếm 2 thanh ghi liên tục. Nên sử dụng thanh ghi khả nhớ khi tắt nguồn. Việc phân bổ địa chỉ thanh ghi cụ thể như sau:

Tên	Trạm 1	Trạm 2	Station n
Giá trị tham chiếu (double word)	S4	S4+2	S4+(n-1)×2
Giá trị sai lệch đầu vào phôi (double word)	S4+2n	S4+2n+2	S4+(2n-1)×2
Giá trị sai lệch đầu ra phôi (double word)	S4+4n	S4+4n+2	S4+(4n-1)×2

◆ Khi giá trị tham chiếu của một trạm được đặt thành 0, điều đó có nghĩa là trạm đó không hoạt động.

◆ Giá trị độ lệch đầu vào phôi và giá trị độ lệch đầu ra phôi chủ yếu được sử dụng để hiệu chỉnh vị trí. Khi giá trị bộ mã hóa của phôi vào và ra khỏi trạm tương ứng được phát hiện không nhất quán với cài đặt trong quá trình sử dụng thực tế, nó có thể được hiệu chỉnh bằng cách điều chỉnh giá trị độ lệch đầu vào phôi và giá trị độ lệch đầu ra phôi. Ví dụ: giá trị tham chiếu của trạm 1 được đặt thành 1000, có nghĩa là phôi đi vào trạm 1 sau khi kích hoạt sườn lên của X4 thông qua 1000 giá trị đếm tốc độ cao. Nếu trong sử dụng thực tế, phôi đi vào trạm 1 chỉ với 990 giá trị đếm tốc độ cao, thì giá trị độ lệch đầu vào phôi có thể được đặt thành -10.

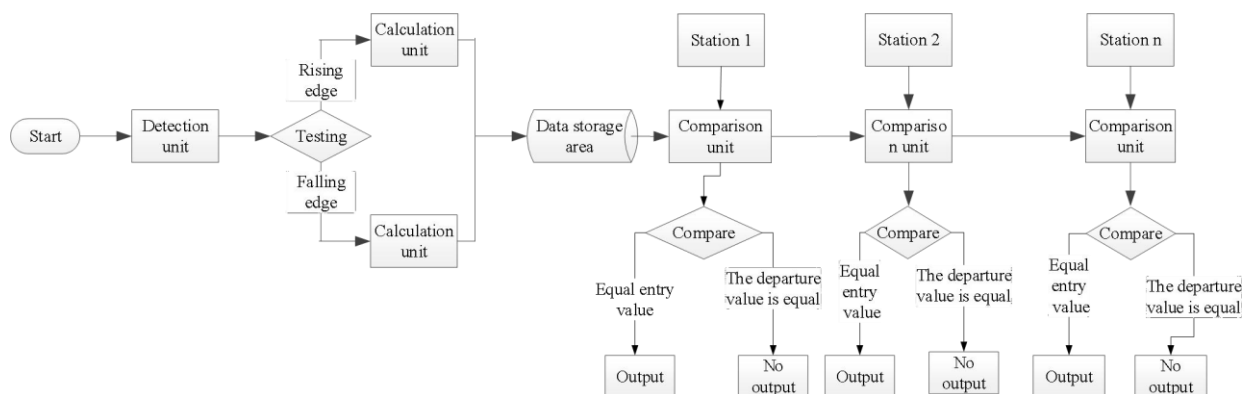
- D1: liên tục chiếm $2n$ thanh ghi 16 bit (single word), thanh ghi $2m \times n$ 32 bit (double word) được sử dụng để lưu trữ giá trị chỉ số chuyển tiếp phôi, giá trị chỉ mục theo sau, giá trị so sánh đầu vào và giá trị so sánh đầu ra của mỗi trạm. Việc phân bổ địa chỉ thanh ghi cụ thể như sau:

Tên	Trạm 1	Trạm 2	Trạm n
Forward index value(word)	D1	D1+1	D1+(n-1)
Follow index value(word)	D1+n	D1+(n+1)	D1+(2n-1)
Giá trị so sánh đầu vào phôi 1(double word)	D1+2n	D1+2n+2	D1+2n+2(n-1)
Giá trị so sánh đầu ra phôi 1 (double word)	D1+4n	D1+4n+2	D1+4n+2(n-1)
.....
Giá trị so sánh đầu vào phôi m (double word)	D1+4mn-2n	D1+4mn-2n+2	D1+4mn-2
Giá trị so sánh đầu ra phôi m (double word)	D1+4mn	D1+4mn+2	D1+4mn+2(n-1)

Lưu ý: D1 chiếm không gian lưu trữ lớn, vui lòng xác nhận xem dung lượng thanh ghi có đủ không. Nếu vẫn chưa đủ, PLC sẽ chỉ lưu trữ dữ liệu trong vùng hữu ích và sẽ không có cảnh báo và nhắc nhở.

- ◆ Khi giá trị so sánh đầu vào và giá trị so sánh đầu ra của một trạm đều bằng 0, điều đó có nghĩa là hành động so sánh của trạm không được thực thi.
- ◆ Giá trị chỉ mục forward (forward index value) sẽ tự động tăng thêm 1 khi kích hoạt mỗi sườn lên và sườn xuống của tín hiệu đầu vào (nếu thời gian lọc > 0 , hãy đợi thời gian lọc rồi tăng thêm 1) và phương pháp tích lũy tuần hoàn được áp dụng. Ví dụ: khi số phôi được xử lý tối đa là $m = 10$, giá trị chỉ số forward sẽ quay vòng theo 0, 1, 2, 3... 19, 0, 1, 2, 3... 19 (giá trị ban đầu là 0). Vì giá trị chỉ số forward sẽ tăng thêm 1 ở cả sườn lên và sườn xuống, nên giá trị chỉ mục chuyển tiếp tối đa là $2*m$.
- ◆ Lưu ý: giá trị chỉ mục follow (following index value) sẽ được đánh giá trước khi thêm 1 vào giá trị chỉ mục chuyển tiếp. Nếu giá trị sau khi cộng 1 bằng giá trị chỉ mục follow thì giá trị mục forward sẽ không được tích lũy và giá trị so sánh lần này sẽ được ghi lại.
- ◆ Giá trị chỉ mục follow sẽ tự động thêm 1 khi phôi vào và ra khỏi trạm. Nói chung, sau khi phôi đã hoàn thành một trạm, giá trị chỉ mục follow của trạm tương ứng là số chẵn.
- ◆ Giá trị so sánh đầu vào được tự động tính toán và lưu trữ trong vùng dữ liệu D1 khi phôi tương ứng kích hoạt sườn lên của tín hiệu đầu vào. Giá trị so sánh đầu vào của trạm thường là:

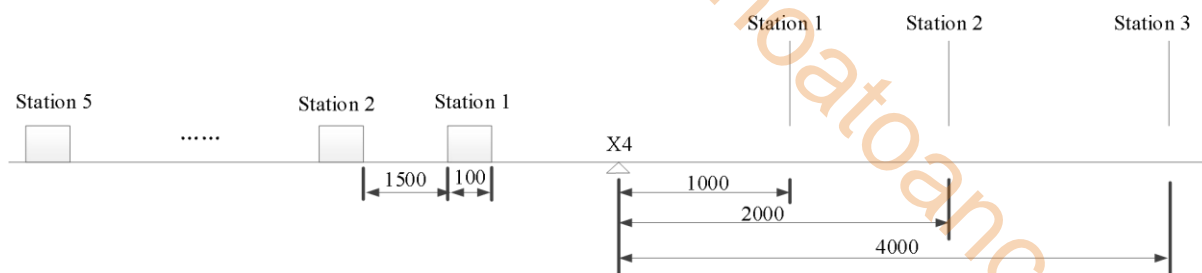
- ◆ Giá trị so sánh phôi m vào trạm n = giá trị đếm của phôi m (tại sườn lên) + giá trị tham chiếu của trạm n + giá trị sai lệch phôi vào trạm n.
- ◆ Giá trị so sánh đầu ra được tự động tính toán và lưu trữ trong vùng dữ liệu D1 khi sườn xuống của tín hiệu đầu vào được kích hoạt bởi phôi tương ứng. Giá trị so sánh đầu ra của trạm thường là:
- ◆ Giá trị so sánh phôi m rời trạm n = giá trị đếm của phôi m (tại sườn xuống) + giá trị tham chiếu trạm n + giá trị sai lệch phôi rời trạm n.
- D2: liên tục chiếm n cuộn dây (tương ứng với số lượng n trạm) và chỉ có thể chỉ định đầu ra cuộn dây Y và M để đánh giá xem phôi tương ứng có vào và ra khỏi trạm hay không. Khi lệnh được thực thi, mỗi trạm sẽ đánh giá xem phôi tương ứng có vào và ra khỏi trạm theo giá trị so sánh đã thiết lập theo giá trị chỉ mục follow hay không. Khi giá trị đếm thời gian thực của phôi tương ứng \geq giá trị so sánh đầu vào, điểm đầu ra tương ứng được đặt thành ON và giá trị chỉ mục follow sẽ tự động tăng thêm 1; Khi giá trị đếm thời gian thực của phôi tương ứng \geq giá trị đầu ra so sánh, điểm đầu ra tương ứng được đặt thành OFF và giá trị chỉ mục follow sẽ tự động tăng thêm 1, nhưng nó sẽ không vượt quá giá trị chỉ mục forward.
- Không có giới hạn về số lần sử dụng các lệnh MSc, nhưng nếu cùng một bộ đếm tốc độ cao cần được sử dụng trong chương trình thì mỗi lệnh phải được đặt trong một quy trình khác nhau và mỗi lần chỉ có thể thực hiện một lệnh.
- Trước khi lệnh được thực thi, vui lòng xác nhận xem bộ đếm tốc độ cao được sử dụng có overflows (tràn) hay không (có thể đánh giá bằng bit cờ tràn bộ đếm tốc độ cao sm130, v.v.) và thực hiện xử lý tương ứng.
- Khi điều kiện trước của lệnh MSC bị ngắt kết nối và kết nối lại, các giá trị trong vùng lưu trữ D1 và D2 sẽ bị xóa về 0 và được đặt thành off.



Ví dụ

Ví dụ, năm phôi hiện có cần được xử lý qua ba trạm. Tín hiệu đầu vào kích hoạt là X4, điểm đầu vào tín hiệu mã hóa là X0 (bộ đếm tốc độ cao tương ứng là HSC0), chiều rộng của mỗi

phôi là 100, khoảng cách giữa các phôi là 1500, khoảng cách giữa các phôi là 1000, khoảng cách giữa phôi là X4, 2000 và khoảng cách giữa các phôi là 4000.



Chương trình như sau:



Địa chỉ phân tử lập trình	Mô tả chức năng
X4	Điểm đầu vào kích hoạt
HSC0	Điểm đầu vào bộ đếm tốc độ cao, nhận tín hiệu mã hóa
HD0	Số lượng trạm, HD0=3 trong ví dụ
HD1	Số lượng phôi tối đa có thể được xử lý, trong ví dụ này là HD1=4
HD2	Thời gian lọc, trong ví dụ là HD2=300ms
HD10(double word)	Giá trị tham chiếu của trạm 1. Trong ví dụ này, HD10=1000
HD12(double word)	Giá trị tham chiếu của trạm 2. Trong ví dụ này, HD12=2000
HD14(double word)	Giá trị tham chiếu của trạm 3. Trong ví dụ này, HD14=4000
HD16(double word)	Giá trị độ vào đầu vào phôi của trạm 1 được đặt về 0
HD18(double word)	Giá trị độ vào đầu vào phôi của trạm 2 được đặt về 0
HD20(double word)	Giá trị độ vào đầu vào phôi của trạm 3 được đặt về 0
HD22(double word)	Giá trị độ lệch đầu ra phôi của trạm 1 được đặt thành 0
HD24(double word)	Giá trị độ lệch đầu ra phôi của trạm 2 được đặt thành 0
HD26(double word)	Giá trị độ lệch đầu ra phôi của trạm 3 được đặt thành 0

Gán địa chỉ kết quả đầu ra:

Tên	Trạm 1	Trạm 2	Trạm 3
Giá trị chỉ mục forward (single word)	HD100	HD101	HD102
Giá trị chỉ mục follow (single word)	HD103	HD104	HD105
Giá trị so sánh đầu vào phôi 1 (double word)	HD106	HD108	HD110

Giá trị so sánh đầu ra phôi 1 (double word)	HD112	HD114	HD116
Giá trị so sánh đầu vào phôi 2 (double word)	HD118	HD120	HD122
Giá trị so sánh đầu ra phôi 2 (double word)	HD124	HD126	HD128
Giá trị so sánh đầu vào phôi 3 (double word)	HD130	HD132	HD134
Giá trị so sánh đầu ra phôi 3 (double word)	HD136	HD138	HD140
Giá trị so sánh đầu vào phôi 4 (double word)	HD142	HD144	HD146
Giá trị so sánh đầu ra phôi 4 (double word)	HD148	HD150	HD152
Cờ đầu ra	M10	M11	M12

Kết quả thực thi chương trình:

Giả sử rằng giá trị bộ đếm tốc độ cao khi phôi 1 kích hoạt cạnh lên của X4 là 1000, giá trị chỉ mục forward, giá trị chỉ mục follow, giá trị so sánh đầu vào phôi và giá trị so sánh đầu ra phôi của mỗi trạm được hiển thị trong bảng sau:

Thông số		Trạm 1	Trạm 2	Trạm 3
Phôi 1	Giá trị chỉ mục forward	X4 rising edge:1	X4 rising edge:1	X4 rising edge:1
		X4 falling edge: 2	X4 falling edge: 2	X4 falling edge: 2
	Giá trị chỉ mục follow	M10 rising edge: 1	M11 rising edge: 1	M12 rising edge: 1
		M10 falling edge: 2	M11 falling edge: 2	M12 falling edge: 2
	Giá trị so sánh đầu vào	HD106=2000	HD108=3000	HD110=5000
Giá trị so sánh đầu ra	HD112=2100	HD114=3100	HD116=5100	
Phôi 2	Giá trị chỉ mục forward	X4 rising edge:3	X4 rising edge:3	X4 rising edge:3
		X4 falling edge:4	X4 falling edge:4	X4 falling edge:4
	Giá trị chỉ mục follow	M10 rising edge: 3	M11 rising edge: 3	M12 rising edge: 3
		M10 falling edge: 4	M11 falling edge: 4	M12 falling edge: 4
	Giá trị so sánh đầu vào	HD118=3600	HD120=4600	HD122=6600
Giá trị so sánh đầu ra	HD124=3700	HD126=4700	HD128=6700	

Phôi 3	Giá trị chỉ mục forward	X4 rising edge:5	X4 rising edge:5	X4 rising edge:5
		X4 falling edge:6	X4 falling edge:6	X4 falling edge:6
	Giá trị chỉ mục follow	M10 rising edge:5	M11 rising edge:5	M12 rising edge:5
		M10 falling edge:6	M11 falling edge:6	M12 falling edge:6
	Giá trị so sánh đầu vào	HD130=5200	HD132=6200	HD134=8200
Giá trị so sánh đầu ra	HD136=5300	HD138=6300	HD140=8300	
Phôi 4	Giá trị chỉ mục forward	X4 rising edge:7	X4 rising edge:7	X4 rising edge:7
		X4 falling edge:0	X4 falling edge:0	X4 falling edge:0
	Giá trị chỉ mục follow	M10 rising edge:7	M11 rising edge:7	M12 rising edge:7
		M10 falling edge:0	M11 falling edge:0	M12 falling edge:0
	Giá trị so sánh đầu vào	HD142=6800	HD144=7800	HD146=9800
Giá trị so sánh đầu ra	HD148=6900	HD150=7900	HD152=9900	
Phôi 5	Giá trị chỉ mục forward	X4 rising edge:1	X4 rising edge:1	X4 rising edge:1
		X4 falling edge:2	X4 falling edge:2	X4 falling edge:2
	Giá trị chỉ mục follow	M10 rising edge:1	M11 rising edge:1	M12 rising edge:1
		M10 falling edge:2	M11 falling edge:2	M12 falling edge:2
	Giá trị so sánh đầu vào	HD106=8400	HD108=9400	HD110=11400
Giá trị so sánh đầu ra	HD112=8500	HD114=9500	HD116=11500	

Lưu ý: Khi X0 bị ngắt kết nối và kết nối lại, tất cả dữ liệu trong bảng trên sẽ bị xóa về 0.

Chương 11: Các câu hỏi thường gặp và giải đáp


Chương này chủ yếu trình bày các câu hỏi và câu trả lời thường gặp về PLC dòng XD/XL.

Câu hỏi 1: Làm thế nào để kết nối PLC với PC?

Trả lời: PLC dòng XD với phiên bản firmware V3.2 trở lên hỗ trợ cổng USB, cổng COM1 (RS232) và cổng COM2 (RS485) để kết nối. Ethernet PLC cũng hỗ trợ cổng Ethernet (RJ45) để kết nối.

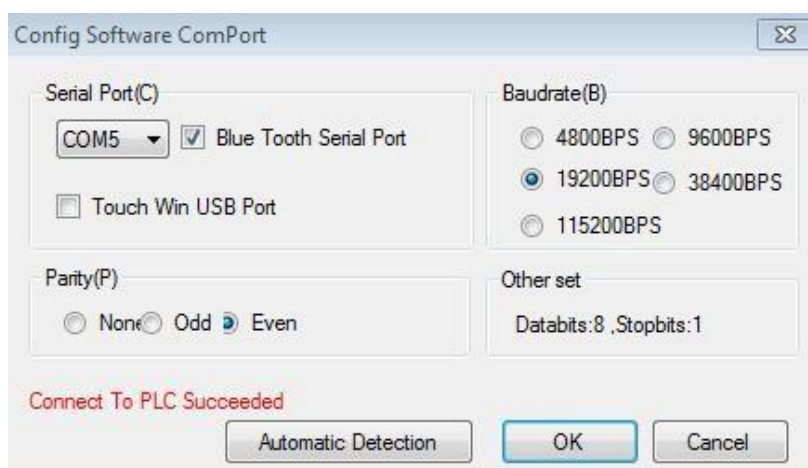
1. Kết nối qua cổng USB và PC (tham khảo Phần 6-1-1)

2. Kết nối qua cổng COM1 (RS232) và PC

Nếu PC của bạn là máy tính để bàn, bạn có thể sử dụng cáp DVP hoặc XVP đặc biệt của công ty chúng tôi để kết nối PC và PLC (Thông thường là PORT1) vì máy tính để bàn thương mại thông thường có cổng nối tiếp 9 chân. Sau khi kết nối DVP chính xác, bật nguồn PLC, nhấp vào 'Config Software ComPort' , cửa sổ sau sẽ xuất hiện:



Chọn cổng nối tiếp giao tiếp chính xác theo cổng nối tiếp thực tế trên PC của bạn.; tốc độ truyền chọn 19200BPS, chọn Even trong Parity, chọn data bits: 8, chọn stop bit: 1; bạn cũng có thể nhấp trực tiếp vào nút “check” trong cửa sổ và các thông số giao tiếp sẽ được chính PLC chọn. “Connect to PLC Succeeded/ Kết nối với PLC thành công” sẽ được hiển thị ở phía dưới bên trái của cửa sổ như bên dưới:



Khi đó có nghĩa là PLC đã được kết nối với PC thành công!

Phương pháp sử dụng của máy tính xách tay có cổng nối tiếp 9 chân cũng giống như máy tính để bàn.

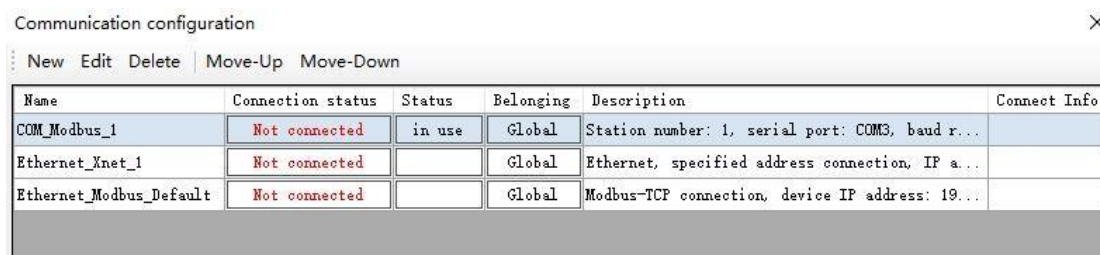
Nếu máy tính xách tay không có cổng nối tiếp 9 chân, người dùng có thể sử dụng bộ chuyển đổi USB để thực hiện kết nối giữa PLC và cổng USB của máy tính xách tay. Đảm bảo cài đặt phần mềm chuyển đổi USB (khuyến nghị mô-đun chuyển đổi USB đặc biệt COM-USB của Xinje, có thể tải xuống phần mềm ổ đĩa chuyển đổi USB trên trang web chính thức của Xinje)!

3. Kết nối qua cổng COM2 (RS485) và PC

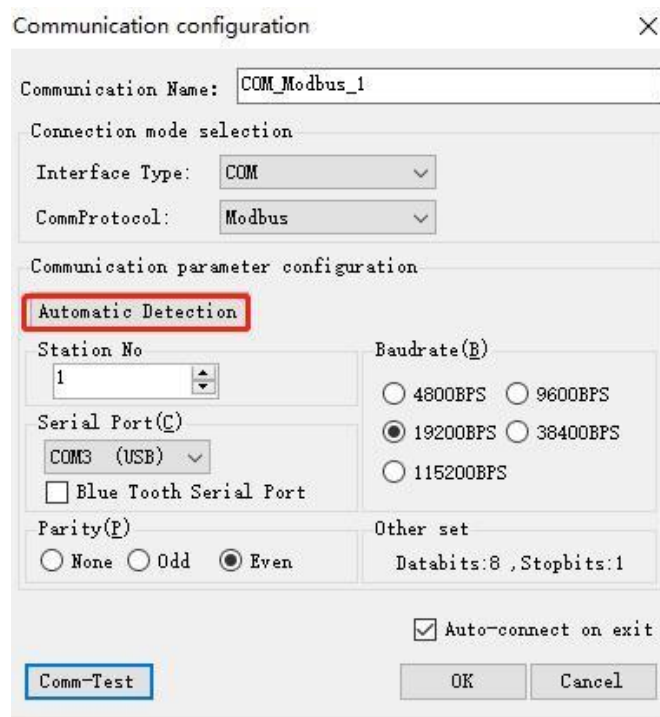
Nếu máy tính được trang bị cổng nối tiếp 9 chân, nó có thể kết nối PC với PLC (thường là cổng com2) thông qua mô-đun chuyển đổi nối tiếp RS485 và cáp XVP. Nếu máy tính chỉ có giao diện USB, nó có thể được kết nối qua cáp USB tới RS485.

Khi hệ thống dây điện được kết nối chính xác, hãy bật nguồn PLC, nhấp vào Config

Software ComPort'  và cửa sổ sau sẽ bật lên:



Chọn cổng nối tiếp giao tiếp chính xác theo cổng nối tiếp thực tế trên PC của bạn.; tốc độ truyền chọn 19200BPS, chọn Even trong Parity, chọn data bits: 8, chọn stop bit: 1; bạn cũng có thể nhập trực tiếp vào nút “check” trong cửa sổ và các thông số giao tiếp sẽ được chính PLC chọn. “Connect to PLC Succeeded” sẽ được hiển thị ở phía dưới bên trái của cửa sổ như bên dưới:



4. Kết nối qua cổng RJ45

1) Cấu hình máy tính

Sau khi cắm cáp mạng, hãy mở "Control panel" → "Network and Internet" → "Network connection".

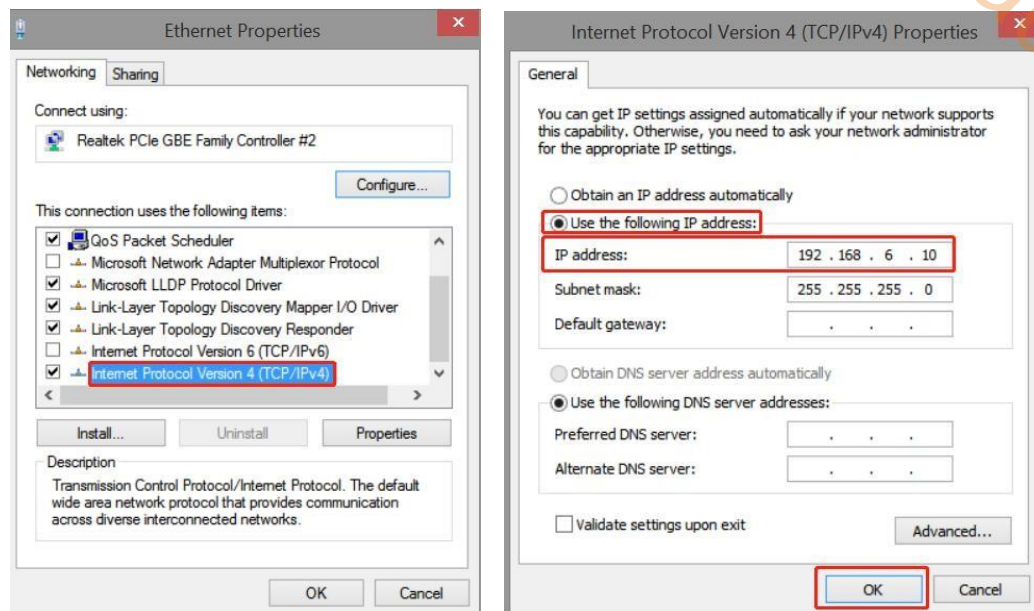
Tìm Ethernet đã được kết nối thành công. Nhấp chuột phải vào Ethernet và nhấp vào thuộc tính. Giao diện thuộc tính Ethernet bật lên. Sau đó làm theo các bước dưới đây:

(1) Nhấp đúp chuột vào "Internet Protocol Version 4 (TCP/IPv4)".

(2) Chọn "use the following IP address".

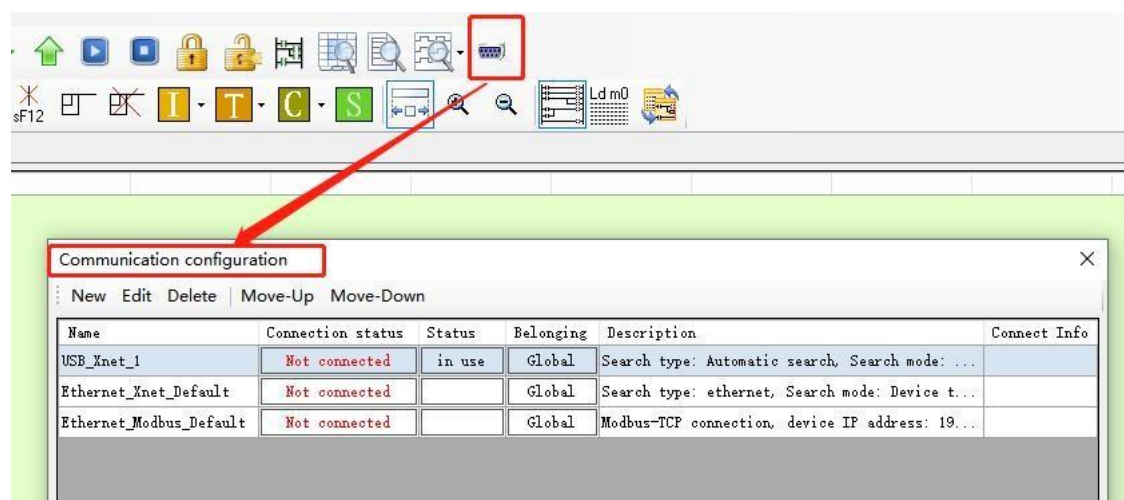
(3) Đặt địa chỉ IP: 192.168.6.xxx, "xxx" có thể đặt tùy ý (trừ 6).

Lưu ý: Chữ số cuối cùng của địa chỉ máy tính và địa chỉ IP của thiết bị PLC không thể được đặt trùng nhau.



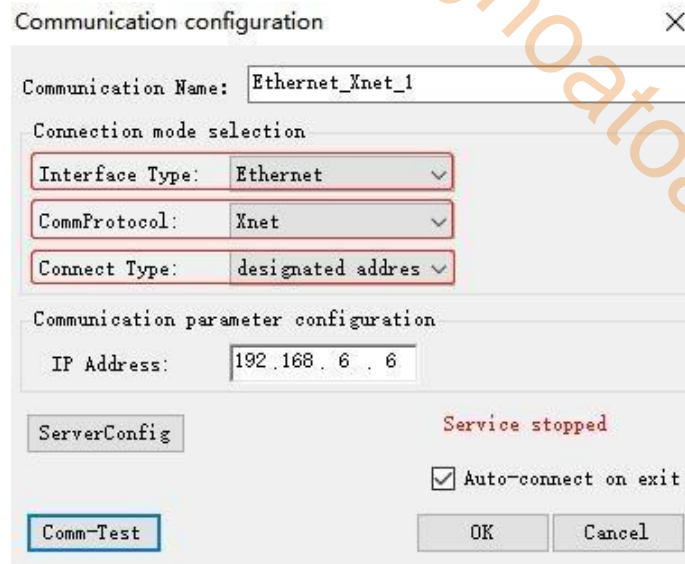
2) Cấu hình PLC

Sau khi kiểm tra cấu hình dây và Ethernet, hãy mở XDPPRO programming tool → nhấp vào communication configuration → nhấp đúp vào Ethernet-Xnet.



Cấu hình theo hình sau:

Chọn giao thức Xnet, địa chỉ IP là địa chỉ IP PLC của bạn. Nhấp vào [Comm-Test], “Connect to PLC succeeded” sẽ được hiển thị.



Nhấn OK sau khi cấu hình và chọn "in use" cho trạng thái tương ứng.

Câu hỏi 2: Tại sao PC không thể kết nối PLC qua cổng RS232 và hiển thị trạng thái ngoại tuyến?

Trả lời:

Một số nguyên nhân phỏng đoán:

Người dùng có thể thay đổi các thông số giao tiếp của PORT1 trong PLC (Không thay đổi các thông số giao tiếp của Port1, nếu không có thể dẫn đến lỗi kết nối giữa PC và PLC!)

Phần mềm chuyển đổi USB được cài đặt không chính xác hoặc cáp chuyển đổi USB bị lỗi.

Cổng giao tiếp PORT1 của PLC bị hỏng.

Thương hiệu cáp giao tiếp download không phải là cáp Xinje XVP.

Giải pháp:

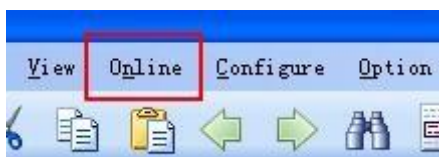
Đầu tiên, sử dụng cáp Xinje XVP để kết nối PC và PLC;

Sau khi xác nhận cáp kết nối là cáp XVP đặc biệt Xinje và bộ chuyển đổi USB đã được sử dụng, bạn có thể sử dụng nó để thử kết nối máy tính để bàn có cổng nối tiếp 9 kim với PLC.

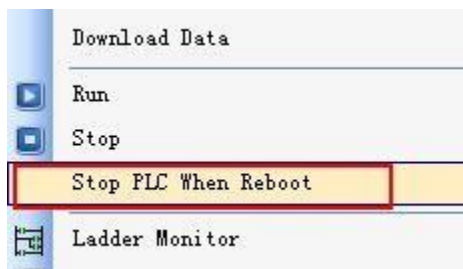
Nếu máy tính để bàn có thể được kết nối chính xác, vui lòng thay đổi cáp chuyển đổi USB có hiệu suất cao hơn hoặc cài đặt lại phần mềm chuyển đổi USB.

Nếu PLC cũng không thể kết nối chính xác với máy tính để bàn, bạn có thể sử dụng chức năng 'stop PLC when reboot/ dừng PLC khi khởi động lại' để dừng PLC và khôi phục PLC về cài đặt gốc, cách thức thao tác như sau:

Bật nguồn PLC và kết nối PLC bằng cáp DVP, sau đó nhấp vào nút “online/ trực tuyến” trên menu phần mềm chỉnh sửa PLC;



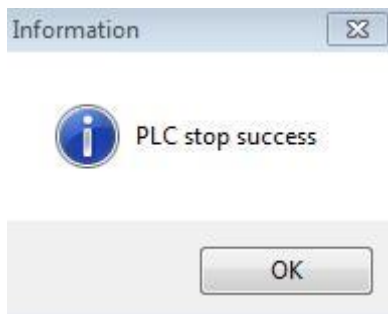
Nhấp vào 'Stop when PLC reboot' từ menu thả xuống



Cửa sổ sau sẽ hiển thị



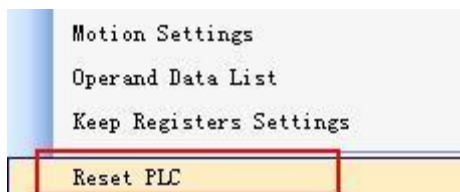
Lúc này, hãy tắt nguồn PLC trong 2-3 giây và bật lại, sau đó cửa sổ 'PLC đã dừng thành công. PLC has been stopped successfully' sẽ hiển thị; nếu cửa sổ không hiển thị sau khi bật nguồn, hãy thử lại vài lần cho đến khi cửa sổ báo dừng thành công hiển thị.



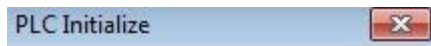
Sau đó nhấp vào nút 'configure';



Nhấp vào 'Reset PLC' trên menu thả xuống;

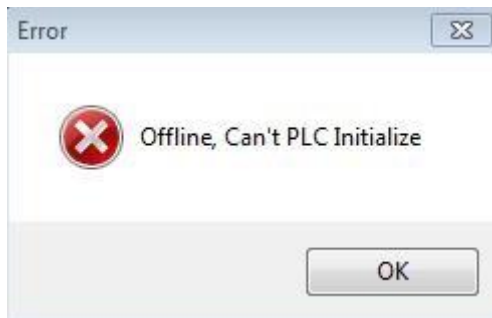


Lúc này, cửa sổ thông tin 'Reset PLC' sẽ hiển thị và điều đó có nghĩa là tất cả các bước 'Dừng khi khởi động lại PLC/ Stop when PLC reboot' đã hoàn tất.



OK

Nếu khởi tạo PLC không thành công sau khi bạn thử vài lần hoặc cửa sổ dưới đây xuất hiện sau khi nhấp vào 'Reset PLC':



Trong cả hai trường hợp, hãy sử dụng công cụ cập nhật hệ thống PLC để cập nhật hệ thống PLC và PLC và PC sẽ được kết nối thành công nếu hệ thống được cập nhật (Để biết thêm các bước về cập nhật hệ thống, vui lòng tham khảo nội dung liên quan đến Câu hỏi 3). Nếu việc cập nhật máy tính để bàn có cổng nối tiếp 9 chân không thành công, rất có thể cổng giao tiếp PLC bị hỏng, vui lòng liên hệ với nhà sản xuất hoặc đại lý.

Câu hỏi 3: Nâng cấp hệ thống PLC dòng XD/XL

Trả lời:

Khi nào PLC cần cập nhật thường xuyên?

Phần mềm PLC đang trong giai đoạn nâng cấp liên tục; nếu phiên bản phần mềm và phần cứng không khớp, PLC sẽ không hỗ trợ chức năng nâng cấp đó. Về phiên bản PLC nào được hỗ trợ lệnh, vui lòng tham khảo sơ lược về lệnh trong sách hướng dẫn này hoặc phụ lục 2 'Yêu cầu phiên bản cho chức năng đặc biệt';

Khi người dùng thay đổi các tham số giao tiếp, PLC và PC không thể kết nối với nhau.

Khi người dùng sử dụng chức năng "tải xuống chương trình bảo mật/program confidential download" nhưng quên mật khẩu.

(Lưu ý: Chương trình PLC sẽ biến mất sau khi cập nhật hệ thống!).

Làm cách nào để cập nhật PLC dòng XD/XL?

Công cụ cập nhật PLC:

'Công cụ chương trình tải xuống PLC dòng XD' và 'tập hệ thống'
(tệp * .sys)

Đóng tất cả các chương trình có thể chiếm cổng nối tiếp

Cắt nguồn PLC, mở công cụ cập nhật dòng XD (nếu người dùng sử dụng công cụ này lần đầu tiên, vui lòng mở đăng ký trước)



Nhấp vào "Open File", chọn model PLC để cập nhật. (Lưu ý: XD3_16.sys phù hợp với model PLC XD3-16, XD3_60.sys phù hợp với model PLC XD3-32 và XD3-60):



Đặt thông số:

Nhấp vào “đặt thông số/ set parameter”, cửa sổ thông số sẽ hiển thị:



Lưu ý: đặt cổng giao tiếp, tốc độ truyền là cài đặt mặc định, không cần thay đổi. Nhấp vào “tải xuống/ download”, cửa sổ sẽ hiển thị như trong hình bên dưới:



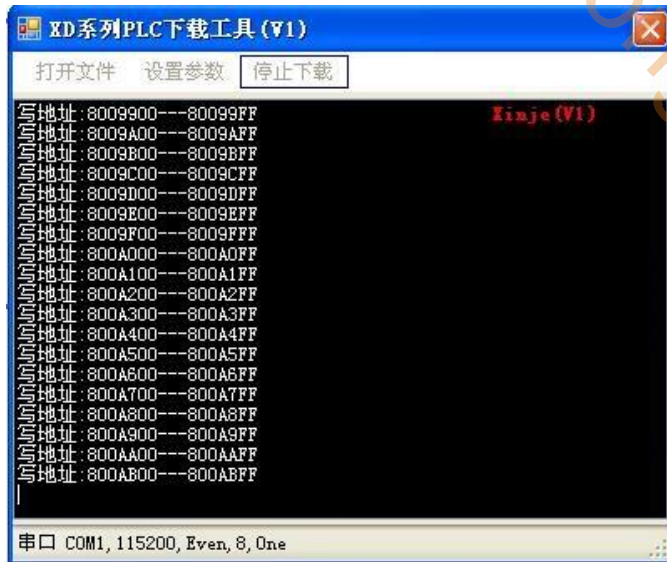
Bật nguồn PLC, công cụ cập nhật sẽ hiển thị các thông tin như bên dưới:



Ngắt nguồn PLC, nối dây **short jumper**, sau đó bật lại PLC.



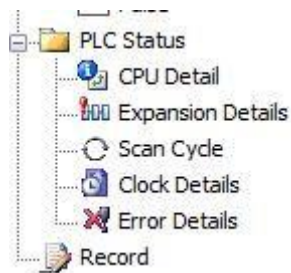
PLC bắt đầu cập nhật, quá trình cập nhật sẽ mất vài phút.

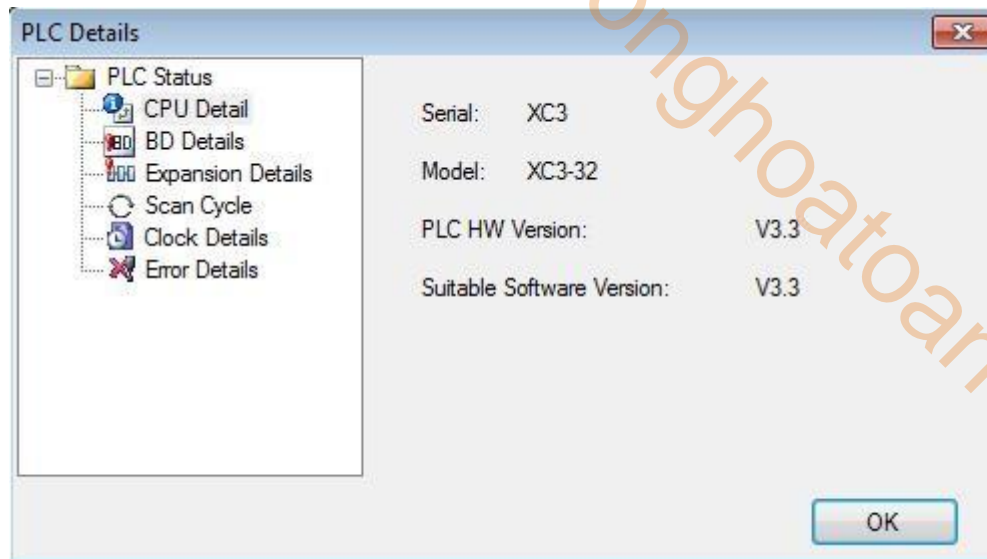


Sau khi cập nhật xong, tắt nguồn PLC, tháo **short jumper**, sau đó bật lại PLC.

Phiên bản phần cứng PLC

Phiên bản phần cứng PLC có thể được nhìn thấy trong “Chi tiết CPU/ CPU detail” trên cửa sổ bên trái trong phần mềm XDPpro (trạng thái trực tuyến PLC)





Short jumper

PLC dòng XD, XL không cần **no need to short the jumper** khi cập nhật:

Lưu ý:

Không cắt nguồn PLC khi đang cập nhật. Nếu xuất hiện lỗi “gửi dữ liệu không thành công, ID không khớp...” vui lòng liên hệ với chúng tôi để được trợ giúp.

Chương trình PLC sẽ bị xóa sau khi cập nhật.

Câu hỏi 4: Chức năng phần tử lập trình bit

Trả lời:

Bao gồm 16 cuộn dây liên tục kiểu single word, VD: DM0 single word bao gồm 16 cuộn dây (bit) M0~M15 như sau:

DM0:

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

Chúng ta có thể sử dụng bit trực tiếp trong thanh ghi.

Ví dụ 1:



Khi M100 từ OFF sang ON, M0 M1 ở trạng thái ON, M2—M15 ở trạng thái OFF

Chế độ khác là thao tác bit của thanh ghi cố định. Ví dụ: D0.0 là bit đầu tiên gồm 16 bit trong thanh ghi D0. Tương tự, D0.1 là bit thứ hai, v.v., như hiển thị bên dưới:

D0:

D0.15	D0.14	D0.13	D0.12	D0.11	D0.10	D0.9	D0.8	D0.7	D0.6	D0.5	D0.4	D0.3	D0.2	D0.1	D0.0
-------	-------	-------	-------	-------	-------	------	------	------	------	------	------	------	------	------	------

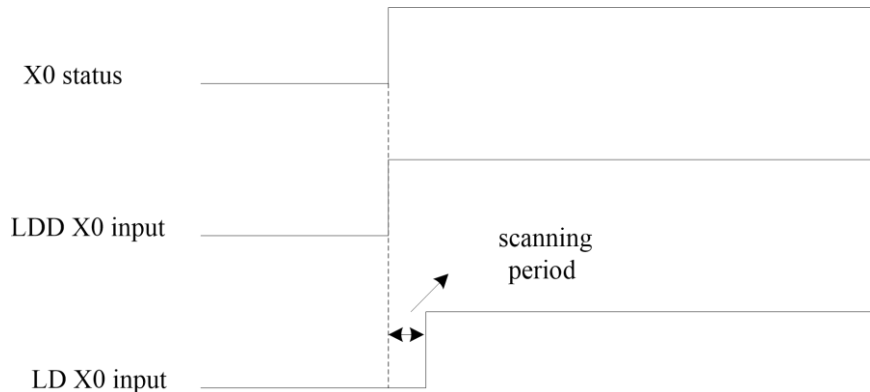
Tương tự, chúng ta có thể sử dụng bit trong thanh ghi D0.

Câu hỏi 5: What's the use of execution instruction LDD/OUTD etc?

Trả lời:

Khi PLC thực thi chương trình, trạng thái của điểm đầu vào sẽ ánh xạ tới thanh ghi hình ảnh. Từ đó trở đi, PLC sẽ làm mới trạng thái đầu vào vào đầu mỗi chu kỳ quét; nếu chúng ta sử dụng lệnh LDD thì trạng thái của điểm đầu vào sẽ không cần ánh xạ tới thanh ghi hình ảnh; tương tự với điểm đầu ra (OUTD).

Lệnh LDD/OUTD thường áp dụng trong trường hợp I/O cần làm mới ngay lập tức, điều này làm cho trạng thái đầu vào và đầu ra tránh được ảnh hưởng của chu kỳ quét.



Sơ đồ trình tự điểm đầu vào X0 của LDD và LD

Câu hỏi 6: Tại sao đèn LED đầu ra nhấp nháy liên tục khi sử dụng lệnh ALT?

Trả lời:

Đối với lệnh ALT và nhiều lệnh tính toán, các lệnh này sẽ thực thi mỗi chu kỳ quét khi điều kiện được đáp ứng (ví dụ: điều kiện là cuộn dây BẬT bình thường). Chúng tôi khuyến nghị rằng điều kiện kích hoạt là sườn lên hoặc sườn xuống.

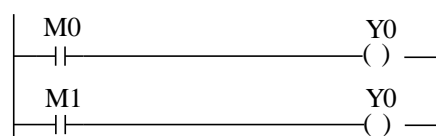
Câu hỏi 7: Tại sao đôi khi M và Y không thể xuất kết quả?

Trả lời:

Xuất kết quả chủ yếu có hai cách: 1. Lệnh OUT; 2. Lệnh SET. Cuộn dây sẽ tiếp tục xuất ra nếu không có lệnh RST.

Thông thường trong chương trình, một cuộn dây M hoặc Y nên sử dụng cùng một cách xuất kết quả. Nếu không thì cuộn dây không thể xuất kết quả.

Ví dụ:



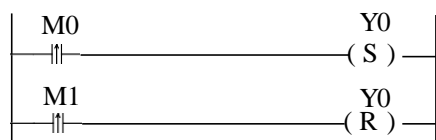
M0 ở trạng thái ON, M1 đang OFF, Y0 không thể xuất

M0 ở trạng thái OFF, M0 đang ON, Y0 sẽ xuất

Lý do: hai cuộn dây khác nhau điều khiển cùng một cuộn dây đầu ra



Y0 sẽ ON trong khoảng thời gian quét



M0 đang ON, Y sẽ tiếp tục xuất nếu M1 đang ON, Y0 đang OFF

Câu hỏi 8: Kiểm tra và thay pin nút bấm trong PCB của PLC

Trả lời:

Điện áp định mức của pin nút bấm là 3V. Điện áp có thể được đo bằng đồng hồ vạn năng. Nếu giá trị của thanh ghi khả nhớ khi mất điện rất lớn, điều đó có nghĩa là pin yếu. Vui lòng thay pin nút. Người dùng có thể sử dụng SM5 và SD5 để phát hiện nguồn điện của pin nút nhằm thay pin kịp thời. Xem Phụ lục 1 và Phụ lục 2 để biết chi tiết.

Câu hỏi 9: Giao tiếp với phần mềm SCADA

Trả lời:

Nếu không có lựa chọn nào cho PLC dòng XD/XL trong phần mềm SCADA, vui lòng chọn giao thức Modbus-RTU và giao tiếp qua cổng RS485. Vui lòng tham khảo Chương 6 sách hướng dẫn sử dụng PLC dòng XD/XL.

Câu hỏi 10: Giao tiếp MODBUS

Trả lời:

Trước hết, hãy đảm bảo rằng ngõ A và B trên PLC được kết nối chính xác với ngõ giao tiếp RS485 của các thiết bị khác. Để sửa đổi các tham số của PORT 2 của PLC, có thể sử dụng các cách sau:

Cách 1: Cấu hình bằng lệnh tham số cấu hình

Để biết các lệnh cụ thể, vui lòng tham khảo Chương 6: Chức năng Giao tiếp của sách hướng dẫn này. Cài đặt thông số giao tiếp của các thiết bị khác nhau thường sẽ khác nhau, vì vậy điều quan trọng là phải chọn chế độ cài đặt tần số chính xác của thiết bị liên lạc, làm rõ địa chỉ và mã chức năng giao tiếp MODBUS tương ứng và một số thiết bị liên lạc cần có tín hiệu hoạt động nhất định trước khi hiển thị cài đặt Tần số

Cách 2: Cấu hình thông qua bảng điều khiển (tham khảo Chương 6: Chức năng Giao tiếp của sách hướng dẫn này để biết phương pháp cấu hình cụ thể).

Câu hỏi 11: Đèn LED của PLC dòng XD/XL (PWR/RUN/ERR)

Trả lời

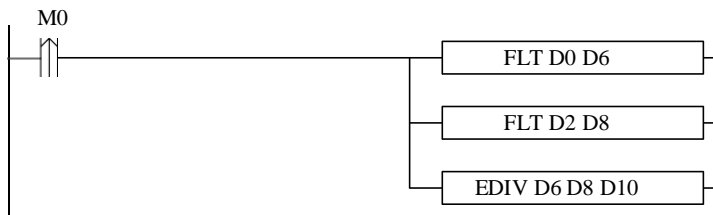
Đèn LED	Vấn đề	Giải pháp
PWR sáng, đèn LED khác tắt.	<ol style="list-style-type: none"> 1. PCB I/O bị ngắn mạch 2. Quá tải đối với nguồn 24V 3. Không nhấp vào RUN để chạy chương trình 	Kiểm tra ngõ I/O nếu có hiện tượng ngắn mạch. Đảm bảo chương trình đang chạy bên trong PLC. Nếu tải quá lớn,

		đối với nguồn điện 24V, hãy liên hệ với chúng tôi để được hỗ trợ.
Ba đèn LED đều TẮT	1. Nguồn điện đầu vào PLC bị ngắn mạch 2. Nguồn PCB của PLC bị hỏng	Kiểm tra nguồn điện đầu vào của PLC. Liên hệ với chúng tôi để được hỗ trợ.
Đèn PWR và ERR sáng	1. Điện áp đầu vào PLC không ổn định. 2. Có dead loop trong chương trình 3. Hệ thống PLC bị lỗi	Kiểm tra điện áp nguồn, kiểm tra xem có dead loop trong chương trình không. Cập nhật phần cứng của PLC. Liên hệ với chúng tôi để được hỗ trợ.

Câu hỏi 12: Kết quả không đúng khi thực hiện phép toán số thực dấu phẩy động

Trả lời:

Hãy chuyển đổi số nguyên thành số thực dấu phẩy động. Ví dụ: EDIV D0 D2 D10. Nếu giá trị của D0 và D2 là số nguyên thì kết quả sẽ bị lỗi (D10). Vui lòng sử dụng lệnh như bên dưới để chuyển đổi số nguyên thành số thực dấu phẩy động.

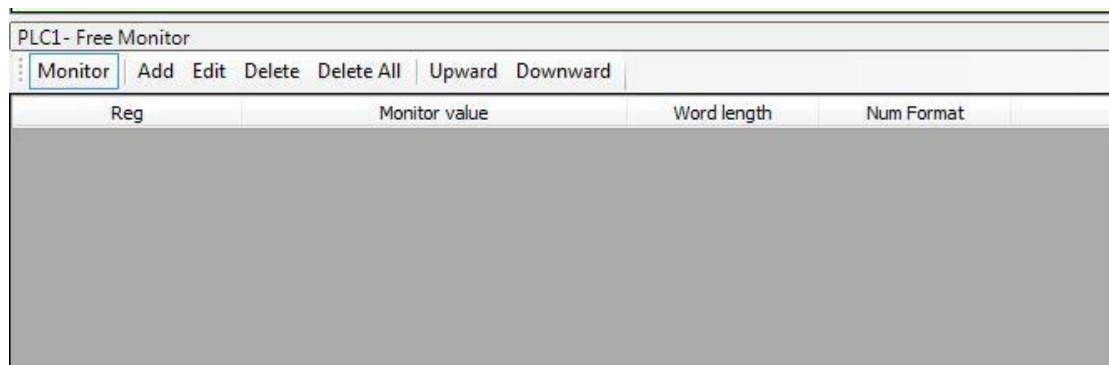


Câu hỏi 13: Tại sao các số thực dấu phẩy động trở thành mã lộn xộn trong cửa sổ điều khiển sơ đồ bậc thang trực tuyến?

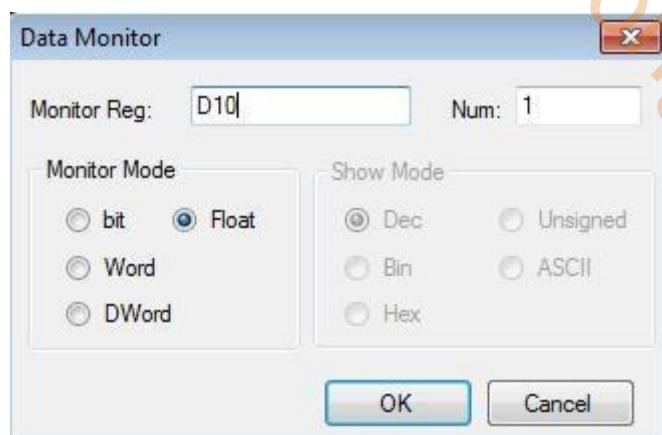
Trả lời:

Vì số thực dấu phẩy động không thể hiển thị trong điều khiển sơ đồ bậc thang trực tuyến, vui lòng điều khiển số thực dấu phẩy động trong chức năng điều khiển miễn phí.

Mở phần mềm XDPpro, nhấp vào điều khiển trực tuyến/miễn phí. Cửa sổ sau sẽ bật lên:



Nhấp vào “add” trong cửa sổ, cửa sổ sau sẽ hiển thị. Đặt chế độ điều khiển thành “số thực dấu phẩy động”. Thanh ghi điều khiển được đặt thành D10. Sau đó nhấp vào Ok.



Câu hỏi 14: Tại sao dữ liệu bị lỗi sau khi sử dụng lệnh DMUL?

Trả lời:

Lệnh DMUL là thao tác $32 \text{ bit} * 32 \text{ bit} = 64 \text{ bit}$, kết quả chiếm 4 word, chẳng hạn như: EMUL D0 D2 D10, hai số nhân đều là 32 bit (D1, D0) và (D3, D2), kết quả là 64 bit (D13, D12, D11, D10) nên D10~D13 sẽ bị chiếm. Nếu các thanh ghi dữ liệu này được sử dụng sau, thao tác sẽ bị lỗi.

Q15: Tại sao hoạt động của điểm đầu ra bị lỗi sau khi PLC chạy một lúc?

Trả lời:

Có thể cổng đầu ra bị lỏng, vui lòng kiểm tra.

Câu hỏi 16: Tại sao mô-đun mở rộng không hoạt động khi đèn báo nguồn SÁNG?

Trả lời:

Có khả năng kết nối của các dải mô-đun và chân PLC hoặc CPU không ổn định. So sánh CPU và mô-đun mở rộng theo cách so sánh chéo để tìm ra vấn đề.

Q17: Tại sao có tín hiệu đầu vào nhưng không thấy bộ đếm tốc độ cao hoạt động?

Trả lời:

Nếu việc đếm tốc độ cao được thực hiện, ngoài việc kết nối xung tốc độ cao với đầu vào bộ đếm tốc độ cao của PLC, chương trình đếm tốc độ cao tương ứng phải được viết cùng với các lệnh chức năng. Để biết chi tiết, vui lòng tham khảo nội dung liên quan trong Chương 5 của sách hướng dẫn này.

Câu hỏi 18: Ưu điểm của ngôn ngữ C so với sơ đồ bậc thang?

Trả lời:

- (1) PLC dòng XD/XL hỗ trợ hầu hết tất cả các chức năng của ngôn ngữ C. Khi nói đến các phép toán phức tạp, ưu điểm của ngôn ngữ C càng rõ ràng.
- (2) Tăng cường tính bảo mật của chương trình (khi sử dụng chế độ lưu trữ nâng cao tệp, ngôn ngữ C không thể tải lên);
- (3) Khối hàm ngôn ngữ C có thể được gọi ở nhiều vị trí và các tệp khác nhau, giúp cải thiện đáng kể hiệu quả của lập trình viên.

Câu hỏi 19: Ngõ đầu ra PLC A, B là gì?

Trả lời:

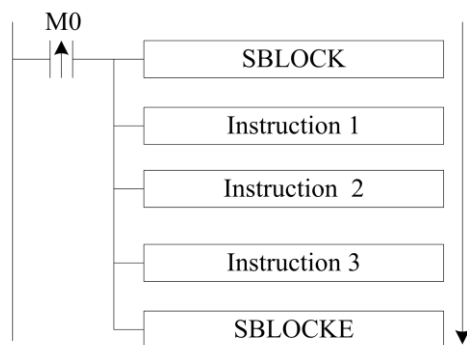
Ngõ đầu ra PLC A, B là các ngõ RS485 của PORT2 trên PLC.

Câu hỏi 20: Câu hỏi 20: Sự khác biệt giữa điều kiện kích hoạt BLOCK hàm tuần tự: kích hoạt sườn lên và tiếp điểm thường đóng là gì?

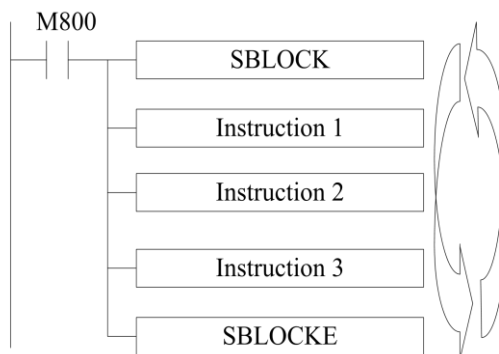
Trả lời:

Kích hoạt sườn lên: khi điều kiện được kích hoạt, khối sẽ thực thi theo thứ tự từ trên xuống dưới;

Normally closed conduction: khi điều kiện được kích hoạt, Block sẽ thực thi theo thứ tự từ trên xuống dưới, quay về đầu và thực hiện lại cho đến khi normally closed conduction bị đứt. Chu kỳ dừng lại khi lệnh cuối cùng kết thúc.



Từ trên xuống dưới, chạy từng lệnh một



từ trên xuống dưới, chạy lệnh theo chu kỳ

Câu hỏi 21: Các chế độ tải xuống của PLC dòng XD/XL và đặc điểm của chúng là gì

Trả lời:

PLC dòng XD/XL có ba chế độ tải xuống, đó là:

Common download mode

Chế độ tải xuống thông thường

Ở chế độ này, bạn có thể dễ dàng tải chương trình từ máy tính về PLC hoặc tải chương trình từ PLC về máy tính. Sẽ rất thuận tiện khi sử dụng chế độ này khi gỡ lỗi thiết bị.

Chế Độ Tải Xuống Có Mật Khẩu

Bạn có thể đặt mật khẩu cho PLC. Khi tải chương trình từ PLC lên máy tính, bạn cần nhập đúng mật khẩu. Trong tùy chọn mật khẩu nâng cao, bạn cũng có thể chọn chức năng "tải chương trình cần giải mã trước/ download the program needs to be decrypted first" (Lưu ý: Phải cẩn thận khi sử dụng thao tác này, nếu quên mật khẩu, PLC của bạn sẽ bị khóa!). Chế độ tải xuống này phù hợp với người dùng khi họ cần giữ bảo mật chương trình thiết bị và có thể gọi chương trình thiết bị bất cứ lúc nào.

Chế Độ Tải Xuống Bảo Mật

Ở chế độ này, chương trình trên máy tính có thể được tải xuống PLC, bất kể người dùng có thể tải chương trình trong PLC lên máy tính bằng cách nào; đồng thời, chương trình người dùng có thể được tải xuống một cách bảo mật, có thể chiếm ít tài nguyên bên trong của PLC hơn, tăng đáng kể dung lượng chương trình của PLC và có thể có tốc độ tải xuống nhanh hơn; sau khi sử dụng chế độ tải xuống này, chương trình sẽ hoàn toàn không thể khôi phục được.

Câu hỏi 22: PLC dòng XD/XL có những loại phương pháp bảo mật nào?

Trả lời:

Xinje PLC có ba phương pháp bảo mật: (1) nhập và xuất các tệp đã tải xuống; (2) tải xuống bảo mật; (3) tải xuống có mật khẩu.

Nhập và xuất file tải xuống: Sau khi lưu chương trình PLC theo cách này, người dùng có thể tải xuống và sử dụng chương trình nhưng không thể xem và chỉnh sửa chương trình.

Tải xuống bảo mật: Sau khi tải xuống bảo mật PLC, chương trình và dữ liệu trong PLC sẽ không được tải lên, nghĩa là "chương trình không tồn tại".

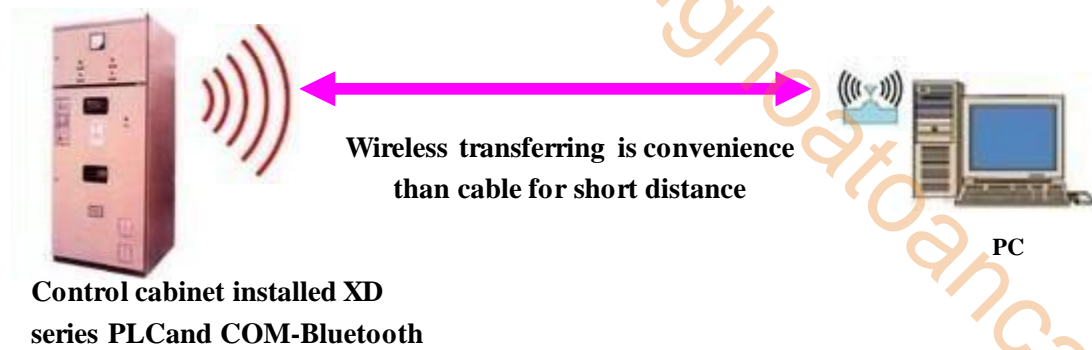
Tải xuống có mật khẩu: Nếu bạn tải xuống chương trình đã đặt mật khẩu cho PLC, bạn cần nhập đúng mật khẩu khi tải lên chương trình PLC; nếu bạn chọn "chương trình tải xuống cần được giải mã trước", bạn cũng cần nhập đúng mật khẩu khi tải chương trình mới xuống PLC. Ở chế độ này, bạn không thể sửa đổi thông tin clock của PLC và tính bảo mật sẽ cao hơn.

Câu hỏi 23: Ưu điểm của PLC dòng XD thay thế cáp download DVP bằng Bluetooth là gì?

Trả lời:

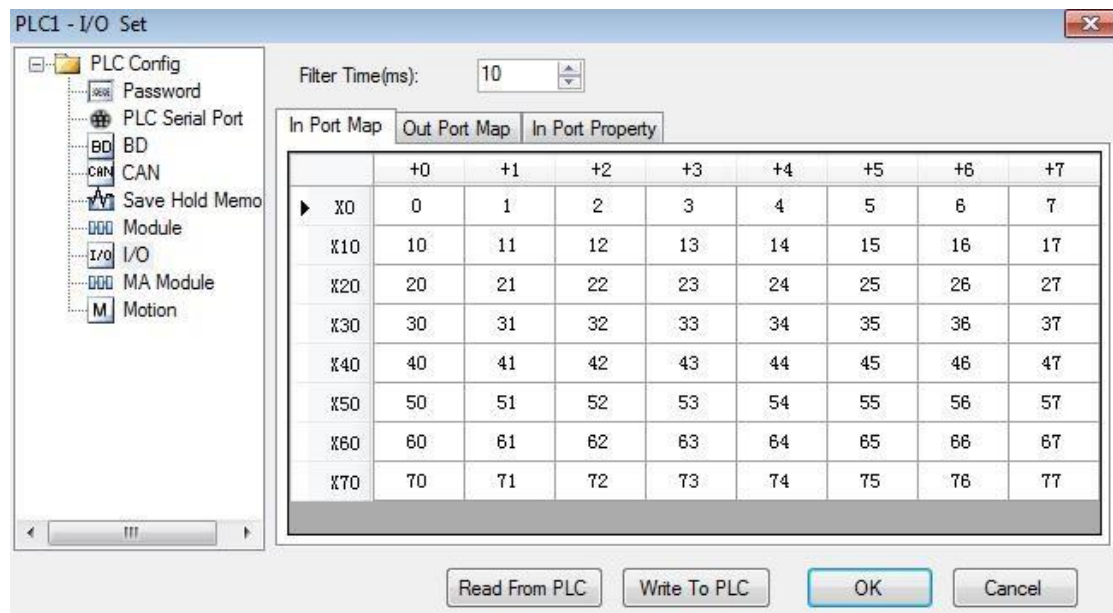
Chức năng Bluetooth của PLC dòng XD có thể thực hiện tải xuống và tải lên chương trình PLC, giám sát và mô phỏng trực tuyến phần mềm cấu hình Twin. Bluetooth có thể thay thế cáp để truyền dữ liệu.

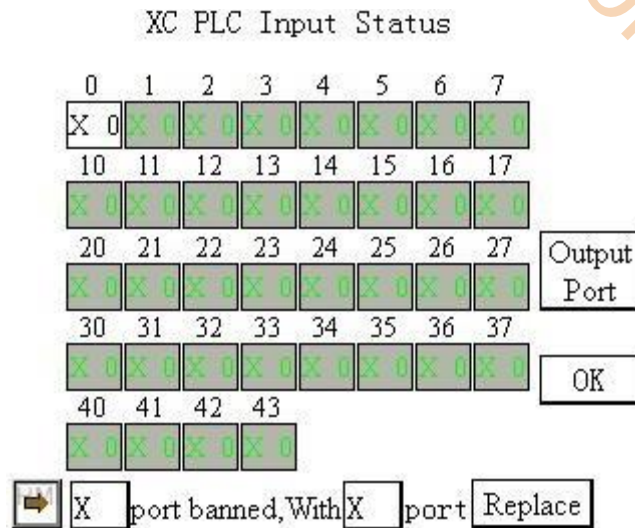
Note: COM-Bluetooth only fit for XINJE PLC.



Câu hỏi 24: Chức năng trao đổi ngõ I/O của PLC

Đôi khi các ngõ I/O của PLC bị hỏng. Người dùng không phải thay đổi chương trình, chức năng trao đổi ngõ I/O của PLC có thể giải quyết được vấn đề. Người dùng có thể trao đổi ngõ thông qua XINJE Touchwin HMI. Mở phần mềm Touchwin lên, đi đến màn hình 60004 (ngõ X) hoặc màn hình 60005 (ngõ Y) để thiết lập trao đổi I/O.





Màn hình trao đổi ngõ I/O của Touchwin HMI

Câu hỏi 25: Chức năng của việc gán địa chỉ gián tiếp PLC dòng XD/XL là gì?

Câu hỏi:

Việc thêm hậu tố **offset/phân bù** sau cuộn dây và thanh ghi dữ liệu (chẳng hạn như X3[D100], M10[D100], D0[D100]) có thể nhận ra chức năng đánh địa chỉ gián tiếp; chẳng hạn như D100=9, X3[D100] đại diện cho X14, M10[D100] đại diện cho M19, D0[D100] đại diện cho D9; Nó thường áp dụng cho số lượng lớn thao tác bit và thanh ghi và lưu trữ.

Câu hỏi 26: PLC dòng XD/XL kết nối với mạng như thế nào?

Trả lời:

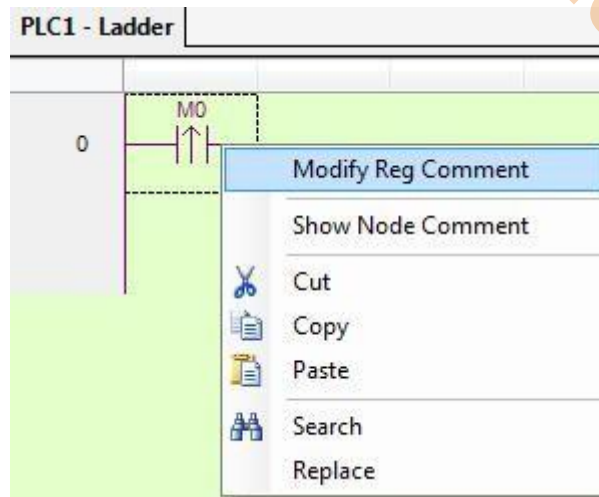
PLC dòng XD/XL có thể kết nối với mạng bằng các mô-đun mở rộng Xinje T-BOX, G-BOX, W-BOX, S-BOX, A-BOX hoặc bảng BD mở rộng có đặc điểm giao tiếp riêng. Để biết thêm chi tiết, vui lòng tham khảo hướng dẫn sử dụng của mô-đun giao tiếp hoặc bảng BD.

Câu hỏi 27: Làm cách nào để thêm ghi chú phần tử lập trình và **Line note** trong phần mềm XDppro?

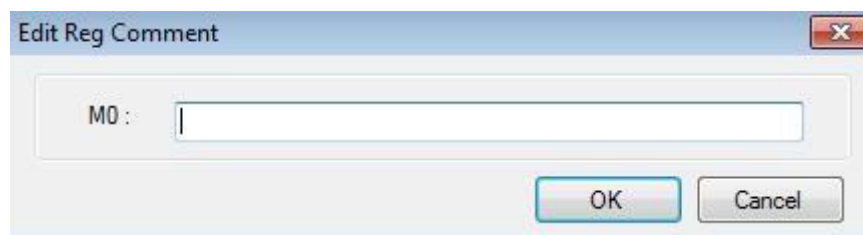
Trả lời:

Ghi chú phần tử lập trình

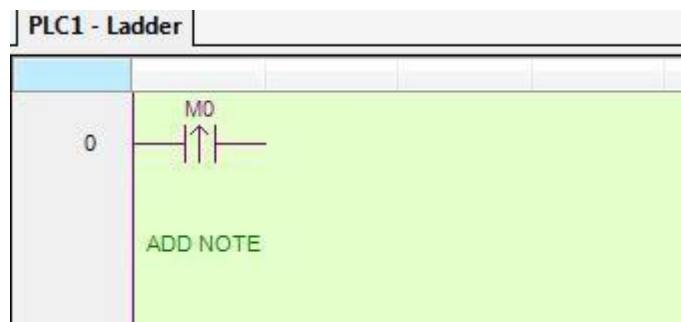
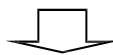
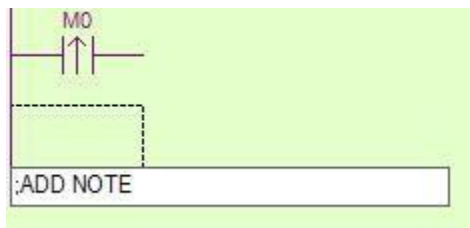
Mở phần mềm XDppro, di chuyển chuột đến phần mềm tương ứng rồi nhấn chuột phải, menu sẽ hiện thị:



Nhấp vào “Modify reg comment” để thêm ghi chú phần tử lập trình như cửa sổ bên dưới



Ghi chú dòng Ghi chú dòng bắt đầu bằng “;”. Bấm đúp vào dòng, sau đó nhập dấu chấm phẩy và nội dung.



Câu hỏi 28: PLC không có chức năng đồng hồ (clock)? Tại sao đồng hồ không chính xác?

Trả lời:

Chức năng đồng hồ PLC dòng XD/XL là tùy chọn và nếu bạn muốn mua PLC có chức năng đồng hồ, vui lòng xác nhận khi mua. Ngược lại, PLC mặc định khi xuất xưởng sẽ không có chức năng đồng hồ.

There are some errors in the clock of XD/XL series PLC. The error is about ± 5 minutes per month. Please calibrate it by HMI or directly in the PLC program.

Nếu bạn sử dụng PLC có chức năng đồng hồ, hãy kiểm tra xem giá trị trong thanh ghi Có một số lỗi trong đồng hồ của PLC dòng XD/XL. Sai số là khoảng ± 5 phút mỗi tháng. Vui lòng hiệu chỉnh nó bằng HMI hoặc trực tiếp trong chương trình PLC.

Phụ Lục Các Thành Phần Mềm Đặc Biệt

Phụ lục này chủ yếu giới thiệu các chức năng của phần tử mềm đặc biệt PLC dòng XD/XL, thanh ghi dữ liệu, FlashROM và phân bố địa chỉ của các phần mở rộng để người dùng tìm kiếm.

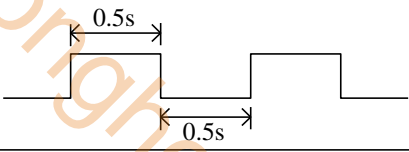
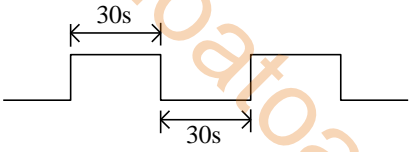
Phụ lục 1 Role trung gian đặc biệt

Trạng thái ban đầu (SM0-SM7)

ID	Chức năng	Mô tả
SM000	Cuộn dây ON khi chạy	
SM001	Cuộn dây OFF khi chạy	
SM002	Cuộn dây xung dương ban đầu	
SM003	Cuộn dây xung âm ban đầu	
SM004	Lỗi chạy PLC	Khi SM4 ở trạng thái ON, nghĩa là có lỗi trong hoạt động của PLC. (Firmware phiên bản V3.4.5 trở lên hỗ trợ chức năng này bằng PLC)
SM005	Cuộn báo động pin yếu	Khi điện áp pin nhỏ hơn 2,5V, SM5 sẽ ở trạng thái ON (lúc này vui lòng thay pin càng sớm càng tốt, nếu không dữ liệu sẽ không được duy trì)
SM007	Online download busy flag	Online download write back flash judge busy flag bit
SM008	Dấu hiệu hoàn thành sạc tụ điện	Hoàn thành sạc tụ điện (hỗ trợ XDH, XLH, XG2)

Thời gian/Clock (SM11-SM14)

ID	Chức năng	Mô tả
SM011	Chu kỳ tần số 10ms	
SM012	Chu kỳ tần số 100ms	

SM013	Chu kỳ tần số 1s	
SM014	Chu kỳ tần số 1 phút	

Cờ (SM20-SM22)

ID	Chức năng	Mô tả
SM020	Bit 0	SM020 ở trạng thái ON khi kết quả phép toán cộng/trừ là 0
SM021	Bit mượn	SM021 ở trạng thái ON khi tràn bit xảy ra trong phép toán trừ
SM022	Bit nhớ	SM022 ở trạng thái ON khi tràn bit xảy ra trong phép toán cộng

Chế độ PC(SM30~M34)

ID	Chức năng	Mô tả
SM030	Khởi tạo PLC	Khôi phục cài đặt gốc
SM032	Đặt lại thanh ghi khả nhớ	Khi SM032 ở trạng thái ON, bộ nhớ ánh xạ ON/OFF của HM, HS và các giá trị hiện tại của HT, HC, HD sẽ được đặt lại.
SM033	Xóa chương trình của người dùng	Khi SM033 ở trạng thái ON, tất cả chương trình của người dùng PLC sẽ bị xóa.
SM034	Chặn tất cả đầu ra	Khi SM034 ở trạng thái ON, tất cả các tiếp điểm bên ngoài PLC sẽ ở trạng thái OFF.

Sơ đồ bậc thang

ID	Chức năng	Mô tả
SM040	Quá trình đang chạy	Đặt ở trạng thái ON khi quá trình đang chạy

Cấm ngắt(SM50-SM90)

ID	Địa chỉ	Chức năng	Mô tả
SM050	I000/I0001	Cấm ngắt đầu vào 0	Sau khi thực thi lệnh EI, lệnh ngắt đầu vào không thể hoạt động độc lập khi M hoạt động, ngay cả khi ngắt được cho phép. Ví dụ: khi SM050 ở trạng thái ON, I000/I0001 bị cấm
SM051	I0100/I0101	Cấm ngắt đầu vào 1	
SM052	I0200/I0201	Cấm ngắt đầu vào 2	
SM053	I0300/I0301	Cấm ngắt đầu vào 3	
SM054	I0400/I0401	Cấm ngắt đầu vào 4	
.....	
SM069	I1900/I1901	Cấm ngắt đầu vào 19	Sau khi thực hiện lệnh EI, việc ngắt bộ định thời không thể hoạt động độc lập khi M hoạt động, ngay cả khi lệnh ngắt được cho phép.
SM070	I40**	Cấm ngắt bộ định thời 0	
SM071	I41**	Cấm ngắt bộ định thời 1	
SM072	I42**	Cấm ngắt bộ định thời 2	
SM073	I43**	Cấm ngắt bộ định thời 3	
SM074	I44**	Cấm ngắt bộ định thời 4	
.....	
SM089	I59**	Cấm ngắt bộ định thời 19	
SM090		Cấm tất cả lệnh ngắt	Cấm mọi lệnh ngắt

Bộ đếm vòng tốc độ cao (SM99)

Địa chỉ	Chức năng	Mô tả
SM099	Kích hoạt tính năng đếm vòng tốc độ cao	SM99 ở trạng thái ON, SD99 tăng thêm 1 tên mỗi 0,1ms, chu kỳ từ 0 đến 32767

Hoàn tất đếm tốc độ cao (SM100-SM109)

Địa chỉ	Chức năng	Mô tả
SM100	Cờ hoàn thành đếm HSC0 (100 đoạn)	
SM101	Cờ hoàn thành đếm HSC2 (100 đoạn)	
SM102	Cờ hoàn thành đếm HSC4 (100 đoạn)	
SM103	Cờ hoàn thành đếm HSC6 (100 đoạn)	
SM104	Cờ hoàn thành đếm HSC8 (100 đoạn)	
SM105	Cờ hoàn thành đếm HSC10 (100 đoạn)	
SM106	Cờ hoàn thành đếm HSC12 (100 đoạn)	
SM107	Cờ hoàn thành đếm HSC14 (100 đoạn)	
SM108	Cờ hoàn thành đếm HSC16 (100 đoạn)	
SM109	Cờ hoàn thành đếm HSC18 (100 đoạn)	

Chiều/ Hướng bộ đếm tốc độ cao (SM110-SM119)

Địa chỉ	Chức năng	Mô tả
SM110	Cờ chiều hoạt động của HSC0	
SM111	Cờ chiều hoạt động của HSC2	
SM112	Cờ chiều hoạt động của HSC4	
SM113	Cờ chiều hoạt động của HSC6	
SM114	Cờ chiều hoạt động của HSC8	
SM115	Cờ chiều hoạt động của HSC10	
SM116	Cờ chiều hoạt động của HSC12	
SM117	Cờ chiều hoạt động của HSC014	
SM118	Cờ chiều hoạt động của HSC16	
SM119	Cờ chiều hoạt động của HSC18	

Lỗi bộ đếm tốc độ cao (SM120-SM129)

Địa chỉ	Chức năng	Mô tả
SM120	Cờ lỗi HSC0	
SM121	Cờ lỗi HSC2	
SM122	Cờ lỗi HSC4	
SM123	Cờ lỗi HSC6	
SM124	Cờ lỗi HSC8	
SM125	Cờ lỗi HSC10	
SM126	Cờ lỗi HSC12	
SM127	Cờ lỗi HSC14	
SM128	Cờ lỗi HSC16	
SM129	Cờ lỗi HSC18	

Cờ tràn bộ đếm tốc độ cao (SM130~SM139)

Địa chỉ	Chức năng	Mô tả
SM130	Cờ tràn HSC0	
SM131	Cờ tràn HSC2	
SM132	Cờ tràn HSC4	
SM133	Cờ tràn HSC6	
SM134	Cờ tràn HSC8	
SM135	Cờ tràn HSC10	
SM136	Cờ tràn HSC12	
SM137	Cờ tràn HSC14	
SM138	Cờ tràn HSC16	
SM139	Cờ tràn HSC18	

Giao tiếp (SM140-SM193)

	Địa chỉ	Chức năng	Mô tả
Công nối tiếp 0	SM140	Cờ thực thi lệnh Modbus	Khi lệnh bắt đầu thực thi, đặt ở trạng thái ON Khi thực hiện xong, đặt ở trạng thái OFF
	SM141	Cờ thực thi lệnh X-NET	Khi lệnh bắt đầu thực thi, đặt ở trạng thái ON Khi thực hiện xong, đặt ở trạng thái OFF
	SM142	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Khi lệnh bắt đầu thực thi, đặt ở trạng thái ON Khi thực hiện xong, đặt ở trạng thái OFF
	SM143	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Khi nhận một khung dữ liệu hoặc nhận thời gian chờ dữ liệu, đặt ở trạng thái ON. Yêu cầu đặt chương trình người dùng ở trạng thái OFF
Công nối tiếp 1	SM150	Cờ thực thi lệnh Modbus	Tương tự như SM140
	SM151	Cờ thực thi lệnh X-NET	Tương tự như SM141
	SM152	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Tương tự như SM142
	SM153	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Tương tự như SM143
Công nối tiếp 2	SM160	Cờ thực thi lệnh Modbus	Tương tự như SM140
	SM161	Cờ thực thi lệnh X-NET	Tương tự như SM141
	SM162	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Tương tự như SM142
	SM163	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Tương tự như SM143
Công nối tiếp 3	SM170	Cờ thực thi lệnh Modbus	Tương tự như SM140
	SM171	Cờ thực thi lệnh X-NET	Tương tự như SM141
	SM172	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Tương tự như SM142
	SM173	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Tương tự như SM143
Công nối tiếp 4	SM180	Cờ thực thi lệnh Modbus	Tương tự như SM140
	SM181	Cờ thực thi lệnh X-NET	Tương tự như SM141
	SM182	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Tương tự như SM142
	SM183	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Tương tự như SM143
Công	SM190	Cờ thực thi lệnh Modbus	Tương tự như SM140

nối tiếp 5	SM191	Cờ thực thi lệnh X-NET	Same to SM141
	SM192	Cờ gửi dữ liệu bằng chế độ giao tiếp tự do	Same to SM142
	SM193	Cờ hoàn tất nhận tín hiệu bằng chế độ giao tiếp tự do	Same to SM143

KHỐI hàm tuần tự (SM300-SM399)

ID	Chức năng	Mô tả
SM300	Cờ chạy BLOCK1	SM300 sẽ ON khi block1 đang chạy
SM301	Cờ chạy BLOCK2	SM301 sẽ ON khi block2 đang chạy
SM302	Cờ chạy BLOCK3	SM302 sẽ ON khi block3 đang chạy
SM303	Cờ chạy BLOCK4	SM303 sẽ ON khi block4 đang chạy
SM304	Cờ chạy BLOCK5	SM304 sẽ ON khi block5 đang chạy
SM305	Cờ chạy BLOCK6	SM305 sẽ ON khi block6 đang chạy
.....	
SM396	Cờ chạy BLOCK97	SM396 sẽ ON khi block97 đang chạy
SM397	Cờ chạy BLOCK98	SM397 sẽ ON khi block98 đang chạy
SM398	Cờ chạy BLOCK99	SM398 sẽ ON khi block99 đang chạy
SM399	Cờ chạy BLOCK100	SM399 sẽ ON khi block100 đang chạy

Kiểm tra lỗi (SM400-SM415)

ID	Chức năng	Mô tả
SM400	Lỗi I/O	Đèn LED ERR vẫn SÁNG, PLC không chạy và xuất kết quả, kiểm tra khi bật nguồn
SM401	Lỗi giao tiếp mô-đun mở rộng	
SM402	Lỗi giao tiếp BD	
.....		
SM405	Không có chương trình người dùng	Kiểm tra mã nội bộ sai
SM406	Lỗi chương trình	Kiểm tra mã hoặc bảng cấu hình triển khai sai
SM407	Lỗi kiểm tra SSFD	Đèn LED ERR vẫn SÁNG, PLC không chạy và xuất kết quả, kiểm tra khi bật nguồn
SM408	Lỗi bộ nhớ	Không thể xóa hoặc ghi trong thanh ghi Flash
SM409	Lỗi tính toán	
SM410	Tràn offset	Offset vượt quá phạm vi của phần tử lập trình
SM411	Tràn FOR-NEXT	Reset khi bật nguồn hoặc người dùng cũng có thể reset thủ công.
SM412	Điền dữ liệu không hợp lệ	Khi offset của thanh ghi tràn, giá trị trả về sẽ là giá trị SM372
SM413	Lỗi checksum được mã hóa	
SM414	Lỗi dữ liệu FLASH	
SM415	Cờ lỗi của đồng hồ thời gian thực RTC	Xác minh ngày và giờ RTC không thành công

Thông báo lỗi (SM450-SM463)

ID	Chức năng	Mô tả
SM450	Kiểm tra lỗi hệ thống	
SM451	Cờ ngắt lỗi cứng	
.....		
SM453	Lỗi thẻ SD	
SM454	Nguồn điện bị ngắt	
SM455	Lỗi lưu dữ liệu khi ngắt nguồn	
SM456	Cờ lỗi tải xuống trực tuyến	
.....		
SM460	ID mô-đun mở rộng không khớp	
SM461	ID mô-đun BD/ED không khớp	
SM462	Extension module communication overtime	
SM463	BD/ED module communication overtime	
SM464	Tràn dữ liệu giao tiếp mô-đun mở rộng	
SM465	Tràn dữ liệu giao tiếp mô-đun BD/ED	

Trạng thái Mô-đun mở rộng, BD(SM500)

ID	Chức năng	Mô tả
SM500	Hoàn tất đọc trạng thái mô-đun	

Phụ lục 2 Thanh ghi dữ liệu đặc biệt

Pin (SD5~SD7)

ID	Chức năng	Mô tả
SD005	Thanh ghi pin	the data will lose when PLC power off. Nó sẽ hiển thị 100 khi điện áp pin là 3V, nếu điện áp pin thấp hơn 2,5V, nó sẽ hiển thị 0, nghĩa là vui lòng thay pin mới ngay lập tức, nếu không dữ liệu sẽ mất khi tắt nguồn PLC.

Đồng hồ/Clock (SD10-SD019)

ID	Chức năng	Mô tả
SD010	Chu kỳ quét hiện tại	100us, us là đơn vị
SD011	Thời gian quét tối thiểu	100us, us là đơn vị
SD012	Thời gian quét tối đa	100us, us là đơn vị
SD013	Giây (thời gian)	0~59 (mã BCD)
SD014	Phút (thời gian)	0~59 (mã BCD)
SD015	Giờ(thời gian)	0~23 (mã BCD)
SD016	Ngày (thời gian)	1~31 (BCD code)
SD017	Tháng (thời gian)	1~12 (mã BCD)

SD018	Năm (Thời gian)	2000~2099 (mã BCD)
SD019	Thứ (Thời gian)	0(Chủ Nhật)~6(Thứ Bảy)(mã BCD)

Cờ (SD020-SD031)

ID	Chức năng	Mô tả
SD020	Loại model	
SD021	model(low-8)series(high-8)	
SD022	Phiên bản hệ thống tương thích (thấp)Phiên bản hệ thống (cao)	
SD023	Phiên bản mô hình tương thích (thấp) Phiên bản mô hình (cao)	
SD024	Thông tin model	
SD025	Thông tin model	
SD026	Thông tin model	
SD027	Thông tin model	
SD028	Phiên bản phần mềm phù hợp	
SD029	Phiên bản phần mềm phù hợp	
SD030	Phiên bản phần mềm phù hợp	
SD031	Phiên bản phần mềm phù hợp	

Sơ đồ bậc thang (SD040)

ID	Chức năng	Mô tả
SD40	Cờ của quá trình thực thi S	

Sơ đồ bậc thang (SD099)

ID	Chức năng	Mô tả
SD99	Bộ đếm vòng tốc độ cao	Khi SM99 được ở trạng thái on, SD99 cứ sau 0,1 mili giây lại tăng thêm 1 trong khoảng từ 0 đến 32767.

Đếm tốc độ cao (SD100-SD109)

ID	Chức năng	Mô tả
SD100	Đoạn hiện tại (Đoạn số n)	HSC00
SD101	Đoạn hiện tại (Đoạn số n)	HSC02
SD102	Đoạn hiện tại (Đoạn số n)	HSC04
SD103	Đoạn hiện tại (Đoạn số n)	HSC06
SD104	Đoạn hiện tại (Đoạn số n)	HSC08
SD105	Đoạn hiện tại (Đoạn số n)	HSC10
SD106	Đoạn hiện tại (Đoạn số n)	HSC12
SD107	Đoạn hiện tại (Đoạn số n)	HSC14

SD108	Đoạn hiện tại (Đoạn số n)	HSC16
SD109	Đoạn hiện tại (Đoạn số n)	HSC18

Lỗi bộ đếm tốc độ cao (SD120-SD129)

ID	Chức năng	Mô tả
SD120	Thông tin lỗi HSC0	
SD121	Thông tin lỗi HSC2	
SD122	Thông tin lỗi HSC4	
SD123	Thông tin lỗi HSC6	
SD124	Thông tin lỗi HSC8	
SD125	Thông tin lỗi HSC10	
SD126	Thông tin lỗi HSC12	
SD127	Thông tin lỗi HSC14	
SD128	Thông tin lỗi HSC16	
SD129	Thông tin lỗi HSC18	

Giao tiếp (SD140~SD199)

	ID	Chức năng	Mô tả
Serial port 0	SD140	Kết quả thực thi lệnh đọc/ghi Modbus	0: thành công 100: lỗi nhận 101: receive overtime 180: lỗi CRC 181: lỗi LRC 182: lỗi trạm 183: lỗi tràn bộ nhớ đệm 400: lỗi mã hàm 401: lỗi địa chỉ 402: lỗi độ dài 403: lỗi dữ liệu 404: máy trạm slave bận 405: lỗi bộ nhớ (xóa bộ nhớ FLASH)
	SD141	Kết quả giao tiếp X-Net	0: thành công 1: communication overtime 2: lỗi bộ nhớ 3: lỗi nhận CRC
	SD142	Kết quả gửi dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn bộ nhớ đệm bằng chế độ giao tiếp tự do
	SD143	Kết quả nhận dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn chiều dài dữ liệu 411: nhận thiếu dữ liệu 412: nhận thiếu dữ liệu 413: lỗi nhận 414: receive overtime 415: không có ký tự bắt đầu 416: không có ký tự kết thúc

	SD144	Số lượng dữ liệu nhận bằng chế độ giao tiếp tự do	Tính bằng byte, không bao gồm ký tự bắt đầu và kết thúc
		
	SD149		
Công nối tiếp 1	SD150	Kết quả thực thi lệnh đọc/ghi Modbus	0: thành công 100: lỗi nhận 101: receive overtime 180: lỗi CRC 181: lỗi LRC 182: lỗi trạm 183: lỗi tràn bộ nhớ đệm 400: lỗi mã hàm 401: lỗi địa chỉ 402: lỗi độ dài 403: lỗi dữ liệu 404: máy trạm slave bận 405: lỗi bộ nhớ (xóa bộ nhớ FLASH)
	SD151	Kết quả giao tiếp X-Net	0: thành công 1: communication overtime 2: lỗi bộ nhớ 3: lỗi nhận CRC
	SD152	Kết quả gửi dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn bộ nhớ đệm bằng chế độ giao tiếp tự do
	SD153	Kết quả nhận dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn chiều dài dữ liệu 411: nhận thiếu dữ liệu 412: nhận thiếu dữ liệu 413: lỗi nhận 414: receive overtime 415: không có ký tự bắt đầu 416: không có ký tự kết thúc
	SD154	Số lượng dữ liệu nhận bằng chế độ giao tiếp tự do	Tính bằng byte, không bao gồm ký tự bắt đầu và kết thúc
		
	SD159		
Công nối tiếp 2	SD160	Kết quả thực thi lệnh đọc/ghi Modbus	0: thành công 100: lỗi nhận 101: receive overtime 180: lỗi CRC 181: lỗi LRC 182: lỗi trạm 183: lỗi tràn bộ nhớ đệm 400: lỗi mã hàm 401: lỗi địa chỉ 402: lỗi độ dài 403: lỗi dữ liệu 404: máy trạm slave bận 405: lỗi bộ nhớ (xóa bộ nhớ FLASH)
	SD161	Kết quả giao tiếp X-Net	0: thành công 1: communication overtime 2: lỗi bộ nhớ

			3: lỗi nhận CRC
	SD162	Kết quả gửi dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn bộ nhớ đệm bằng chế độ giao tiếp tự do
	SD163	Kết quả nhận dữ liệu bằng giao tiếp tự do	0: thành công 410: gửi tràn chiều dài dữ liệu 411: nhận thiếu dữ liệu 412: nhận thiếu dữ liệu 413: lỗi nhận 414: receive overtime 415: không có ký tự bắt đầu 416: không có ký tự kết thúc
	SD164	Số lượng dữ liệu nhận bằng chế độ giao tiếp tự do	Tính bằng byte, không bao gồm ký tự bắt đầu và kết thúc
		
	SD169		
Công nối tiếp 3	SD170~SD179		
Công nối tiếp 4	SD180~SD189		
Công nối tiếp 5	SD190~SD199		

Khối hàm tuần tự (SD300-SD399)

ID	Chức năng	Mô tả
SD300	Thực thi lệnh của BLOCK1	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD301	Thực thi lệnh của BLOCK2	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD302	Thực thi lệnh của BLOCK3	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD303	Thực thi lệnh của BLOCK4	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD304	Thực thi lệnh của BLOCK5	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD305	Thực thi lệnh của BLOCK6	Giá trị sẽ được sử dụng khi BLOCK điều khiển
.....
SD396	Thực thi lệnh của BLOCK95	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD397	Thực thi lệnh của BLOCK96	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD398	Thực thi lệnh của BLOCK97	Giá trị sẽ được sử dụng khi BLOCK điều khiển
SD399	Thực thi lệnh của BLOCK98	Giá trị sẽ được sử dụng khi BLOCK điều khiển

Kiểm tra lỗi (SD400-SD413)

ID	Chức năng	Mô tả
SD400		
SD401	Lỗi giao tiếp của mô-đun mở rộng số	Có nghĩa là mô-đun n bị lỗi
SD402	Lỗi giao tiếp của mô-đun BD/ED số	
SD403	Lỗi FROM/TO	
SD404	Lỗi PID	
SD405	Không có chương trình người dùng	
SD406	Loại lỗi chương trình người dùng	
SD407	Lỗi SSDF	
SD408	Lỗi xóa thanh ghi flash	
SD409	Lỗi tính toán	1: Lỗi chia cho 0 2: Địa chỉ toán hạng phía trước lệnh MRST, MSET nhỏ hơn toán hạng phía sau 3: Các bit dữ liệu ENCO, DECO của lệnh mã hóa và giải mã vượt quá giới hạn. 4: Lỗi mã BDC 7: Lỗi dấu căn
SD410	The number of offset register D when offset crosses the Boundary	
SD411		
SD412	Giá trị điền dữ liệu không hợp lệ (16 bit thấp)	
SD413	Giá trị điền dữ liệu không hợp lệ (16 bit cao)	
SD414	Lỗi dữ liệu thanh ghi flash	
SD415	Lỗi đồng hồ thời gian thực RTC	1: Bộ nguồn RTC có tình trạng điện áp thấp và cần được ghi lại 2: RTC ghi dữ liệu và chip đồng hồ không phản hồi tín hiệu ACK 3: Ghi dữ liệu ngày giờ không hợp lệ

Kiểm tra lỗi (SD450-SD465)

ID	Chức năng	Mô tả
SD450	1: Watchdog act (Mặc định 200ms) 2: Áp dụng khối điều khiển bị lỗi 3: Địa chỉ không hợp lệ	
SD451	Lỗi phản ứng: 1: Lỗi thanh ghi 2: Lỗi bus 3: Lỗi sử dụng	
SD452	Lỗi phản ứng	
SD453	Lỗi thẻ SD	
SD454	Thời gian tắt nguồn	
SD455		

SD456		
SD460	ID mô-đun mở rộng không khớp	
SD461	ID mô-đun BD/ED không khớp	
SD462	Extension module communication overtime	
SD463	BD/ED module communication overtime	
SD464	Tràn dữ liệu truyền thông của mô-đun mở rộng số.	
SD465	Tràn dữ liệu truyền thông của mô-đun BD/ED số.	

Trạng thái của mô-đun mở rộng, BD (SD500-SD516)

ID	Chức năng	Mô tả
SD500	Mô-đun số. Các mô-đun mở rộng: #10000~10015 BD: #20000~20001 ED: #30000	
SD501~516	Mô-đun mở rộng, trạng thái BD /ED	16 thanh ghi

Thông tin mô-đun (SD520-SD823)

ID	Chức năng	Mô tả	Ghi chú
SD520~SD535	Thông tin mô-đun mở rộng	Mô-đun mở rộng 1	Mỗi mô-đun mở rộng, BD, ED chiếm 16 thanh ghi
.....	
SD760~SD775	Thông tin mô-đun mở rộng	Mô-đun mở rộng 16	
SD776~SD791	Thông tin mô-đun BD	Mô-đun BD 1	
SD792~SD807	Thông tin mô-đun BD	Mô-đun BD 2	
SD808~SD823	Thông tin mô-đun BD	Mô-đun ED 1	

Thông tin lỗi mô-đun mở rộng

ID	Chức năng	Mô tả	
SD860	Số lần lỗi khi đọc mô-đun		Mô-đun mở rộng 1
SD861	Lỗi đọc mô-đun	Lỗi địa chỉ mô-đun Lỗi độ dài dữ liệu được chấp nhận của mô-đun. Lỗi chẵn lẻ CRC của mô-đun khi PLC nhận dữ liệu. Lỗi ID mô-đun. Module overtime error.	
SD862	Số lần lỗi khi ghi mô-đun		
SD863	Lỗi ghi mô-đun		

SD864	Số lần lỗi khi đọc mô-đun		Mô-đun mở rộng 2
SD865	Lỗi đọc mô-đun	Lỗi địa chỉ mô-đun Lỗi độ dài dữ liệu được chấp nhận của mô-đun. Lỗi chẵn lẻ CRC của mô-đun khi PLC nhận dữ liệu. Lỗi ID mô-đun. Module overtime error.	
SD866	Số lần lỗi khi ghi mô-đun		
SD867	Lỗi ghi mô-đun		
.....			
SD920	Số lần lỗi khi đọc mô-đun		Mô-đun mở rộng 16
SD921	Lỗi đọc mô-đun	Lỗi địa chỉ mô-đun Lỗi độ dài dữ liệu được chấp nhận của mô-đun. Lỗi chẵn lẻ CRC của mô-đun khi PLC nhận dữ liệu. Lỗi ID mô-đun. Module overtime error.	
SD922	Số lần lỗi khi ghi mô-đun		
SD923	Lỗi ghi mô-đun		
SD930	Thời lượng đọc/ghi ED hiện tại	100us, đơn vị: us	Mô-đun ED 1
SD931	Thời lượng đọc/ghi ED tối thiểu		
SD932	Thời lượng đọc/ghi ED tối đa		
SD933	Thời lượng lỗi đọc/ghi ED		
SD934	Tổng số lần đọc và ghi ED (16 bit thấp)		
SD935	Tổng số lần đọc và ghi ED (16 bit cao)		
SD936	Thời lượng đọc/ghi BD1 hiện tại	100us, đơn vị: us	Mô-đun BD 1
SD937	Thời lượng đọc/ghi BD1 tối thiểu		
SD938	Thời lượng đọc/ghi BD1 tối đa		
SD939	Số lần lỗi đọc/ghi BD1		
SD940	Tổng số lần đọc và ghi BD1 (16 bit thấp)		
SD941	Tổng số lần đọc và ghi BD1 (16 bit cao)		
SD942	Thời lượng đọc/ghi BD2 hiện tại		Mô-đun BD 2
SD943	Thời lượng đọc/ghi BD2 tối thiểu		

SD944	Thời lượng đọc/ghi BD2 tối đa		
SD945	Số lần lỗi đọc/ghi BD2		
SD946	Tổng số lần đọc và ghi BD2 (16 bit thấp)		
SD947	Tổng số lần đọc và ghi BD2 (16 bit cao)		

Thông tin phiên bản(SD990~SD993)

ID	Chức năng	Mô-tả	Ghi chú
SD990	Ngày phiên bản firmware	16-bit thấp	
SD991	Ngày biên soạn phiên bản firmware	16-bit cao	
SD992	Ngày biên soạn phiên bản FPGA	16-bit thấp	
SD993	Ngày biên soạn phiên bản FPGA	16-bit cao	

Chức năng đặc biệt (HSD50~HSD60)

ID	Chức năng	Mô tả
HSD50	Lưu giữ dữ liệu thời gian ghi lại sau khi mất điện ^{*1}	Single word, đơn vị:1ms
HSD51	Phát hiện mất điện	CPU working time after power failure,unit:100us
HSD52	Thời gian hoạt động PLC lần trước đó (16 bit thấp)	Double word, đơn vị:1s
HSD53	Thời gian hoạt động PLC lần trước đó (16 bit cao)	
HSD54	Thời gian hoạt động PLC hiện tại (16 bit thấp)	Double word, đơn vị:1s
HSD55	Thời gian hoạt động PLC hiện tại (16 bit cao)	
HSD58	Số lần xóa thanh ghi flash	

Ghi lỗi (HSD80~HSD179)

ID	Chức năng	Mô tả
HSD79	Error list index value	
HSD80~HSD84	Article 1 error message	
HSD85~HSD89	Article 2 error message	
HSD90~HSD94	Article 3 error message	
HSD95~HSD99	Article 4 error message	
HSD100~HSD104	Article 5 error message	

HSD105~HSD109	Article 6 error message	(1)PLC dòng XDC chỉ hỗ trợ lưu trữ 4 thông báo lịch sử lỗi; (2) Chức năng này yêu cầu phần mềm lập trình phiên bản v3.5.3 (20190326) trở lên. (3) Chế độ chuyển động H hỗ trợ tối đa 20 thông báo lỗi và chế độ chuyển động C hỗ trợ tối đa 4 thông báo lỗi.	
HSD110~HSD114	Article 7 error message		
HSD115~HSD119	Article 8 error message		
HSD120~HD124	Article 9 error message		
HSD125~HSD129	Article 10 error message		
HSD130~HD134	Article 11 error message		
HSD135~HSD139	Article 12 error message		
HSD140~HD144	Article 13 error message		
HSD145~HSD149	Article 14 error message		
HSD150~HD154	Article 15 error message		
HSD155~HSD159	Article 16 error message		
HSD160~HSD164	Article 17 error message		
HSD165~HSD169	Article 18 error message		
HSD170~HSD174	Article 19 error message		
HSD175~HSD179	Article 20 error message		
HSD180~HSD184	Article 21 error message		Chế độ chuyển động H được mở rộng tới 76 thông báo lỗi (hỗ trợ phiên bản firmware v3.7.2 trở lên)
HSD185~HSD189	Article 22 error message		
.....			
HSD450~HSD454	Article 75 error message		
HSD455~HSD459	Article 76 error message		

Lưu ý:

※ 1: HSD50 là “maintain data write back time after power failure” ở phiên bản 3.7.2 trở lên.

Phụ lục 3 Thanh ghi Flash đặc biệt

Thanh ghi dữ liệu FLASH đặc biệt SFD

* có nghĩa là nó chỉ hoạt động sau khi cấp nguồn lại cho PLC

Lọc I

ID	Chức năng	Mô tả
SFD0*	Thời gian lọc đầu vào	
SFD2*	Thời gian chạy của Watchdog, giá trị mặc định là 200ms	

Cấu hình chức năng đặc biệt

ID	Chức năng	Mô tả
SFD3*	Cấu hình chức năng đặc biệt (mặc định: 0x0000)	Bit0: Power down memory register exception handling.

		<p>0: hệ thống sẽ xóa nó; 1: Không xử lý Bit1: Thực thi chương trình người dùng trong chương trình con ngắt ngoài. 0: thực hiện nhiệm vụ; 1: Thực thi lệnh ngắt (ở chế độ này, chương trình con ngắt của người dùng không thể chứa khối hàm ngôn ngữ C. Chế độ này thường được sử dụng trong những trường hợp yêu cầu hiệu suất thời gian thực cao của tín hiệu bên ngoài. Bit2: có tăng mức ưu tiên ngắt bên ngoài hay không. 0: không tăng; 1: tăng (tăng đến mức độ cao nhất)</p>
--	--	---

Ảnh xạ I

ID	Chức năng	Mô tả	
SFD10*	I00 tương ứng với X**	Ngõ vào 0 tương ứng với X**	0xFF có nghĩa là ngõ vào bị lỗi, 0xFE có nghĩa là ngõ vào không hoạt động
SFD11*	I01 tương ứng với X**		
SFD12*	I02 tương ứng với X**		
.....		
SFD73*	I77 tương ứng với X**	Giá trị mặc định là 77 (Octonary)	

Ảnh xạ O

ID	Chức năng	Mô tả	
SFD74*	O00 tương ứng với Y**	Ngõ ra 0 tương ứng với Y**, Giá trị mặc định là 0	0xFF có nghĩa là ngõ vào bị lỗi, 0xFE có nghĩa là ngõ vào không hoạt động
.....		
SFD137*	O77 tương ứng với Y**	Giá trị mặc định là 77 (Octonary)	

Thuộc tính I

ID	Chức năng	Mô tả	
SFD138*	Thuộc tính I00	Thuộc tính của ngõ vào 0	0: logic + Khác: logic -
SFD139*	Thuộc tính I01		
.....		
SFD201*	Thuộc tính I77		

Đếm tốc độ cao

ID	Chức năng	Mô tả
SFD310	Cấu hình sườn đếm một pha HSC0	0: đếm sườn lên 1: đếm sườn xuống, 2: Cả hai sườn lên và sườn xuống đều được đếm

SFD311	Cấu hình sườn đếm một pha HSC2	0: đếm sườn lên 1: đếm sườn xuống, 2: Cả hai sườn lên và sườn xuống đều được đếm
SFD312	Cấu hình sườn đếm một pha HSC4	0: đếm sườn lên 1: đếm sườn xuống, 2: Cả hai sườn lên và sườn xuống đều được đếm
SFD313	Cấu hình sườn đếm một pha HSC6	0: đếm sườn lên 1: đếm sườn xuống, 2: Cả hai sườn lên và sườn xuống đều được đếm
SFD320	Số lần gấp tần số HSC0	2: gấp 2 lần tần số ; 4: gấp 4 lần tần số (hiệu quả ở chế độ đếm pha AB)
SFD321	Số lần gấp tần số HSC2	Như trên
SFD322	Số lần gấp tần số HSC4	Như trên
SFD323	Số lần gấp tần số HSC6	Như trên
SFD324	Số lần gấp tần số HSC8	Như trên
SFD325	Số lần gấp tần số HSC10	Như trên
SFD326	Số lần gấp tần số HSC12	Như trên
SFD327	Số lần gấp tần số HSC14	Như trên
SFD328	Số lần gấp tần số HSC16	Như trên
SFD329	Số lần gấp tần số HSC18	Như trên
SFD330	Lựa chọn bit của HSC tuyệt đối và tương đối (24 đoạn)	bit0 tương ứng với HSC0, bit1 tương ứng với HSC 2, v.v., bit9 tương ứng với HSC18 0: tương đối 1: tuyệt đối
SFD331	Ngắt tuần hoàn 24 đoạn đếm tốc độ cao	bit0 tương ứng với HSC0, bit1 tương ứng với HSC 2, v.v., bit9 tương ứng với HSC18 0: single 1: loop
SFD332	Chức năng CAM	bit0 tương ứng với HSC0, bit1 tương ứng với HSC 2, v.v., bit9 tương ứng với HSC18 0: không hỗ trợ chức năng CAM 1: hỗ trợ chức năng CAM

Cấu hình mô-đun mở rộng

ID	Chức năng	Mô tả
SFD340	Trạng thái cấu hình mô-đun mở rộng (#1#2)	Trạng thái cấu hình của mô-đun mở rộng 1 và 2
SFD341	Trạng thái cấu hình mô-đun mở rộng (#3#4)	Trạng thái cấu hình của mô-đun mở rộng 3 và 4
.....
SFD347	Trạng thái cấu hình mô-đun mở rộng (#15#16)	Trạng thái cấu hình của Mô-đun mở rộng 15 và 16
SFD348	Trạng thái cấu hình mô-đun BD (#1#2)	Trạng thái cấu hình của mô-đun BD 1 và 2
SFD349	Trạng thái cấu hình mô-đun ED (#1)	Trạng thái cấu hình của Mô-đun ED 1
SFD350	Cấu hình mô-đun mở rộng	Cấu hình của Mô-đun mở rộng 1
:		
SFD359		

SFD360	Cấu hình mô-đun mở rộng	
⋮		Cấu hình của mô-đun mở rộng 2
SFD369		
⋮	⋮	
SFD500		
⋮	Cấu hình mô-đun mở rộng	Cấu hình của Mô-đun mở rộng 16
SFD509		
SFD510		
⋮	Cấu hình mô-đun BD	Cấu hình của mô-đun BD 1
SFD519		
SFD520		
⋮	Cấu hình mô-đun BD	Cấu hình của mô-đun BD 2
SFD529		
SFD530		
⋮	Cấu hình mô-đun ED	Cấu hình mô-đun ED 1
SFD539		

Giao tiếp

ID	Chức năng	Ghi chú
SFD600	Số bit bộ đệm bằng chế độ giao tiếp tự do COM1	0: 8-bit 1: 16-bit
SFD610	Số bit bộ đệm bằng chế độ giao tiếp tự do COM2	0: 8-bit 1: 16-bit
SFD620	Số bit bộ đệm bằng chế độ giao tiếp tự do COM3	0: 8-bit 1: 16-bit
SFD630	Số bit bộ đệm bằng chế độ giao tiếp tự do COM4	0: 8-bit 1: 16-bit
SFD640	Số bit bộ đệm bằng chế độ giao tiếp tự do COM5	0: 8-bit 1: 16-bit

Phụ lục 4 Bảng xung đột tài nguyên PLC

Khi PLC được sử dụng trong thực tế, có thể xảy ra xung đột do một số tài nguyên được sử dụng cùng lúc. Phần này sẽ liệt kê các tài nguyên có thể gây ra xung đột trong mỗi mô hình PLC. Phần này chủ yếu đề cập đến việc đếm tốc độ cao, định thời chính xác và đầu ra xung .

Định thời chính xác	Đếm tốc độ cao				Xuất xung
XD2-16, XD3-16, XD5-16, XL3-16/32, XL5-16, XL5E-16					
ET0	-	-	-	-	-
ET2					
ET4					
ET6					
ET8	HSC0				
ET10		HSC2			
ET12			HSC4		
ET14					Y0
ET16					Y0
ET18					Y1
ET20					Y1
ET22					
ET24					
XD3-24/32/48/60, ZG3-30					
ET0					
ET2					
ET4					
ET6					
ET8					
ET10					
ET12	HSC0				
ET14		HSC2			
ET16			HSC4		
ET18					Y0
ET20					Y0
ET22					Y1
ET24					Y1
XD5-24/32/48/60, XDM-24/32/48/60, XD5E-24/30/48/60, XDME-30/60, XL5-32, XL5E-32/64					
ET0	-	-	-	-	-
ET2				HSC6	
ET4			HSC4		
ET6		HSC2			
ET8	HSC0				
ET10					Y3
ET12					Y3
ET14					Y2
ET16					Y2
ET18					Y1
ET20					Y1
ET22					Y0
ET24					Y0
XDC-24/32/48/60					

ET0	-	-	-	HSC6	-
ET2			HSC4		
ET4		HSC2			
ET6	HSC0				
ET8					Y3
ET10					Y3
ET12					Y2
ET14					Y2
ET16					Y1
ET18					Y1
ET20					Y0
ET22					Y0
ET24					

※ 1: Bảng này phải được đọc theo chiều ngang, không thể sử dụng cùng lúc hai tài nguyên trong mỗi hàng, nếu không sẽ gây ra xung đột.

Phụ lục 5 Danh sách cấu hình chức năng PLC

Phần này dùng để kiểm tra cấu hình của từng model. Thông qua bảng này, chúng ta có thể đánh giá loại sản phẩm một cách dễ dàng.

○ Có thể lựa chọn × Không hỗ trợ √ Hỗ trợ

Model	Cổng USB	Cổng 232	Cổng 485	Cổng RJ45	Mô-đun mở rộng	BD mở rộng	ED mở rộng trái	Kênh HSC Pha đơn/AB		Xuất xung (T/RT)		Ngắt ngoài
								OC	Differential	normal	Differential	
Dòng XD1												
XD1-10	×	2	×	×	×	×	×	×	×	×	×	6
XD1-16	×	2	×	×	×	×	×	×	×	×	×	6
XD1-24	×	2	1	×	×	×	×	×	×	×	×	10
XD1-32	×	2	1	×	×	×	×	×	×	×	×	10
Dòng XD2												
XD2-16	×	2	1	×	×	×	1	3	×	2	×	6
XD2-24	×	2	1	×	×	1	1	3	×	2	×	10
XD2-32	×	2	1	×	×	1	1	3	×	2	×	10
XD2-42	×	2	1	×	×	1	1	3	×	2	×	10
XD2-48	×	2	1	×	×	2	1	3	×	2	×	10
XD2-60	×	2	1	×	×	2	1	3	×	2	×	10
Dòng XD3												
XD3-16	1	1	1	×	10	×	1	3	×	2	×	6
XD3-24	1	1	1	×	10	1	1	3	×	2	×	10
XD3-32	1	1	1	×	10	1	1	3	×	2	×	10
XD3-48	1	1	1	×	10	2	1	3	×	2	×	10
XD3-60	1	1	1	×	10	2	1	3	×	2	×	10
Dòng XD5												
XD5-16	1	1	1	×	16	×	1	3	×	2	×	6

Model	Công USB	Công 232	Công 485	Công RJ45	Mô-đun mở rộng	BD mở rộng	ED mở rộng trái	Kênh HSC Pha đơn/AB		Xuất xung (T/RT)		Ngắt ngoài
								OC	Differential	normal	Differential	
XD5-24	1	1	1	×	16	1	1	3	×	2	×	10
XD5-32	1	1	1	×	16	1	1	3	×	2	×	10
XD5-42	1	1	1	×	16	1	1	3	×	2	×	10
XD5-48	1	1	1	×	16	2	1	3	×	2	×	10
XD5-60	1	1	1	×	16	2	1	3	×	2	×	10
XD5-24T4	1	1	1	×	16	1	1	4	×	4	×	10
XD5-24D2T2	1	1	1	×	16	1	1	2	2	2	2	10
XD5-32T4	1	1	1	×	16	1	1	4	×	4	×	10
XD5-48T4	1	1	1	×	16	2	1	4	×	4	×	10
XD5-48D4T4	1	1	1	×	16	2	1	4	4	4	4	10
XD5-48T6	1	1	1	×	16	2	1	6	×	6	×	10
XD5-60T4	1	1	1	×	16	2	1	4	×	4	×	10
XD5-60T6	1	1	1	×	16	2	1	6	×	6	×	10
XD5-60T10	1	1	1	×	16	2	1	10	×	10	×	10
Dòng XDM												
XDM-24T4	1	1	1	×	16	1	1	4	×	4	×	10
XDM-32T4	1	1	1	×	16	1	1	4	×	4	×	10
XDM-60T4	1	1	1	×	16	2	1	4	×	4	×	10
XDM-60T4L	1	1	1	×	16	2	1	4	×	4	×	10
XDM-60T10	1	1	1	×	16	2	1	10	×	10	×	10
Dòng XDC												
XDC-24	×	2	1	×	16	1	1	4	×	2	×	10
XDC-32	×	2	1	×	16	1	1	4	×	2	×	10
XDC-48	×	2	1	×	16	2	1	4	×	2	×	10
XDC-60	×	2	1	×	16	2	1	4	×	2	×	10
Dòng XD5E												
XD5E-24	×	1	1	2	16	1	1	3	×	2	×	10
XD5E-30	×	1	1	2	16	1	1	3	×	2	×	10
XD5E-30T4	×	1	1	2	16	1	1	4	×	4	×	10
XD5E-48	×	1	1	2	16	2	1	3	×	2	×	10
XD5E-60	×	1	1	2	16	2	1	3	×	2	×	10
XD5E-60T4	×	1	1	2	16	2	1	4	×	4	×	10
XD5E-60T6	×	1	1	2	16	2	1	6	×	6	×	10
XD5E-60T10	×	1	1	2	16	2	1	10	×	10	×	10
Dòng XDME												
XDME-30T4	×	1	1	2	16	1	1	4	×	4	×	10
XDME-60T4	×	1	1	2	16	2	1	4	×	4	×	10
XDME-60T10	×	1	1	2	16	2	1	10	×	10	×	10
Dòng XDH												
XDH-30A16	×	1	1	2	16	×	1	4	×	4	×	10
XDH-30A16L	×	1	1	2	16	×	1	4	×	4	×	10

Model	Cổng USB	Cổng 232	Cổng 485	Cổng RJ45	Mô-đun mở rộng	BD mở rộng	ED mở rộng trái	Kênh HSC Pha đơn/AB		Xuất xung (T/RT)		Ngắt ngoài
								OC	Differential	normal	Differential	
XDH-60A32	×	1	1	2	16	1	1	4	×	4	×	10
XDH-60T4	×	1	1	2	16	1	1	4	×	4	×	10
Dòng XL1												
XL1-16	×	2	1	×	×	×	×	×	×	×	×	6
XL1-16T-U	1	1	1	×	×	×	×	×	×	×	×	6
Dòng XL3												
XL3-16	1	1	1	×	10	×	1	3	×	2	×	6
XL3-32	1	1	1	×	10	×	1	3	×	2	×	10
Dòng XL5												
XL5-16	1	1	1	×	16	×	1	3	×	2	×	6
XL5-32	1	1	1	×	16	×	1	3	×	2	×	10
XL5-32T4	1	1	1	×	16	×	1	4	×	4	×	10
XL5-64T10	1	1	1	×	16	×	1	10	×	10	×	10
Dòng XL5E												
XL5E-16	×	1	1	2	16	×	1	3	×	2	×	6
XL5E-32	×	1	1	2	16	×	1	3	×	2	×	10
XL5E-32T4	×	1	1	2	16	×	1	4	×	4	×	10
XL5E-64T6	×	1	1	2	16	×	1	6	×	6	×	10
XL5E-64T10	×	1	1	2	16	×	1	10	×	10	×	10
Dòng XLME												
XLME-32T4	×	1	1	2	16	×	1	4	×	4	×	10
XLME-64T10	×	1	1	2	16	×	1	10	×	10	×	10
Dòng XLH												
XLH-24A16	×	1	1	2	16	×	1	4	×	4	×	10
XLH-24A16L	×	1	1	2	16	×	1	4	×	4	×	10

Lưu ý:

1. Dòng XL1-16T với phiên bản phần cứng thấp hơn H4 chỉ có một cổng RS232 (COM1).
- 2: Tất cả các model đều được trang bị chức năng đồng hồ tiêu chuẩn.

GAC CÔNG TY CỔ PHẦN
TỰ ĐỘNG HÓA TOÀN CẦU TỰ ĐỘNG HÓA TOÀN CẦU

Lô 17-F1 khu đô thị Đại Kim, phường Đại Kim, quận Hoàng
Mai, Hà Nội
Tel: 0967.837.815
Email: toancau@tudonghoatoancau.com
Website: www.tudonghoatoancau.com