N THẾ SAN (Biên dịch) - TS. NGUYỄN NGỌC PHƯƠNG (Hiệu đính)
ỌA CƠ KHÍ CHẾ TẠO MÁY
ỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

Hướng Dẫn THIẾT KẾ MẠCH & LẬP TRÌNH PLC

New EDITION

NHÀ XUẤT BẢN ĐÀ NẴNG

Biên dịch: **TRẦN THẾ SAN**
Hiệu đính: **TS. NGUYỄN NGỌC PHƯƠNG**
Khoa Cơ Khí Chế Tạo Máy
Trường Đại Học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh

HƯỚNG DẪN
THIẾT KẾ MẠCH & LẬP TRÌNH
PLC

NHÀ XUẤT BẢN ĐÀ NẴNG

Lời nói đầu

Bộ điều khiển lập trình (PLC — Programmable Logic Controller), bước phát triển mới trong kỹ thuật điều khiển, được sử dụng ngày càng rộng rãi. Các vấn đề cơ bản về lý thuyết mạch và điều khiển điện - điện tử là cơ sở để phát triển bộ điều khiển lập trình. Các học viên trường công nhân kỹ thuật, sinh viên cao đẳng và đại học chuyên ngành điện và cơ khí đã và đang được học về điều khiển lập trình, cả lý thuyết và thực hành, do các ứng dụng ngày càng rộng rãi của PLC trong thực tiễn. Cuốn sách này trình bày dưới dạng các bài tập lớn (gồm 24 bài) chia làm hai phần.

Phần I: Điều khiển động cơ bằng cơ điện tử. *Phần này có năm bài tập lớn, trình bày tóm tắt lý thuyết và thực hành các mạch điều khiển cơ bản.*

Phần II: Lập trình. *Phần này gồm 19 bài tập lớn.*

Tất cả các bài tập được sắp xếp theo thứ tự từ dễ đến khó, từ đơn giản đến phức tạp và có tính thực tiễn. Mỗi bài đều nêu rõ mục đích, nội dung, tài liệu đọc thêm, vật tư, và các bước thực hiện. Kèm theo mỗi bài tập là các thí nghiệm và câu hỏi. Các bài tập và thí nghiệm đều có thể được thực hiện với sự hướng dẫn của thầy cô giáo hoặc học viên tự thực hiện để nâng cao trình độ.

Cuối sách là phụ lục về các lệnh dùng trong lập trình logic và danh mục các linh kiện thông dụng.

Mục đích cuốn sách là phục vụ rộng rãi bạn đọc, từ các học viên, sinh viên để làm tài liệu học tập, đến các thầy cô giáo để tham khảo. Ngoài ra, mọi bạn đọc yêu thích điện điều khiển và lập trình logic đều có thể tìm được ở đây nhiều điều thú vị.

•

PHẦN I

ĐIỀU KHIỂN ĐỘNG CƠ BẰNG CƠ ĐIỆN TỬ

Trong phần này, các bài tập lớn sẽ giúp bạn làm quen với các linh kiện cơ điện tử và sơ đồ logic bậc thang. Các mẫu logic bậc thang được giới thiệu nhằm cung cấp ý tưởng cơ bản về nguyên lý vận hành các mạch điện được dùng trong ứng dụng thực tế. Ở đây đặc biệt nhấn mạnh việc sử dụng các điều khiển điện áp thấp để bảo đảm môi trường an toàn. Ngày nay, các bộ điều khiển lập trình đã thay thế hầu hết các relay điện tử và bộ thời chuẩn. Tuy nhiên, sự phân tích logic vẫn giữ nguyên giá trị, và kiến thức cơ bản về lĩnh vực này vẫn là cơ sở để hiểu các kỹ thuật lập trình.

NỘI DUNG. Sau khi hoàn tất phần này bạn sẽ có khả năng:

- Thiết kế và kết nối các sơ đồ bậc thang biểu thị quan hệ logic của các tín hiệu nhập và xuất.
- Xây dựng mạch điều khiển sử dụng nút nhấn và bộ khởi động của động cơ với điện áp toàn phần.
- Nghiên cứu các nguyên tắc và các quy định cơ bản của Tiêu Chuẩn Điện Hoa Kỳ (NEC) về mạch điều khiển động cơ.
- Áp dụng các công tắc chọn kiểu nhấn – xoay cho nhiều hệ thống logic.
- Phân tích mạch trễ thời gian – on và thời gian – off, áp dụng chúng trong mạch điều khiển công nghiệp.
- Hiểu công dụng của quá tải trong các bộ khởi động của động cơ ba pha.
- Xác lập các mạch điều khiển thuận và đảo chiều để tạo ra chuỗi thứ tự chuyên biệt.
- Chọn các bộ khởi động cuộn dây điện áp thấp IEC hợp lý.
- Nhận biết và nối kết các module nhập/xuất của bộ điều khiển lập trình.
- Lập trình các tham số của bộ truyền động AC điều chỉnh tần số.

Bài tập lớn 1. 0

CÁC MẠCH DỪNG – KHỞI ĐỘNG

MỤC ĐÍCH

Thiết kế và nối kết mạch nút nhấn dừng – khởi động dùng để
khiển bộ khởi động điện áp toàn phần.

NỘI DUNG

Sau khi hoàn tất bài tập này bạn sẽ có khả năng:

- Hiểu các bộ khởi động định mức NEMA và IEC điện áp t
phần.
- Hiểu các mạch dừng – khởi động.
- Nối kết các tiếp điểm relay quá tải và bao kín.
- Hiểu mục 430 của NEC về động cơ, mạch động cơ, và bộ
khiển.

Tài liệu đọc thêm: Động cơ điện, mạch bảo vệ, mạch điều khiển.

VẬT LIỆU

Nguồn điện 24V và 120V AC.

(2) Bộ khởi động IEC, với các cuộn 24V AC và các công tắc phụ
NO (thường đóng/thường mở).

(2) Nút nhấn dừng – khởi động.

(1) Nút nhấn khởi động.

(3) Đèn thí nghiệm 24V.

(1) Đèn thí nghiệm 120V.

(3) Relay với 2 công tắc NC và 1 công tắc NO.

IEC và NEMA. Các tổ chức chuyên ngành

IEC (International Electric – Technical Commission — Ủy ban Đi
– Kỹ thuật Quốc tế) thành lập năm 1906 nhằm nỗ lực tập trung hóa
tiêu chuẩn về trang thiết bị điện trong các nước công nghiệp. Cơ qu

bầu cử gồm các quốc gia thành viên, mỗi quốc gia có một phiếu bầu vào ủy ban.

IEC là tổ chức quốc tế. Các hoạt động của ủy ban theo truyền thống chủ yếu là đưa ra các khuyến nghị về các tham số thiết kế sản phẩm và các quy trình kiểm nghiệm. Các nhà sản xuất có thể kiểm nghiệm và công bố thông tin kỹ thuật cung cấp cơ sở cho khách hàng để so sánh theo định mức cho trước. Các tiêu chuẩn IEC phản ánh nhu cầu và triết lý kỹ thuật của các quốc gia thành viên. Trên cơ sở đó, các nhà sản xuất Mỹ thành lập Hiệp Hội Quốc Gia Các Nhà Sản Xuất Điện (National Electrical Manufacturers Association — NEMA) năm 1926.

NEMA đưa ra các tiêu chuẩn thiết kế và các thông số kỹ thuật kiểm nghiệm để thiết lập tiêu chuẩn hóa cho công nghiệp điện Bắc Mỹ.

Triết lý thiết kế

Nói chung, thường có quan niệm sai lầm cho rằng các tiêu chuẩn IEC và NEMA rất khác nhau do các thiết kế khác nhau. Trên thực tế, thiết kế phản ánh môi trường kinh tế và chính trị trong đó thiết kế hình thành. Cả hai loại thiết kế này đều nhằm đáp ứng các nhu cầu thị trường sử dụng các triết lý riêng.

Các bộ khởi động NEMA được thiết kế nhằm dễ dàng lựa chọn và thích hợp với khoảng ứng dụng rộng. Các tiêu chuẩn NEMA về cơ bản là *tiêu chuẩn lựa chọn*. Các nhà sản xuất thiết kế thiết bị của họ để lựa chọn phù hợp hệ thống các kích cỡ. Với các bộ khởi động NEMA, kích cỡ có cùng định mức đối với ứng dụng thích hợp bất kỳ, bất kể nhà sản xuất. Các thiết kế NEMA có mức hiệu suất cao trong khoảng ứng dụng rộng. Cách tiếp cận này làm cho sản phẩm có tính dự phòng cao, do vậy thường có kích thước lớn.

Các tiêu chuẩn IEC về cơ bản là *tiêu chuẩn vận hành* do các bộ khởi động phải đạt được các yêu cầu vận hành với định mức cho trước. Với các bộ khởi động IEC, sự lựa chọn chủ yếu dựa trên ứng dụng và điều kiện vận hành. Khi lựa chọn bộ khởi động IEC, bạn phải xét các yếu tố về tải, sử dụng, và tuổi thọ về điện của ứng dụng đó. Các cấp AC3 và AC4 được dùng phổ biến khi lựa chọn bộ khởi động IEC cho các động cơ công nghiệp tiêu chuẩn. Các nhà sản xuất kiểm nghiệm sản phẩm của họ theo các tiêu chuẩn này để công bố định mức hoặc tuổi thọ tiếp điểm. Điều khác biệt là các bộ khởi động NEMA được chế tạo để lựa chọn theo hệ thống các kích cỡ. Với các bộ khởi động IEC, sự lựa chọn dựa trên các cấp sử dụng và đường cong tuổi thọ – tải. Cả hai loại sản phẩm NEMA và IEC đều có thể được chọn để có sự vận hành tối ưu trong khoảng rộng

các ứng dụng. Cả hai đều cần được chọn một cách cẩn thận để bảo đảm sự vận hành hợp lý.

Bảo vệ quá tải

Bảo vệ quá tải là phần tích hợp trong bộ khởi động, bảo vệ động cơ khi dòng điện tăng đột ngột. Các relay quá tải thường đi kèm với bộ khởi động theo định mức IEC có cấp bảo vệ 10 và được thiết kế để vận hành trong 10 giây với 6 lần dòng điện tải toàn phần. Các relay quá tải này sử dụng phần tử nhiệt tích hợp, do đó không yêu cầu cung cấp phần tử nhiệt. Để tăng tính linh hoạt, các bộ quá tải IEC sử dụng sự hiệu chỉnh kiểu thanh trượt để điều chỉnh xác lập quá tải đối với các định mức dòng tải toàn phần khác nhau tùy theo định mức động cơ.

Các bộ khởi động theo định mức NEMA cần có thêm phần tử nhiệt cho relay quá tải. Các phần tử nhiệt này được quy định theo số cấp an toàn biểu thị thời gian tối đa (tính theo giây) relay hoạt động khi có dòng điện gấp 6 lần dòng định mức.

Relay quá tải cấp 10 hoạt động không quá 10 giây khi dòng điện gấp 6 lần định mức. Cấp 10 được đề nghị dùng cho các động cơ với thời gian khóa rotor ngắn, chẳng hạn các bơm nước ngầm.

Relay quá tải cấp 20 hoạt động không quá 20 giây nếu cường độ dòng điện đến 600% định mức. Loại này thường dùng cho các ứng dụng chung.

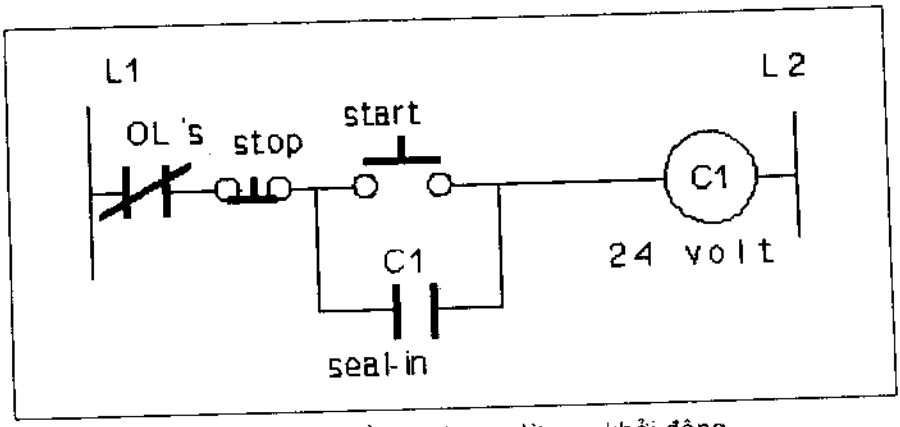
Relay quá tải cấp 30 có thời gian vận hành không quá 30 giây, thường dùng cho các động cơ truyền động tải quán tính cao đòi hỏi thời gian gia tốc.

Sơ đồ bậc thang

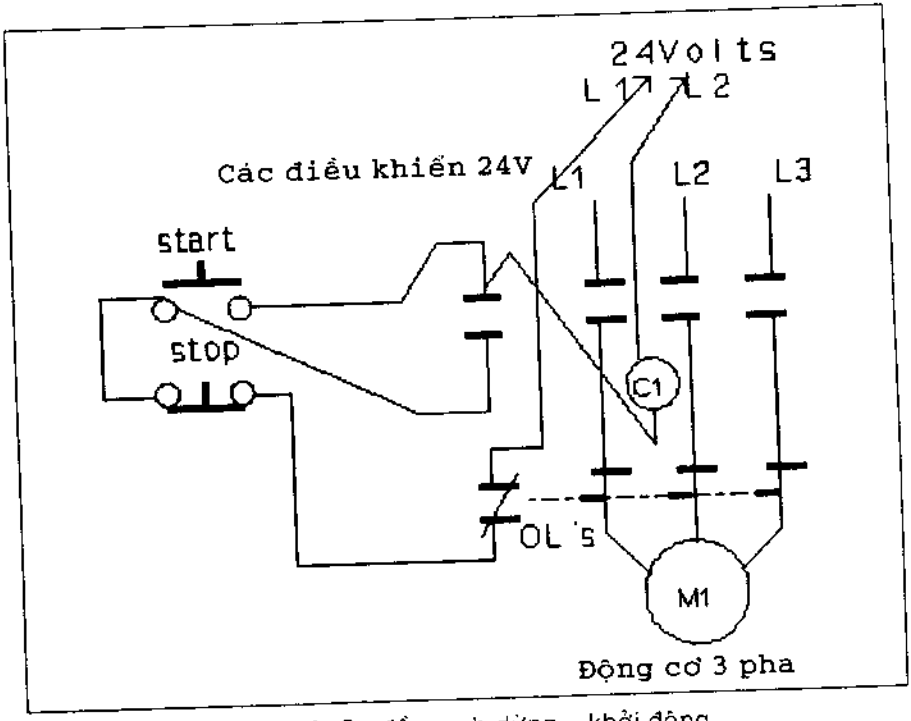
Phương pháp cơ bản để biểu thị ngôn ngữ điều khiển là sơ đồ bậc thang và sơ đồ mạch. Sơ đồ bậc thang chỉ biểu thị mạch cần thiết cho sự vận hành cơ bản của mạch điều khiển, không trình bày quan hệ chi tiết giữa các linh kiện trong mạch điều khiển, mục đích là nhấn mạnh sự vận hành điện của mạch điều khiển (Hình 1.1).

Sơ đồ bậc thang về cơ bản có hai phần: (1) nguồn công suất được biểu thị bằng hai vạch đậm song song và thẳng đứng, (2) dòng điện đi qua các bộ phận trong mạch, chẳng hạn nút nhấn, bộ giới hạn, tiếp điểm, cuộn dây, công tắc quá tải. Dòng điện được vẽ bằng đường ngang nối giữa các đường biểu thị nguồn.

Sơ đồ mạch có mọi linh kiện trong hệ thống, biểu thị quan hệ giữa



Hình 1.1. Sơ đồ bậc thang dừng – khởi động.



Hình 1.2. Sơ đồ mạch dừng – khởi động.

các linh kiện này. Các nối kết giữa bộ khởi động, các công tắc và nút nhấn đều được nêu chi tiết trên sơ đồ mạch. (Hình 1.2).

Nút nhấn dừng – khởi động

Hình 1.1 minh họa nút nhấn dừng - khởi động. Bạn hãy quan sát các phép toán logic của mạch dừng – khởi động. Nút dừng thường đóng được mắc nối tiếp với dây nguồn. Khi nhấn, nút dừng sẽ ngắt mạch làm cho cuộn dây không được cấp điện. Nút khởi động thường mở cũng mắc nối tiếp với dây nguồn (về logic là phép AND với nút dừng). Khi nhấn, nút khởi động nối mạch để cấp điện cho cuộn dây. Các bộ quá tải cũng mắc nối tiếp với dây nguồn. Điều này tạo ra bộ ba nhập phép toán logic AND với các nút nhấn dừng và khởi động. Các tiếp điểm làm kín mắc song song với nút khởi động (phép logic OR với nút khởi động). Khi cuộn dây được cấp nguồn, các tiếp điểm làm kín sẽ đóng để duy trì dòng điện trong mạch.

Bài tập lớn và các thí nghiệm liên quan sẽ minh họa các khái niệm logic và kỹ thuật có thể được dùng với các bài tập khác.

An toàn. An toàn luôn luôn là vấn đề hàng đầu. Sử dụng các thói quen làm việc an toàn không đòi hỏi kỹ thuật đặc biệt, chỉ yêu cầu sử dụng trang thiết bị đúng phương pháp và tuân thủ các quy định an toàn. Bạn cần nắm vững các quy định này trong phòng thí nghiệm.

Bạn hãy áp dụng các quy định an toàn dưới đây khi thực hiện các bài tập trong sách này.

1. Không chạm vào dây nóng. Ngắt điện nguồn trước khi thực hiện các thí nghiệm.
2. Sử dụng các bộ điều khiển điện áp thấp (không quá 30V) mỗi khi có thể.
3. Chỉ đóng mạch nguồn để thí nghiệm khi được phép của người có trách nhiệm.

CHƯƠNG TRÌNH LOGIC

1. Nút nhấn dừng tạm thời thường đóng. Khi nhấn, mạch bị ngắt, động cơ dừng.
2. Nút nhấn khởi động tạm thời thường mở. Khi nhấn, mạch sẽ đóng, động cơ khởi động.
3. Công tắc thường mở, đóng/mở mạch bộ khởi động.
4. Contactor động cơ điện áp toàn phần IEC với cuộn dây 24V, AC hoặc DC. Relay nhiệt (bảo vệ) quá tải được chọn với khoảng dòng điện tương ứng động cơ.

5. Nhấn nút khởi động (Start) sẽ khởi động và đóng mạch trong động cơ cho đến khi nhấn nút dừng (Stop) để dừng động cơ.

ÁC BƯỚC THỰC HIỆN

1. Nối mạch điều khiển theo mạch dừng – khởi động như trên các Hình 1.1 và 1.2.

   Chú ý: Trong bài tập này có thể không cần nối với động cơ, do mục đích chỉ là làm quen với các mạch điều khiển.

2. Quan sát nút nhấn thứ nhất (dừng, Stop). Đây là công tắc thường đóng và mắc nối tiếp với dây nguồn. Khi nhấn nút này, mạch bị ngắt nguồn.

3. Xem nút nhấn kế tiếp (khởi động, Start). Đây là công tắc thường mở và mắc nối tiếp với dây nguồn (AND với nút dừng). Khi nhấn, nút này sẽ nối mạch đến cuộn dây.

4. Kỹ thuật thứ hai được dùng trong bài tập này là sử dụng công tắc đóng của bộ khởi động (các công tắc nhớ của bộ khởi động). Các công tắc này được mắc song song với nút nhấn khởi động. Các công tắc đóng khi nhấn nút khởi động và cuộn dây được cấp nguồn. Công tắc vẫn đóng để duy trì dòng điện cho cuộn dây sau khi nhả nút khởi động.

5. Các công tắc quá tải là rất quan trọng trong mạch điều khiển. Các công tắc thường đóng này sẽ mở để bảo vệ động cơ khi có quá tải. Công tắc quá tải mắc nối tiếp với cuộn dây (AND với nút nhấn dừng và khởi động).

6. Khi bạn hoàn tất các bước nêu trên, hãy kiểm tra lại mạch. Sau khi đã chắc chắn, bạn có thể cho mạch hoạt động với bộ khởi động của động cơ, kiểm tra các mạch dừng và bảo vệ quá tải.

ÔN TẬP

Sau khi hoàn thành phần thứ nhất của bài tập này, bạn hãy ôn lại kiến thức đã học bằng cách thực hiện các thí nghiệm áp dụng. Mỗi thí nghiệm đều yêu cầu vẽ sơ đồ bậc thang trước khi nối mạch.

Thí nghiệm 1.1. HAI BỘ KHỞI ĐỘNG ĐỘNG CƠ

Mục đích

Thí nghiệm này yêu cầu bạn cải tiến Bài tập 1 như sau:

1. Vẽ sơ đồ bậc thang và mắc bộ khởi động thứ hai vào mạch của Bài tập 1 (bộ khởi động IEC, 24V AC).
2. Sử dụng mạch điều khiển nút nhấn dừng – khởi động của Bài tập 1 để điều khiển cả hai bộ khởi động.
3. Nếu một bộ khởi động mở công tắc đóng hoặc công tắc (bảo vệ) quá tải, cả hai bộ khởi động đều dừng.

Sơ đồ bậc thang

Câu hỏi

1. Hãy xác định chỉ số catalog của các contactor và relay quá tải (kiểu mở) IEC dùng cho động cơ AC, 3-pha, 1800 v/ph. Chọn các contactor với cuộn dây 24V. Sử dụng catalog của nhà sản xuất.

   A. 460V, 7.5 hp, cuộn DC, dòng tải toàn phần (FLC) 11A. _____

   B. 230V, 5hp, cuộn AC FLC 15.2 A. _____

   C. 575V, 25 hp, cuộn DC FLC 27 A. _____

   D. 230V, 3 hp, cuộn AC FLC 9.6 A. _____

   E. 46V, 15 hp, cuộn AC FLC 21 A. _____

2. Hãy xác định chỉ số catalog của bộ khởi động điện áp toàn phần IEC (vỏ kim loại) dùng cho động cơ AC, 3-pha. Chọn bộ khởi động với cuộn dây AC 24V và biến áp mạch điều khiển. Sử dụng catalog của nhà sản xuất.

   Số catalog

   A. 460V, 7.5 hp, FLC 11A. _____

   B. 460V, 15hp, FLC 21 A. _____

   C. 575V, 25 hp, FLC 27 A. _____

í nghiệm 1.2. HAI CẶP DỪNG – KHỞI ĐỘNG

đích

Thí nghiệm này yêu cầu bạn cải tiến Thí nghiệm 1.1 như sau:

1. Vẽ sơ đồ bậc thang, mắc thêm một nút dừng thường đóng và một nút khởi động thường mở. Điều này sẽ cho phép bạn dừng và khởi động hai bộ khởi động từ hai vị trí riêng rẽ.
2. Bổ sung đèn báo ở từng cặp nút nhấn. Đèn sáng khi bộ khởi động hoạt động, đèn có cùng điện áp với bộ khởi động.

đồ bậc thang

Thí nghiệm 1.3. KHỞI ĐỘNG LẦN LƯỢT

Mục đích

Thí nghiệm này yêu cầu bạn thiết kế và chỉnh sửa Thí nghiệm 1.2 như sau:

1. Vẽ sơ đồ bậc thang và nối lại từng bộ khởi động theo mạch nút nhấn dừng – khởi động riêng có đèn báo.
2. Bộ khởi động thứ hai chỉ có thể khởi động khi bộ khởi động thứ nhất hoạt động.
3. Bổ sung nút đẩy – kéo dừng khẩn cấp (E-Stop), khi đẩy sẽ ngắt mạch. Để khởi động lại phải kéo nút này (xác lập lại).
4. Mắc đèn báo, cho biết bộ khởi động thứ nhất không hoạt động (đèn sáng khi ngắt mạch bộ này), nối đèn với điện 120V.

Sơ đồ bậc thang

Câu hỏi

1. Tìm các bộ khởi động điện áp toàn phần (NEMA) và kích cỡ bộ cấp nhiệt cho các động cơ. Sử dụng catalog của nhà sản xuất.

   Bộ cấp nhiệt

Động cơ	Cỡ bộ khởi động	Cấp 10	20	30
A. 5hp, 460V, FLC 7.6 A.	_____	_____	___	___
B. 7.5hp, 230V, FLC 22 A.	_____	_____	___	___
C. 15hp, 230V, FLC 42 A.	_____	_____	___	___
D. 20hp, 575V, FLC 22 A.	_____	_____	___	___
E. 30hp, 460V, FLC 40 A.	_____	_____	___	___
F. 100hp, 575V, FLC 99 A.	_____	_____	___	___

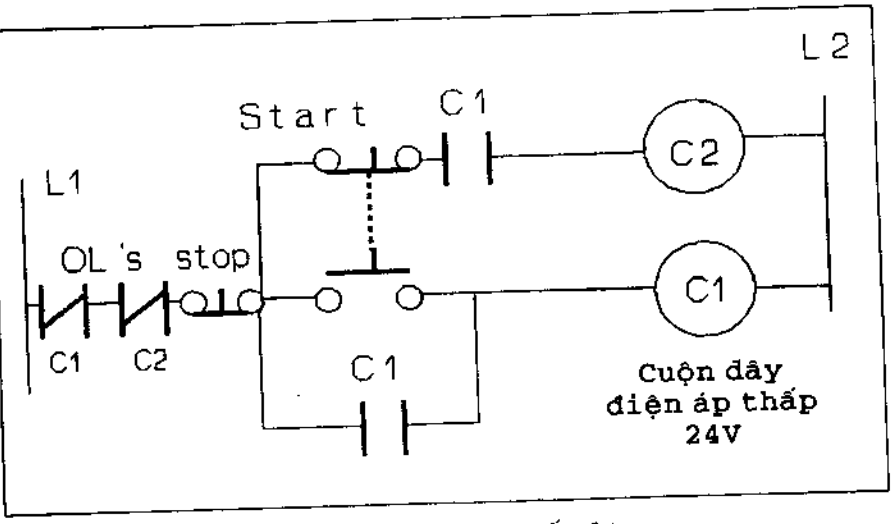
ní nghiệm 1.4. THỨ TỰ NÚT NHẤN ĐƠN

c đích

Bạn hãy nối sơ đồ mạch được nêu trên Hình 1.3:

1. Sử dụng một nút nhấn thường đóng/thường mở để khởi động hai bộ khởi động theo thứ tự (có khoảng trễ giữa hai sự khởi động của từng bộ). Nhấn nút khởi động (Start) sẽ cấp nguồn cho bộ khởi động #1, nhá nút này sẽ khởi động bộ khởi động #2.
2. Sử dụng hai bộ khởi động điện áp toàn phần điều khiển điện áp thấp.
3. Sử dụng một nút nhấn dừng để dừng các động cơ. Công tắc quá tải bất kỳ sẽ dừng cả hai động cơ.

ơ đồ bậc thang



Hình 1.3. Thứ tự nút nhấn đơn.

Câu hỏi

1. Bạn hãy chọn cụm nút nhấn dừng – khởi động/đèn báo tải tiêu chuẩn từ catalog điều khiển công nghiệp.

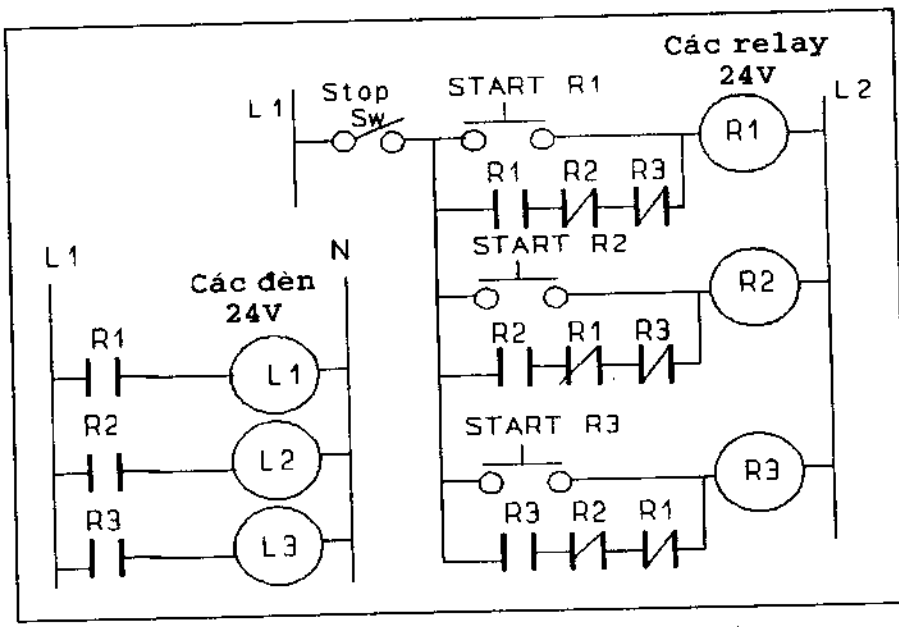
Thí nghiệm 1.5. CÁC ĐÈN KHÔNG THEO THỨ TỰ

Mục đích

Bạn hãy nối mạch được nêu trên Hình 1.4

1. Ba relay điện áp thấp với hai công tắc N/O (thường mở) và hai công tắc N/C (thường đóng).
2. Ba nút khởi động tạm, mỗi nút điều khiển một relay điện áp thấp. Các relay này điều khiển ngõ ra đèn. Chỉ một đèn được cấp nguồn khi nhấn một nút. Khi nhấn nhiều hơn một nút, các đèn tương ứng sẽ sáng.

Sơ đồ bậc thang



Hình 1.4. Các đèn điều khiển relay điện áp thấp.

Câu hỏi

1. Sử dụng catalog điều khiển công nghiệp của nhà sản xuất, bạn hãy chọn relay có thể dùng được để điều khiển các đèn trong thí nghiệm trên.

Bài tập lớn 2.0

CÁC MẠCH CHẠY - DỪNG THEO CHU KỲ

ỤC ĐÍCH

Thiết kế và nối các nạch nút nhấn dừng – khởi động và chạy sử dụng ộ chọn và các công tắc nhấn – xoay.

ỘI DUNG

Sau khi hoàn tất bài tập này bạn sẽ có khả năng:

- Thiết kế và nối các mạch dừng – khởi động chạy.
- Nghiên cứu và mắc các bộ nhấn – xoay (công tắc chọn nhấn – xoay).

TÀI LIỆU ĐỌC THÊM

Để biết chi tiết về các công tắc sẽ dùng trong bài tập này, bạn hãy xem Phụ lục B "Các bộ nhấn – xoay và các linh kiện khối tiếp điểm" ở cuối sách.

VẬT LIỆU

Nguồn điện 24V và 120V AC

(1) Bộ khởi động IEC với cuộn dây 24V AC.

(1) Bộ công tắc chọn nhấn – xoay ba vị trí (C-H 10250 T261-7)

(1) Khối tiếp điểm (10250-T1)

(1) Nút nhấn dừng – khởi động/chạy.

Chạy – dừng

Thuật ngữ chạy – dừng được hiểu là sự khởi động và dừng động cơ lặp lại nhiều lần theo các khoảng thời gian tương đối ngắn. Sự chạy – dừng được dùng để định vị vật liệu bằng cách dịch chuyển chúng theo các khoảng ngắn mỗi khi nhấn nút chạy – dừng. Khi nối mạch chạy dừng, các tiếp điểm phải được cô lập (ngắt) khỏi mạch. Có nhiều phương pháp để thực hiện điều này: (1) Công tắc chọn (2-vị trí) có thể được dùng để

mở/đóng (bằng tay) một phần của mạch. Điều này được gọi là bộ chọn chạy/dừng. (2) Nút nhấn riêng rẽ tạm cấp điện cho cuộn relay. Các tiếp điểm relay có thể được dùng để đóng mạch chạy. Nút nhấn chạy sẽ cấp điện cho bộ khởi động, không cần các tiếp điểm. Điều này được gọi là relay chạy/dừng. (3) Nút nhấn chọn nhấn – xoay cung cấp các chức năng phối hợp cho nút nhấn và công tắc chọn nêu trên. Điều này được gọi là chạy/dừng kiểu nhấn – xoay. Cả ba mạch đều được trình bày trong bài tập này.

Các bộ nhấn – xoay

Bộ nhấn – xoay (công tắc chọn nhấn – xoay) cung cấp hai và ba chức năng điều khiển có thể nhận được ở nơi thường yêu cầu một phần tử nút nhấn. Vành ngoài có thể xoay đến hai hoặc ba vị trí để đóng hoặc ngắt các mạch điều khiển. Các mạch này có thể đóng bằng cách nhấn một nút nhấn theo một trong ba vị trí.

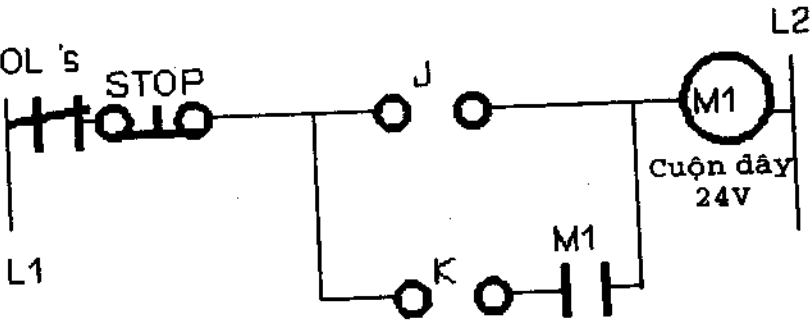
Các bước dưới đây có thể được sử dụng khi chọn nút nhấn – xoay:

1. Bước thứ nhất khi chọn nút là thiết kế mạch điện theo các phương pháp bạn thường dùng trong thiết kế mạch với các sơ đồ bậc thang tiêu chuẩn. Mạch này tương tự nút dừng – khởi động cơ bản (Hình 1.1). Sự khác biệt duy nhất trong bài tập này là các tiếp điểm cần ngắt khỏi mạch ở vị trí chạy. Khi thiết kế các mạch với công tắc nhấn – xoay, bạn dùng bảng chức năng để xác định sự vận hành của công tắc sẽ rất thuận tiện. Bạn hãy xác lập tính logic của mạch sử dụng các khoảng trống để biểu thị tiếp điểm hở và ký hiệu bằng chữ in, chẳng hạn J và K (Hình 2.1a).

2. Sau khi thiết kế mạch, bước kế tiếp là khai triển bảng chức năng (logic) cho mạch đó. Bảng này còn được gọi là bảng chân trị về các phép toán logic. Để lập bảng chân trị, bạn sắp xếp các chức năng vận hành của công tắc với ký hiệu X hoặc 1 biểu thị vị trí đóng (nhấn) công tắc (D) sử dụng O để biểu thị vị trí ngắt (mở) bình thường (N).

   Nút khởi động thường mở sẽ là O, X; ví trí mở (ngắt) là O và đóng là X. nút dừng thường đóng sẽ là X, O, các kết hợp hai nút này sẽ là tổ hợp X,X, O,O.

   Bạn cần nhớ mọi phép toán logic ở đây cần phải đồng nhất với từng vị trí của công tắc chọn nhấn – xoay. Một nút nhấn có thể có tối đa ba phép toán logic khả dĩ (Hình 2.1b). Dòng trên cùng là thường đóng hoặc thường mở (N) và nhấn (D) của từng vị trí.



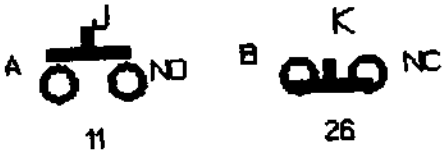
Công tắc nhấn - xoay 3 - vị trí

CAM	COMB	CONT	RUN L		OFF M		JOG R	
			N	D	N	D	N	D
7	11	J	0	1	0	0	0	1
7	26	K	1	1	0	0	0	0

Cách bố trí tiếp điểm

N= Bình thường

D= Nhấn



Hình 2.1. chạy – ngắt – dùng với công tắc xong ba – vị trí.

3. Bước kế tiếp là so sánh bảng chân trị với chuỗi thứ tự mạch của nhà sản xuất để chọn sự phối hợp tương ứng. Ví dụ, bảng trên

Hình 2.1b có thể so sánh với Bảng 2, Chọn khối Tiếp Điểm và Cam đối với công tắc ba vị trí trong Phụ lục B. Số tổ hợp 11 tương ứng dòng ngang thứ nhất trên bảng chân trị (OX – OO - XO). Số tổ hợp 26 tương ứng dòng thứ hai trên bảng này (XX – OO - OO). Bạn hãy ghi các số tổ hợp vào bảng chân trị. Bước kế tiếp là tìm mã cam chung có thể vận hành hai tổ hợp nêu trên.

4. Bạn hãy xem các cột bên phải (Bảng 2, Phụ lục B) của các tổ hợp bạn đã chọn để tìm ký hiệu công tắc. Nếu có ký hiệu công tắc, mã cam được ghi ở đều cột sẽ thực hiện phép logic của tổ hợp đó. Mã cam bạn chọn phải có khả năng thực hiện toàn bộ các tổ hợp trên bảng chân trị, đối với mạch của bạn mã số cam là 7. bước tiếp theo là chọn bộ vận hành đồng nhất với mã cam 7. Ký hiệu bộ vận hành này là C-N 102 50T261-7, đây là bộ vận hành 3 vị trí (xem bảng Mã Cam và Bộ Vận Hành, Phụ Lục B).

5. Đến đây bạn cần xác lập thứ tự các tiếp điểm cho bộ vận hành và mã cam. Để chọn các tiếp điểm, bạn hãy xem Bảng 2a, phụ lục B theo các tổ hợp đã chọn. Tổ hợp 11 cần có công tắc thường mở (NO), tổ hợp 26 cần dùng công tắc thường đóng (NC).

Các ký tự A và B biểu thị vị trí của khối tiếp điểm do hai mạch chiếm chỗ trong quan hệ với nút định vị của bộ vận hành. Nút định vị ở mặt trước của bộ vận hành. Để vận hành đúng, mọi tiếp điểm A phải thẳng hàng với nút này (Bảng Các Linh Kiện Khối Tiếp Điểm, Phụ Lục B). Bộ tiếp điểm dùng cho Bài tập này là khối công tắc 1025O-1, NC/NO. Bạn hãy lắp hoàn chỉnh bộ này và kiểm tra sự vận hành bằng điện kế, kiểm tra sự hoạt động của các tiếp điểm. Bộ công tắc phải vận hành theo đúng bảng chân trị.

Bài tập và các thí nghiệp sẽ minh họa các khái niệm logic và các kỹ thuật có thể được dùng với vài bài tập khác trong cuốn sách này.

CHƯƠNG TRÌNH LOGIC

1. Nút nhấn dừng thường đóng, khi nhấn nút này mạnh sẽ ngắt và động cơ dừng lại.

2. Công tắc chọn nhấn – xoay ba vị trí, khi nhấn vị trí bên trái, động cơ khởi động và tiếp tục chạy. Khi vị trí giữa là OFF, nhấn nút này sẽ không kích hoạt động cơ. Ở vị trí bên phải, động cơ sẽ chạy/dừng khi nhấn nút.

3. Công tắc thường mở được dùng để đóng/ngắt bộ khởi động của động cơ.

4. Bộ khởi động điện áp toàn phần với dây 24V và các công tắc (bảo vệ) quá tải mắc nối tiếp với cuộn dây.

CÁC BƯỚC THỰC HIỆN

1. Bạn hãy hoàn tất sơ đồ mạnh theo sơ đồ bậc thang (Hình 2-1) của cụm nút nhấn chạy – ngắt – dừng.
2. Nối mạch chạy - ngắt – dừng theo sơ đồ mạch (Hình 2-1).
3. Quan sát nút nhấn thứ nhất (dừng). Đây là công tắc thường đóng (NC) mắc nối tiếp với nguồn. Khi nhấn nút này, mạch sẽ ngắt (hở), nguồn sẽ không vào mạch.
4. Bạn hãy lắp công tắc nhấn – xoay ba – vị trí. Bạn cần có bộ vận hành 3-vị trí với mã cam 7. Lắp khối công tắc NO/NC (thường mở/ thường đóng) vào bộ vận hành.
5. Bộ chọn nhấn – xoay cần được mắc theo sơ đồ mạch. Công tắc A thường mở là phép logic AND với nút dừng. Công tắc B thường đóng được mức nối tiếp (AND) với công tắc bộ khởi động. Chức năng của nút này là đóng mạch cuộn dây (ON) ở vị trí bên trái, ngắt (OFF) ở vị trí giữa, và đóng (ON) ở vị trí bên phải.
6. Kỹ thuật thứ hai được dùng trong Bài tập là công tắc của bộ khởi động (các tiếp điểm nhớ của bộ khởi động), mắc song song với nút nhấn khởi động. Công tắc đóng khi nhấn nút khởi động và cuộn dây được cấp điện. Trong mạch này, công tắc B thường đóng của bộ nhấn – xoay mắc nối tiếp (AND) với công tắc của bộ khởi động. Công tắc B sẽ ngắt công tắc bộ khởi động ở vị trí chạy/dừng và vị trí OFF.
7. Điều quan trọng là công tắc A phải thẳng hàng với nút trên vành ngoài ở mặt trước của bộ vận hành. Nếu ngược lại, thứ tự vận hành của mạch sẽ đảo ngược.
8. Các công tắc (bảo vệ) quá tải là rất quan trọng trong mạch điều khiển. Công tắc thường đóng sẽ mở (ngắt mạch) để bảo vệ động cơ nếu có quá tải. Các công tắc này mắc nối tiếp với cuộn dây (AND với nút nhấn dừng và khởi động).
9. Sau khi hoàn tất các bước nêu trên, bạn hãy kiểm tra mạch, và cho mạch hoạt động chỉ với bộ khởi động của động cơ (chưa mắc động cơ vào mạch); kiểm tra quá tải và mạch dừng bộ khởi động.

ÔN TẬP

Sau khi hoàn tất phần này, bạn cần ôn lại và kiểm tra kiến thức đã học bằng cách thực hiện các bài thí nghiệm.

> **CHÚ Ý:** Nối động cơ vào mạch là chưa thực sự cần thiết, nội dung của bài tập là làm quen với các kỹ thuật mạch điều khiển. Để an toàn, bạn nên dùng các bộ điều khiển điện áp thấp (không quá 30V) mỗi khi có thể.

Thí nghiệm 2.1. CHẠY /DỪNG – NGẮT – TỰ ĐỘNG

Mục đích

Thiết kế và vẽ sơ đồ bậc thang với bảng chân trị theo các thông số sau:

Sử dụng công tắc nhấn – xoay ba – vị trí, nút dừng đẩy – kéo thường đóng, công tắc đơn cực dùng cho chế độ tự động, bộ khởi động với cuộc dây điện áp thấp. Sự vận hành công tắc ba – vị trí: Vị trí giữa, ngắt toàn bộ mạch; vị trí bên trái, nhấn công tắc sẽ làm cho bộ khởi động chạy/dừng; xoay công tắc sang phải, sẽ kích hoạt mạch tự động. Bộ khởi động sẽ vận hành công tắc tự động đơn cực.

Sơ đồ bậc thang và bảng chân trị

Câu hỏi

Bạn hãy thiết kế sơ đồ bậc thang với bảng chân trị của mạch hoạt động như sau:

- Vị trí bên trái: nhấn, bộ khởi động sẽ cho phép động cơ chạy/dừng #1
- Vị trí bên giữa: nhấn, bộ khởi động sẽ cho phép động cơ chạy/dừng #2
- Vị trí bên phải: nhấn, bộ khởi động sẽ cho phép động cơ chạy/dừng #3

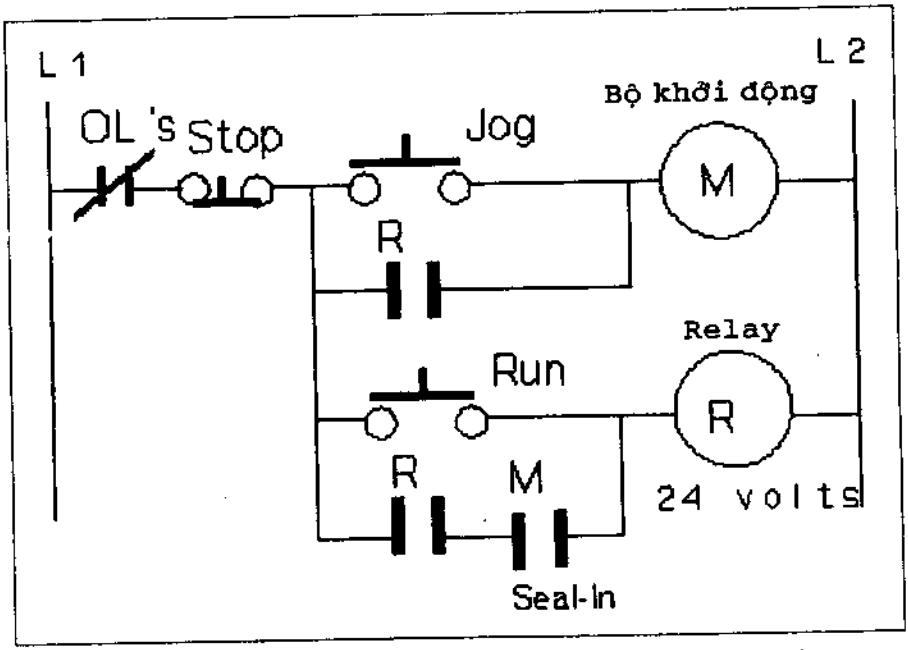
Thí nghiệm 2.2. RELAY CHẠY – DỪNG 1

Mục đích

Nối mạch relay chạy – dừng (Hình 2.2) sử dụng các thông số sau:

1. Dùng relay với hai công tắc thường mở cùng với công tắc bộ khởi động trong mạch chạy.
2. Dùng nút nhấn khởi động tiêu chuẩn thường mở để chạy. Nhấn nút này sẽ cấp điện cho relay và bộ khởi động.
3. Nút nhấn tiêu chuẩn thường mở thứ hai dùng cho chế độ chạy – dừng. Nhấn nút này động cơ chỉ hoạt động theo kiểu chạy – dừng, relay không được cấp nguồn.

Sơ đồ bậc thang



Hình 2.2. Relay chạy – dừng sử dụng nút nhấn tiêu chuẩn

Câu hỏi

Thiết kế mạch dừng, sử dụng công tắc nhấn – xoay hai – vị trí (dừng/dừng an toàn), xác định bảng chân trị và chỉ số catalog của công tắc này (xem Phụ lục B) vị trí bên trái, vành xoay hoạt động như nút nhấn dừng bình thường; vị trí bên phải sẽ ngắt mạch (không vận hành).

Thí nghiệm 2.3. RELAY CHẠY – DỪNG 2

Mục đích

Bạn hãy chỉ chỉnh sửa Thí nghiệm 2.2 theo cách thức dưới đây:

1. Nhấn nút chạy- dừng sẽ kích hoạt bộ khởi động chạy – dừng. Sự vận hành này có cùng chuỗi thứ tự như trong Thí nghiệm 2.2
2. Nhấn nút chạy sẽ không khởi động bộ khởi động chạy. Để chạy động cơ, cần nhấn cả nút chạy và nút chạy – dừng (Gợi ý: chỉ cần thay đổi một nối kết trên mạch relay chạy – dừng trong Thí nghiệm 2.2).

Sơ đồ bậc thang

Câu hỏi

Thiết kế mạch và bảng chân trị sử dụng công tắc nhấn – xoay hai – vị trí để điều khiển bộ khởi động (chạy – chạy/dừng). Mạch vận hành như sau:

1. Xoay sang vị trí bên trái và nhấn sẽ chạy bộ khởi động
2. Xoay sang vị trí bên phải và nhấn sẽ chạy/dừng bộ khởi động
3. Động cơ phải dừng để chuyển sang chế độ chạy/dừng.
4. Vẽ sơ đồ bậc thang và lập bảng chân trị.

Thí nghiệm 2.4. RELAY CHẠY – DỪNG 3

Mục đích

Thiết kế và vẽ sơ đồ bậc thang của mạch sử dụng công tắc nhấn – xoay ba – vị trí để điều khiển hai bộ khởi động như sau:

1. Vị trí bên trái và nhấn sẽ chạy/dừng bộ khởi động #1
2. Vị trí bên giữa và nhấn sẽ chạy/dừng bộ khởi động #2
3. Vị trí bên phải và nhấn sẽ chạy đều cả hai bộ khởi động #1 và #2. các động cơ phải dừng để chuyển sang chế độ chạy/ dừng. Lập bản chân trị.

Sơ đồ bậc thang

Câu hỏi

Thiết kế sơ đồ bậc thang và lập bảng chân trị cho mạch sử dụng công tắc chọn ba – vị trí. (Đây không phải là công tắc chọn nhấn - xoay). Bạn hãy dùng catalog điều khiển công nghiệp của nhà sản xuất (Phụ lục B) để chọn công tắc chọn ba – vị trí. Bạn hãy chọn núm tiêu chuẩn với lò xo trả về từ trái và phải, vận hành theo thứ tự như sau:

Trái = chạy, solenoid #1; giữa = Off; phải = chạy, solenoid # 3. Chỉ một solenoid hoạt động vào thời điểm bất kỳ.

Bài tập 3.0

MẠCH THUẬN/ĐẢO

MỤC ĐÍCH

Thiết kế và mắc mạch nút nhấn thuận – ngắt - đảo sử dụng công tắc chọn nhấn – xoay.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Mắc các bộ khởi động đảo chiều với điện áp toàn phần.
- Hiểu và biết cách mắc các mạch thuận/đảo chiều.
- Mắc các công tắc xoay ba – vị trí.

TÀI LIỆU ĐỌC THÊM

Để biết chi tiết các công tắc được dùng trong Bài tập này, bạn hãy đọc Phục lục B: Các bộ nhấn – xoay và các linh kiện khối tiếp điểm.

VẬT LIỆU

Điện nguồn 24V và 120V AC.

(1) Bộ khởi động thuận/đảo (IEC, cuộn dây 24V).

(1) Nút nhấn dừng - thuận/đảo.

(1) Công tắc nhấn – xoay ba – vị trí (C–H 10250 T261-8).

(2) Khối tiếp điểm (C–H 10250-T1).

(1) Khối tiếp điểm (C–H 10250-T3).

Bộ khởi động đảo chiều

Nhiều ứng dụng điều khiển đòi hỏi động cơ ba pha có khả năng chạy theo chiều thuận và đảo chiều. Để thực hiện điều này, bộ khởi động điện áp toàn phần là thiết bị dùng cho cả chiều thuận và chiều ngược (Bộ khởi động thuận - nghịch). Các bộ khởi động này còn có thể được dùng trong

ứng dụng đứng/ngắt mạch. Bộ khởi động thuận – nghịch có chế độ khóa liên động cả cơ và điện để duy trì trạng thái đóng một cánh đồng thời.

Do hầu hết các bộ khởi động đảo chiều từ tính đều có bảo vệ khóa liên động cơ – điện, nhưng một số mạch điều khiển còn có hệ thống an toàn thứ cấp để bảo đảm khóa liên động về điện. Hình 3.1 minh họa mạch sử dụng bộ nhấn – xoay 3 – vị trí. Mạch này dùng bộ nhấn – xoay 3 – vị trí để điều khiển: (1) vị trí bên trái và nhấn nút nhấn – xoay, động cơ chạy theo chiều thuận. (2) vị trí giữa, động cơ bị ngắt (OFF). Điều này bảo đảm khóa liên động điện giữa chiều thuận và ngược. (3) xoay sang vị trí phải và nhấn nút xoay, động cơ sẽ đảo chiều.

CHƯƠNG TRÌNH LOGIC

1. Nút nhấn dừng thường đóng. Khi nhấn sẽ ngắt mạch, động cơ dừng.
2. Nút nhấn xoay 3 – vị trí.
   Vị trí trái = chiều thuận
   Vị trí giữa = OFF
   Vị trí phải = chiều ngược.
3. Các bộ khởi động đảo chiều điện áp toàn phần: (1) các cuộn 24V. (2) Các tiếp điểm phụ thường đóng. (3) Các tiếp điểm thường mở. (4) bảo vệ quá tải.

CÁC BƯỚC THỰC HIỆN

1. Mắc mạch bộ khởi động, nhận biết các linh kiện và công tắc theo sơ đồ bậc thang (Hình 3.1). Mạch có công tắc nhấn – xoay 3 – vị trí dùng cho mạch thuận – ngắt – đảo.
2. Lắp bộ vận hành và các công tắc trong mạch. Bạn cần dùng bộ vận hành mã số cam 8, lắp ba khối tiếp điểm NO/NC (thường mở/thường đóng) vào bộ vận hành. Các tiếp điểm A của cả ba khối tiếp điểm phải thẳng hàng với núm xoay ở mặt trước của bộ vận hành. Trước khi nối mạch, bạn cần kiểm tra các linh kiện. Sử dụng đồng hồ (điện kế) vạn năng, bạn hãy kiểm lại thứ tự tiếp điểm của bộ công tắc theo bảng chân trị trên Hình 3.1
3. Quan sát nút nhấn thứ nhất (dừng). Đây là loại thường đóng, phải mắc nối tiếp với dây nguồn. Khi nhấn nút này, mạch sẽ bị ngắt khỏi điện nguồn.
4. Công tắc chọn nhấn – xoay thực hiện mọi thao tác trong mạch.

Xoay sang trái và nhấn sẽ cấp nguồn cho bộ khởi động thuận. Vị trí giữa là OFF. Xoay sang phải và nhấn sẽ cấp cho bộ khởi động đảo chiều.

5. Công tắc đóng mạch cho cả hai bộ khởi động thuận và đảo được mắc hoàn toàn như nút khởi động. Các công tắc này luôn luôn mắc song song với nút nhấn khởi động. Công tắc này đóng khi nút khởi động được nhấn và cấp nguồn cho cuộn dây, công tắc vẫn đóng để duy trì dòng điện đi qua cuộn dây khi nhả nút khởi động.

6. Bộ bảo vệ quá tải là rất quan trọng trong mạch điều khiển. Đây là các công tắc thường đóng, sẽ ngắt mạch để bảo vệ động cơ khi có quá tải. Bộ bảo vệ quá tải được mắc nối tiếp với cuộn dây (phép logic AND với nút nhấn dừng và khởi động).

7. Trên các mạch thuận/đảo, bộ các tiếp điểm phụ thường đóng được dùng để khóa liên động về điện cho các bộ khởi động.

8. Sau khi hoàn tất các bước nêu trên, bạn hãy kiểm tra mạch điện; chỉ cho mạch chạy với bộ khởi động, kiểm tra quá tải và mạch dừng. Nếu mạch điều khiển vận hành chuẩn xác, bạn có thể nối mạch với động cơ tương ứng. Bài tập và các thí nghiệm này không bắt buộc phải nối với động cơ.

ÔN TẬP

Sau khi hoàn tất phần này cùng với các kiểm nghiệm, bạn hãy chuyển sang các bài thí nghiệm. Các thí nghiệm yêu cầu thiết kế và vẽ sơ đồ bậc thang trước khi nối mạch.

Thí nghiệm 3.1. MẠCH THUẬN/ ĐẢO TIÊU CHUẨN

Mục đích

Mắc mạch thuận/đảo với các đặc tính như sau:

1. Nút nhấn dừng (STOP) thường đóng, hai nút nhấn khởi động (START) thường mở, một bộ khởi động thuận/đảo.
2. Một nút khởi động thuận và một nút khởi động đảo.
3. Nhấn nút thuận hoặc nút thảo, động cơ khởi động tương ứng theo chiều thuận hoặc đảo. Để khởi động theo chiều ngược lại khi trong chế độ chạy, động cơ phải dừng lại trước.

Sơ đồ bậc thang

Câu hỏi

1. Công dụng của khóa liên động trong bộ khởi động thuận/đảo?
2. Hãy xác định chỉ số sản xuất của các bộ khởi động công tắc IEC thuận/đảo và các relay quá tải dùng cho động cơ 3 – pha. Chọn các bộ công tắc với các cuộn dây 24V AC (kiểu kín).

   A. 460V, 5hp, dòng tải toàn phần (FLC) 7.6A. _____

   _____

   B. 460V, 10hp, FLC 14A. _____

   _____

   C. 460V, 15hp, FLC 21A. _____

   _____

   D. 575V, 30hp, FLC 32A. _____

   _____

Thí nghiệm 3.2.
DỪNG ĐẢO CHIỀU – CHẠY – DỪNG ĐẢO CHIỀU

Mục đích

Thiết kế sơ đồ bậc thang và bảng chân trị để điều khiển các bộ khởi động thuận/đảo #1 và #2 với các đặc tính như sau:

1. Một công tắc chọn nhấn – xoay 3 – vị trí, một nút nhấn dừng chính, hai bộ khởi động thuận/đảo.
2. Để chuyển từ chạy sang chạy/dừng, mạch phải dừng trước khi chuyển chế độ. Một trong hai bộ khởi động bị quá tải sẽ dừng cả hai động cơ.
3. Chọn công tắc xoay thích hợp để thực hiện sự vận hành:
   Vị trí trái = nhấn, chạy/dừng cho bộ khởi động ngược chiều #1
   Vị trí giữa = nhấn, chạy cho cả hai bộ khởi động thuận #1 và #2
   Vị trí phải = nhấn, chạy/dừng cho bộ khởi động ngược chiều #2

Sơ đồ bậc thang

Câu hỏi

1. Từ catalog của nhà sản xuất, hãy chọn các bộ khởi động thuận/đảo điện áp toàn phần (NEMA 1) và kích cỡ relay nhiệt cho các động cơ, biết nhiệt độ môi trường bằng nhiệt ở bộ điều khiển và động cơ.

	Kích cỡ	Cấp 10	Cấp 20
A. 5 hp, 230 V, FLC 15.2 A	_____	_____	_____
B. 7.5 hp, 460V, FLC 11 A	_____	_____	_____
C. 15 hp, 460V, FLC 21 A	_____	_____	_____
D. 20 hp, 460V, FLC 27 A	_____	_____	_____
E. 30 hp, 575V, FLC 32 A	_____	_____	_____

Thí nghiệm 3.3. BỐN BỘ KHỞI ĐỘNG CHẠY THUẬN

Mục đích

Thiết kế bảng chân trị và vẽ sơ đồ bậc thang của mạch với các đặc tính sau:

1. Sử dụng công tắc nhấn – xoay 3 – vị trí. Chọn công tắc từ catalog của nhà sản xuất. Một nút nhấn dừng chính, ba bộ khởi động vận hành động cơ #1, #2, #3 với các tiếp điểm phụ NC (thường đóng) và một bộ khởi động #4 dùng cho động cơ bơm chất bôi trơn.

2. Chọn công tắc nhấn – xoay 3 – vị trí, vận hành như sau:
   Vị trí trái = nhấn, khởi động #4 và #3
   Vị trí giữa= nhấn, khởi động #4 và #2
   Vị trí phải = nhấn, khởi động #4 và #1
   Khóa liên động các vị trí để mỗi lần chỉ vận hành một vị trí.

Sơ đồ bậc thang

Câu hỏi

1. Từ catalog của nhà sản xuất, chọn tổ hợp các bộ khởi động đảo chiều (NEMA4) và kích cỡ relay nhiệt cho các động cơ:

	Kích cỡ	Số catalog	Cấp quá tải 20
A. 1hp, 230V, FLC 3.6A	_____	_____	_____
B. 3hp, 230V, FLC 9.6A	_____	_____	_____
C. 5hp, 230V, FLC 15.2A	_____	_____	_____
D. 10hp, 230V, FLC 28A.	_____	_____	_____

Bài tập lớn 4.0

CÁC MẠCH THỜI CHUẨN

MỤC ĐÍCH

Nối hai bộ thời chuẩn trễ với mạch thuận/đảo

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Đọc và diễn dịch sơ đồ bậc thang có các bộ thời chuẩn
- Thiết kế và mắc các mạch logic thời gian trễ.
- Chọn các bộ thời điểm vận hành trễ và ngắt trễ.
- Hiểu và mắc các mạch module I/O (nhập/xuất) PLC (điều khiển logic lập trình)

VẬT LIỆU

Nguồn điện 24V và 120V.

(1) Bộ khởi động thuận/đảo.

(1) Nút nhấn dừng - thuận/đảo.

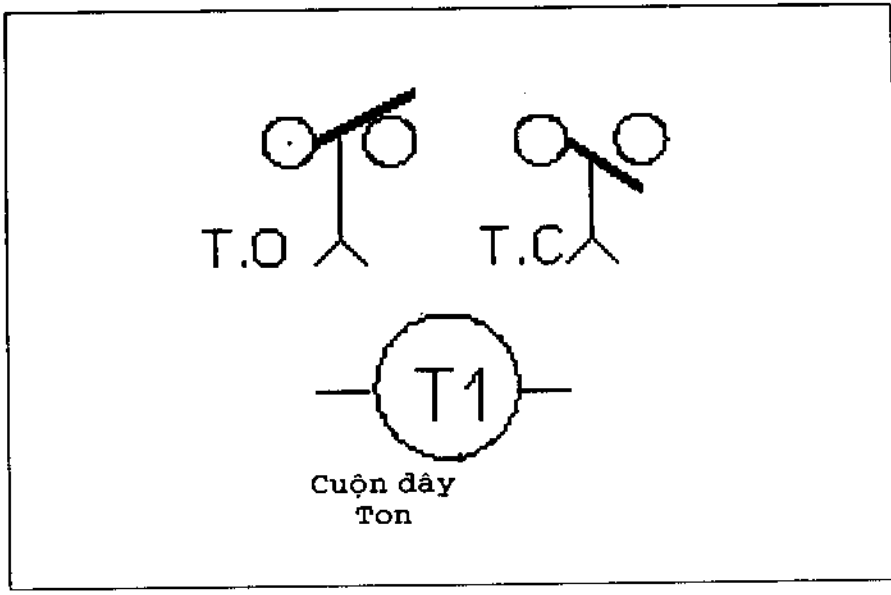
(2) Bộ thời chuẩn ngắt trễ.

(2) Bộ thời chuẩn đóng (vận hành) trễ.

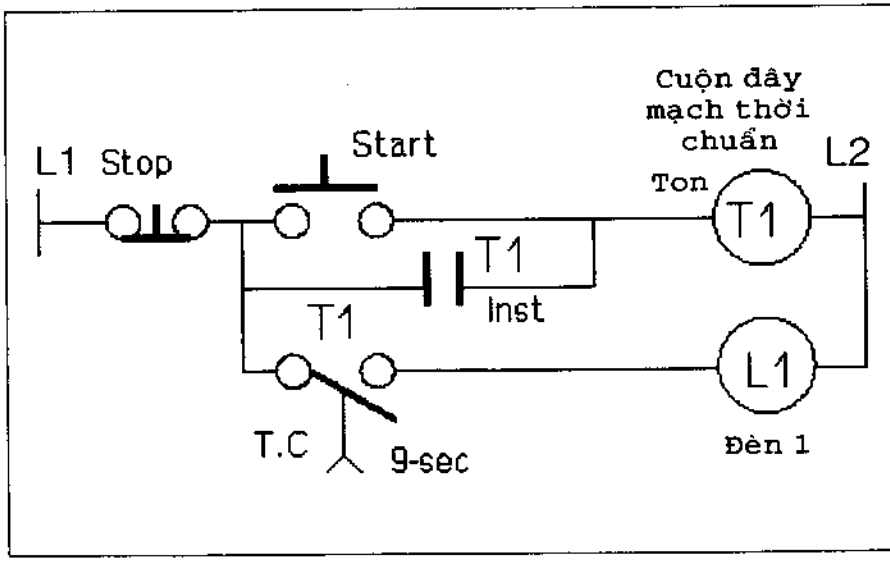
Bộ thời chuẩn

Các relay thời chuẩn được dùng để cung cấp sự trễ thời gian theo hai phương pháp. Thứ nhất, thời chuẩn đóng, tạo sự trễ thời gian sau khi cuộc dây được cấp điện; thứ hai, thời chuẩn ngắt, tạo sự trễ thời gian sau khi cuộn dây bị ngắt điện.

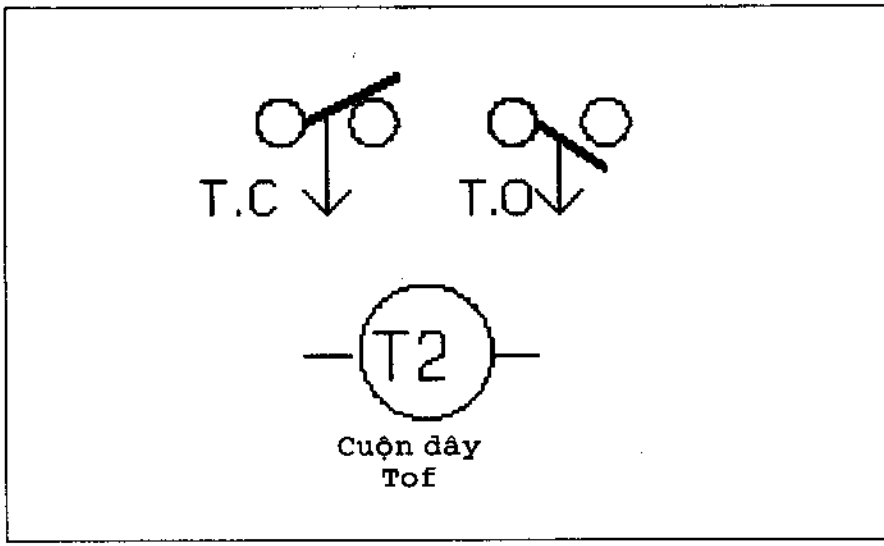
Các ký hiệu thời chuẩn đóng được nêu trên Hình 4.1a. Khi cấp điện cho cuộn dây TR, khoảng thời gian được xác định trước sẽ trôi qua trước khi các tiếp điểm thay đổi trạng thái. Khi ngắt điện cuộn dây này, các tiếp điểm trở về vị trí ban đầu một cách tức thời. Hình 4.1b minh họa mạch thời chuẩn đóng (TON). Khi nút nhấn khởi động (START) vận hành, cuộn T1 đóng, thanh ngang đóng, cho phép bắt đầu hoạt động thời chuẩn.



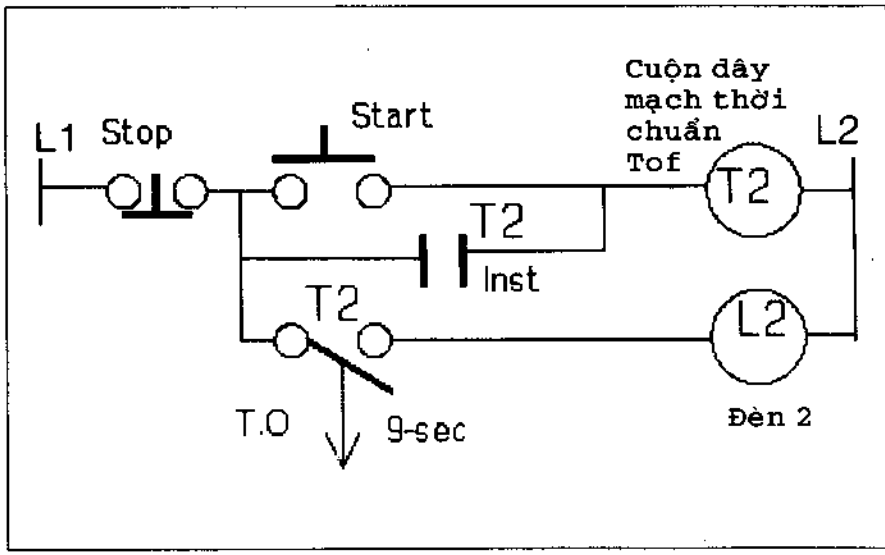
Hình 4.1a. Các ký hiệu thời chuẩn đóng (TON)



Hình 4.1b. Mạch trễ thời chuẩn đóng.



Hình 4.2a. Các ký hiệu thời chuẩn ngắt (TOF)



Hình 4.2b. Mạch trễ thời chuẩn ngắt

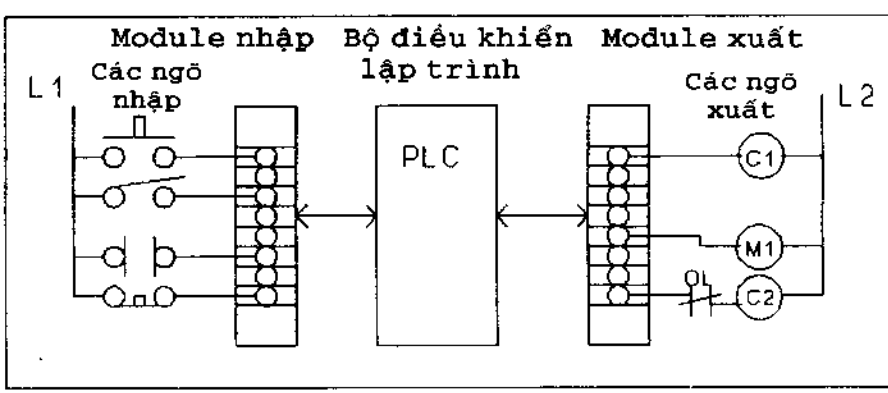
Sau khoảng thời gian xác định (ở đây là 9 giây), công tắc thời chuẩn NO (thường mở) sẽ đóng để vận hành đèn. Khi nút nhấn dừng (Stop) mở, thanh ngang ngắt, các tiếp điểm thời chuẩn trở về vị trí ban đầu (bình thường) hầu như tức thời.

Các ký hiệu thời chuẩn ngắt được nêu trên Hình 4.2a. Khi cấp điện cho cuộn dây, các tiếp điểm lập tức thay đổi trạng thái. Các tiếp điểm thời chuẩn duy trì trạng thái này trong thời gian cuộn dây có điện. Khi ngắt điện cuộn dây, các tiếp điểm trở về trạng thái ban đầu (bình thường) sau khoảng thời gian được xác định trước.

Mạch thời chuẩn ngắt (TOF) được minh họa trên Hình 4.2b. Khi nút nhấn khởi động (START) vận hành, cuộn T2 đóng và thanh ngang đóng, xảy ra hoạt động tức thời. Điều này làm đóng các tiếp điểm thời chuẩn và làm cho đèn sáng. Khi nút nhấn dừng (STOP) mở, thanh ngang mở, bắt đầu hoạt động thời chuẩn. Sau khoảng thời gian xác định (ví dụ 9 giây), các tiếp điểm thời chuẩn NO (thường mở) sẽ mở để ngắt mạch (tắt đèn).

Các module nhập/xuất (I/O) PLC (điều khiển logic lập trình)

Bộ điều khiển logic lập trình (PLC) là mạch điều khiển logic bán dẫn dùng trong công nghiệp. Các phép toán logic được xác định bằng chương trình do người dùng nhập vào, chuyên biệt sự vận hành của các thiết bị xuất đáp ứng thiết bị nhập. Do chương trình được lưu trong bộ nhớ đọc và ghi, sự thay đổi trong quá trình điều khiển được thực hiện bằng cách lập trình lại. Bộ điều khiển liên tục giám sát trạng thái các thiết bị được mắc với ngõ vào. Dựa trên chương trình của người dùng, bộ PLC điều khiển các thiết bị mắc ở ngõ ra (Hình 4.3).



Hình 4.3. Các nối kết module I/O

Nói chung, có hai kiểu module I/O

1. Module nhập, cảm biến các mức điện áp của thiết bị nhập và cung cấp cho bộ xử lý các thông tin trạng thái mức logic của thiết bị nhập. Các thiết bị và linh kiện nhập thường là nút nhấn, công tắc chọn, bộ giới hạn điện áp, các cảm biến, và các bộ chuyển mạch.

2. Module xuất, điều khiển các thiết bị ở ngõ ra, dựa trên các lệnh mức logic từ bộ xử lý. Thiết bị ngõ ra có thể là bộ khởi động, solenoid, màn hiểu thị, đèn, bộ cảnh báo,...

CHƯƠNG TRÌNH LOGIC

1. Nút nhấn dừng (STOP) thường đóng (NC), khi nhấn, mạch sẽ ngắt (False) làm dừng động cơ.

2. Hai nút nhấn khởi động (START) thường mở (NO), một dùng cho chiều thuận, và một dùng cho chiều ngược.

3. Bộ khởi động đảo chiều điện áp toàn phần, gồm (1) cuộn 24V. (2) Các tiếp điểm phụ thường đóng, dùng cho khóa liên động. (3) Các tiếp điểm thường mở (4) Relay và tiếp điểm bảo vệ quá tải.

4. Hai relay thời chuẩn (FOF), dùng cho bộ khởi động thuận và bộ khởi động đảo chiều. Đây là hai khối riêng rẽ lắp trên từng bộ khởi động.

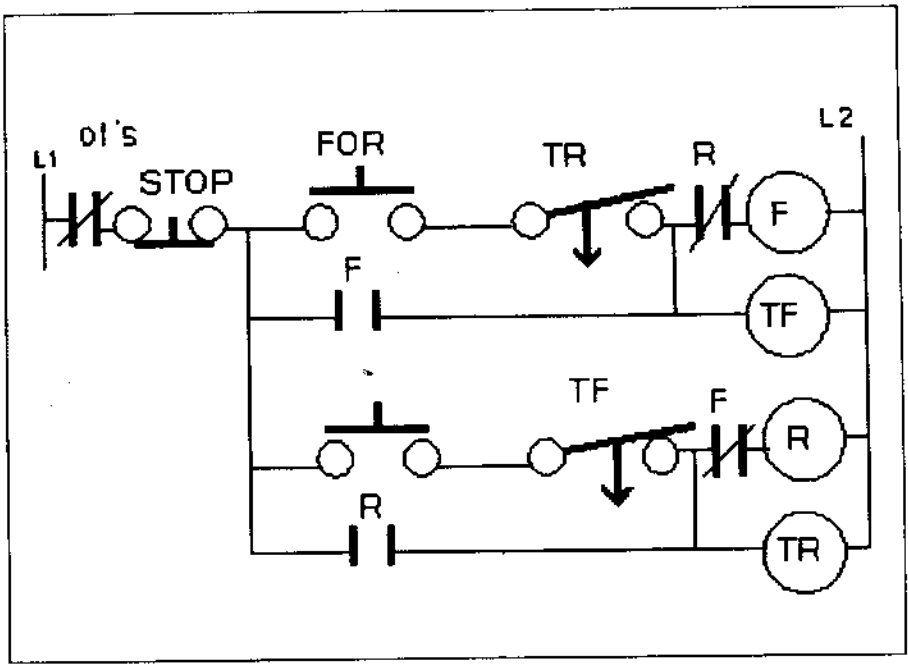
5. Mạch này vận hành theo mạch thuận/đảo tiêu chuẩn (Bài tập 3, Thí nghiệm 3.1), có thêm hai bộ thời chuẩn ngắt, ngăn chặn động cơ khởi động theo chiều ngược lại quá nhanh sau khi dừng.

CÁC BƯỚC THỰC HIỆN

1. Mắc mạch thuận/đảo theo sơ đồ bậc thang (Hình 4.4)

2. Mạch này có hai bộ thời chuẩn ngắt. Một cuộn dây thời chuẩn mắc song song với cuộn khởi động thuận, cuộn kia mắc song song với bộ khởi động đảo. Mỗi bộ thời chuẩn đều có tiếp điểm thường đóng (NC) để khóa liên động các bộ khởi động thuận và đảo.

3. Quan sát các tiếp điểm của bộ thời chuẩn, đây là loại thường đóng (NC), đóng theo thời gian. Khi cuộn thời chuẩn thuận (TF) được cấp điện, các tiếp điểm (TF) tương ứng mắc nối tiếp với cuộn đảo sẽ mở và đóng thời gian. Điều này làm làm cho cuộn đảo không được cấp điện với thời gian xác định sau khi cuộn thuận bị



Hình 4.4. Mạch thời chuẩn ngắt thuận/đảo

ngắt điện nguồn. Đối với cuộn đảo, tình huống là ngược lại. Khi cuộn đảo được cấp điện, cuộn thuận bị khóa liên động, không khởi động trong khoảng thời gian xác định.

4. Sau khi hoàn tất các bước nêu trên, bạn hãy kiểm tra lại chương trình, chỉ chạy chương trình với bộ khởi động, kiểm tra quá tải và mạch dừng. Nếu hoạt động tốt, bạn có thể thử nối với động cơ tương ứng.

ÔN TẬP

Sau khi hoàn tất phần đầu của Bài tập 4.0, bạn có thể chuyển sang các bài thí nghiệm để củng cố và nâng cao kiến thức. Bạn cần vẽ sơ đồ bậc thang cho từng thí nghiệm trước khi nối mạch.

Thí nghiệm 4.1.
KHỞI ĐỘNG THEO THỨ TỰ THỜI CHUẨN 1

Mục đích

Thiết kế và nối mạch điều khiển khởi động lần lượt ba bộ khởi động. Mạch vận hành như sau:

1. Sử dụng nút nhấn START để sử dụng chuẩn thứ tự. Nút STOP đẩy - kéo để dừng khẩn cấp và dừng tạm thời khi dừng vận hành tiêu chuẩn. Hai bộ thời chuẩn đóng (TON) và ba bộ khởi động với cuộn dây điện áp thấp 24 VAC.
2. Bộ khởi động #1 sẽ khởi động bằng cách nhấn nút START. Sau 10 giây, bộ khởi động #2 sẽ khởi động với một bộ thời chuẩn. Cuối cùng, sau 10 giây tiếp theo, bộ khởi động #3 sẽ hoạt động với bộ thời chuẩn thứ hai. Nhấn nút nhấn dừng bất kỳ đều dừng cả ba bộ khởi động.

Sơ đồ bậc thang

Câu hỏi

1. Sử dụng catalog công nghiệp, chọn hai bộ thời chuẩn đóng đáp ứng các yêu cầu của thí nghiệm.
2. Sử dụng catalog công nghiệp, chọn các contactor và relay quá tải IEC. Dùng cho thí nghiệm nêu trên, sử dụng các bộ thời chuẩn kèm theo contactor (động cơ 5hp, 460V, PLC 7.6A), xác định kiểu loại và số catalog.

Thí nghiệm 4.2.
KHỞI ĐỘNG THEO THỨ TỰ THỜI CHUẨN 2

Mục đích

Thiết kế sơ đồ bậc thang và cải tiến mạch điện trong Thí nghiệm 4.
để có thêm các tính năng dưới đây:

1. Giữ nguyên thứ tự khởi động như trong Thí nghiệm 4.1.
2. Thay đổi thứ tự dừng, nhấn nút dừng (STOP) tiêu chuẩn sẽ dừn
ngay hai bộ khởi động #2 và #3, bộ khởi động #1 còn chạy thêi
5 giây (TOF) trước khi dừng.
3. Nhấn nút E-stop sẽ ngắt mạch (OFF) toàn bộ mạch sử dụng ha
bộ thời chuẩn đóng (TON) và hai bộ thời chuẩn ngắt (TOF).

Sơ đồ bậc thang

Câu hỏi

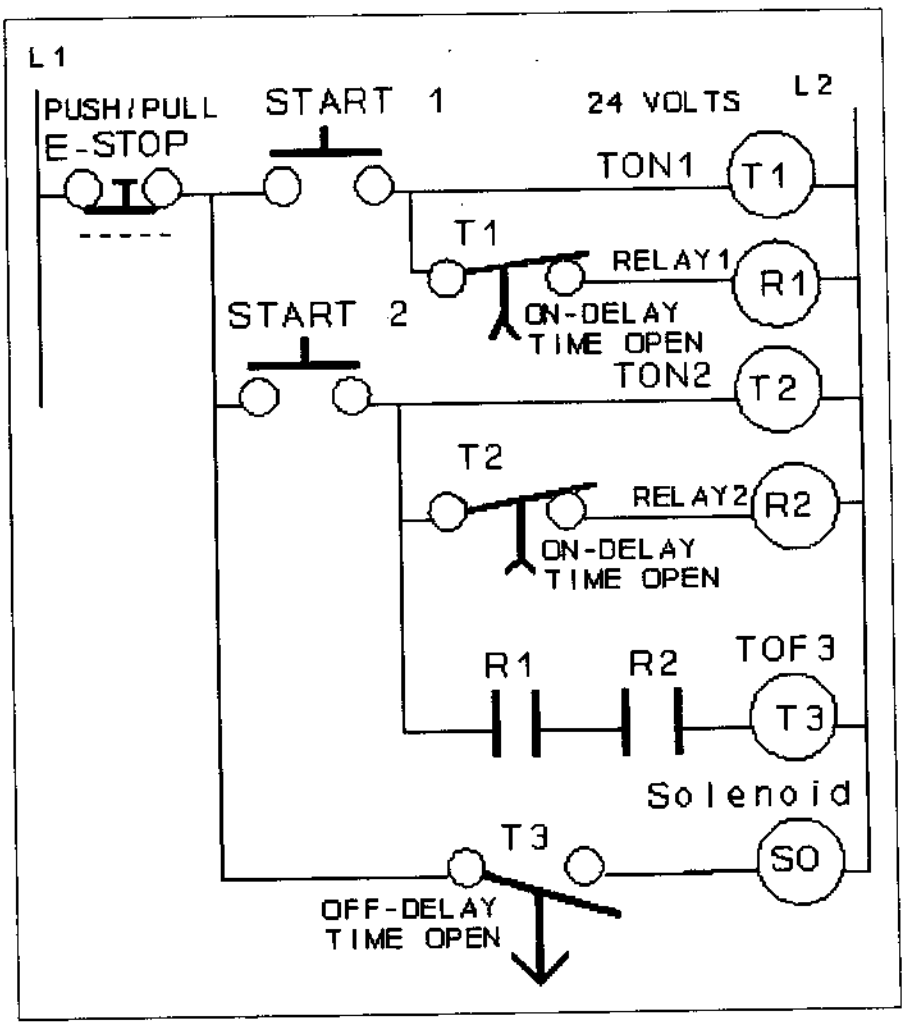
1. Sử dụng catalog công nghiệp, chọn bộ thời chuẩn ngắt (TOF) thỏ
các yêu cầu của thí nghiệm nêu trên.

Thí nghiệm 4.3. BỘ NHẤN THỜI CHUẨN

Mục đích

Nối mạch thời chuẩn theo sơ đồ (Hình 4.5) gồm:

1. Hai nút nhấn START để điều khiển cuộn solenoid (SO).



Hình 4.5. Mạch nhấn thời chuẩn.

2. Cả hai nút nhấn được nhấn đồng thời trong thời gian cho trước (khoảng 0.1 đến 10 giây). Nếu cả hai không được nhấn trong khoảng thời gian cho trước của các bộ thời chuẩn T1 và T2, cuộn solenoid sẽ không kích hoạt.
3. Khi được kích hoạt, cuộn solenoid sẽ vận hành trong khoảng thời gian xác định, sau đó sẽ dừng hoạt động (khoảng 1 đến 180 giây).
4. Mạch sử dụng hai bộ thời chuẩn đóng (TON) và một bộ thời chuẩn ngắt (TOF), 24V AC.

Sơ đồ bậc thang

Câu hỏi

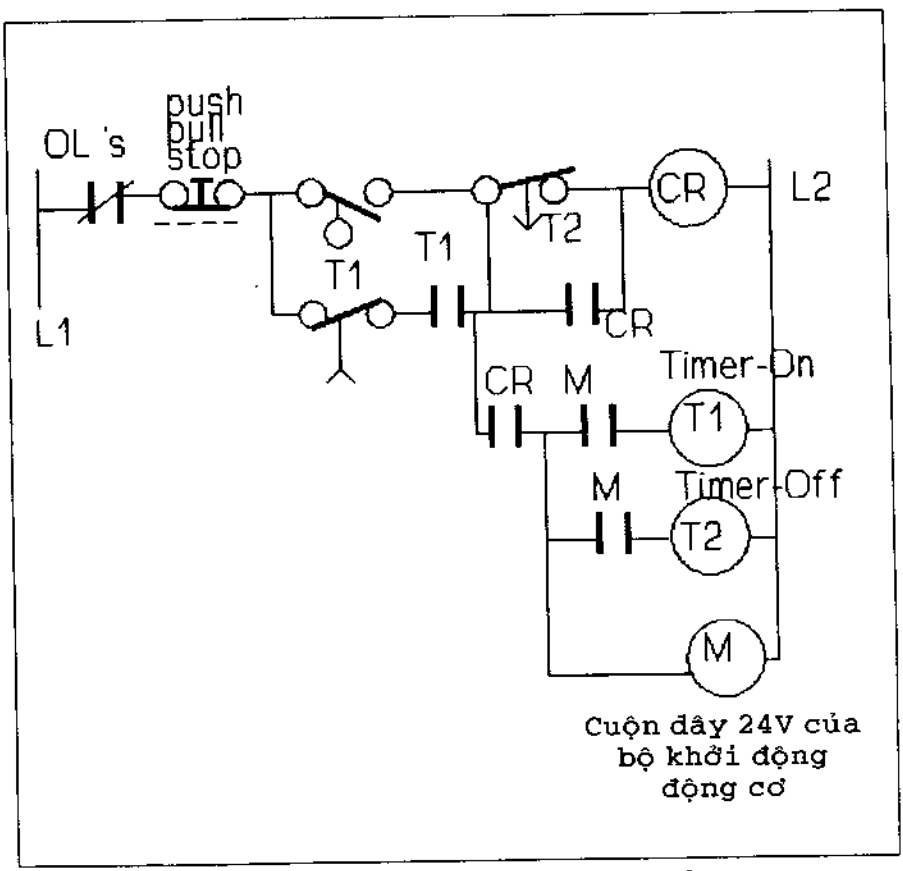
1. Sử dụng catalog công nghiệp, chọn hai bộ TON và một bộ TOF thỏa các yêu cầu của Thí nghiệm.

Thí nghiệm 4.4. MẠCH CHUYỂN MẠCH THỜI CHUẨN

Mục đích

Nối mạch thời chuẩn theo sơ đồ (Hình 4.6). Sau khi nối mạch, cần thực hiện:

1. Đóng công tắc chọn (ON) sau đó ngắt ngay (OFF), mạch vẫn được cấp điện (ON) trong thời gian cho trước trên bộ thời chuẩn đóng (TON).



Hình 4.6. Mạch chuyển mạch thời chuẩn

2. Đóng công tắc chọn (ON) ngay sau khi mạch ngắt nguồn, mạc vẫn bị ngắt (OFF) trong khoảng thời gian cho trước trên bộ thi chuẩn ngắt (TOF).
3. Mạch được thiết kế để điều khiển thời gian đóng/ngắt (ON/OFI của công tắc chọn và tránh khởi động sai.

Sơ đồ bậc thang

Câu hỏi

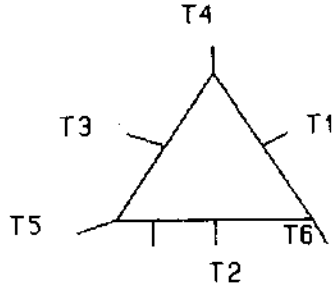
1. Sử dụng catalog công nghiệp, chọn bộ thời chuẩn bán dẫn (TON đáp ứng các yêu cầu của thí nghiệm.

Thí nghiệm 4.5. MẠCH DIỄN DỊCH HAI TỐC ĐỘ

Mục đích

Thí nghiệm này chỉ yêu cầu thiết kế sơ đồ bậc thang để điều khiển động cơ ba pha, hai tốc độ, công suất không đổi. Hình 4.7 minh họa nối kết động cơ. Mạch vận hành như sau:

1. Mạch kích hoạt công tắc nhấn – xoay 3 – vị trí.
2. Người vận hành có thể chọn tốc độ mong muốn bằng cách nhấn nút nhấn cao hoặc thấp.
3. Nhấn nút nhấn thấp, động cơ khởi động và chạy với tốc độ thấp. Nhấn nút nhấn cao, động cơ khởi động với tốc độ thấp chạy với tốt độ cao sau khoảng thời gian cho trước (1.5 đến 30 giây).
4. Người vận hành phải nhấn nút nhấn STOP (dừng) trước khi đổi tốc độ từ cao sang thấp.



SPEED	L1 L2 L3	OPEN	TOGETHER
LOW	T1--T2--T3	———	T4--T5--T6
HIGH	T6--T4--T5	all others	———

Hình 4.7. Nối kết động cơ 2- tốc độ công suất không đổi.

Sơ đồ bậc thang

Câu hỏi

1. Sử dụng sơ đồ phần cứng trong sổ tay hướng dẫn lắp đặt PL
hãy nêu sơ đồ nối kết cuộn dây của bộ khởi động động cơ điện
thấp nối với module xuất điện áp thấp.

2. Sử dụng sơ đồ phần cứng trong sổ tay hướng dẫn lắp đặc PL
hãy nêu sơ đồ nối kết các tiếp điểm thường mở (NO) của bộ kh
động nối với module nhập điện áp thấp. Sau đó nối nút nh
STOP thường đóng, nút nhấn START thường mở, và tiếp điể
phụ thường đóng với module nhập.

Bài tập lớn 5.0

HỆ THỐNG TRUYỀN ĐỘNG AC

MỤC ĐÍCH

Nghiên cứu mạch điều khiển tốc độ động cơ 3 pha – sử dụng truyền động động cơ AC tần số biến thiên..

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Lập cấu hình các xác lập ban đầu của bộ truyền động AC
- Lập trình các thông số của bộ truyền động AC để điều khiển tốc độ động cơ 3- pha.
- Nghiên cứu các chức năng vận hành của bộ truyền động AC.

VẬT LIỆU

(1) Bộ truyền động tần số biến thiên, 1336 – B003 – EAD – FA2 – L2 hoặc tương đương.

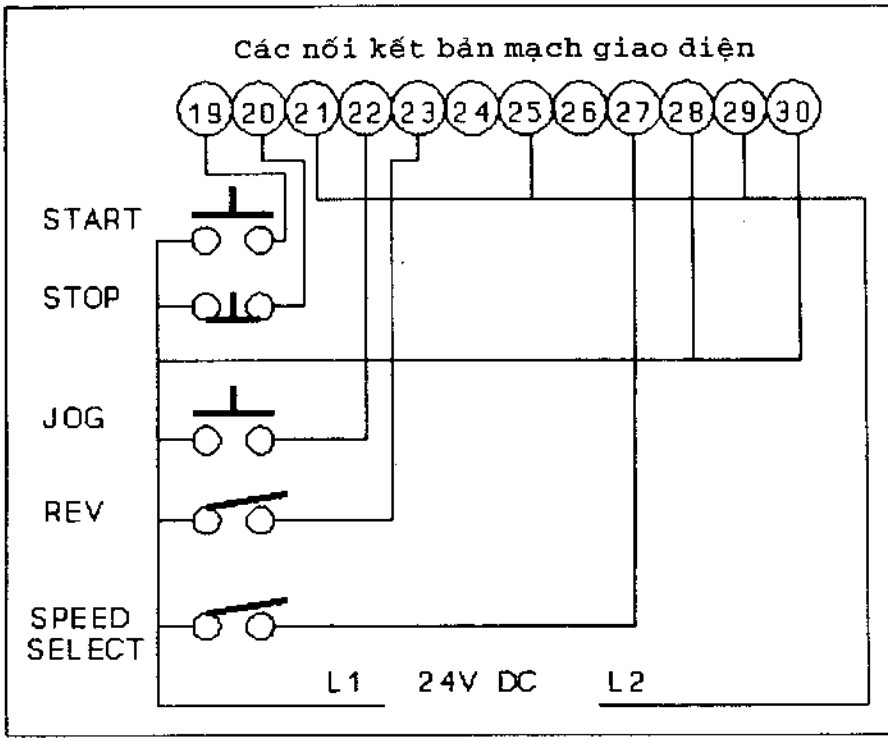
(1) Nút nhấn stop – start và chạy/dừng.

(1) Công tắc đảo chiều.

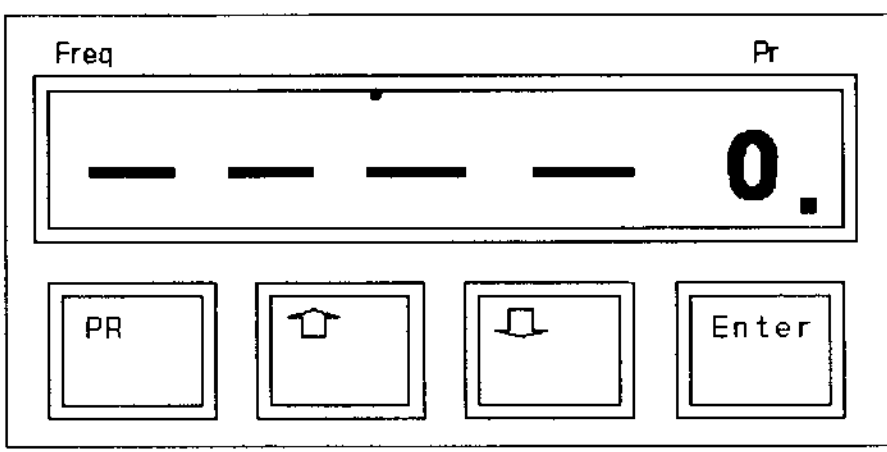
(1) Công tắc chọn tốc độ 2 – vị trí.

BỘ TRUYỀN ĐỘNG AC

Bộ truyền động AC tần số biến thiên nói chung có hai nhóm chức năng cơ bản, các điều khiển vận hành và các thông số lập trình. Các điều khiển vận hành cho phép người vận hành khởi động, dừng, tự động/bằng tay, đổi chiều và tốc độ của bộ điều khiển bằng cách chỉnh chiết áp kế hoặc các thiết bị tương tự. Bộ truyền động có thể được vận hành trực tiếp từ panel điều khiển lắp ở phía trước hoặc thông qua giao diện sử dụng các thiết bị điều khiển ngoại vi (Hình 5.1). Các thông số lập trình có thể được lập trình tại chỗ, trên panel điều khiển hoặc thông qua cổng truyền thông nối tiếp sử dụng các thiết bị tương ứng. Bốn nút nhấn trên màn hiểu thị và panel lập trình được dùng để quan sát và lập trình các thông số. Các thông số có thể được quan sát trong khi bộ truyền động



Hình 5.1. Nối kết giao diện



Hình 5.2. Panel lập trình dùng cho bộ truyền động AC.

hoạt động nhưng không thay đổi được. Số thập phân xuất hiện ở góc xa bên phải biểu thị chế độ lập trình (Hình 5.2).

Nút nhấn PR được dùng để chuyển từ hiển thị vận hành sang hiển thị quan sát các thông số. Khi trong chế độ hiển thị quan sát, nút nhấn được dùng để tăng dần các thông số (theo gia số). Nút nhấn ENTER được dùng để chuyển từ quan sát sang lập trình chỉ khi hiển thị thông số 0 (zero). Nút ENTER còn được dùng để lưu các giá trị hiển thị. Các nút nhấn INCREMENT và DECREMENT được dùng để cuộn lên hoặc xuống giá trị thông số được nhập. Nhấn đồng thời cả hai nút này sẽ kết thúc lập trình.

Bài tập

Bài tập này giúp bạn làm quen với các thuật ngữ và thông số dùng trong lập trìnhb bộ truyền động AC. Từng chức năng điều khiển đều có thông số riêng, là một phần trong chương trình chung để vận hành bộ truyền động AC. Bộ truyền động AB (Allen - Bradley) 1336 được dùng để lập trình các thông số này.

Bạn cần nắm vững Bài tập này, do nội dung sẽ được dùng tiếp trong Bài tập 20, Digital to Analog.

LẬP TRÌNH LOGIC

1. Bạn hãy xem kỹ sơ đồ mạch và tuân theo các hướng dẫn an toàn của nhà sản xuất (Bộ AB 1336).
2. Nối các nút nhấn điều khiển từ xa (dừng – điều khiển – chạy/dừng, công tắc chọn, công tắc đảo chiều) vào bản mạch giao diện.

CÁC BƯỚC THỰC HIỆN

1. Kiểm tra, tuân thủ mọi yêu cầu vận hành và an toàn trước khi kích hoạt bộ truyền động AC.
2. Xoay công tắc chọn của bộ truyền động đến chế độ vận hành bằng tay để vận hành bộ truyền động với các điều khiển trực tiếp.
3. Sử dụng panel hiển thị tại chỗ và lập trình để quan sát và lập trình các thông số của bộ truyền động. Tham khảo sổ tay lập trình để thay đổi các thông số liệt kê ở phần dưới. Ghi lại các giá trị thông số được chọn.

Thông số/ chức năng

5,6. *Chọn tốc độ* xác định một trong sáu nguồn sẽ điều khiển tần số ra của bộ truyền động với công tắc chọn tốc độ.

0 = chiết áp kế panel điều khiển tại chỗ. 1 = nguồn 0-10V DC. 2 = nguồn 4-20 mA. 3 = nguồn xung. 4 = Ngõ nhập nối tiếp. 5 = chiết áp kế tốc độ ở xa.

Bạn hãy xác lập thông số 5 cho nguồn 0 đến + 10V DC.

Xác lập thông số 6 cho chiết áp kế tại chỗ hoặc ở xa.

16, 17, 18, 19, 20. *Các thông số vận hành* xác lập tần số vận hành bộ truyền động và điện áp.

16 = _____ xác lập tần số vận hành tối thiểu 0-120Hz

17 = _____ xác lập tần số cơ bản theo các giá trị trên từng tên động cơ.

18 = _____ xác lập điện áp cơ bản theo các giá trị trên từng tên động cơ.

19 = _____ xác lập tần số vận hành cực đại 0-250Hz

20 = _____ điện áp cực đại được xác lập bằng hoặc lớn hơn điện áp cơ bản của thông số 18.

10, 12, 13. Thông số *chọn dừng* (stop) chọn chế độ dừng động cơ 0—dừng từ từ, 1—dừng bằng thắng (phanh) DC, 2—dừng theo xung.

10 = _____ xác lập chế độ dừng từ từ (0), dừng bằng thắng DC (1), hoặc dừng theo xung (2).

12 = _____ xác lập thời gian trễ (0-15giây) khi dừng

13 = _____ xác lập điện áp 0 – 115V DC cấp cho động cơ để dừng (thắng).

7, 30. *Thời gian tăng tốc* xác định thời gian để động cơ tăng (tốc độ) từ 0 Hz đến tần số cực đại.

7 = _____ xác lập 1 – 600 giây theo thời gian tăng tốc #1 từ 0 Hz đến tần số cực đại.

30 = _____ xác lập 1 – 600 giây theo thời gian tăng tốc # 2, nếu được sử dụng.

8, 31. *Thời gian giảm tốc* xác định thời gian giảm (tốc độ) từ tần số cực đại đến 0 Hz.

8 = _____ xác lập 1–600 giây theo thời gian giảm tốc # 1 từ tần số cực đại đến 0 Hz.

31 = _____ xác lập 1 – 600 giây theo thời gian giảm tốc # 2, nếu được dùng.

32, 33, 34, 35. *Bỏ qua các dải tần số* thiết lập các khoảng tần số tại đó bộ truyền động không vận hành. Mỗi dải xác lập một khoảng tần số tại đó bộ truyền động không vận hành.

32 = _____ bỏ qua dải tần số # 1.

33 = _____ bỏ qua dải tần số # 2.

34 = _____ bỏ qua dải tần số # 3.

35 = _____ xác lập khoảng cộng hoặc trừ các dải tần số 0-15 Hz.

43, 44. *Thời gian và tần số sóng*. Bộ truyền động không theo thời gian tăng tốc, chỉ chuyển trực tiếp sang xác lập sóng. Bộ truyền động duy trì tần số sóng theo thời gian cho trước, sau đó sẽ giảm hoặc tăng. Điều này cho phép động cơ đạt được module khởi động cao hơn.

43 = _____ xác lập tần số bộ truyền động đạt được khi khởi động từ 0 đến 120 Hz.

44 = _____ xác lập tần số bộ truyền động duy trì tần số sống (khởi động) từ 0 đến 10 giây.

45. *Điều biến chiều rộng xung (PWM)*. Thông số này điều chỉnh tần số sóng mang tối thiểu được dùng để tạo tần số xuất PWM. Tăng PWM sẽ giảm tiếng ồn của động cơ và tính ổn định trên động cơ có tải nhẹ. Giảm PWM sẽ tăng tiếng ồn động cơ và tính ổn định động cơ.

45 = _____ xác lập tần số sóng mang từ 0.4 kHz đến 2 kHz.

38. *Dòng quá tải* xác lập bảo vệ quá tải cho động cơ theo phần trăm cường độ dòng điện. Xác lập từ 50 đến 100% dòng định mức. Thời chuẩn bắt đầu khi dòng ra vượt quá thông số này.

38 = _____ % dòng ra định mức.

21, 22, 23, 24. *Đảo chiều và chạy/dừng.* Chọn nguồn xác định chiều và chế độ chạy/dừng của động cơ. Các xác lập này điều khiển nơi vận hành của bộ truyền động từ các thiết bị nhập từ xa hoặc tại chỗ.

21 = _____ chạy tại chỗ

22 = _____ nguồn đảo.

23 = _____ nguồn chạy/dừng.

24 = _____ tần số được dùng khi nhấn chạy/dừng.

Sau khi lập trình bộ truyền động, bạn hãy kiểm tra các thông số bằng cách vận hành bộ truyền động với panel điều khiển tại chỗ.

PHẦN II

LẬP TRÌNH

Hiểu chi tiết các lệnh lập trình là rất quan trọng đối với nhà công nghệ khi thiết kế và lập trình hệ thống điều khiển. Do đó, phần này sẽ nhấn mạnh các lệnh logic được dùng trong bộ điều khiển lập trình và ứng dụng trong các mạch điều khiển công nghiệp. Các ví dụ về ứng dụng hệ thống được giới thiệu để cung cấp ý tưởng cơ bản về sử dụng các lệnh logic trong thực tiễn.

NỘI DUNG

Sau khi hoàn tất Phần này bạn sẽ có khả năng:

- Thiết kế và lập trình các mạch điều khiển logic.
- Hiểu và áp dụng các lệnh đếm, thời chuẩn, relay, để xây dựng hệ thống điều khiển.
- Thực hiện và áp dụng các lệnh tập tin, tính toán, so sánh, để thiết kế hệ thống điều khiển phức tạp.
- Nghiên cứu công dụng của bộ ghi dịch chuyển và bộ tạo thứ tự để vận hành hệ thống điều khiển lặp.
- Sử dụng các tập tin bit để tìm lỗi trong các quá trình tự động.
- Áp dụng công tắc xoay bánh xe nhỏ trong các chức năng lập trình.
- Sử dụng lệnh kiểm tra giới hạn để tạo chuỗi thuận và đảo.
- Nhận biết các dạng dữ liệu cơ bản và dịch chuyển thời gian trong bộ ghi dịch chuyển.
- Biết các ứng dụng chuyển đổi digital – analog và analog – digital.
- Xác định các đặc tính cơ bản của lệnh PID.
- Nghiên cứu khái niệm lập trình văn bản cấu trúc.

GIỚI THIỆU

Lập trình là quá trình đơn giản báo cho bộ điều khiển cách thức thực hiện tác vụ một cách chính xác. Để thực hiện điều này, bạn phải "nói" với PLC bằng ngôn ngữ riêng, PLC có thể hiểu. Sau đó PLC sẽ thực thi từng lệnh nhập, không thực hiện điều không được ra lệnh, do đó, bạn phải báo cho PLC từng tác vụ lần lượt theo các bước để PLC thực hiện.

Trong Phần này, sẽ sử dụng các lệnh cần thiết có khả năng báo cho PLC theo ngôn ngữ riêng của PLC. Từ các lệnh này, bạn sẽ xây dựng một số chương trình tương tự lệnh logic được dùng trong sơ đồ bậc thang đã xét ở Phần I.

Trước khi bắt đầu lập trình, bạn cần xét khả năng hướng đến các kết quả cần thiết mà bạn muốn thực hiện. Điều này có thể được thực hiện bằng cách tổ chức công việc theo chuỗi thứ tự các bước vận hành. Chuỗi thứ tự liệt kê dưới đây minh họa các bước này.

1. Xác định mạch

Trong một số trường hợp, mạch được xác định có thể là hệ thống công nghiệp phức tạp đòi hỏi nhóm kỹ sư thiết kế và xây dựng. Trong các trường hợp khác, công việc có thể chỉ đơn giản là mắc thiết bị ON/OFF cơ bản. Nói chung, xuất phát điểm luôn luôn là ý tưởng về các kết quả mong muốn. Sau đó, cùng với kiến thức về mạch điều khiển, phần mềm và phần cứng, bạn có thể hình dung những điều cần thực hiện và tìm hướng giải quyết.

Xác định mạch là rất quan trọng, từ đó bạn có thể nhận biết nhóm đại lượng cơ bản, cần thiết để viết chương trình.

1. Các tín hiệu nhập cần cung cấp.
2. Các tín hiệu xuất cần điều khiển.
3. Phép toán logic cần dùng trong điều khiển mạch.

Hiện có nhiều mạch điều khiển được dùng trong ứng dụng điều khiển động cơ. Các kiểu mạch phổ biến được sử dụng trong các Bài tập thuộc phần này.

2. Viết chuỗi thứ tự

Sau khi xác định mạch và nối các thiết bị nhập, xuất vào module I/O, bạn có thể viết chương trình cho ứng dụng cụ thể. Bạn có thể đạt được cùng kết quả với một số chương trình mà thợ điện có thể nối một mạch điều khiển theo nhiều cách để thực thi các kết quả như nhau. Điều quan trọng là bạn viết chương trình vận hành an toàn và chuẩn xác. Bước thứ nhất là viết chuỗi thứ tự vận hành các thiết bị nhập và xuất. Bạn cần quyết định điều các thiết bị phải thực hiện và các điều khiển nào là đúng (bảng chân trị) trước khi vận hành các thiết bị đó. Điều này sẽ giảm khó khăn khi lập trình, giúp bạn hình dung tính logic cần thiết để điều khiển mạch. Bạn hãy xem lại Bài tập 1.0, Logic Lập trình.

3. Viết và tải chương trình

Sau khi xác định mạch, bạn có thể viết hoặc mã hóa. Mã hóa gồm chuỗi các lệnh có thể được viết và do bộ điều khiển PLC thực thi. Cách tốt nhất là viết logic chương trình và xác định các địa chỉ lệnh bạn sẽ sử dụng. Điều này giúp bạn nhận biết hướng giải quyết từ đầu và chuẩn bị tải chương trình.

Khi tải chương trình, toàn bộ truyền thông của bạn được thực hiện thông qua bàn phím. Điều quan trọng là bạn phải biết chức năng từng phím. Bạn hãy xem sổ tay hướng dẫn của nhà sản xuất để biết thông tin về các chức năng phím của bộ PLC. Khi nhập lệnh vào PLC bạn cần chọn chế độ PROGRAM, mọi ngõ ra đều bị ngắt nguồn trong chế độ này và máy được điều khiển bằng PLC sẽ không vận hành.

4. Kiểm tra và chạy chương trình

Sau khi nhập chương trình, bạn hãy kiểm tra chương trình này. Trong chế độ kiểm tra (TEST), chương trình được kiểm tra với các điều kiện mô phỏng sự vận hành (các lệnh chớp sáng trên màn hiển thị). Các thiết bị nhập đều hoạt động và được bộ xử lý nhận biết, nhưng thiết bị xuất không được cấp nguồn. Mọi thiết bị xuất đều không hoạt động trong chế độ này. Bộ mô phỏng thí nghiệm có thể được dùng để thực hiện và minh họa các Bài tập. Nếu kết quả kiểm tra đạt yêu cầu, bạn có thể chuyển sang chế độ chương trình (PROGRAM) và ghi lại các lệnh. Điều này nhằm tránh các nhầm lẫn hoặc sai sót, nếu sau này bạn muốn chỉnh sửa hoặc mở rộng chương trình. Tại thời điểm này, bạn có thể lưu chương trình lên đĩa riêng (đĩa mềm, CD).

Sau khi bảo đảm các bước cần thiết đều được thực hiện đáp ứng mọi yêu cầu về an toàn và vận hành, bạn có thể chạy chương trình.

An toàn

1. Các bộ khởi động nối mạch và kiểm tra trước khi nối với tải hoặc động cơ.
2. Mọi giới hạn phải được kiểm tra về vận hành, kiểm lại các bộ giới hạn, dừng, và quá tải.
3. Không làm việc với dây nóng (đang có điện), luôn luôn ngắt nguồn trước khi thực hiện các nối kết.
4. Tuân thủ các yêu cầu an toàn và vận hành của nhà sản xuất.
5. Kiểm tra cẩn thận mỗi khi nối kết và chạy chương trình.

Bài tập lớn 6.0

STOP – START (PLC)

MỤC ĐÍCH

Viết chương trình trên PLC cho mạch điều khiển nút nhất Stop – Start điều khiển bộ khởi động của động cơ 3 – pha.

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Sử dụng các lệnh XIC (kiểm tra đóng ngõ vào) để lập trình các nút nhấn Stop và Start.
- Lập trình bộ khởi động và các tiếp điểm.
- Sử dụng lệnh khóa/mở khóa.
- Sử dụng lệnh MCR.
- Lập trình các hàm AND, OR, NOT, và MEMORY.

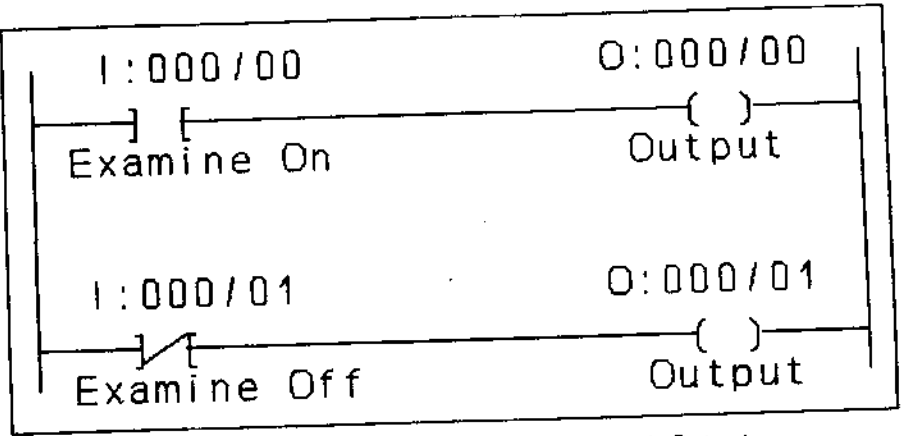
TÀI LIỆU

Bạn hãy tham khảo Phụ lục A: "Các lệnh điều khiển chương trình và kiểu relay" để biết về các lệnh XIC, XIO, OTE, OTL, OUT, và MCR được dùng trong Bài tập này.

THẢO LUẬN

Các bộ PLC có nhiều chức năng tương tự hệ thống điều khiển relay. Các chức năng điều khiển tương tự hệ thống relay được cung cấp bằng lệnh xuất và lệnh examine (kiểm tra).

Có hai lệnh examine (kiểm tra): ON (-] [-) và OFF (-] / [-). Khi lệnh *examine off* hoặc *examine on* được cấp địa chỉ bit trong bộ nhớ, lệnh này có thể gián tiếp kiểm tra trạng thái của thiết bị nhập tương ứng. Trạng thái của bit này sẽ là 1 hoặc 0, phản ánh điều kiện ON (đúng) hoặc OFF (sai) của thiết bị nhập/xuất (Hình 6.1). Lệnh xuất báo cho bộ xử lý kích hoạt bit nhớ có địa chỉ khi các điều kiện bậc thang là đúng (ON). Bit nhớ này sẽ xác định trạng thái ON hoặc OFF của thiết bị xuất khi hướng đến ngõ ra.

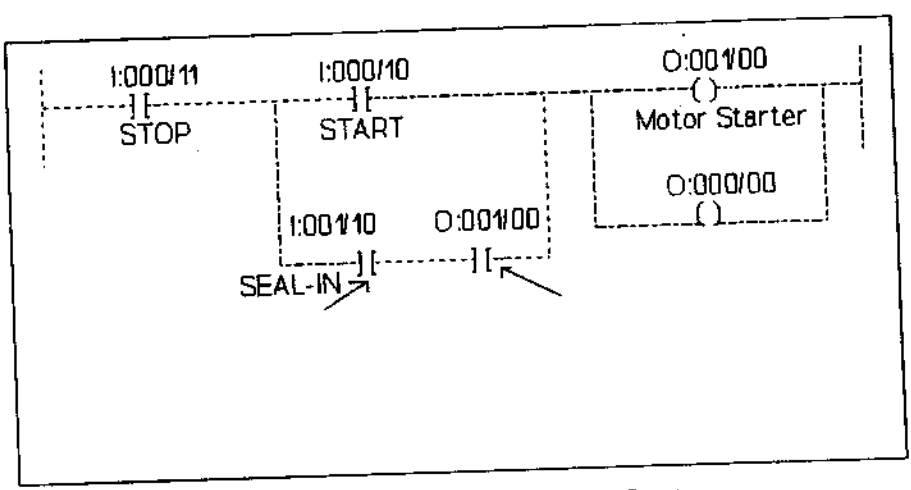


Hình 6.1. Các lệnh Examine ON/OFF và Output

Nút nhấn Stop – Start

Trong lập trình, cùng một lệnh được dùng cho cả nút nhấn Stop và Start, do các lệnh không phân biệt giữa công tắc thường mở và thường đóng, chỉ tìm điều kiện ON hoặc OFF của thiết bị đó và xác lập bit trong bộ nhớ tùy theo điều khiển cụ thể.

Trên Hình 6.2, bạn có thể thấy nút Stop được lập trình với lệnh XIC thường mở (-] [-) và nút Start cũng được lập trình với lệnh XIC thường



Hình 6.2. Chương trình Stop – Start

mở (-] [-). Nút Stop thường đóng (I:000/11) là đúng (ON) khi không được nhấn. Nút start thường mở (I:000/10) là đúng (ON) khi được nhấn. Điều này tạo ra cùng điều kiện cho cả nút Stop và Start.

An toàn

Có nhiều phương pháp lập trình mạch điều khiển với bộ khởi động điều khiển động cơ. *Safety* (an toàn) là yếu tố quyết định trong phương pháp lập trình.

Phương pháp thứ nhất là lập trình các tiếp điểm thực của bộ khởi động theo kiểu nối tiếp (AND) với bit nội hướng đến ngõ ra. Do chương trình và I/O đều quét nhiều lần, điều quan trọng là có cả tiếp điểm thực và tiếp điểm nội bên trong mạch. Trên Hình 6.2, địa chỉ I:001/10 biểu thị tiếp điểm thực của bộ khởi động.

Công dụng của tiếp điểm thực là ngắt mạch điều khiển đối với điều kiện bất kỳ có thể ngắt bộ khởi động, chẳng hạn quá tải. Tiếp điểm nội không nhận biết sự hở mạch bộ khởi động, do đó vẫn đóng. Sự xác lập lại bằng tay sau khi quá tải sẽ khởi động động cơ một cách tức thời nhưng có thể không an toàn.

Phương pháp thứ hai là nối các tiếp điểm quá tải của bộ khởi động theo kiểu thiết bị nhập, sau đó lập trình các tiếp điểm này trong mạch điều khiển. Nhiều hệ thống thiết bị đòi hỏi tiếp điểm quá tải và các tiếp điểm thực phải nối ở ngõ vào và lập trình trong mạch.

Các Bài tập và các thí nghiệm trong sách này, có thể sử dụng một hoặc cả hai phương pháp an toàn. Bài tập 6.0 và các thí nghiệm kèm theo sẽ minh họa các khái niệm logic và các kỹ thuật sẽ được ứng dụng trong nhiều bài tập khác.

LOGIC CHƯƠNG TRÌNH

1. Lệnh nhập cho nút nhấn Stop thường đóng khi được nhấn, mạch sẽ sai (OFF) và động cơ dừng.
2. Lệnh nhập cho nút nhấn Start thường mở, khi được nhấn, mạch sẽ đứng (ON) và động cơ khởi động.
3. Tiếp điểm thực ở bộ khởi động, được coi là ngõ nhập và được mắc vào module nhập.
4. Bộ khởi động điện áp toàn phần với cuộn dây 24V DC và các tiếp điểm quá tải mắc theo kiểu thiết bị nhập.

5. Lệnh xuất đối với đèn điều khiển 24V, hoạt động (ON) khi động cơ vận hành (ON – khi hoạt động).

6. Nhấn nút Start sẽ nối mạch động cơ cho đến khi nhấn nút stop.

CHÚ Ý: bạn hãy đọc Thí nghiệm 6.6 trước khi bắt đầu Bài tập này.

CÁC BƯỚC THỰC HIỆN

1. Bạn hãy lập thư mục chương trình, có thể đặt tên là PROJECT 6 và đổi tên tập tin là stop – start. Thực hiện tập tin giám sát (f8) và xây dựng chương trình.

2. Viết chương trình PLC cho mạch stop – start (Hình 6.2)

   CHÚ Ý: Kiểm tra địa chỉ của từng bộ khởi động trong trạm làm việc của bạn. Các địa chỉ bộ khởi động ở trạm của bạn có thể khác các địa chỉ trong chương trình.

3. Bạn hãy quan sát lệnh thứ nhất -] [- I: 000/11, đây là lệnh mở XIC hướng đến ngõ vào nút nhấn Stop thường đóng. Bộ xử lý quét ngõ vào này và thấy ngõ vào đang đóng. Do đó, bộ xử lý xác lập bit lệnh XIC là 1 và cho phép dòng điện đi qua để đến lệnh kế tiếp.

4. Lệnh kế tiếp -] [- I: 000/10, đây là lệnh XIC hướng đến nút nhấn Start thường mở. Bộ xử lý quét ngõ vào nút Start và coi ngõ này là mở. Chỉ khi nhấn nút Start (đóng) sẽ xác lập bit của lệnh XIC là 1 và nối mạch. Đây là kỹ thuật hơi khó phân tích, cùng kiểu lệnh được dùng cho cả nút Stop và Start. Bạn có thể kiểm chứng thao tác này khi cho mạch hoạt động.

5. Kỹ thuật thứ hai trong Bài tập cũng là tiếp điểm thực (-] [- I:001/01), chúng được lập trình theo chuỗi nối tiếp với bit nội hướng đến ngõ ra.

6. Sau khi hoàn tất các bước nêu trên, bạn có thể kiểm tra chương trình. Trong chế độ TEST bạn sẽ thấy chương trình chỉ vận hành trên màn hình, bộ khởi động và tiếp điểm thực không hoạt động.

   Nếu mọi thứ đều có vẻ chuẩn xác trong chế độ TEST, bạn hãy chuyển sang chế độ RUN và vận hành chương trình chỉ với bộ khởi động (không cần nối vào động cơ). Bạn hãy kiểm tra quá tải và mạch Stop. Nếu mạch vận hành chuẩn xác, bạn có thể cho chạy với động cơ, nhưng phải tuân thủ mọi quy định về an toàn.

7. Bước tiếp theo là ghi lại các lệnh và sơ đồ mạch. Bạn có thể sử dụng nội dung trên Hình 6.2 để tham khảo. Bạn có thể đặt tên

cho tập tin này là Project 6, stop – start, và lưu lên đĩa mềm để bảo quản.

8. Sau khi hoàn tất phần thứ nhất của Bài tập 6, bạn hãy thực hiện các thí nghiệm. Mỗi thí nghiệm đều cần được ghi lại một cách riêng rẽ và lưu lên đĩa mềm.

9. Sau khi hoàn tất các thí nghiệm, bạn có thể lập báo cáo và ghi nhận xét. Đây sẽ là tài liệu rất hữu ích để bạn tham khảo hoặc chỉnh sửa khi cần thiết.

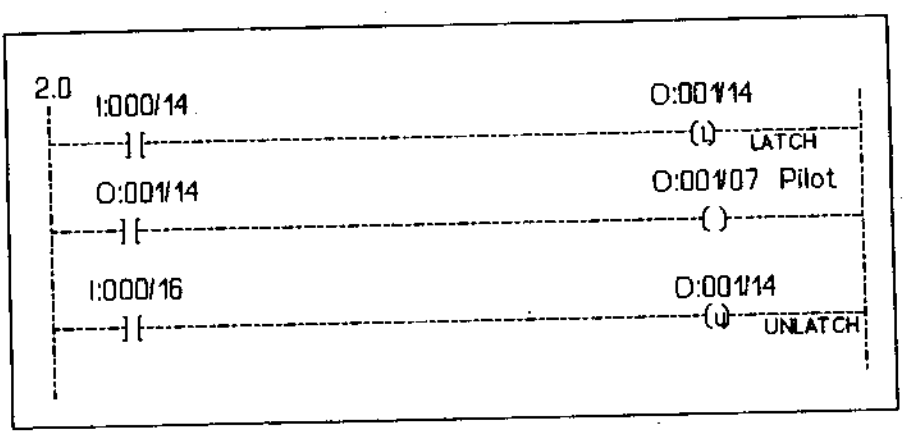
Thí nghiệm 6.1. KHỞI ĐỘNG THEO THỨ TỰ

Bạn hãy thiết kế mạch sau đây và bổ sung vào chương trình Project 6, xem lại Thí nghiệm 1.3 để biết sơ đồ bậc thang của Thí nghiệm này.

1. Bộ khởi động thứ hai (#2) với nút Stop – Start riêng, không có khả năng khởi động trừ khi bộ khởi động thứ nhất đang chạy.
2. Đèn báo cho biết động cơ (#1) là OFF, nghĩa là đèn này sáng (ON) khi động cơ (#1) OFF.
3. Sử dụng các địa chỉ thích hợp cho các nút nhấn và bộ khởi động.

Thí nghiệm 6.2. KHÓA/MỞ KHÓA

Bạn hãy nhập chương trình khóa/mở khóa được nêu trên Hình 6.3. Mục đích của thí nghiệm này là giúp bạn làm quen với lệnh khóa/mở khóa.



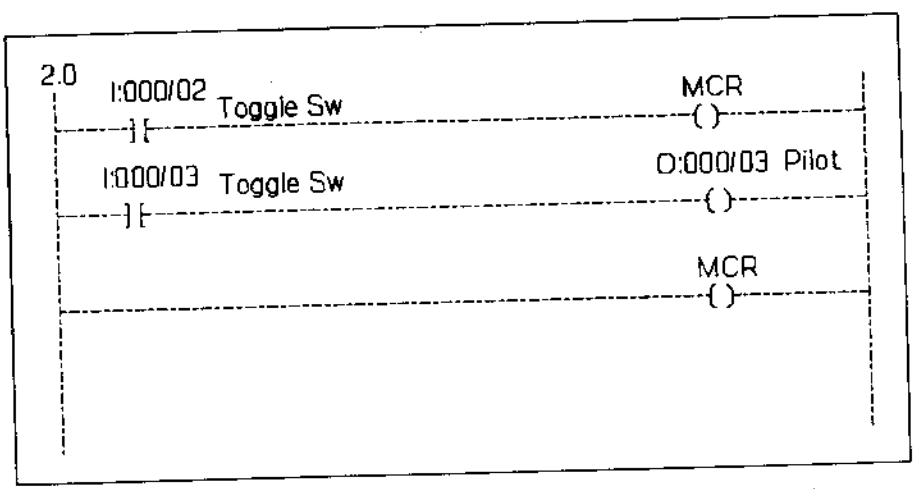
Hình 6.3. Chương trình khóa/mở khóa

1. Các lệnh khóa và mở là các lệnh xuất có khả năng ngăn chặn, thường được dùng cho bit bảng dữ liệu bất kỳ được điều khiển.
2. Sau khi nhập chương trình, bạn hãy thử vận hành chương trình này. Bạn cần nhớ sử dụng các địa chỉ thích hợp với bộ mô phỏng hiện có.

3. Trước hết bạn hãy nhấn nút khóa (latch) và quan sát ngõ ra khóa (latch), ngõ ra phải vẫn giữ ON kể cả sau khi bạn nhã nút này để ngắt.
4. Kế tiếp, bạn hãy nhấn nút mở khóa (un latch), điều này sẽ mở khóa ngõ ra cho đến khi nhấn nút khóa (latch).
5. Đặc tính này là rất quan trọng, do bạn có thể cố định một bit với xung tạm thời. Với đặc tính đó, bạn có thể khóa hoặc mở khóa các bit vào nhiều thời điểm sử dụng bộ đếm thời chuẩn. Các lệnh này sẽ được dùng trong nhiều thí nghiệm kế tiếp.

Thí nghiệm 6.3
CÀI ĐẶT LẠI ĐIỀU KHIỂN CHÍNH (MCR)

Bạn hãy nhập chương trình MCR được nêu trên Hình 6.4. Mục đích của thí nghiệm này là giúp bạn làm quen với lệnh MCR.



Hình 6.4. Chương trình cài đặt lại điều khiển chính (MCR).

1. Lệnh cài đặt lại điều khiển chính (MCR) là lệnh xuất, cho phép bộ xử lý kích hoạt hoặc ngăn chặn một vùng trong sơ đồ bậc thang phù hợp với logic chương trình.
2. Sau khi nhập chương trình này, bạn hãy thử vận hành.

3. Trước hết bạn hãy kích hoạt công tắc Toggle điều khiển lệnh MCR kiểm chứng sự vận hành của mạch ON/OFF trong vùng MCR. Các mạch trong vùng MCR phải được kích hoạt.

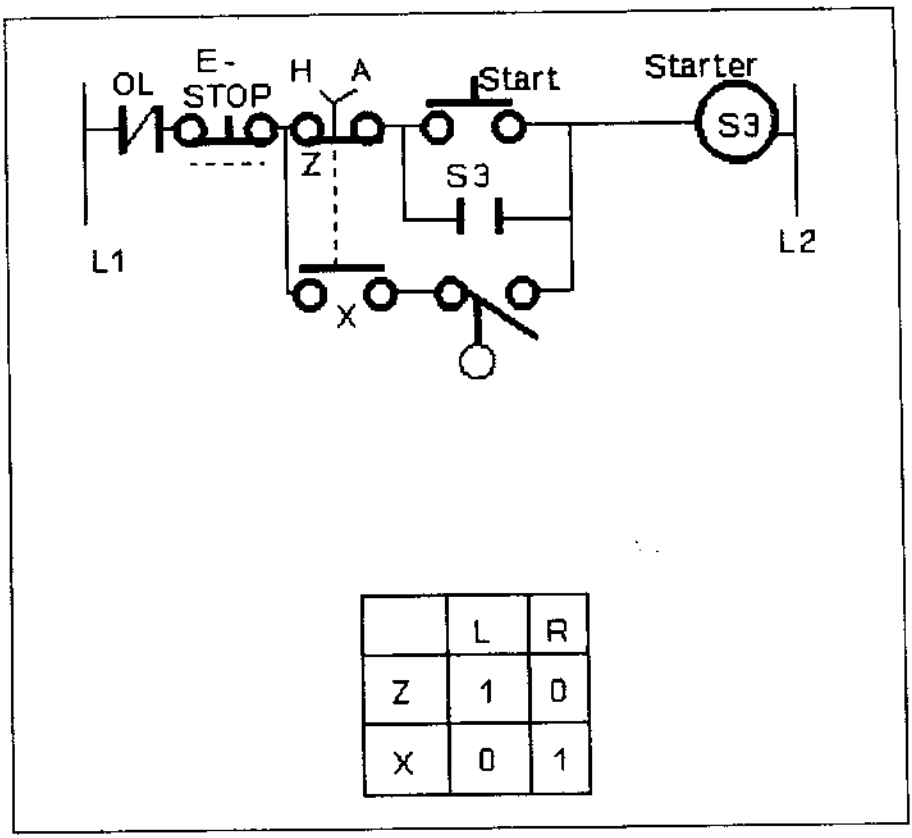
4. Kế tiếp, bạn ngắt công tắc điều khiển lệnh MCR và thử vận hành mạch trong vùng MCR, mạch đó phải bị chặn, chỉ có phần mạch bên ngoài vùng MCR là hoạt động.

5. Bạn có thể sử dụng các lệnh MCR để tạo các vùng chương trình để ngắt mọi tín hiệu ra không bị chặn trong vùng đó.

   Chú ý: Điều quan trọng cần biết là lệnh này không được dùng để thay cho relay điều khiển chính với chức năng dừng (stop) khẩn cấp. Bạn hãy đọc kỹ các quy định an toàn trong sổ tay hướng dẫn về tập hợp lệnh.

Thí nghiệm 6.4. TỰ ĐỘNG – BẰNG TAY

Bạn hãy chuyển sơ đồ mạch trên Hình 6.5 thành sơ đồ PLC, lập trình mạch này với các thông số dưới đây.



	L	R
Z	1	0
X	0	1

Hình 6.5. Mạch tự động – bằng tay.

1. Sử dụng bộ chuyển mạch Toggle làm công tắc chọn. Bộ chuyển mạch Toggle hướng đến lệnh XIC (-] [-) vàXIO (-] / [-) có thể được dùng làm công tắc chọn 2 vị trí. Bạn hãy chọn nút nhấn Stop - Start để điều khiển bằng tay hoặc chọn công tắc mức chất lỏng để điều khiển tự động bộ khởi động (S3). Sử dụng bộ chuyển mạch Toggle để mô phỏng công tắc mức chất lỏng.

2. Bạn hãy thực hiện các bước của mạch Stop – Start trong Bài tập 6 đối với các tiếp điểm thực.

3. Mạch Stop là mạch đẩy/kéo E-stop. Nếu không có loại mạch này bạn hãy sử dụng công tắc ON/OFF cho mạch stop.

Thí nghiệm 6.5.
CÁC ĐÈN SÁNG KHÔNG THEO THỨ TỰ

Bạn hãy chuyển đổi và lập trình cho mạch điện trong Thí nghiệm 1.5. Thí nghiệm này dùng ba nút Start để điều khiển đèn. Bạn hãy tuân theo các bước dưới đây.

1. Nhấn nút Start bất kỳ sẽ cấp nguồn và đóng mạch cho đèn tương ứng. Mỗi lần chỉ sáng một đèn. Bạn hãy dùng lệnh xuất (OTE) để kích hoạt các đèn.

2. Khi nhấn đồng thời nhiều nút, các đèn tương ứng sẽ sáng, nhưng chỉ một đèn tiếp tục sáng sau khi nhả các nút.

Thí nghiệm 6.6. MẠCH LOGIC THÍ NGHIỆM

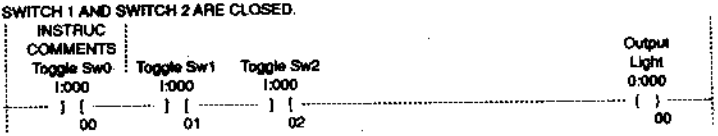
Thí nghiệm này có thể được thực hiện trước trong Bài tập 6.0 để giúp bạn đạt được kỹ năng lập trình và làm quen với phần mềm. Đây còn là thí nghiệm bổ sung để thực hành lập trình các mạch sơ đồ bậc thang, phối hợp các khái niệm logic bậc thang, nhưng chỉ cần dùng các lệnh examine ON/OFF cơ bản và lệnh xuất (Hình 6.6). Bạn cần đọc kỹ các quy định về an toàn trước khi tiến hành thí nghiệm.

Bạn hãy tuân theo các bước dưới đây:

1. Tạo thư mục chương trình riêng, đặt tên là EXP 6-6 và tên tập tin là Logic.
2. Lập trình bậc thang thứ nhất của Thí nghiệm 6.6 (logic AND), bạn cần dùng các địa chỉ thích hợp.
3. Sau khi hoàn tất hai bước trên, bạn chuyển sang chế độ RUN và vận hành chương trình, kiểm tra các khái niệm logic AND trong chương trình.
4. Nếu mọi thứ đều vận hành ổn định và bạn hiểu các khái niệm trong bậc thang này, bạn hãy chuyển sang bậc thang kế tiếp trong chương trình.
5. Trong chế độ PROGRAM, bạn hãy nhập bậc thang logic OR, vận hành chế độ RUN và kiểm tra chương trình.
6. Tiếp tục quy trình này cho đến khi hoàn tất chương trình.
7. Bạn hãy lưu chương trình lên đĩa, ghi lại, và in nếu cần.

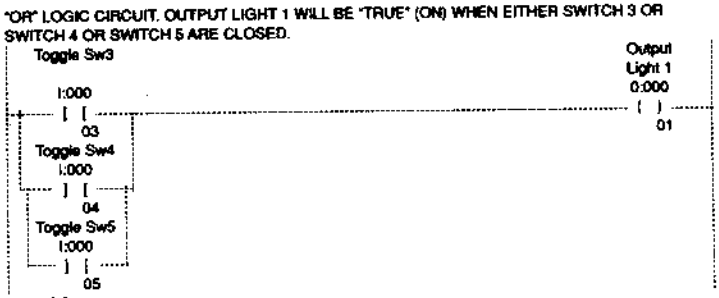
EXPERIMENT 6.6 LOGIC CIRCUITS

Rung 2:0
EXPERIMENT 6.6 LOGIC CIRUIT—ILLUSTRATING THE BASIC LOGIC FUNCTIONS.
OPTIONAL LAB
"AND" LOGIC CIRCUIT, OUTPUT LIGHT WILL BE "TRUE" (ON) ONLY WHEN SWITCH 0 AND SWITCH 1 AND SWITCH 2 ARE CLOSED.



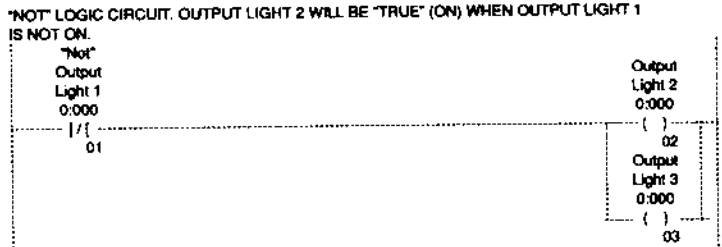
Rung 2:1

"OR" LOGIC CIRCUIT. OUTPUT LIGHT 1 WILL BE "TRUE" (ON) WHEN EITHER SWITCH 3 OR SWITCH 4 OR SWITCH 5 ARE CLOSED.



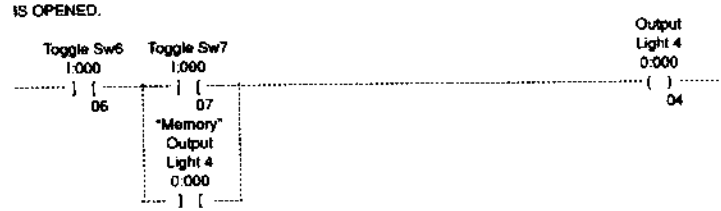
Rung 2:2

"NOT" LOGIC CIRCUIT. OUTPUT LIGHT 2 WILL BE "TRUE" (ON) WHEN OUTPUT LIGHT 1 IS NOT ON.



Rung 2:3

"MEMORY" WITH "AND/OR" LOGIC CIRCUIT. THE EXAMINE ON INSTRUCTION THAT IS ADDRESSED TO THE OUTPUT LIGHT 4 (0:000/04) REMAINS ON, EVEN AFTER TOGGLE SWITCH 7 IS OPENED.



Hình 6.6. Chương trình các hàm logic cơ bản.

Bài tập lớn 7.0

CHẠY VÀ CHẠY DỪNG

MỤC ĐÍCH

Viết chương trình sử dụng bit để điều khiển nút nhấn của bộ khởi động chạy liên tục hoặc chạy/dừng.

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng.

- Sử dụng lệnh bit nội/từ ngữ (B3/0)
- Sử dụng bit thay cho các thao tác relay.
- Lập trình các mạch chạy – chạy/dừng.
- Nghiên cứu các lệnh dịch chuyển và chỉnh sửa bit.

TÀI LIỆU

Để biết về các lệnh được dùng trong Bài tập này bạn nên xem Phụ Lục A: Các lệnh Chỉnh sửa/Dịch chuyển Bit và kiểu relay. XIC, XIO, OTE, BTD, MOV, và MVM.

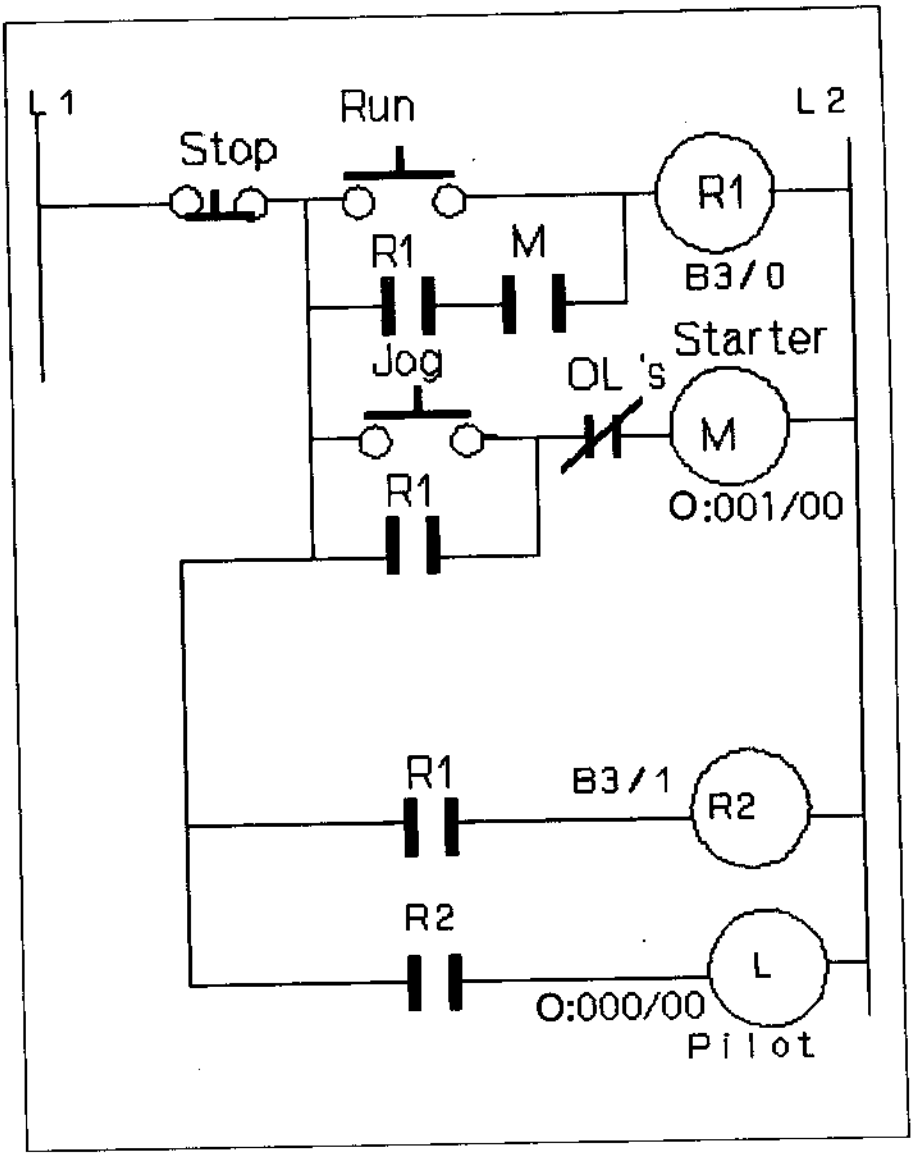
THẢO LUẬN

Chạy/dừng là thuật ngữ biểu thị sự khởi động và dừng động cơ thường xuyên trong lúc khoảng thời gian ngắn. Chạy/ dừng được dùng để định vị vật liệu theo các khoảng cách nhỏ mỗi khi động cơ khởi động. Chạy/dừng được thực hiện bằng cách loại bỏ hoặc rẽ mạch các tiếp điểm đóng mạch trong mạch điều khiển.

Trong Bài tập 6.0, Bạn sử dụng nút nhấn để điều khiển bộ khởi động chạy liên tục. Trong Bài tập này, bạn sẽ dùng nút nhấn để điều khiển bộ khởi động chạy/dừng và nút nhấn để điều khiển relay đóng mạch và chạy động cơ. Kiểu mạch này được gọi là relay chạy/dừng (Hình 7.1).

Trong lập trình PLC, có nhiều trường hợp các lệnh là nội đối với máy tính. Các lệnh này thực thi hàm logic nội nhưng không điều khiển ngõ ra một cách trực tiếp. Lệnh thực hiện điều này được gọi là *lệnh bit/từ ngữ*. Lệnh bit có thể được lập địa chỉ theo địa chỉ bit đơn (B3/0) hoặc địa chỉ từ ngữ 16-bit (B3:0).

Trên Hình 7.1, lệnh bit (B3/0) không điều khiển trực tiếp thiết bị xuất, chỉ được dùng trong chương trình một cách logic để điều khiển các lệnh khác. Trong trường hợp này, lệnh được dùng để đóng mạch, tương



Hình 7.1. Mạch chạy – chạy/dừng

tự relay điện – cơ và các tiếp điểm liên quan được dùng để đóng mạch điều khiển.

LOGIC CHƯƠNG TRÌNH

1. Lệnh nhập cho nút Stop chính thường đóng. Khi nhấn Stop, mạch sẽ ngắt và động cơ dừng.
2. Lệnh nhập cho nút RUN thường mở.
3. Lệnh nhập cho nút chạy dừng thường mở.
4. Lệnh nhập cho cuộn dây bộ khởi động. Bảo vệ quá tải mắc nối tiếp với cuộn này.
5. Lệnh nhập cho đèn báo, đóng mạch (ON) chỉ khi động cơ ở chế độ RUN.
6. Tín hiệu xuất lưu bit từ ngữ được dùng để đóng mạch động cơ
7. Tiếp điểm đóng mạch thực từ bộ khởi động, được coi là ngõ vào và mắc tiếp với bit nội đóng mạch.
8. Nhấn nút RUN sẽ đóng mạch động cơ cho đến khi nhấn nút Stop.
9. Nhấn nút chạy/dừng (Jog) chỉ cho phép động cơ chạy khi nhấn nút nhấn, nhả nút chạy/dừng (Jog) sẽ dừng động cơ.

CÁC BƯỚC THỰC HIỆN

1. Lập thư mục chương trình mới (PROJECT 7) và đặt tên tập tin là Run – jog, quan sát màn hình để chỉnh sửa tập tin chương trình.
2. Viết chương trình trên PLC cho mạch run/jog (Hình 7.1). Sử dụng các địa chỉ lệnh trên hệ thống của bạn. Tham khảo sổ tay dữ liệu của nhà sản xuất về các lệnh lập trình.
3. Quan sát lệnh xuất – ( )- (B3/0) trên thang thứ nhất. Địa chỉ tập tin bit này được dùng để thay cho cuộn relay trong sơ đồ mạch điện. Mục đích là cung cấp các lệnh -] [- đóng mạch phụ tương tự các tiếp điểm đóng mạch relay. Tập tin bit có thể được dùng để lập trình logic nội. Khi được coi là bit đơn, cần dùng vạch nghiêng thuận (B3/0), nếu được lập trình là từ ngữ 16-bit, bạn dùng dấu hai chấm (B3:0).
4. Chú ý, công tắc đóng mạch thực -| [- I: 000/02 vẫn được lập trình trong mạch để tránh vô ý khởi động khi chỉnh lại quá tải. Ngõ ra – ( ) – O:000/00 trong thang kế tiếp là địa chỉ bộ khởi động thực.

5. Khái niệm tiếp theo là tiếp điểm phụ (R) điều khiển relay thứ hai (R2). Trong nối kết dây, đây có thể là đèn và relay điện áp khác. Với PLC, chỉ cần sử dụng bit nội (B3/1) cho relay 2.

6. Sau khi hoàn tất các bước nêu trên, bạn có thể chấp nhận bậc thang này và kiểm tra chương trình. Trong chế độ TEST bạn chỉ thấy chương trình vận hành trên màn hình. Nếu mọi việc đều có vẻ hoạt động tốt trong chế độ kiểm tra (Test), bạn hãy chuyển sang chế độ RUN (chạy) và vận hành chương trình với các bộ khởi động (không cần nối với động cơ). Nếu chương trình hoạt động tốt, bạn hãy nối với động cơ và cho chạy thử.

7. Bước kế tiếp là ghi lại các lệnh và bậc thang. Bạn hãy đặt tên, chẳng hạn Project 7 Run/Jog, cho tập tin, sau đó lưu lên đĩa mềm.

8. Bước cuối cùng là lập báo cáo. Bạn ghi rõ tiêu đề báo cáo và in ra từ máy in.

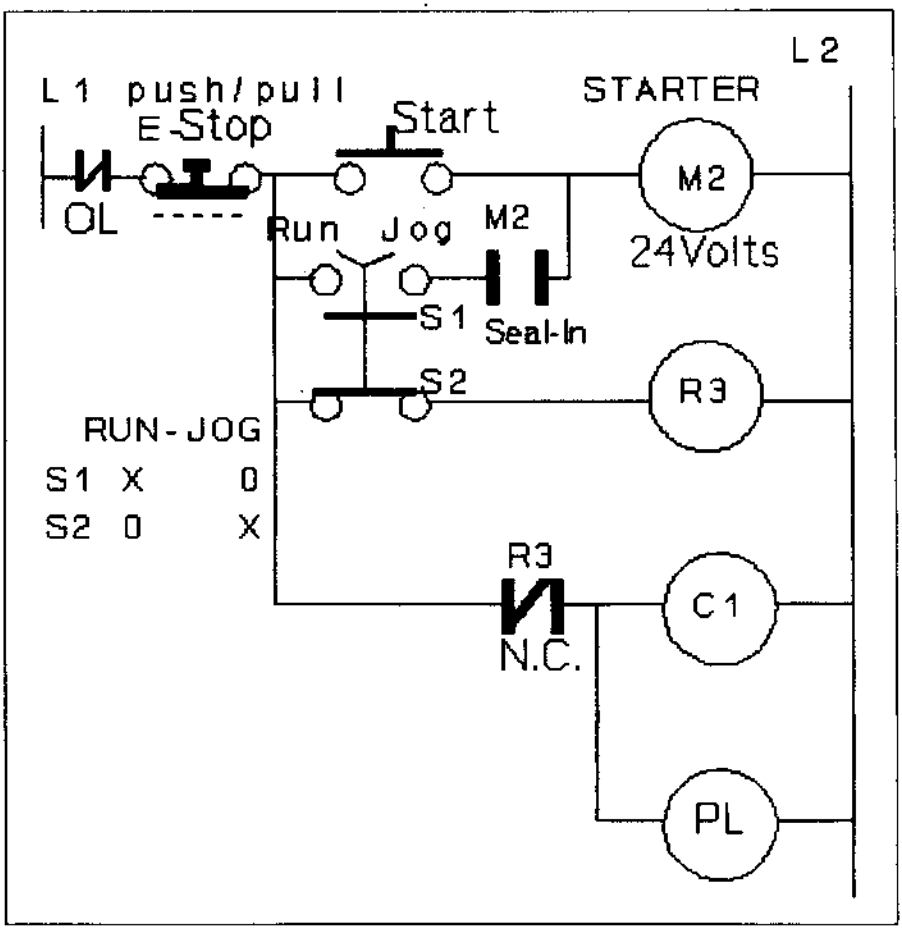
ÔN TẬP

Sau khi hoàn tất phần đầu của Bài Tập 7, bạn chuyển sang các thí nghiệm để củng cố và nâng cao kiến thức. Mỗi thí nghiệm đều cần ghi lại để làm tài liệu. Sau khi hoàn tất các thí nghiệm, bạn hãy lưu toàn bộ lên đĩa mềm.

Thí nghiệm 7.1. BỘ CHỌN CHẠY/DỪNG

Với các bậc thang riêng rẽ, bạn hãy tiếp tục lập trình mạch được nêu trên Hình 7.2. Bạn hãy dùng công tắc chọn 2 vị trí và nút nhấn Start tạm thời để điều khiển động cơ 2 theo thứ tự như sau:

1. Xoay bộ chọn sang trái và nhấn nút Start, động cơ 2 sẽ khởi động và chạy liên tục. Contactor chạy và đèn báo sáng liên tục cho đến khi nhấn nút E-stop.

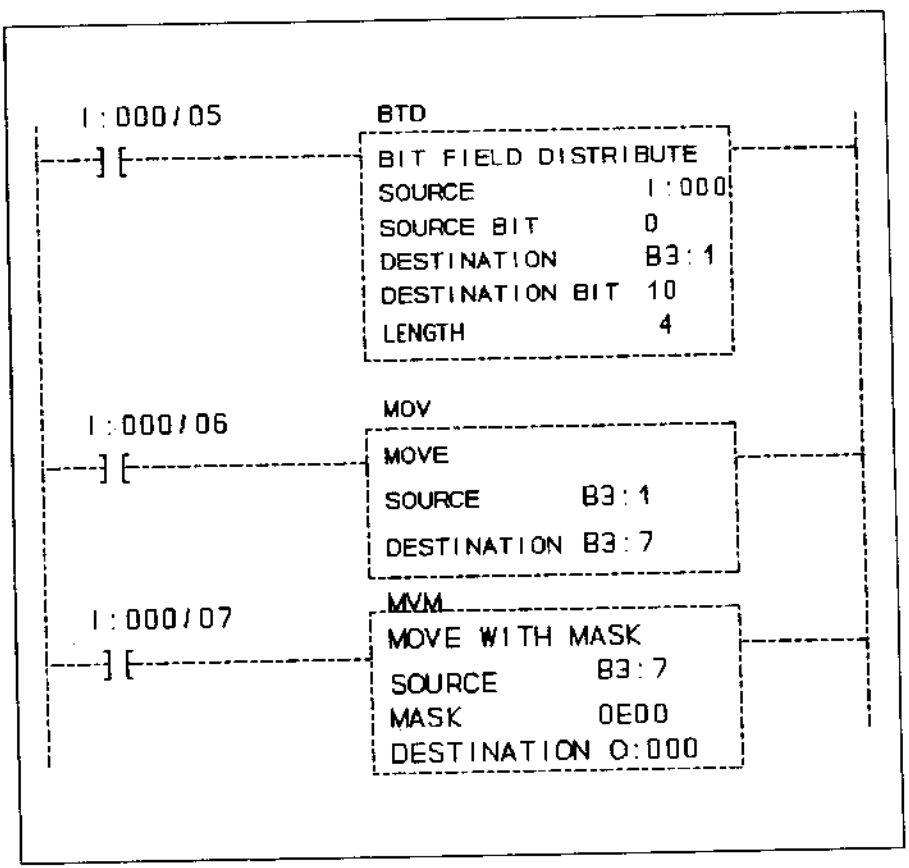


Hình 7.2. Mạch bộ chọn chạy bit

2. Xoay bộ chọn sang phải và nhấn nút Start, bộ khởi động của động cơ 2 sẽ chạy/dừng (jog). Contactor và đèn báo bị ngắt (OFF) ở vị trí này.

Thí nghiệm 7.2. DỊCH CHUYỂN BIT

Mục đích của thí nghiệm này là làm quen với quá trình dịch chuyển thông tin trong chương trình. Bạn hãy lập trình mạch trên Hình 7.3, sử dụng các lệnh phân phối trường bit (BTD), dịch chuyển (MOV), và dịch chuyển ẩn (MVM) để thực hiện chức năng dưới đây:



Hình 7.3. Các lệnh dịch chuyển bit.

CHÚ Ý: Bạn hãy kiểm tra lại các địa chỉ nhập/xuất, chúng có thể khác với sơ đồ trên Hình 7.3, bạn cần sử dụng các địa chỉ thích hợp.

Sau khi nhập chương trình này, bạn hãy chuyển sang chế độ RUN và quan sát sự dịch chuyển bit trên màn hình trong chế độ nhị phân. Bạn hãy kiểm tra chương trình như sau:

1. Thay đổi các bit nguồn (source) với các switch 00 đến 03. Dịch chuyển thông tin đó với switch 05. Kích hoạt switch 05 sẽ dịch chuyển bốn bit thấp của I:0 sang B3:1, bắt đầu từ bit 10.
2. Đóng switch 06 để dịch chuyển các bit ở B3:1 sang B3:7.
3. Kích hoạt switch 07 sẽ dịch chuyển các bit từ B3:7 sang các đèn ngõ ra O:000. Các đèn này chỉ vận hành khi giá trị (bit) ẩn cho phép. Trong trường hợp này, giá trị ẩn cho phép ba trong số các bit được lọc qua.
4. Tiếp tục quy trình này cho đến khi bạn nắm vững sự dịch chuyển bit trong chương trình.

Bài tập lớn 8.0

BĂNG TẢI BAO

MỤC ĐÍCH

Viết chương trình điều khiển thời gian ON/OFF của băng tải bao, sử dụng bộ thời chuẩn đóng TON.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Hiểu khái niệm về bộ thời chuẩn.
- Nghiên cứu các lệnh nhảy (JMP) và ghi nhận (LBL)
- Nâng cao kiến thức về phương pháp sử dụng bộ thời chuẩn để điều khiển quá trình.

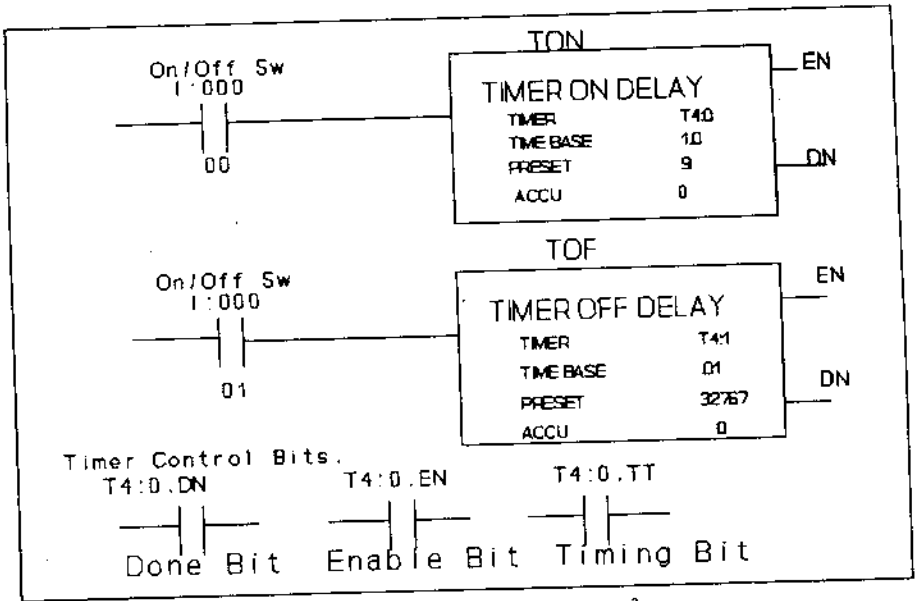
TÀI LIỆU

Để biết chi tiết các lệnh được dùng trong Bài tập này, bạn hãy tham khảo Phụ lục A, "Bộ thời chuẩn và các lệnh điều khiển chương trình" TO, TOF, RTO, JMP, và LBL.

THẢO LUẬN

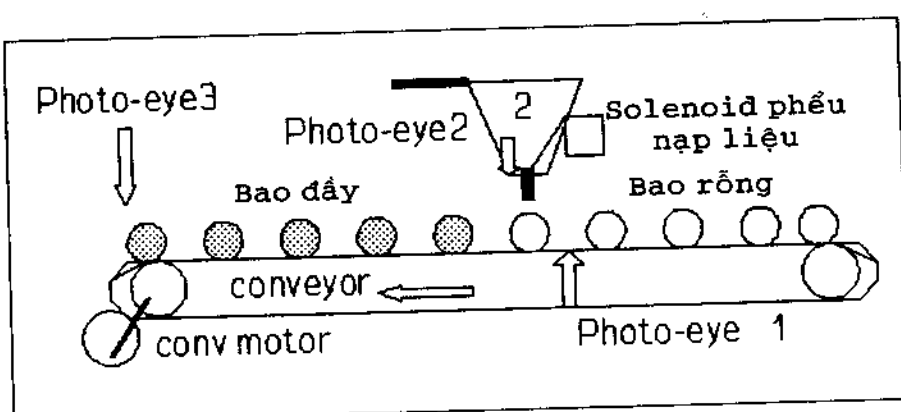
Các lệnh thời chuẩn là các lệnh xuất bên trong đối với bộ xử lý, chúng cung cấp nhiều khả năng tương tự các relay thời chuẩn và các linh kiện bán dẫn thời chuẩn/đếm. Thường được quy định theo các lệnh nhập, bộ thời chuẩn liên tục theo dõi các khoảng thời gian (Hình 8.1). Giá trị được xác định trước (PR) chuyên biệt số khoảng thời chuẩn cần đạt được trước khi bộ xử lý xác lập bit toàn tất (DN) và chương trình bậc thang bắt đầu thực thi tác vụ. Giá trị tích lũy (AC) là số lượng khoảng thời chuẩn trôi qua.

Lệnh thời chuẩn đóng (TON) có thể được dùng để kích hoạt thiết bị (ON) hoặt ngắt (OFF) sau khoảng thời gian xác định. Khi điều kiện thang là đúng, bộ thời chuẩn tăng dần giá trị tích lũy theo từng khoảng gia số. Khi giá trị tích lũy đạt giá trị cho trước, bộ thời chuẩn sẽ dừng và xác lập các bit điều khiển.



Hình 8.1. Các lệnh thời chuẩn

Lệnh thời chuẩn ngắt (TOF) khác với lệnh TON, bắt đầu tính thời gian ngay khi mạch thang là sai (OFF). Khi điều kiện thang là sai, lệnh TOF liên tục tính thời gian cho đến khi đạt được giá trị tích lũy cho trước, khi đó lệnh TOF xác lập các bit điều khiển.



Hình 8.2. Hệ thống băng tải đóng bao

Bài tập

Bài tập này sử dụng các bộ thời chuẩn và tế bào quang điện để điều khiển băng tải trong hệ thống đóng bao (Hình 8.2). Các bao rỗng dịch chuyển trên băng tải, khi đến vị trí thích hợp băng tải dừng và bắt đầu quá trình đóng bao. Sau khi đóng bao, băng tải khởi động và các bao tiếp tục chạy trên băng tải để đến công đoạn kế tiếp.

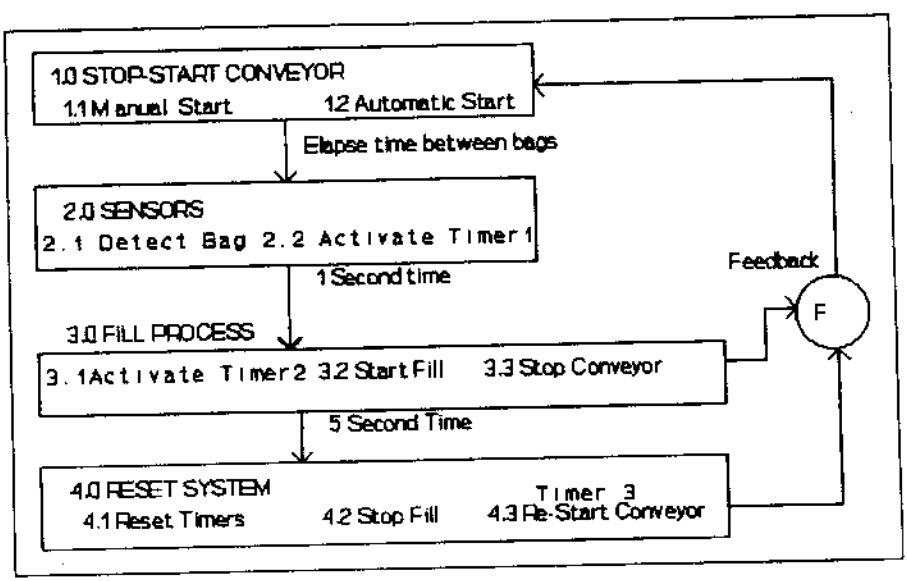
LOGIC CHƯƠNG TRÌNH

Bạn hãy xem các phần tử trên lưu đồ băng tải (Hình 8.3) để sắp xếp các bậc thang trong chương trình.

1. Phần tử 1.0 xác định phương pháp được dùng để khởi động băng tải.

   A. Sử dụng lệnh MCR vận hành bằng bộ chuyển mạch Toggle trên bậc thang thứ nhất. Kế tiếp, bạn dùng mạch Stop – Start với cùng bộ chuyển mạch ngõ ra của băng tải. Bạn dùng bộ điều khiển từ tính để vận hành băng tải.

   B. Sự khởi động tự động có thể là bit thời chuẩn thường mở (T4:3TT) mắc song song (OR) với tiếp điểm đóng của bộ khởi động.



Hình 8.3. Lưu đồ băng tải

2. Phần tử 2.0, tế bào quang điện, là bộ phận kế tiếp được kích hoạt khi bao thứ nhất đến khu vực đóng bao. Sử dụng bộ chuyển mạch Toggle để mô phỏng tế bào quang điện. Điều này sẽ kích hoạt bộ thời chuẩn đóng TON (T4:1) với giá trị xác lập sẵn là một giây, để bao đi vào đúng tâm của máng nạp liệu (đóng bao), bạn hãy dùng bit TT để đóng mạch bộ thời chuẩn.

3. Bước kế tiếp, Phần tử 3.0 có ba chức năng:

   A. Kích hoạt bộ thời chuẩn TON thứ hai T4:2

   B. Dừng băng tải

   C. Bắt đầu nạp liệu vào bao trong khoảng 5 giây, bảo đảm bao ở đúng vị trí.

4. Phần tử 4.0 là khoảng thời gian xác lập lại (reset). Khi bộ thời chuẩn T4: 2 dừng:

   A. Quá trình nạp liệu vào bao dừng lại.

   B. Các bộ thời chuẩn được xác lập lại.

   C. Băng tải tái khởi động và chạy cho đến khi bao kế tiếp đến khu vực tế bào quang. Bộ thời chuẩn T4: 3 được dùng để khởi động lại băng tải. Các bộ khởi động điện - cơ kích hoạt chậm hơn so với các lệnh PLC do đó cần có lượng thời gian tối thiểu để kích hoạt cuộn dây.

5. **Chú ý:** Có thể thêm tế bào quang điện để kích hoạt bộ đếm số lượng bao đã nạp liệu. Tùy chọn này có thể được bổ sung vào Bài tập 8.0 sau khi bạn nghiên cứu Bài tập 10 "Bộ đếm thuận/ngược". Bài tập này có thể thực hiện bằng cách sử dụng bộ mô phỏng, các đèn ngõ ra có thể được dùng cho selenoid và bộ khởi động của động cơ băng tải. Các nút nhấn có thể được dùng cho mạch Stop – Start và bộ chuyển mạch Toggle thay cho các điều khiển quang điện.

CÁC BƯỚC THỰC HIỆN

1. Viết chương trình trên PLC cho hệ thống đóng bao tự động như lưu đồ trên Hình 8.3 và hệ thống băng tải (Hình 8.2). Các bậc thang thời chuẩn được giải thích dưới đây. Bạn cần hoàn tất phần chương trình còn lại để mạch hoạt động.

2. Bước thứ nhất trong chương trình là mạch khởi động bằng tay và tự động Bậc thang đầu là lệnh MCR vận hành bằng công tắc

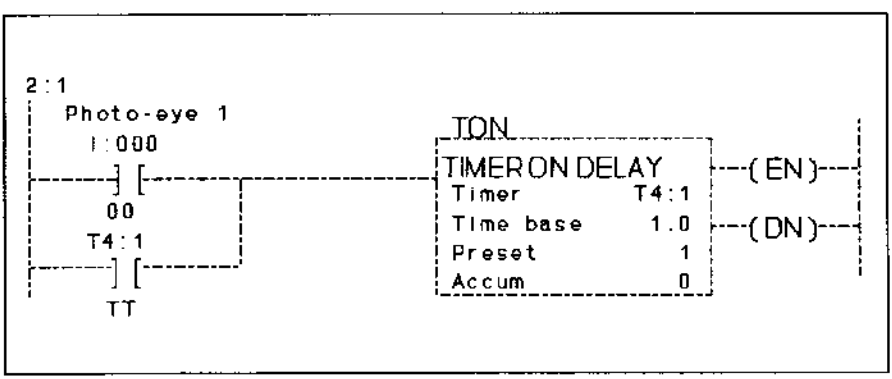
Stop. Bậc thang kế tiếp, sử dụng mạch nút nhấn Stop – Start để khởi động và dừng băng tải. Mạch này tương tự mạch trên Hình 6.2.

Công tắc Stop có cùng địa chỉ như công tắc nhập trên thang MCR. Dùng một trong các bộ khởi động và tiếp điểm đóng mạch để chạy băng tải.

3. Bước tiếp theo là quá trình nạp liệu vào bao. Khi các bao rỗng trên băng tải tiến đến vị trí tế bào quang điện sẽ xảy ra các thao tác như sau:

   Tế bào quang 1 kích hoạt bộ thời chuẩn đóng T4:1, sử dụng T4:1 TT bit của bộ thời chuẩn này song song với tế bào quang 1 để định chuẩn thời gian (Hình 8.4). Bộ thời chuẩn này phải bổ sung thêm một giây để bao đến đúng vị trí máng nạp liệu.

   Chú ý: Bạn có thể sử dụng thêm một tế bào quang để kích hoạt bộ thời chuẩn đếm ngược nếu bao bị kẹt ở phía trước tế bào quang.

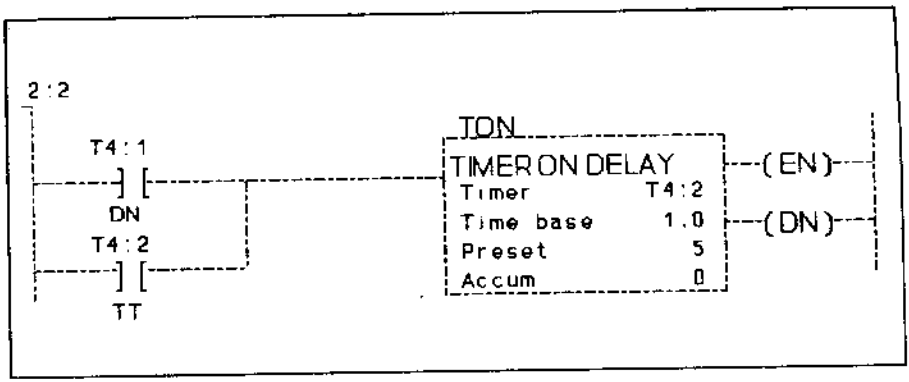


Hình 8.4. Thang thời chuẩn đưa bao vào vị trí đóng bao

4. Bộ thời chuẩn đóng thứ hai (T4:2) được kích hoạt bằng bộ T4:1. Done bit. Bạn đóng mạch bộ thời chuẩn này bằng TT bit. Thời gian xác lập trước là 5 giây (Hình 8.5). Bộ T4:2 điều khiển cuộn solenoid về thời gian đóng bao.

   Chú ý: Chèn T4:2.TT bit thường đóng từ bộ thời chuẩn này trong mạch khởi động băng tải để dừng băng tải trong khi nạp liệu vào bao.

Trong một thang riêng, bạn hãy dùng T4:2 EN bit để kích hoạt cuộn solenoid nạp liệu với thời gian nạp 5 giây. Trong thang này, tế bào quang



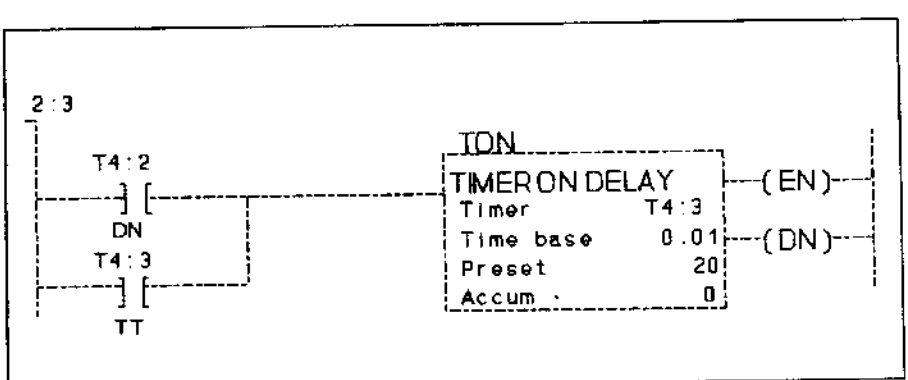
Hình 8.5. Thang bộ thời chuẩn nạp liệu

2 cần được chèn theo bộ giới hạn để tránh nạp liệu nếu bao không được định vị chính xác và T4:2 DN bit để trách nạp liệu lâu hơn thời gian cho trước.

5. Sau khi nạp liệu, bộ thời chuẩn thứ ba T4:3 được dùng để khởi động băng tải một cách tự động khi bao đã đầy. Chỉnh bộ thời chuẩn này theo 20 giây và xác lập cơ sở thời gian là 0.01 giây. Đóng mạch bộ này bằng T4:3.TT bit (Hình 8.6).

   Chú ý: Sử dụng T4:3. TT bit từ bộ thời chuẩn này song song với mạch nút nhấn khởi động để tái khởi động băng tải và lặp lại quá trình.

6. Sau khi hoàn tất các bước nêu trên, bạn hãy chạy chương trình. Khởi động chương trình với nút nhấn Start, thử đóng và mở công



Hình 8.6. Thang bộ thời chuẩn cài đặt lại

tắc tế bào quang điện 1, kiểm tra các bộ thời chuẩn để quan sát sự hoạt động của hệ thống. Thực hiện các bước trong logic chương trình để kiểm tra mạch. Nếu mọi thứ đều vận hành tốt, bạn hãy ghi lại các lệnh và lưu chương trình lên đĩa.

ÔN TẬP

Sau khi hoàn tất phần đầu của Bài tập 8, để củng cố và nâng cao kiến thức bạn hãy thực hiện các thí nghiệp kèm theo. Sau khi làm xong các thí nghiệm bạn hãy lưu toàn bộ chương trình lên đĩa mềm, bước cuối cùng là viết và in báo cáo.

Thí nghiệm 8.1.
THỨ TỰ ĐỘNG CƠ THEO THỜI CHUẨN

Bạn cần viết chương trình cho mạch khởi động lần lượt ba động cơ. Bạn hãy chỉnh sửa và lập trình mạch điện trong Thí nghiệm 4.2 đáp ứng các yêu cầu của Thí nghiệm 8.1. Nhấn nút Start sẽ cấp điện cho các bộ thời chuẩn #4, #5, #6 theo thứ tự như sau:

1. Bộ thời chuẩn ngắt (TOF) #4 kích hoạt bộ khởi động #1 để khởi động động cơ #1 hầu như tức thời.
2. Khi bộ thời chuẩn đóng (TON) #5 đạt thời gian cho trước sẽ cấp điện cho bộ khởi động #2 và bộ thời chuẩn #6.
3. Khi bộ thời chuẩn đóng (TON) #6 đạt thời gian cho trước, bộ khởi động #3 được cấp điện để khởi động động cơ #3, hoàn tất chuỗi thứ tự.
4. Khi cả ba động cơ đều chạy, nhấn nút Stop sẽ làm cho:

   A. Các động cơ #2 và #3 dừng

   B. Động cơ #1 tiếp tục chạy trong 7 giây.

Sau khi động cơ #1 dừng, cả ba động cơ đều được ngắt điện nguồn.

Chú ý: Trong các thí nghiệm ở đây, bạn hãy tiếp tục từ bộ thời chuẩn cuối trong Bài tập này và lập địa chỉ cho các bộ thời chuẩn theo thứ tự tăng dần. Đặt các lệnh nhảy (JMP) đến nhãn (LBL) như trên Hình 8.7 bao quanh sơ đồ mạch của Thí nghiệm 8.1. Điều đó cho phép bạn tạo một vùng xung quanh Thí nghiệm này

```
        Toggle Sw                          JMP
-------] [-----------------------------( )-------
          7                                1
        LBL                            Storage Bit
-------] [-----------------------------( )-------
          1                              B3/7
```

Hình 8.7. Các lệnh Jump (JUP) to Label (LBL)

và tái sử dụng các nút nhấn. Khi điều kiện thang đối với ngõ ra JMP là đúng, bộ xử lý chuyển đến lệnh LBL tương ứng và tiếp tục thực thi chương trình từ nhãn đó.

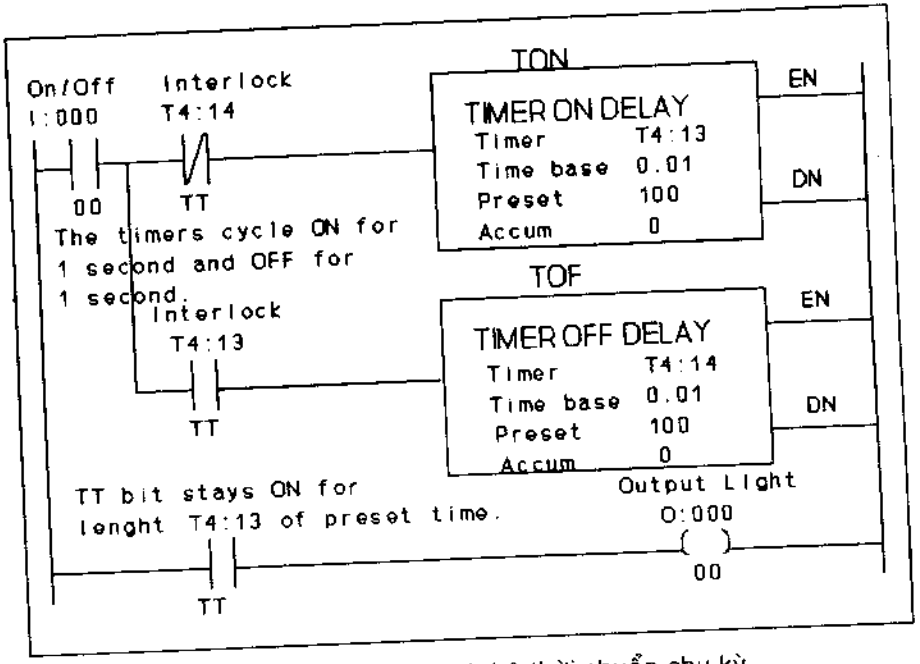
Thí nghiệm 8.2.
MẠCH THỜI CHUẨN DẬP CẮT

Máy dập cắt có mạch an toàn do công nhân phải dùng cả hai tay để vận hành máy. Bạn hãy chỉnh sửa và lập trình mạch được nêu trên Hình 4.4 để đáp ứng các yêu cầu của thí nghiệm này.

1. Hai nút thời chuẩn TON điều khiển thời lượng của từng nút nhấn trong máy dập cắt.
2. Hai bộ thời chuẩn TON điều khiển thời lượng của từng nút nhấn khi được nhấn.
3. Đặt lệnh JMP to LBL bao quanh thí nghiệm này tương tự Thí nghiệm 8.1.

Thí nghiệm 8.3. BỘ THỜI CHUẨN CHU KỲ

Bạn hãy lập trình cho mạch được nêu trên Hình 8.8. Mạch sử dụng các bộ thời chuẩn TON/TOF để vận hành theo kiểu dao động (liên tục lặp lại). Bạn hãy chạy chương trình và quan sát các giá trị tích lũy thời chuẩn. Chúng phải thay đổi các giá trị tích lũy 0 đến 100. Đèn nhấp nháy theo chu kỳ sáng trong 1 giây và tắt trong 1 giây.



Hình 8.8. Chương trình bộ thời chuẩn chu kỳ

Sau khi hoàn tất và kiểm tra mạch này, bạn hãy nâng cấp và chỉnh sửa, chọn và sao mạch dao động vào tập tin dán (paste), đặt tên cho tập tin là OSC.

Thí nghiệm 8.4. KHỞI ĐỘNG KHÔNG THEO THỨ TỰ

Bạn hãy thiết kế và lập trình mạch vận hành ba bộ khởi động. Mỗi bộ khởi động đều có nút nhấn Stop – Start riêng. Do dòng điện khởi động cao và các yêu cầu của máy, mạch cần vận hành như sau:

1. Nhấn nút Start bất kỳ sẽ khởi động bộ khởi động nối với nút Start đó.
2. Khi bộ khởi động được áp nguồn (ON), các bộ khởi động còn lại không thể khởi động trong khoảng thời gian được xác lập sẵn ở bộ thời chuẩn.
3. Khi khoảng thời gian xác lập sẵn trôi qua, một trong hai động cơ còn lại có thể khởi động.
4. Khi động cơ thứ hai khởi động, động cơ thứ ba không khởi động trong khoảng thời gian cho trước.
5. Sau khoảng thời gian định trước, động cơ thứ ba có thể khởi động. Cả ba bộ khởi động đều được cấp nguồn (ON).
6. Giữa hai lần khởi động bất kỳ phải có khoảng thời gian xác định, bất kể thứ tự các bộ khởi động được khởi động.

Bài tập lớn 9.0

THUẬN ĐẢO (TOF)

MỤC ĐÍCH

Lập trình các lệnh thời chuẩn ngắt (TOF) để tránh các bộ khởi động thuận/ đảo tái khởi động theo chiều ngược lại trong khoảng thời gian cho trước.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Lập trình các lệnh thời chuẩn ngắt (TOF) với bộ khởi động thuận/ đảo.
- Nghiên cứu các lệnh JMP và LBL.

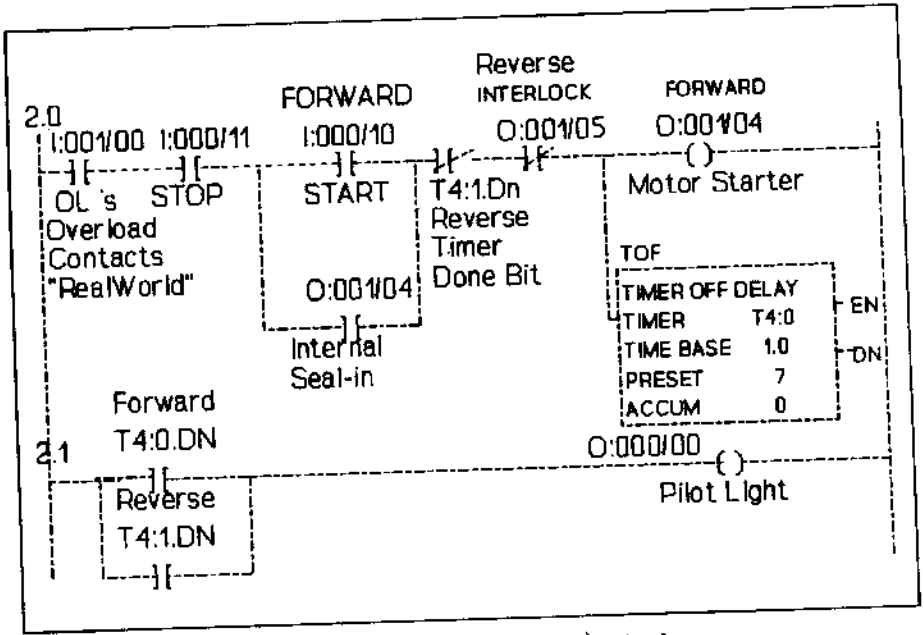
TÀI LIỆU

Để biết chi tiết các lệnh được dùng trong bài tập này, bạn hãy đọc Phụ lục A "Các lệnh điều khiển chương trình và thời chuẩn", TON, TOF, JMP, LBL.

THẢO LUẬN

Trong nhiều trường hợp, do bản chất của trang thiết bị, không được phép đảo chiều động cơ quá nhanh. Bài tập này sử dụng hai điều khiển thời chuẩn ngắt (TOF) theo thời lượng động cơ bị ngắt mạch (OFF) sau khi dừng. Mạch ngăn chặn sự tái khởi động theo chiều ngược trong khoảng thời gian cho trước.

Hình 9.1 minh họa thời chuẩn ngắt (TOF) được dùng để ngăn chặn động cơ tái khởi động theo chiều ngược quá nhanh. Việc dừng động cơ làm cho thang thuận trở thành sai (OFF). Các lệnh thời chuẩn ngắt (TOF) bắt đầu tính thời gian ngay khi điều kiện thang này là sai. Khi điều kiện thang là sai, TOF tiếp tục tính thời gian cho đến khi giá trị tích lũy bằng giá trị cho trước. Trong khoảng thời gian định trước, mạch điều khiển ngăn cản sự tái khởi động theo chiều ngược trong khoảng thời gian xác định.



Hình 9.1. Chương trình chiều thuận.

Bài tập

Mục đích của bài tập này là lập trình bộ khởi động thuận và đảo sử dụng lệnh thời chuẩn ngắt (TOF) để khóa liên động chiều ngược, không cho phép khởi động sau khi dừng. Khoảng thời gian trôi qua sau khi dừng từ một chiều quay, trước khi tái khởi động theo chiều ngược lại. Bài tập 4, Hình 4.3 là tương đương điện – cơ của chương trình này.

Kiểu lệnh thứ hai được dùng trong Bài tập này là lệnh nhảy (Jump) đến nhãn (Label). Sự nhảy đến nhãn sẽ tránh thời gian quét bằng cách bỏ qua phần của chương trình cho đến khi cần thiết. Điều này cho phép bạn sử dụng cùng các công tắc trong hai vị trí khác nhau.

LOGIC CHƯƠNG TRÌNH

1. Lệnh nhập dùng cho nút Stop thường đóng để dừng động cơ.
2. Lệnh nhập dùng cho nút Start thường mở để khởi động thuận và đảo.
3. Các lệnh nội dùng cho khóa liên động điện thường đóng để tránh chạy đồng thời cả hai chiều.

4. Bảo vệ quá tải cho cả chiều thuận và ngược.
5. Đèn báo, cho biết động cơ có thể tái khởi động khi đèn tắt (OFF).

CÁC BƯỚC THỰC HIỆN

1. Lập thư mục chương trình, đặt tên là PROJECT 9 và tên tập tin là forward reverse. Viết chương trình trên PLC cho mạch thuận/đảo, sử dụng các địa chỉ thích hợp.
2. Bước thứ nhất, nhập chương trình được nêu trên Hình 9.1. Thang thứ nhất là các điều khiển chỉ theo chiều thuận. Thang thứ hai là các điều khiển để kích hoạt đèn báo theo chiều thuận hoặc ngược. Đèn này sáng khi bộ khởi động không được kích hoạt.
3. Kế tiếp, nhập bản sao của bậc thang thứ nhất (2.0) để điều khiển theo chiều ngược, khác biệt duy nhất là các địa chỉ tương ứng cho mọi lệnh trong bậc thang chiều ngược.
4. Bạn hãy chú ý các bit chặn –[/]- T4:0. DN và T4:1.DN. Các bit này sẽ chặn mạch và không cho chạy theo chiều ngược lại trong khoảng thời gian được xác định sẵn sau khi dừng động cơ.
5. Kỹ thuật thứ hai trong Bài tập là các khóa liên động –[/]- O:001/04 và O:001/05 được đưa vào bậc thang thuận và ngược, bảo đảm ngăn chặn động cơ đồng thời chạy theo cả hai chiều.

```
Toggle Switch JMP

----------]    [----------------------(     )----------

              00                              1

PROJECT 9. FOR/REV, OFF-DELAY.

LBL Storage Bit

----------]    [----------------------(     )----------

              1                               B3/10
```

Hình 9.2. Các lệnh Jump (nhảy) to Label (đến nhãn)

6. Bước tiếp theo là đặt vùng Jump to Label (Hình 9.2) bao quanh chương trình. Điều này cho phép bạn bỏ qua phần này của chương trình khi cần. Khi bậc thang có lệnh JMP là đúng, bộ xử lý nhảy qua các bậc thang để đến bậc thang chứa lệnh LBL với cùng số. Bộ xử lý thực thi chương trình từ bậc thang này.

7. Sau khi hoàn tất các bước nêu trên (đừng quên lưu lại) bạn hãy kiểm tra chương trình với các bộ khởi động. Nếu mọi thứ đều hoạt động chuẩn xác, bạn có thể cho chạy với động cơ. Khi chạy chương trình trình, nhấn nút Stop sẽ ngăn chặn động cơ tái khởi động theo chiều ngược trong 7 giây.

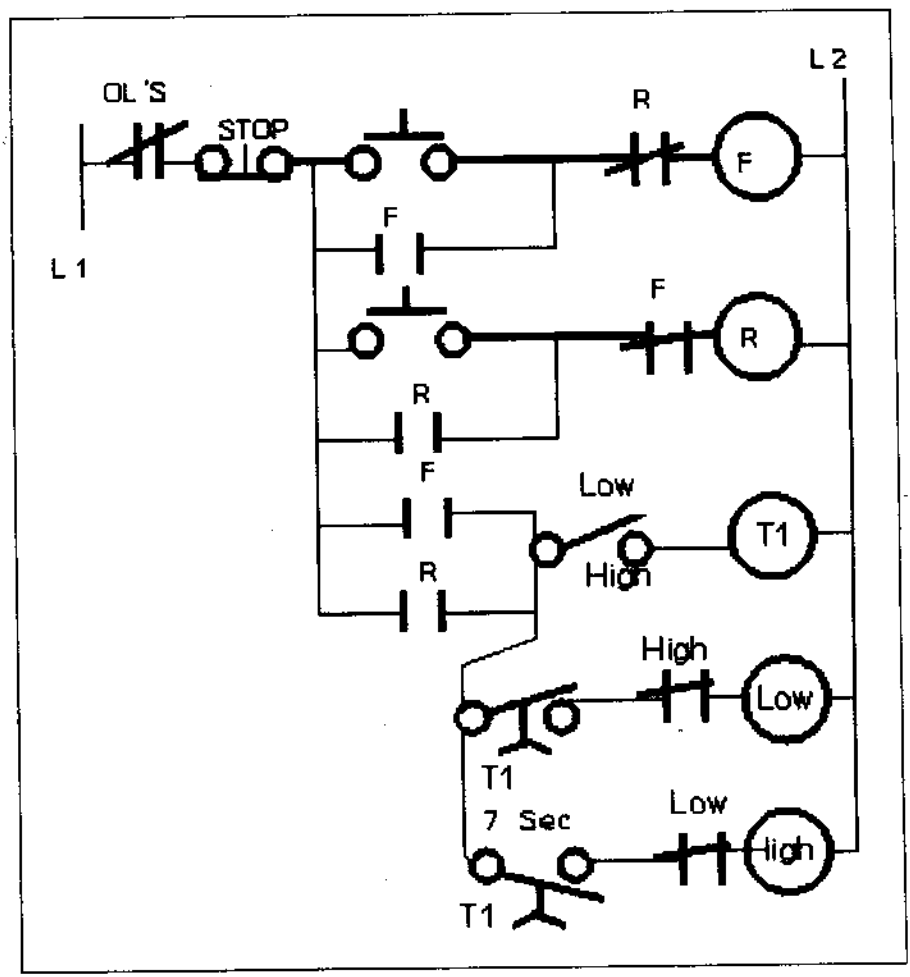
ÔN TẬP

Sau khi xong phần đầu của Bài tập 9.0, bạn hãy tiến hành các thí nghiệm, viết và in chương trình (nhớ lưu lên đĩa).

Thí nghiệm 9.1. THUẬN/ĐẢO VÀ THẤP – CAO

Bạn hãy chuyển mạnh điều khiển relay trên Hình 9.3 thành chương trình PLC. Mạch này điều khiển động cơ dây quấn riêng rẽ, hai tốc độ, 3-pha. Mạch vận hành như sau:

1. Nút Stop và nút nhấn thuận/đảo điều khiển chiều của động cơ.
2. Công tắc Toggle đơn cực điều khiển sự chọn tốc độ thấp và cao của động cơ.



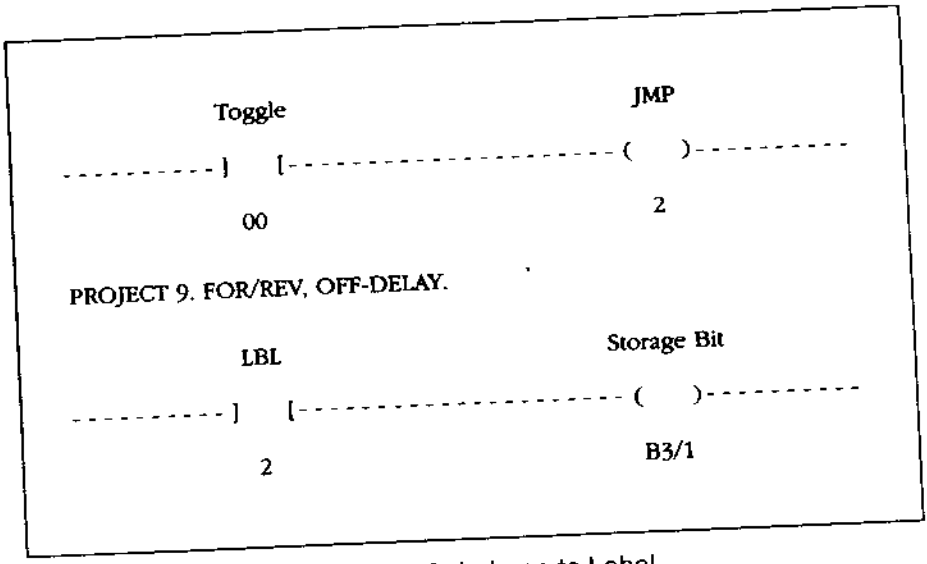
Hình 9.3. Mạch Thuận/Đảo Thấp – Cao

3. Nhấn nút nhấn thuận hoặc đảo luôn luôn điều khiển với tốc độ thấp. Khi chọn tốc độ, động cơ điều khiển và duy trì tốc độ thấp.

4. Chọn tốc độ cao, động cơ khởi động với tốc độ thấp, sau khoảng thời gian định trước, động cơ tự động chuyển sang tốc độ cao.

5. Đặt vùng JMP to LBL (Hình 9.4) bao quanh thí nghiệm này.

6. Bạn hãy dùng cùng địa chỉ thời chuẩn như của bộ thời chuẩn trong Bài tập này để kiểm tra lệnh Jump to Label. Sau khi hoàn tất, bạn hãy chạy mạch này và kiểm tra lệnh Jump to Label. Công tắc Toggle (00) sẽ điều khiển Bài tập 9.0 hoặc thí nghiệm 9.1. Khi chương trình này vận hành, chương trình trong Bài tập 9.0 sẽ không hoạt động.



Hình 9.4. Lệnh Jump to Label.

Bài tập lớn 10. 0

BỘ ĐẾM THUẬN/NGƯỢC

MỤC ĐÍCH

Viết chương trình sử dụng lệnh bộ đếm Up/Dn (thuận/ngược) để đếm và loại bỏ số nguyên các bộ phận trong hệ thống xử lý.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Hiểu khái niệm bộ đếm Up/Dn
- Hiểu và sử dụng các lệnh so sánh.
- Sử dụng số nguyên với lệnh BCD (TOD)

TÀI LIỆU

Bạn hãy tham khảo Phụ lục A: "Các lệnh thời chuẩn/đếm, so sánh và chuyển đổi", CTU, CTD, RES, TON, RTO, EQU, LEQ, LES, và TOD.

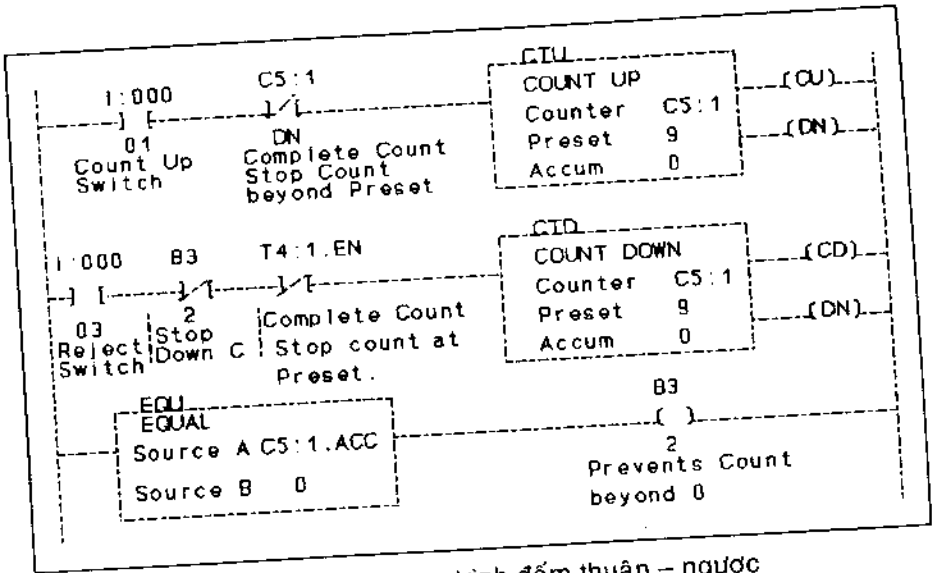
THẢO LUẬN

Lệnh đếm ngược có thể đi đôi với lệnh đếm thuận để tạo thành bộ đếm Up/Dn (thuận/ngược). Trừ khi logic chương trình được dùng để ngăn chặn đếm xuôi vượt quá giá trị cho trước, chúng sẽ tiếp tục đếm vượt quá giá trị này. Sử dụng (25:1.DN) done bit thường đóng trong bậc thang đếm thuận sẽ ngăn cản bộ đếm vượt quá giá trị cho trước. (Hình 10.1).

Tương tự, để dừng đếm ngược, cần dùng bit lưu thường đóng (B3/2). Bậc thang riêng với lệnh nhập bằng zero sẽ kích hoạt bit xuất (B3/2) để dừng đếm ngược ở zero. Bit (T4:1.EN) được dùng trong đếm ngược để ngăn chặn đếm ngược tiếp theo khi giá trị tích lũy đếm thuận bằng giá trị cho trước.

Bài tập

Trong Bài tập này bạn sẽ lập trình bộ đếm Up/Dn được dùng để đếm số lượng sản phẩm với bộ đếm thuận. Bộ đếm ngược sẽ giảm số đếm khi có phế phẩm bị loại bỏ.



Hình 10.1. Chương trình đếm thuận – ngược

Hai bộ thời chuẩn được dùng trong mạch:

1. Thứ nhất là bộ thời chuẩn TON được dùng để cài đặt lại hệ thống khi đếm xong. Bộ này sẽ làm trễ sự cài đặt lại theo thời lượng cho trước.
2. Thứ hai là bộ thời chuẩn ngăn chặn, được dùng để kích hoạt hệ thống cảnh báo nếu bộ đếm vượt quá thời lượng cho trước ở bộ thời chuẩn. Số lượng sản phẩm được xác định trước phải hoàn tất trong khung thời gian định sẵn.

LOGIC CHƯƠNG TRÌNH

1. Lệnh CTU đếm số lượng sản phẩm hoàn tất từ quy trình này.
2. Lệnh CTD đếm số lượng sản phẩm bị loại bỏ.
3. Lệnh Equal (bằng) ngăn chặn đếm ngược khi đạt đến zero.
4. Lệnh TON cài đặt lại quy trình sau 5 giây.
5. Bộ thời chuẩn ngăn chặn, được dùng để theo dõi thời gian chế tạo sản phẩm. Lệnh lớn hơn hoặc bằng (GEQ) điều khiển bộ thời chuẩn ngăn chặn.

6. Bộ cảnh báo thời gian quá mức để hoàn tất các sản phẩm.

7. Chuỗi thứ tự các sự kiện:

   A. Bộ đếm thuận đếm 9 sản phẩm hoàn tất và bộ đếm ngược đếm các phế phẩm bị loại bỏ.

   B. Sau 9 sản phẩm hoàn tất, hệ thống dừng trong 5 giây và cài đặt lại.

   C. Bộ thời chuẩn ngăn chặn sẽ kích hoạt hệ thống cảnh báo khi thời gian xử lý các sản phẩm hoàn tất quá dài.

CÁC BƯỚC THỰC HIỆN

1. Nhập chương trình mạch Up/Dn vào PLC (Hình 10.1). Mục đích của phần thứ nhất trong Bài tập này là xem xét và kiểm chứng sự vận hành của bộ đếm Up/Dn.

2. Chuyển sang chế độ RUN và kiểm tra mạch. Bạn có khả năng đếm thuận đến giá trị định trước là 9 và không vượt qua giá trị này. Sau đó bạn hãy thử đếm ngược đến zero, chương trình phải không cho phép đếm dưới zero.

3. Kế tiếp, bạn nhập các bậc thang dưới đây.

   A. Đối với bậc thang kế, bạn hãy nhập lệnh nhập (C5:1. DN) để điều khiển bộ thời chuẩn TON (T4:1), và xác lập thời gian 5 giây.

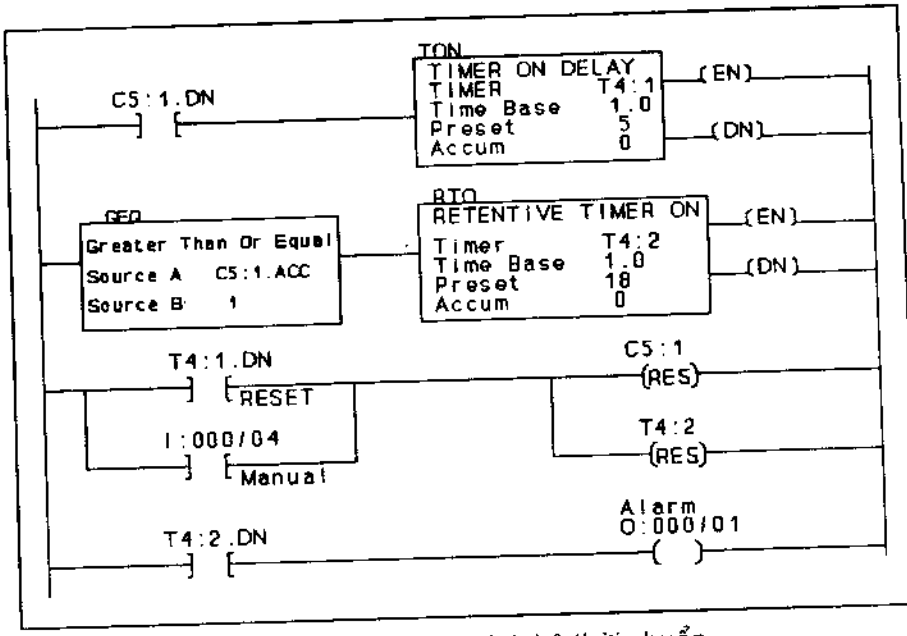
   B. Nhập lệnh lớn hơn hoặc bằng (GEQ) để điều khiển bộ thời chuẩn ngăn chặn (T4:2).

   C. Bạn dùng done bit (T4:1.DN) của bộ thời chuẩn để cài đặt lại bộ đếm và bộ thời chuẩn ngăn chặn, sau đó lập trình cài đặt lại bằng tay trên bậc thang này.

   D. Done bit (T4:2) của bộ thời chuẩn ngăn chặn, được dùng để kích hoạt bộ cảnh báo.

   E. Khi hoàn tất, phần thứ hai của chương trình sẽ có dạng như trên Hình 10.2.

4. Sau khi nhập chương trình, bạn hãy chuyển sang chế độ RUN và tiến hành kiểm tra, bạn hãy thử công tắc đếm thuận. Công tắc đếm ngược mỗi lần sẽ giảm số đếm là 1 khi bạn đóng hoặc mở công tắc này.



Hình 10.2. Chương trình bộ thời chuẩn.

5. Bạn hãy lưu ý lệnh lớn hơn hoặc bằng (GEQ). Lệnh này kích hoạt bộ thời chuẩn ngăn chặn ở lần đếm thứ nhất của bộ đếm Up/Dn. Mục đích là duy trì theo dõi thời lượng xử lý số lượng sản phẩm này. Nếu thời lượng quá dài trước khi đến số 6, bộ cảnh báo sẽ được kích hoạt.
6. Quy trình được cài đặt lại một cách tự động khi đạt đến số đếm cài đặt trước. Bạn hãy quan sát sự cài đặt lại bằng tay, điều này cho phép bạn cài đặt lại hệ thống trong trường hợp mạch có sự cố.
7. Mạch này cho phép bạn nghiên cứu khái niệm về bộ đếm Up/Dn, các bộ thời chuẩn và các lệnh so sánh. Khi bạn hoàn tất và kiểm tra xong mạch này, bạn hãy chuyển sang chế độ RUN và vận hành chương trình, sau đó ghi lại và lưu.

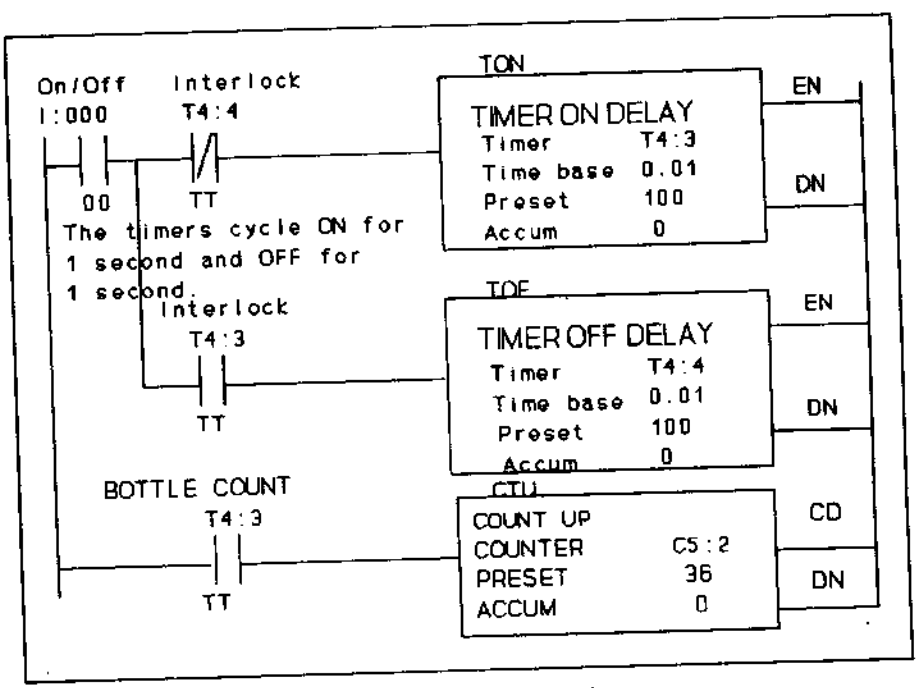
ÔN TẬP

Sau khi hoàn thành phần đầu của Bài tập, bạn hãy sử dụng kiến thức đã học để thực hiện các thí nghiệm kèm theo. sau khi hoàn tất các thí nghiệm bạn nên lưu lại và viết báo cáo hoàn chỉnh.

Thí nghiệm 10.1. BỘ ĐẾM CHAI 1

Bạn có thể thiết kế lại một phần của chương trình hiện hữu trong phân xưởng đóng chai. Hình 10.3 là phần thiết kế lại. Bạn hãy chỉnh sửa phần chương trình này theo các thông số sau:

1. Sao chép bộ thời chuẩn chu kỳ, dán tập tin OSC từ Thí nghiệm 8.3 để mô phỏng bộ đếm chai tự động.
2. Lập trình nhỏ hơn hoặc bằng 11 để hiển thị đèn báo chớp-nháy.
3. Lập trình giữa hoặc bằng 12 đến 24 để hiển thị đèn báo màu vàng. (Sử dụng bit điều khiển từ bộ thời chuẩn chu kỳ để chớp nháy đèn báo).
4. Lập trình giữa hoặc bằng 25 đến 35 để hiển thị đèn báo màu đỏ.
5. Khi số đếm bằng 36, tạm dừng hệ thống trong 10 giây. Sau chu kỳ 10 giây, cài đặt lại và khởi động lại hệ thống.
6. Đếm tổng số các thùng (mỗi thùng 36 chai).

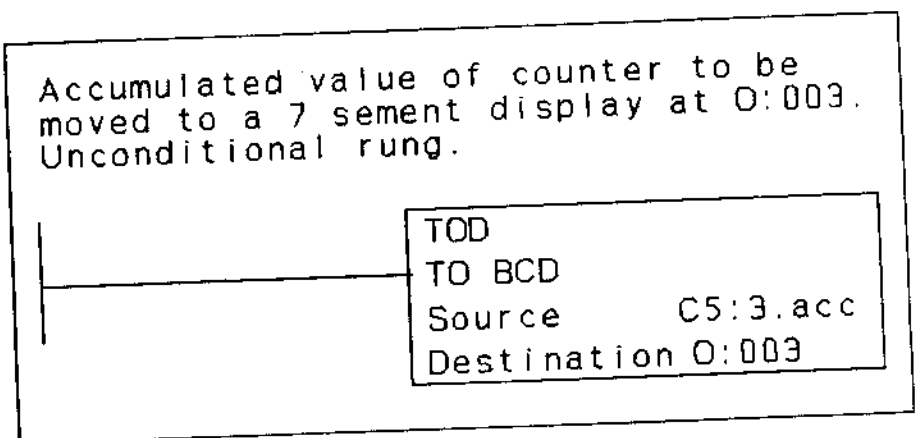


Hình 10.3. Chương trình đếm chai.

Thí nghiệm 10.2. BỘ ĐẾM CHAI 2

Bạn cần thiết kế và lập trình mạch điều khiển trong phân xưởng đóng chai với các yêu cầu như sau:

1. Hệ thống đếm, yêu cầu đếm từng chai và các bộ 6-chai. Các bộ thời chuẩn chu kỳ được dùng để mô phỏng số đếm chai. Bạn hãy dùng bộ đếm thuận (tăng) để đếm đến 24 chai riêng rẽ. Bộ đếm ngược (giảm) đếm từng bộ 6-chai (gồm 4 bộ trong một chu kỳ).
2. Dừng hệ thống trong 5 giây khi đếm ngược (giảm) đến zero (gồm 4 bộ, mỗi bộ 6 chai). Khởi động lại và lặp lại quy trình sau 5 giây.
3. Cài đặt lại và lặp lại (sử dụng lệnh Move để cài đặt giá trị tích lũy bộ đếm ngược trở lại giá trị cài đặt sẵn là 4).
4. Hiển thị kết quả bộ đếm ngược trên hiển thị 7-đoạn. Bạn hãy xem lệnh TOD trên Hình 10.4. Lệnh này chuyển giá trị nhị phân thành giá trị BCD.



Hình 10.4. Lệnh Integer to BCD.

Bài tập lớn 11. 0

THUẬN/ĐẢO (THEO THỨ TỰ)

MỤC ĐÍCH

Lập trình các lệnh kiểm tra giới hạn sử dụng các giá trị thời gian được xác lập trước để điều khiển các chiều thuận và đảo của bộ khởi động.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Lập trình các thông số cửa sổ sử dụng lệnh kiểm tra giới hạn.
- Sử dụng lệnh thời chuẩn với lệnh kiểm tra giới hạn.

TÀI LIỆU

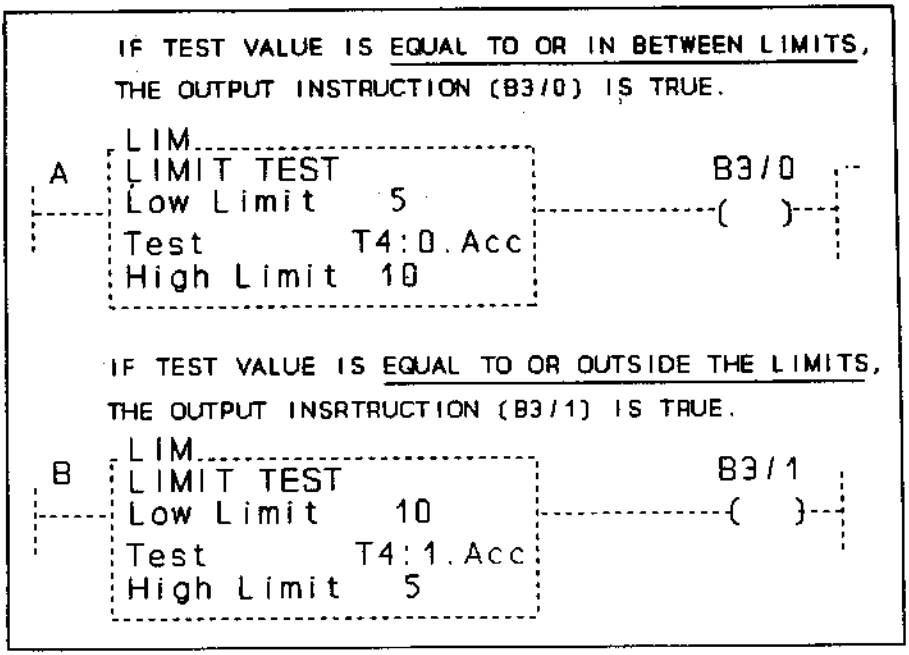
Bạn hãy xem Phụ lục A: Các lệnh thời chuẩn và so sánh TON, TOF, RTO, EQU, LIM.

THẢO LUẬN

Lệnh kiểm tra giới hạn

Một trong các lệnh đa năng nhất trong lập trình PLC là lệnh kiểm tra giới hạn. Lệnh này có thể được dùng trong nhiều chương trình cần có khoảng xác định các giá trị trên và dưới. Lệnh giới hạn có thể được coi là lệnh cửa sổ (giá trị giới hạn trên và dưới). Có hai phương pháp lập trình cơ bản đối với lệnh kiểm tra giới hạn. Phương pháp thứ nhất là tạo cửa sổ làm cho bậc thang là đúng trong các giá trị giới hạn dưới và trên. Điều này được thực hiện bằng cách nhập số nhỏ vào giới hạn dưới và số lớn vào giới hạn trên. (Thang A trên Hình 11.1). Bậc thang kiểm tra giới hạn là đúng khi giá trị trong giá trị tích lũy thời chuẩn (T4:0.Acc) bằng hoặc giữa các giới hạn này.

Phương pháp lập trình thứ hai là nhập giá trị lớn vào giới hạn dưới khi so sánh với giá trị giới hạn trên. Khi giá trị kiểm tra (thời chuẩn T4:1.Acc) bằng hoặc bên ngoài các giá trị giới hạn, bậc thang là đúng. (Thang B trên Hình 11.1). Giới hạn dưới được xác lập là 10 và giới hạn trên là 5. Điều này tạo ra cửa sổ làm cho bậc thang bị sai khi giá trị kiểm tra trong phạm vi các xác lập giới hạn.



Hình 11.1. Các lệnh kiểm tra giới hạn.

Bài tập

Trong Bài Tập này bạn sẽ lập trình bộ khởi động của động cơ ba – pha chạy theo chiều thuận trong khoảng thời gian được xác lập trước. Sau khi khoảng thời gian này trôi qua, động cơ đảo chiều và chạy theo chiều ngược trong thời lượng xác định được lập sẵn. Lệnh thời chuẩn TON và hai lệnh kiểm tra giới hạn được dùng để điều khiển các bộ khởi động thuận và đảo. Các bộ khởi động chỉ vận hành trong khoảng thời gian do lệnh giới hạn xác lập.

Do mạch này đóng (ON) và ngắt (OFF) một cách tự động, việc sử dụng tiếp điểm thực trong mạch là không thích hợp, chỉ cần mắc các tiếp điểm quá tải của bộ khởi động ở ngõ vào và lập trình theo các giới hạn trong mạch.

> **Chú ý:** Khái niệm về sử dụng khung giá ở xa được giới thiệu trong Bài tập này. Các địa chỉ phản ánh sự sử dụng số khung giá xa là 1. Nếu không có khung giá xa, bạn có thể dùng các địa chỉ khung giá của bộ mô phỏng tại chỗ.

LOGIC CHƯƠNG TRÌNH

1. Các lệnh nhập đối với nút nhấn stop – start. Sử dụng lưu trữ bit từ ngữ để đóng mạch điều khiển.

   CHÚ Ý: Các mạch tự động đòi hỏi nút nhấn đẩy/kéo để bảo đảm dừng an toàn.

2. Lệnh nhập examine off cho các tiếp điểm thực. Các tiếp điểm thực không được dùng trong mạch này. Do đó các tiếp điểm quá tải được đưa vào theo dạng ngõ nhập và được chèn vào chương trình ở vị trí tiếp điểm đóng.

3. Bộ thời chuẩn ngăn chặn, được dùng cùng với lệnh kiểm tra giới hạn đối với thứ tự của bộ khởi động thuận/đảo.

4. Lệnh cấp nguồn xuất cho từng bộ khởi động thuận và đảo.

5. Thứ tự vận hành như sau:

   A. Thuận được đóng mạch (ON) trong 5 giây và ngắt mạch (OFF) trong 3 giây.

   B. Đảo được đóng mạch (ON) trong 5 giây và ngắt mạch (OFF) trong 3 giây.

   C. Thứ tự này lặp lại một cách tự động trừ khi nhấn nút Stop.

6. Nút nhấn cài đặt lại bằng tay để cài đặt lại bộ thời chuẩn và chuyển động cơ sang trạng thái tạm dừng, đây là vị trí đảo xa nhất.

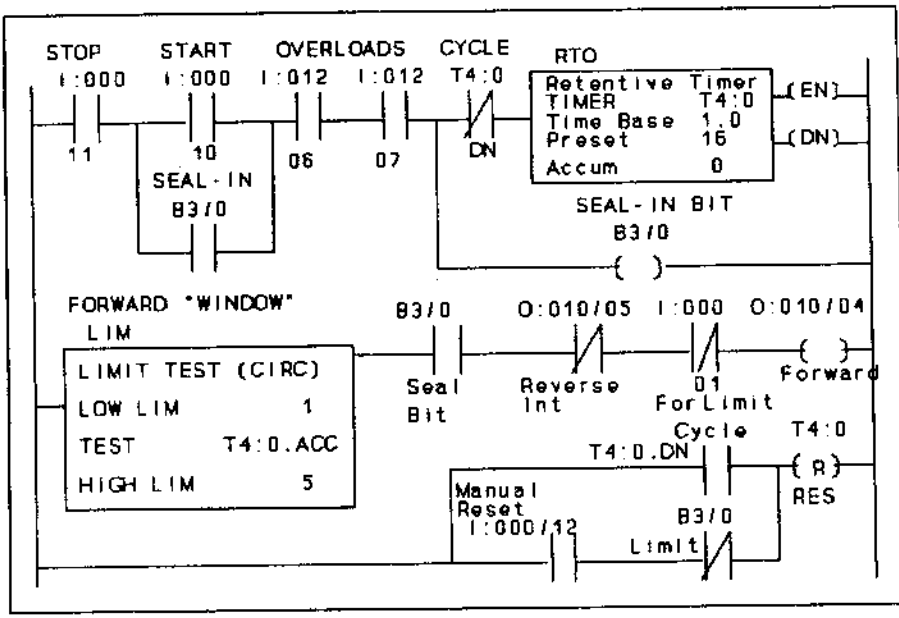
CÁC BƯỚC THỰC HIỆN

1. Viết chương trình trên PLC cho chu kỳ khởi động thuận/đảo theo logic lập trình nêu trên. Bạn hãy nhập chương trình được nêu trên Hình 11.2 với hai bậc thang đầu.

   Bạn hãy lưu ý lệnh kiểm tra giới hạn, lệnh này được dùng chỉ để kích hoạt bộ khởi động thuận. Khi bậc thang này là đúng, động cơ chạy 5 giây theo chiều thuận.

   CHÚ Ý: Bạn hãy quan sát địa chỉ của các bộ khởi động. Các bộ khởi động được mắc vào khung nhánh 1 thuộc nhóm 1, bạn cần bảo đảm sử dụng đúng các địa chỉ.

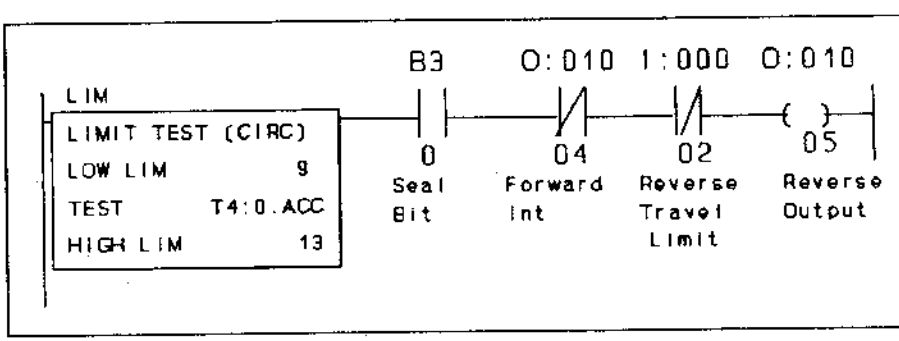
2. Bước kế tiếp là lập trình kiểm tra giới hạn cho bộ khởi động đảo (tương tự bậc thang thuận). Bạn hãy dùng các địa chỉ thích hợp



Hình 11.2. Chương trình chiều thuận

cho bộ khởi động đảo và xác lập cửa sổ lệnh giới hạn là 9 đến 13 giây. Khi hoàn tất, bậc thang của bạn tương tự bậc thang kiểm tra giới hạn được nêu trên Hình 11.3.

3. Trong các bậc thang thuận và đảo, bạn hãy đặt các giới hạn an toàn thích hợp. Đây sẽ là các khóa điện liên động và các giới hạn hành trình nhằm giới hạn khoảng cách hành trình theo chiều



Hình 11.3. Bậc thang xuất chiều ngược

thuận và chiều đảo. Bạn hãy sử dụng các bộ chuyển mạch toggle để mô phỏng các giới hạn hành trình thuận và đảo.

4. Chú ý, sự cài đặt lại bằng tay sử dụng nút nhấn thường mở. Sự cài đặt lại chỉ có thể xảy ra sau khi dừng mạch. Nhấn nút này sẽ kích hoạt lệnh cài đặt lại để cài đặt lại bộ thời chuẩn ngăn chặn. Cần có khóa liên động để ngăn chặn sự cài đặt lại trong khi chạy. Bạn hãy dùng bit B3/0 để khóa liên động nút cài đặt lại.

5. Sau khi hoàn tất các bước nêu trên, bạn hãy lưu và kiểm tra chương trình của mình. Nếu mọi thứ đều vận hành chuẩn xác, bạn hãy chạy chương trình với các bộ khởi động. Nếu mạch vận hành tốt, bạn hãy nối với động cơ.

6. Tại điểm này, mạch của bạn vận hành như sau: (A) Động cơ vận hành theo chiều thuận trong 5 giây; (B) Dừng trong 3 giây; (C) khởi động lại theo chiều ngược trong 5 giây; (D) dừng trong 3 giây; (E) chu kỳ lặp lại.

7. Kế tiếp, bạn có thể nghiên cứu khả năng mở công tắc quá tải. Nếu điều này xảy ra, bộ thời chuẩn ngăn chặn sẽ tạm dừng tính thời gian. Khi bạn cài đặt lại quá tải và khởi động động cơ, quá trình sẽ tiếp tục tính từ điểm dừng. Mạch này được thiết kế để khởi động từ khoảng thời gian tạm dừng do sự cố (quá tải, ...).

   Chú ý: Nếu được, bạn nên áp dụng mạch thắng (hãm) trên động cơ. Bạn có thể phối hợp Thí nghiệm 11.1 và 11.2 vào Bài tập này. Hai thí nghiệm này sẽ bổ sung sự thắng động cơ và chuyển đến vị trí dừng cho chương trình của Bài tập 11.0.

8. Bước cuối cùng là viết báo cáo, lưu và in chương trình.

Thí nghiệm 11.1. THẮNG ĐIỆN

Bạn có thể bổ sung hệ thống thắng điện vào mạch thuận/đảo trong Bài tập 11.0. Hệ thống thắng được dùng để dừng động cơ khi chạy theo chu kỳ giữa thuận và đảo. Đèn báo có thể được dùng để mô phỏng thắng điện nếu bạn không có hệ thống này. Dưới đây là các thông số:

Bạn hãy lập trình thêm hai lệnh cửa sổ để kiểm tra bộ thời chuẩn ngăn chặn của Bài tập 11.0. Bạn có thể dùng các lệnh bằng (Equal) hoặc kiểm tra giới hạn, nối các lệnh này song song để điều khiển bộ thời chuẩn đóng (thời chuẩn thắng), chỉnh hai lệnh bằng (Equal) hoặc kiểm tra giới hạn theo khoảng 7 và 15 giây.

Thời lượng cài đặt trước cho bộ thời chuẩn này là 1 giây, với cơ sở thời gian là 0.01 giây. Sử dụng TT bit điều khiển của thời chuẩn đóng để kích hoạt cuộn dây contactor của thắng điện (O:010/02).

Mục đích của bộ thời chuẩn này là thay đổi thời lượng thắng và bảo đảm cuộn contactor của thắng điện chỉ được kích hoạt tạm thời trong thời lượng được xác định trước.

Thí nghiệm 11.2. CÀI ĐẶT LẠI VỊ TRÍ DỪNG

Bài tập 11.0 có thể được bổ sung với chương trình cài đặt lại vị trí dừng. Mục đích là thắng và chuyển động cơ đến vị trí dừng khi nhấn nút Stop, với vị trí dừng là vị trí ngược chiều xa nhất. Bạn hãy dùng công tắc giới hạn hành trình đảo để thắng động cơ ở đúng vị trí dừng.

1. Bước thứ nhất là nhập chương trình được nêu trên Hình 11.4. Bộ thời chuẩn ngắt (TOF) và lệnh kiểm tra giới hạn được dùng để đưa động cơ đến vị trí dừng khi nhấn nút Stop. (Hình 11.4, các bậc thang 1.0 và 2.0). Bạn kích hoạt bộ thời chuẩn thắng khi động cơ đạt đến giới hạn hành trình ngược chiều (Hình 11.4, bậc thang 3.0).

   CHÚ Ý: Bộ thời chuẩn này là không cần thiết nếu dùng bộ thời chuẩn thắng trong Thí nghiệm 11.1.

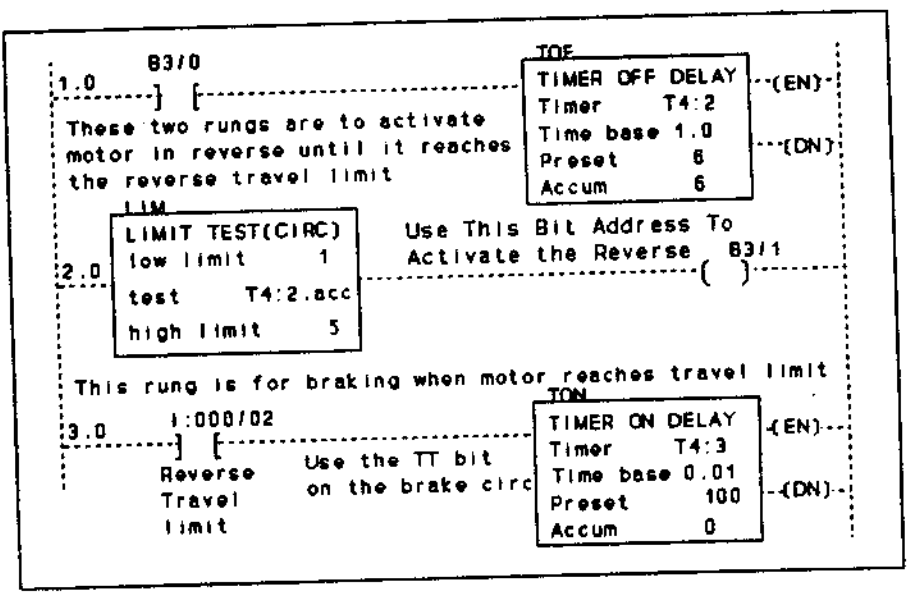
2. Bước thứ hai là bổ sung các lệnh vào chương trình để hoàn tất các thông số:

   A. Khi nhấn nút nhấn Stop hiện hữu, bạn kích hoạt bộ thời chuẩn thắng được dùng trong Thí nghiệm 11.1, và cài đặt lại bộ thời chuẩn ngăn chặn.

B. Lập trình nút cài đặt lại bằng tay hiện hữu hướng đến sự kích hoạt ngõ ra đảo. Điều này sẽ tạo ra sự điều khiển chạy/dừng động cơ theo chiều ngược cho đến khi kích hoạt công tắc giới hạn hành trình đảo.

C. Bạn hãy chú ý bit xuất B3/1 thuộc bậc thang 2.0 trên Hình 11.4. Bạn hãy lập trình địa chỉ bit này với lệnh nhập để kích hoạt ngõ ra đảo.

Bạn hãy kiểm tra và chạy chương trình nhưng chỉ sử dụng các bộ khởi động (để an toàn, bạn chưa nên nối với động cơ). Nếu chuỗi thứ tự của chương trình là chuẩn xác, bạn có thể nối và chạy thử với động cơ.



Hình 11.4. Chương trình dừng

Bài tập lớn 12.0

CHU KỲ ÉP

MỤC ĐÍCH

Lập trình các lệnh thời chuẩn/đếm cùng với các lệnh khác để điều khiển quy trình (ép các bản vi mạch).

NỘI DUNG

Sau khi hoàn tất Bài Tập này bạn sẽ có khả năng:

- Nghiên cứu và sử dụng các lệnh số học.
- Hiểu về điểm trôi và dữ liệu số nguyên.
- Nghiên cứu khái niệm bậc thang được phép.

TÀI LIỆU

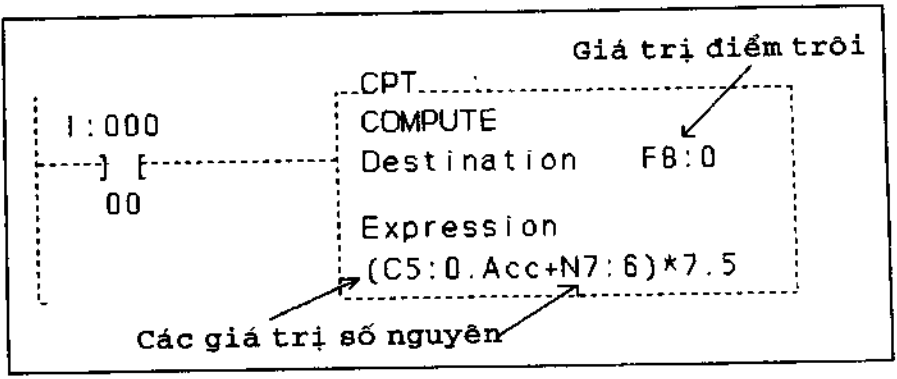
Để biết về từng lệnh trong Bài tập này, bạn hãy đọc Phụ Lục A: Các lệnh thời chuẩn/ đếm, so sánh, số và chuyển đổi, CTU, CTD, RES, CMP, EQU, LES, LIM, ADD, DIV, MUL, CPL, TOD.

THẢO LUẬN

Bộ xử lý hỗ trợ và vận hành hai tập tin dữ liệu được dùng cho các phép toán số học. Thứ nhất là tập tin số nguyên, với địa chỉ từ N7: 0 đến 999. Tập tin số nguyên sử dụng phần tử một – từ 16 bit. Khoảng giá trị có thể được lưu trong phần tử từ ngữ nguyên là –32.768 đến +32.768. Các tập tin số nguyên chỉ lưu số nguyên. Bộ xử lý sẽ làm tròn các giá trị bằng hoặc lớn hơn 0,5 thành số nguyên kế tiếp và bỏ các giá trị nhỏ hơn 0,5.

Thứ hai là tập tin điểm trôi với các địa chỉ từ F8:0 đến 999. Tập tin điểm trôi sử dụng phần tử 2 - từ ngữ, số 32 bit. Khoảng các giá trị có thể lưu trong phần tử điểm trôi là $\pm 1.1754944e^{-38}$ đến $\pm 3.4028237\ e^{+38}$. Các tập tin điểm trôi lưu các số nguyên và số thập phân.

Bạn có thể phối hợp các giá trị nguyên và giá trị điểm trôi nếu kết quả của phép toán không vượt quá khoảng số nguyên của giá trị nguyên. Độ chính xác có thể bị giảm khi đổi từ kiểu giá trị này sang kiểu kia. Bạn hãy xem Hình 12.1 về lệnh tính toán sử dụng các giá trị nguyên và điểm trôi.

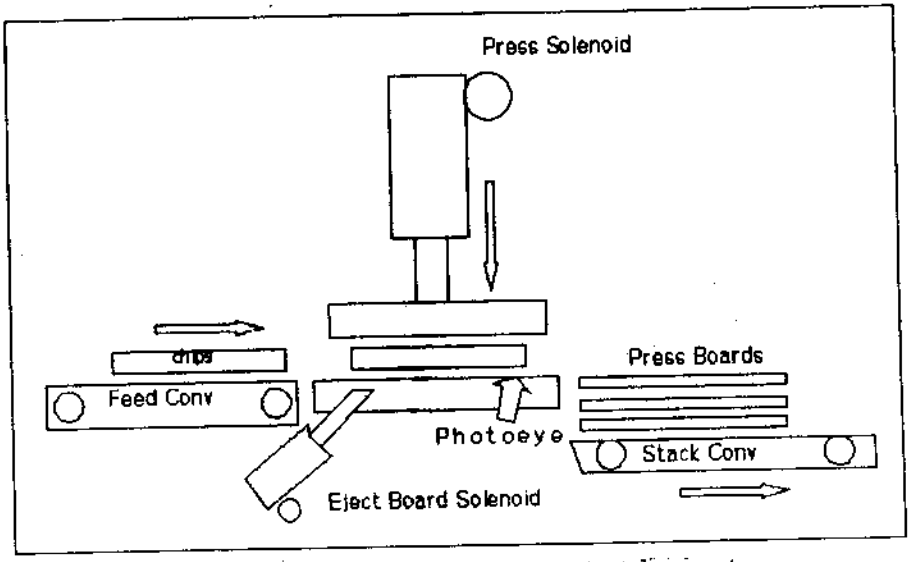


Hình 12.1. Các lệnh tính toán

ASCII và BCD chỉ được dùng để hiển thị. Bộ xử lý diễn dịch chúng theo số nguyên. Bạn dùng lệnh tính toán (TOD) để đổi số nguyên thành BCD nếu muốn hiển thị các giá trị 7 – đoạn cho bộ xử lý.

Bài tập

Chương trình này sử dụng các lệnh bạn đã học trong các Bài tập trước. Nhiệm vụ là lập trình mạch điều khiển để chuyển các vi mạch rời thành bản mạch. Các vi mạch rời dịch chuyển trên băng tải nạp và dừng



Hình 12.2. Hệ thống ép vi mạch lên bản mạch.

kế bên máy ép thủy lực. Khi các vi mạch đến đó, chúng được đưa vào vị trí ép lên bản mạch. Sau khi ép, bản mạch được đưa lên băng tải xếp chồng. Khi đủ 4 bản mạch, chúng được chuyển xuống để cắt và tiếp tục xử lý (Hình 12.2).

LOGIC CHƯƠNG TRÌNH

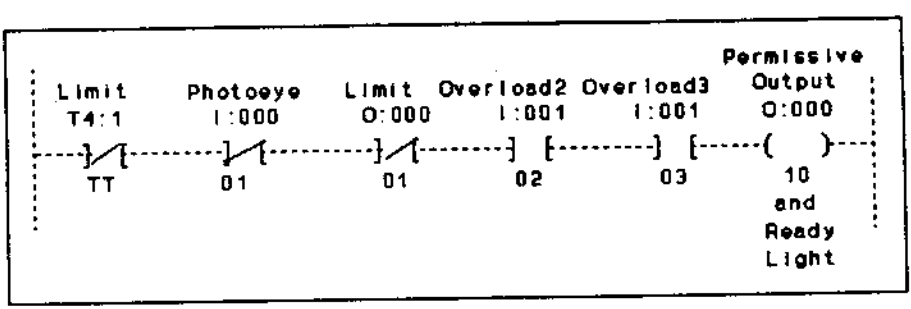
Dưới đây là thứ tự các sự kiện

1. Băng tải nạp khởi động và chạy trong 3 giây để chuyển các vi mạch đến khu vực ép.
2. Khi băng tải nạp hoàn tất, cuộn solenoid của máy ép được kích hoạt với thời gian ép là 6 giây.
3. Khi ép hoàn tất, cuộn solenoid khí nén được kích hoạt trong 1 giây để đưa bản mạch ra khỏi khu vực ép.
4. Các bản mạch được đặt lên băng tải xếp chồng. Sau khi xếp chồng 4 bản mạch trên băng tải, các bản mạch được chuyển sang khu vực xử lý kế tiếp. Thời gian dịch chuyển các bản mạch là 4 giây.

Trong lập trình, với nhiều điều kiện khác nhau cần đáp ứng để đạt kết quả. Để thực hiện điều này, bậc thang cho phép có thể được lập trình với tất cả các giới hạn đó. Khi mọi giới hạn đều thỏa, bit cho phép sẽ được kích hoạt. Bit này có thể được dùng trong một chương trình. Bạn hãy xem Hình 12.3, đây là bậc thang cho phép, được dùng trong chương trình. Trong ví dụ này, bít xuất được dùng thay cho bit nội. Mục đích là làm sáng đèn báo sẵn sàng.

CÁC BƯỚC THỰC HIỆN

1. Bạn được yêu cầu thiết kế và lập trình mạch điều khiển máy ép



Hình 12.3. Bậc thang cho phép khi thỏa các điều kiện.

các vi mạch lên bản mạch. Quy trình được nêu trên Hình 12.2. Bài tập này có tính linh hoạt trong thiết kế. Bạn có thể tự thiết kế hoặc sử dụng mạch ví dụ làm hướng dẫn để hoàn tất chương trình của bạn.

2. Để bắt đầu, bạn hãy lập trình lệnh JMP (nhảy) to LBL (đến nhãn). Lệnh này cho phép bạn nhảy qua chương trình của Bài tập 12.0.

3. Đối với bậc thang kế tiếp, bạn dùng bộ chuyển mạch toggle để kích hoạt lệnh MCR. Mạch stop – start có thể được dùng để khởi động băng tải nạp và quy trình. Bạn hãy dùng cùng địa chỉ cho nút stop và bộ chuyển mạch MCR.

4. Bước thứ nhất là băng tải nạp được kích hoạt từ bộ thời chuẩn #1 để chuyển các vi mạch rời đến máy ép, cần 3 giây để đưa vi mạch đến máy.

   Các vi mạch không được phép dịch chuyển khi chưa thỏa các điều khiển giới hạn dưới đây.

   A. Máy ép đang ở chu kỳ OFF

   B. Bản mạch đã được lấy ra khỏi máy ép.

   C. Tế bào quang điện không bị chặn (không có bản mạch trên máy ép)

   D. Các tiếp điểm quá tải không ngắt mạch. (Hình 12.3).

5. Bước kế tiếp là bộ thời chuẩn #2 có thể được dùng để kích hoạt cuộn solenoid của máy ép. Thời gian ép là 6 giây. Lệnh xuất được dùng cho cuộn solenoid này. Bạn dùng các đèn để mô phỏng solenoid máy ép. Khi bộ thời chuẩn kích hoạt cuộn solenoid máy ép bị ngắt (OFF), cylinder thủy lực của máy ép trở về vị trí ban đầu.

6. Sau khi ép xong, bộ thời chuẩn #3 có thể được dùng để kích hoạt lệnh xuất cho cuộn solenoid khí nén. Thời gian lấy bản mạch ra là 1 giây, việc lấy bản mạch (bằng khí nén) chỉ được thực hiện ở chu kỳ OFF của solenoid máy ép.

   Chú ý: Sau khi bộ thời chuẩn #3 hoàn tất, băng tải nạp khởi động lại một cách tự động, quy trình lặp lại.

7. Cuối cùng, bộ thời chuẩn #4 được dùng để kích hoạt băng tải xếp chồng. Thời lượng để dịch chuyển và lấy chồng các bản mạch

hoàn tất là 4 giây. Băng tải này chỉ dịch chuyển khi số lượng bản mạch hoàn tất bằng 4. Bạn cài đặt lại bộ đếm và chờ 4 bản mạch kế tiếp. Hình ảnh trên hiển thị 7- đoạn sẽ cho biết kết quả đếm của tập tin số nguyên.

8. Sau khi hoàn tất và kiểm tra xong mạch này, bạn hãy kiểm chứng chương trình theo các điều kiện đã nêu trong Bài tập và trong phần logic lập trình.

ÔN TẬP

Sau khi làm xong phần này, bạn hãy thực hành các thí nghiệm để củng cố và nâng cao kiến thức đã học. Khi hoàn thành các thí nghiệm bạn hãy lưu chương trình và viết báo cáo.

Thí nghiệm 12.1. CHUYỂN ĐỔI NHIỆT ĐỘ

Mục đích là chuyển giá trị nhiệt độ được ghi theo thang độ Celsius thành giá trị Fahrenheit và sử dụng giá trị Fahrenheit trong tình huống điều khiển. Điều này đòi hỏi sử dụng các lệnh số học và các lệnh so sánh.

Dưới đây là các thông số cần thiết để hoàn tất chương trình.

1. Sử dụng giá trị tích lũy của bộ đếm thuận để mô phỏng ngõ nhập dùng cặp nhiệt điện đo giá trị độ Celsius (Giá trị tích lũy bộ đếm = giá trị ^{0}C). Nhập bộ đếm ngược với cùng địa chỉ như bộ đếm thuận để giảm giá trị Celsius.

2. Mạch bộ đếm chu kỳ trong tập dán (paste) của bạn hoặc các công tắc vận hành bằng tay có thể được dùng để tạo bước cho các bộ đếm thuận và ngược.

3. Lập trình các lệnh số học cần thiết để rút ra công thức chuyển đổi từ độ Celsius sang độ Fahrenheit.

$$F = \frac{9}{5}C + 32$$

   Hiển thị kết quả độ Fahrenheit trên màn hình 7-đoạn. Đối với lệnh số học, bạn có thể dùng tập tin số nguyên hoặc tập tin điểm trôi. Bạn hãy bắt đầu lập trình địa chỉ số nguyên bất kỳ trong các tập tin từ N7:20.

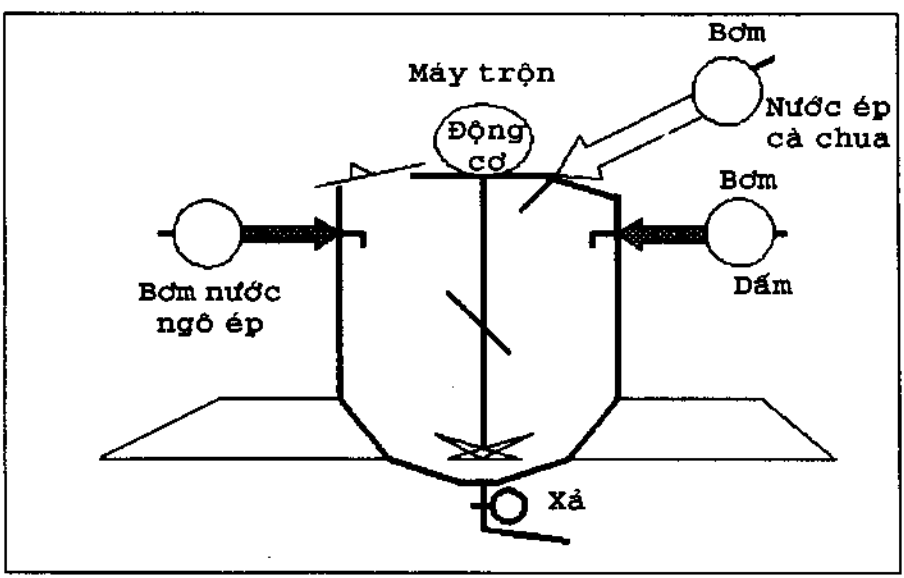
4. Bạn hãy lập trình các lệnh so sánh để thực hiện yêu cầu sau: khi giá trị Fahrenheit lớn hơn hoặc bằng 214^{0}F, bạn hãy giảm nhiệt độ xuống giá trị 178^{0}F. Khi giá trị Fahrenheit nhỏ hơn hoặc bằng 178^{0}F, bạn hãy tăng đến 214^{0}F.

Bạn hãy dùng các lệnh so sánh này để điều khiển các bộ đếm Up/Dn để giảm và tăng giá trị độ Celsius. Khi hoàn tất, bạn hãy chuyển sang chế độ RUN và thử chạy chương trình. Nhiệt độ của bạn phải thay đổi lần lượt giữa 178 và 214^{0}F.

Thí nghiệm 12.2. NƯỚC SỐT

Hai thành phần cần thiết trong công thức nước sốt là 3 lít dấm và 3 lít nước ngô (bắp) ép. Cả hai thành phần này được bơm vào thùng trộn (Hình 12.4). Bạn hãy lập trình mạch này để đáp ứng các thông số dưới đây:

1. Mỗi lít dấm và nước ngô ép cần 5 giây để bơm. Bạn hãy dùng bộ đếm để đếm số lít.
2. Khi 3 lít dấm và 3 lít nước ngô ép được bơm vào thùng trộn, bạn hãy kích hoạt máy trộn. Thời gian trộn là 30 phút (hoặc 30 giây) trong khi đó, các thành phần khác lần lượt được đưa vào thùng trộn.
3. Khi các thành phần được pha trộn, bạn bổ sung 10 lít nước cà chua ép. Mỗi lít cần 1 giây để bơm. Khi hoàn tất, bạn xả hỗn hợp bằng tay và cài đặt lại hệ thống.



Hình 12.4. Quy trình trộn nước sốt.

Bài tập lớn 13.0

THUMBWHEEL

MỤC ĐÍCH

Lập trình lệnh đọc chuyển khối trong bộ điều khiển công tắc thumbwheel bên ngoài.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Hiểu khái niệm đọc chuyển khối.
- Sử dụng các bộ chuyển đổi thumbwheel thực tiễn.
- Sử dụng bộ thời chuẩn ON và OFF trong cùng một chương trình.
- Sử dụng module đa công để điều khiển công tắc thumbwheel.

TÀI LIỆU

Bạn hãy đọc Phụ lục A: Các lệnh thời chuẩn, so sánh, và chuyển khối, TON, TOF, FRD, BTR.

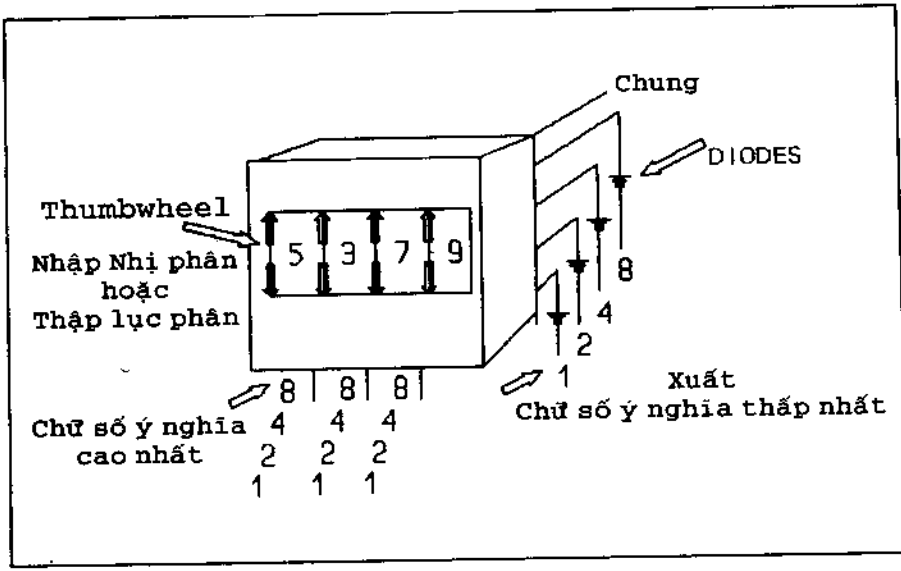
THẢO LUẬN

Thumbwheel

Bộ chuyển đổi thumbwheel là bộ phận rất hữu dụng trong lập trình các mạch điều khiển. Chúng có thể được dùng để nhập số thập phân từ một đến bốn chữ số vào bộ xử lý. Số này có thể được dùng để thay đổi giá trị xác lập trước trong bộ thời chuẩn, bộ đếm, hoặc số nguyên để xử lý.

Số thập phân hiển thị trên một phần của bộ chuyển đổi thumbwheel tạo ra số xuất có 4 chữ số, là số BCD (nhị phân) hoặc số HEX (thập lục phân). Ví dụ, phần bộ chuyển đổi thumbwheel BCD (nhị phân) với số 9 thập phân sẽ hiển thị số nhị phân 1001 ở ngõ ra.

Các phần bộ chuyển đổi thumbwheel riêng rẽ có thể kết nhóm để tạo thành thumbwheel nhiều phần. Hình 13.1 minh họa 4 phần kết hợp, tạo thành bộ chuyển đổi thumbwheel 4 – chữ số. Khi bạn nối bộ chuyển đổi



Hình 13.1. Bộ chuyển đổi thumbwheel 4- chữ số

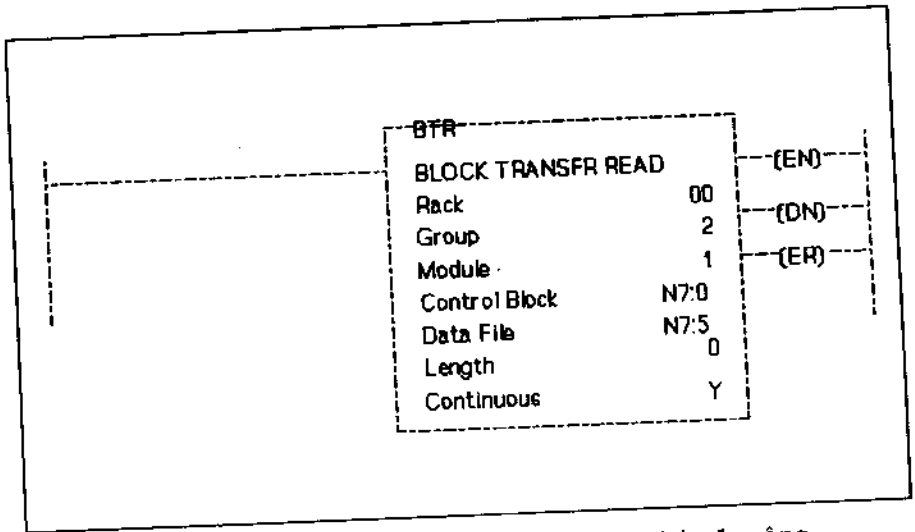
thumbwheel 4 chữ số vào module nhập, mỗi số của thumbwheel này đòi hỏi 4 tín hiệu nhập và một ngõ chung. Do đó, thumbwheel 4 – chữ số cần 16 nối kết ngõ vào và một nối kết chung.

Phương pháp thứ hai là nối kết bộ chuyển đổi thumbwheel với module đa công. Đây là module "thông minh" đọc và tạo đa công cho thông tin từ bộ chuyển đổi thumbwheel vào bộ xử lý. Các ưu điểm bao gồm: (1) Khi cần vài bộ thumbwheel 4 – chữ số, module đa công được dùng để giảm số nối kết ngõ vào. (2) Giảm thể tích khung hộp chứa. (3) Module đa công có thể hoạt động với bốn bộ chuyển đổi thumbwheel 4 – chữ số. Các nối kết thumbwheel được ghi rõ trong các bộ mô phỏng hoặc tài liệu của nhà sản xuất.

Lệnh chuyển khối

Chuyển khối được coi là tập hợp các lệnh và là kỹ thuật lập trình được dùng để chuyển nhiều dữ liệu trong một lần quét I/O. Chuyển khối có thể được thực thi theo thao tác đọc, viết hoặc hai chiều, tùy theo module xử lý. Module nhập sử dụng thao tác đọc, module xuất sử dụng thao tác viết và module hai chiều sử dụng cả thao tác đọc và viết.

Trong Bài tập này module đa công được dùng để đọc dữ liệu từ bộ chuyển đổi thumbwheel. Lệnh đọc chuyển khối được dùng để đọc và



Hình 13.2. Lệnh đọc chuyển khối trong module đa công.

truyền dữ liệu này từ module đa công đến bộ xử lý. Trên Hình 13.2, lệnh đọc chuyển khối được xác lập theo chế độ liên tục để thiết lập thao tác đọc liên tục ở module đa công (từ bộ chuyển đổi thumbwheel).

Bài tập

Bài tập gồm hai phần. Thứ nhất, bạn dùng lệnh đọc chuyển khối để truyền thông với module đa công. Điều này tạo cho bạn khả năng sử dụng hai bộ thumbwheel để thay đổi giá trị xác lập trước của các bộ thời chuẩn trong chương trình. Thứ hai, bạn lập trình mạch thời chuẩn để điều khiển hoạt động ON/OFF của bộ chuyển đổi. Bộ thời chuẩn đóng (TON) và bộ thời chuẩn ngắt (TOF) được dùng để điều khiển thời lượng tối thiểu bộ khởi động ở trạng thái ON và OFF. Bạn hãy xem lại Hình 4.5, Thí nghiệm 4.4.

LOGIC CHƯƠNG TRÌNH

1. Bậc thang không điều kiện với lệnh đọc chuyển khối để đọc các giá trị của bộ chuyển đổi thumbwheel. Điều này chỉ được dùng với module thông minh, chẳng hạn module đa công, để đọc bộ chuyển đổi thumbwheel.
2. Công tắc áp lực (công tắc toggle có thể được dùng thay cho công tắc áp lực) để kích hoạt lệnh xuất cho bộ khởi động (có thể dùng

đèn mô phỏng bộ khởi động) theo chế độ ON và OFF một cách tự động.

3. Bộ thời chuẩn đóng (TON) để bảo đảm bộ khởi động vẫn ở chế độ ON sau khi khởi động trong thời lượng tối thiểu (thời gian tối thiểu là 3 giây).

4. Bộ thời chuẩn ngắt (TOF) để bảo đảm bộ khởi động vẫn ở chế độ OFF với thời lượng tối thiểu (3 giây) sau khi mạch dừng lại (Stop).

5. Bộ chuyển đổi thumbwheel 4 – chữ số thay đổi giá trị xác lập trước của các bộ thời chuẩn. Lệnh FRD chuyển giá trị BCD (nhị phân) thành giá trị nguyên.

CÁC BƯỚC THỰC HIỆN

Phần I

1. Phần thứ nhất của Bài tập này cần lập trình lệnh đọc chuyển khối được nêu trên Hình 13.2. Sau khi ghòan tất, bạn hãy chuyển sang chế độ RUN, chuyển màn hình sang menu giám sát dữ liệu. Trong menu này, bạn thấy các số nhị phân N7:6, N7:7, và N7:9. Các tập tin số nguyên này là địa chỉ của bốn bộ chuyển đổi thumbwheel 4- chữ số.

   Bạn hãy quan sát mỗi khi quay các chữ số thumbwheel chúng sẽ đổi thành các bit nhị phân trên màn hình. Bạn hãy quay số trên thumbwheel và nhập số BCD từ màn hình giám sát dữ liệu trong Bảng Dữ Liệu 13.1

Bảng 13.1. Dữ liệu Thumbwheel

INTEGER THUMB #	BCD # THUMB #	BCD #
N7:6	9999	1111
N7:7	7777	8888
N7:8	5555	6666
N7:9	3333	4444

2. Sau khi hoàn tất phần thứ nhất và biết sử dụng bộ chuyển đổi thumbwheel, bạn chuyển sang phần thứ hai.

Phần II

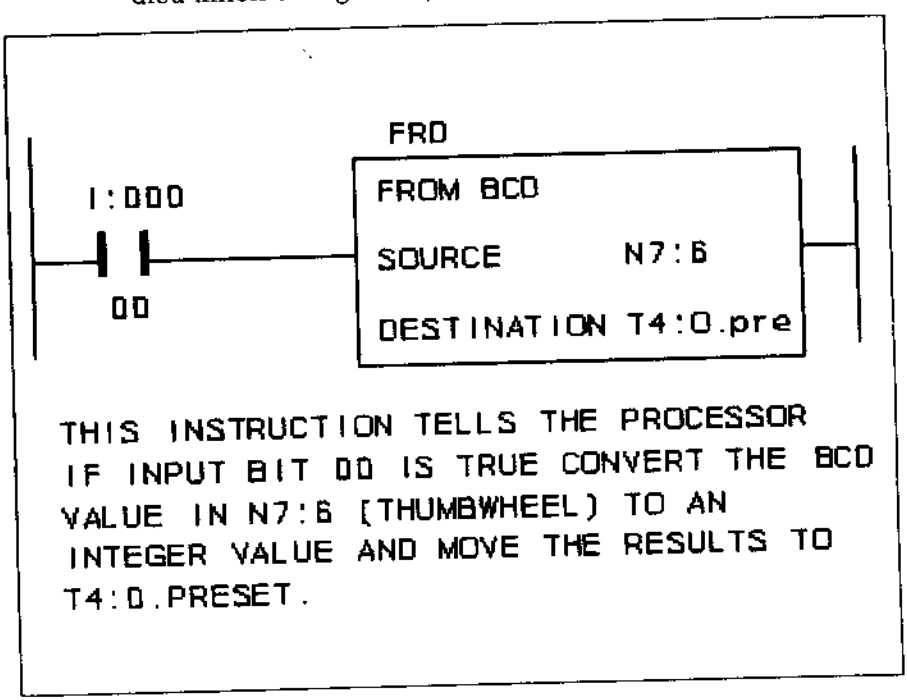
3. Phần này yêu cầu bạn chuyển đổi và lập trình mạch relay trong Thí nghiệm 4.5 của Bài tập 4. Mạch này sử dụng bộ thời chuẩn để điều khiển các điều kiện ON/OFF của công tắc ngõ vào.

4. Bạn hãy lập trình bộ chuyển đổi thumbwheel để thay đổi giá trị cài đặt trước của cả hai bộ thời chuẩn. Bạn chuyển đổi các giá trị xuất trước khi xử lý chúng với chương trình FRD được dùng để đổi giá trị BCD thành giá trị nguyên (Hình 13.3).

5. Sau khi nhập chương trình, bạn hãy kiểm tra sự vận hành của công tắc ngõ vào theo các bước dưới đây.

   A. Bạn bật công tắc ON và chuyển ngay sang OFF. Ngõ ra vẫn đóng mạch ON trong thời lượng được xác lập trước, bộ TON điều khiển thời gian mạch duy trì ON trong công tắc là OFF.

   B. Bạn tắt công tắc OFF và chuyển ngay sang ON. Ngõ ra vẫn ngắt mạch OFF trong thời lượng được xác lập trước, bộ TOF điều khiển thời gian mạch duy trì OFF kể vị trí công tắc.



Hình 13.3. Lệnh FRD

6. Bạn hãy thay đổi thời lượng xác định trước của bộ thời chuẩn với các bộ chuyển đổi thumbwheel và quay lại bước 5, sau đó lưu và in chương trình.

ÔN TẬP

Sau khi hoàn thành phần này, bạn cần củng cố và nâng cao kiến thức bằng cách thực hiện các Thí nghiệm kèm theo. Thí nghiệm 13.1 là phần tích hợp bổ sung cho Bài tập này.

Thí nghiệm 13.1. KHỞI ĐỘNG KHÔNG CHUẨN

Bạn hãy chỉnh sửa Bài tập 13.0 với các thông số dưới đây:

1. Kích hoạt đèn cảnh báo và dừng hệ thống nếu có hơn ba lần thử khởi động liên tiếp bằng cách vận hành công tắc trong khoảng thời gian tối thiểu cho trước (quá nhiều khởi động trong khung thời gian xác định). Bạn cần dùng công tắc bằng tay để cài đặt hệ thống.

2. Sử dụng bộ thời chuẩn và bộ đếm để duy trì thời lượng quy định tối thiểu đối với số lần thử khởi động từ công tắc. Bạn hãy lập trình bộ thumbwheel thứ hai để thay đổi thời lượng tối thiểu của bộ thời chuẩn này.

Thí nghiệm 13.2. CHUYỂN ĐỔI NHIỆT ĐỘ 2

Mục tiêu của thí nghiệm:

1. Nghiên cứu các tính năng biên tập tiên tiến.
2. Cung cấp phương tiện nhập giá trị thang nhiệt độ Celsius trong phương trình toán với bộ chuyển đổi thumbwheel.

Trong Bài tập 12, bạn lập trình mạch chuyển các giá trị Celsius sang giá trị Fahrenheit. Bạn hãy xem lại Thí nghiệm 12.1 – chuyển đổi nhiệt độ 1. Với các tính năng trên tập tiên tiến, bạn hãy chọn và chép chương trình của thí nghiệm 12.1 lên đĩa và đặt tên tập tin là Temp.

Bước thứ hai là khôi phục và trở về Bài tập 13.0. Với các tính năng biên tập tiên tiến, bạn hãy chép tập tin Temp vào Bài tập 13.0. Sau khi hoàn tất, bạn hãy chỉnh sửa chương trình thí nghiệm này như sau:

1. Xóa tất cả các bậc thang liên quan đến bộ đếm được dùng để mô phỏng giá trị Celsius.
2. Lập trình bộ chuyển đổi thumbwheel để hiển thị giá trị Celsius trong phương trình toán. Khi hoàn tất, bạn có khả năng nhập giá trị Celsius từ bộ chuyển đổi thumbwheel và giá trị này sẽ được chuyển thành Fahrenheit.
3. Bạn chỉnh sửa các lệnh so sánh để kích hoạt đèn báo khi nhiệt độ F lớn hơn 212°F, và đèn thứ hai khi nhiệt độ dưới 180°F.

Bài tập lớn 14.0

ĐỘNG CƠ ĐA TỐC ĐỘ

MỤC ĐÍCH

Viết chương trình điều khiển mạch thứ tự sử dụng các lệnh kiểm tra giới hạn.

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Điều khiển chuỗi thứ tự tốc độ của động cơ nhiều tốc độ.
- Nghiên cứu và viết các chương trình chuỗi thứ tự.
- Hiểu công dụng của lệnh kiểm tra giới hạn.

TÀI LIỆU

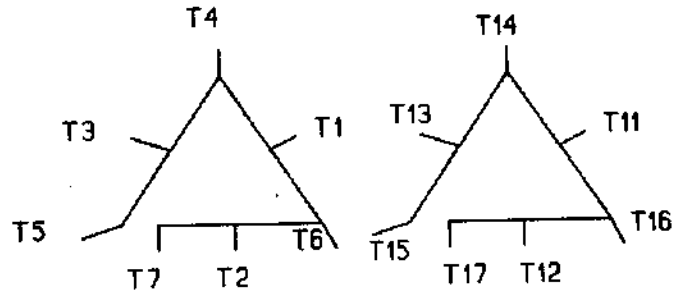
Để biết chi tiết về các lệnh được dùng trong Bài tập này, bạn hãy đọc Phụ lục A. Các lệnh thời chuẩn/đếm và so sánh, TON, CTU, LIM.

THẢO LUẬN

Bài tập được thực hiện với động cơ 3-pha, 4–tốc độ, 2-cuộn dây, công suất không đổi. Do động cơ 4-tốc độ có thể hơi khó tìm, các khái niệm lập trình trong Bài tập này có thể được áp dụng với một số khái niệm khác. Một lựa chọn là sử dụng bốn bộ khởi động hoặc đèn thay cho động cơ 4-tốc độ. Thí nghiệm 14.1 sẽ giúp bạn hiểu rõ hơn các khái niệm này.

Chú ý nối kết động cơ trên Hình 14.1. Tốc độ thấp và trung bình – thấp đòi hỏi bộ khởi động và contactor cho từng tốc độ. Tốc độ trung bình cao và cao chỉ cần bộ khởi động. Tổng cộng có 4 bộ khởi động 3-pha và 2 contactor (có thể sử dụng hai bộ khởi động 2 – tốc độ).

Cần lập trình động cơ sao cho người vận hành có thể khởi động động cơ từ trạng thái nghỉ đến tốc độ bất kỳ. Khi khởi động, động cơ luôn luôn bắt đầu từ tốc độ thấp đến tốc độ được chọn. Để thay đổi tốc độ (tăng hoặc giảm), người vận hành phải dừng động cơ.



SPEED	L1 L2 L3	OPEN	TOGETHER
LOW	T1--T2--T3	all others	T4,T5,T6,T7
MEDLOW	T11-T12-T13	all others	T14,T15,T16,T17
MEDHIGH	T6--T4,T5--T7	all others	
HIGH	T16-T14T15-T17	all others	

Hình 14.1. Nối kết động cơ 4-tốc độ, 2-cuộn dây, công suất không đổi.

LOGIC CHƯƠNG TRÌNH

1. Nút Stop chính thường đóng để dừng động cơ ở tốc độ bất kỳ. Động cơ phải dừng ở tốc độ bất kỳ để thay đổi tốc độ.
2. Bốn nút Start thường mở, thấp, trung bình – thấp, trung bình – cao, và cao. Bạn hãy dùng các nút nhấn trên bộ mô phỏng.
3. Nhấn một trong bốn nút Start làm cho động cơ khởi động ở tốc độ thấp và tăng dần qua từng cấp tốc độ cho đến khi đạt tốc độ được chọn.
4. Bạn hãy lập trình cửa sổ cho từng tốc độ, xác lập các giới hạn trong từng cửa sổ sao cho từng tốc độ trong chuỗi thứ tự vẫn duy trì ON trong ít nhất là 5 giây. Khi đạt được tốc độ đã chọn, động cơ sẽ tiếp tục chạy ở tốc độ đó.
5. Để thay đổi tốc độ đang chạy (tăng hoặc giảm), động cơ phải dừng lại.
6. Bốn bộ khởi động và hai contactor được sử dụng, bạn hãy dùng các tiếp điểm đóng thực từ từng bộ khởi động trong mạch, bảo đảm khóa liên động từng tốc độ. Các địa chỉ bộ khởi động được

dùng trong Bài tập ở trên khung giá xa #1. Bạn hãy chọn và sử dụng địa chỉ thích hợp.

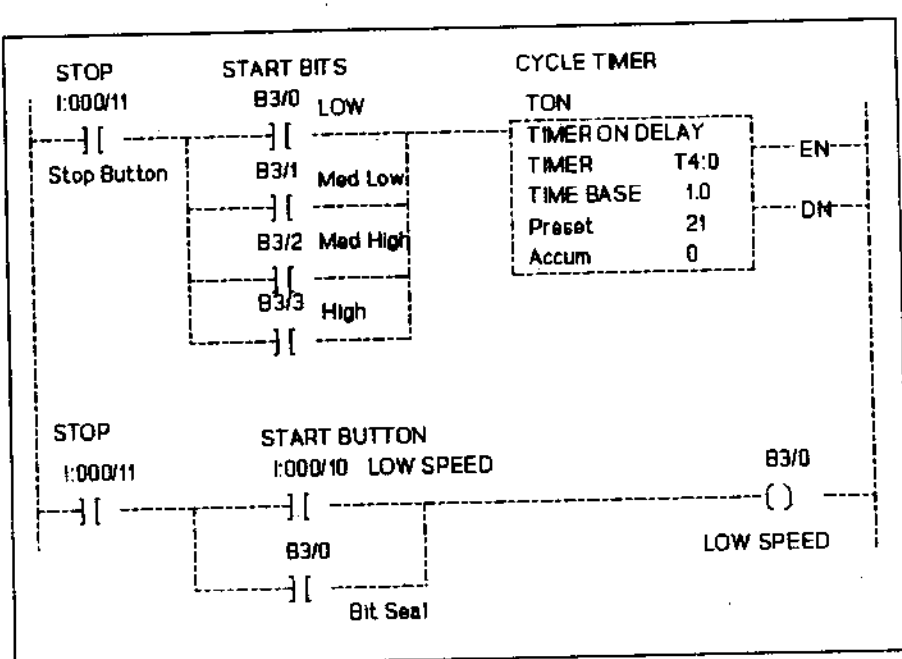
7. Bạn lập trình đèn báo riêng cho từng tốc độ. Một đèn được đặt trong bộ điều khiển, ba đèn ở bộ mô phỏng.

CÁC BƯỚC THỰC HIỆN

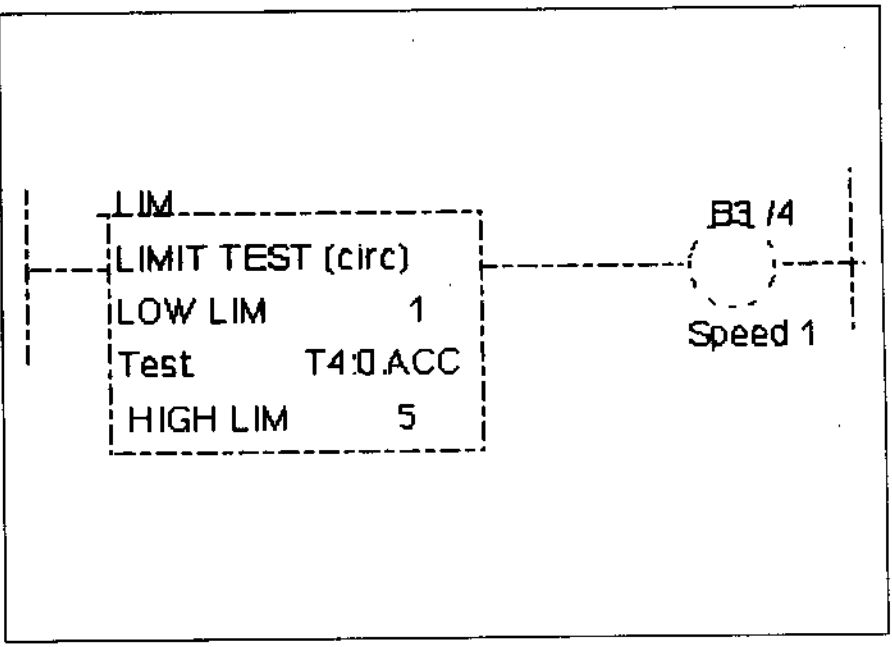
1. Bạn xây dựng chương trình trên PLC để điều khiển động cơ 4- tốc độ. Bạn hãy nhập các bậc thang trên Hình 14.2, 14.3, và 14.4

   CHÚ Ý: Các bậc thang trên Hình 14.2, 14.3 và 14.4 là chương trình chỉ dành cho tốc độ thấp. Bạn phải lập trình và bổ sung các bậc thang với địa chỉ thích hợp cho ba tốc độ còn lại.

   Đối với bước thứ nhất, bạn hãy nhập bậc thang trên Hình 14.2. Bậc thang này là chương trình bộ thời chuẩn chu kỳ. Bộ thời chuẩn này lập chu kỳ thông qua các lệnh kiểm tra giới hạn để kích hoạt các tốc độ khác nhau. Bậc thang thứ hai cho thấy bậc thang Stop và Start vận hành bit lưu trữ (B3/0). Nút Start I:00/10



Hình 14.2. Các bậc thang Start tốc độ thấp.



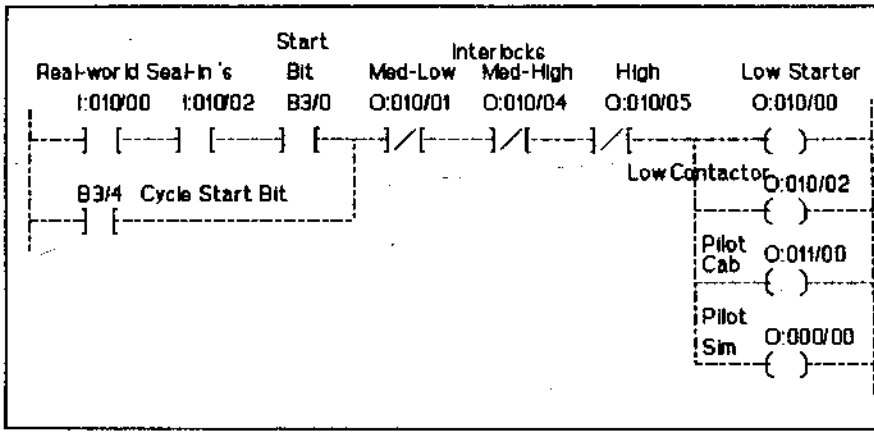
Hình 14.3. Bậc thang kiểm tra giới hạn.

kích hoạt bit lưu trữ (B3/0). Bit này đóng mạnh bậc thang thời chuẩn chu kỳ và được dùng để đóng các tiếp điểm thực được nêu trên Hình 14.4.

Bạn cần lập trình thêm ba bậc thang tương tự bậc thang thứ hai cho ba tốc độ còn lại.

2. Bạn hãy lập trình bậc thang tốc độ thấp trên Hình 14.3, cộng thêm ba bậc thang tương tự bậc thang này nhưng với các địa chỉ khác. Các bậc thang này tạo ra 4 cửa sổ sử dụng các lệnh kiểm tra giới hạn. Các cửa sổ này xác định thời lượng động cơ chạy theo từng tốc độ. Các thang này điều khiển bốn bit lưu trữ, B3/4 đến B3/7. Bạn xác lập các giới hạn của 4 cửa sổ để cho phép: (A) 1 đến 5; (B) 6 đến 10; (C) 11 đến 15; và (D) 16 đến 20.

3. Các bậc thang cần thiết để điều khiển bộ khởi động thực tiễn là bước cuối. Trên 4 bậc thang này, các bit lưu trữ được dùng trong các bậc thang trước kết hợp với các tiếp điểm đóng thực để điều khiển các bộ khởi động. Hình 14.4 trình bày sơ đồ nối tốc độ thấp.



Hình 14.4. Các bậc thang đóng tốc độ thấp.

Bạn hãy xem Bảng 14.1 để biết địa chỉ của ba tốc độ còn lại.

Bảng 14.1. Các địa chỉ.

	BỘ MÔ PHỎNG				
TỐC ĐỘ	KHỞI ĐỘNG	CONTACTOR	ĐÓNG THỰC	ĐÈN BÁO	ĐÈN BÁO
Thấp	O:010/00	O:010/02	I:010/00	O:011/00	O:000/00
Hơi thấp	O:010/01	O:010/03	I:010/01	O:011/01	O:000/01
Hơi cao	O:010/04		I:010/04	O:011/04	O:000/04
Cao	O:010/05		I:010/05	O:011/05	O:000/05

4. Sau khi hoàn tất chương trình này, bạn hãy kiểm tra và cho chạy thử với các bộ khởi động (không nối với động cơ). Nhấn nút tốc độ cao sẽ xảy ra chuỗi thứ tự sau: (A) Bộ khởi động tốc độ thấp sẽ ON trong 5 giây, sau đó OFF, (B) Bộ khởi động trung bình – thấp ON trong 5 giây, sau đó OFF; (C) Bộ khởi động trung bình – cao ON trong 5 giây, sau đó OFF; và (D) Bộ khởi động cao ON và vẫn duy trì ON cho đến khi nhấn nút Stop. Nếu mọi thứ đều vận hành chuẩn xác, bạn có thể nối và chạy động cơ với các tốc độ khác nhau.

5. Bạn hãy xem lại tất cả các bước trong Bài tập này. Chức năng của lệnh kiểm tra giới hạn là rất quan trọng. Các khái niệm và kỹ

thuật được dùng trong Bài tập này thường được ứng dụng trong nhiều kiểu mạch điều khiển. Bạn hãy lưu chương trình lên đĩa để bảo quản.

6. Sau khi hoàn tất, bạn hãy lập báo cáo, lưu và in.

ÔN TẬP

Sau khi hoàn thành phần đầu của Bài tập 14, để kiểm tra kiến thức về các khái niệm và kỹ thuật đã học, bạn hãy thực hành Thí nghiệm 14.1.

Thí nghiệm 14.1. ĐÈN SÁNG THEO THỨ TỰ

Thí nghiệm này có thể được hoàn tất với Bài tập 14.0. Mục đích của Thí nghiệm là: (1) Xây dựng mạch thứ tự được điều khiển theo biến. (2) Sử dụng ngõ nhập thực tiễn (thumbwheel) cho chương trình điều khiển tốc độ theo thứ tự. (3) Thực hiện kiểm tra giới hạn, có các lệnh thời chuẩn và tương tác với nhau trong chương trình vận hành.

Thiết kế và vận hành mạch thứ tự sử dụng các lệnh kiểm tra giới hạn vận hành theo cách thức dưới đây:

1. Tám đèn theo chuỗi thứ tự trong chu kỳ liên tục. Khi công tắc toggle vận hành bộ thời chuẩn đóng mạch ON, các đèn bắt đầu sáng theo chu kỳ.
2. Ở cuối mỗi chu kỳ, bộ thời chuẩn tăng một done bit cho bộ đếm. Giá trị tích lũy của bộ đếm sẽ tăng qua từng lệnh kiểm tra giới hạn và kích hoạt các đèn theo chuỗi thứ tự.
3. Tốc độ của chu kỳ phụ thuộc vào thời gian cài đặt trước của bộ thời chuẩn. Bạn dùng bộ chuyển đổi thumbwheel để thay đổi thời lượng này của bộ thời chuẩn.

Dưới đây là các thông số lập trình.

1. Dùng 8 lệnh kiểm tra giới hạn để vận hành 8 đèn ở ngõ ra. Toàn bộ kiểm tra giới hạn chỉ kiểm tra cùng một bộ đếm (C5:5). Các xác lập giới hạn như sau: thấp và cao của lệnh kiểm tra giới hạn thứ nhất là một và hai, lệnh thứ hai là ba và bốn, lệnh thứ ba là bốn và năm, v.v....
2. Bạn lập trình thời chuẩn đóng (TON) để tiếp tục chu kỳ. Xác lập cơ sở thời gian của bộ thời chuẩn theo 0.01 giây, dùng công tắc

Toggle để kích hoạt bộ thời chuẩn này. Để khởi động, bạn hãy xác lập bộ thời chuẩn theo 100.

3. Lập trình done bit của bộ thời chuẩn để kích hoạt bộ đếm thuận C5:5 xác lập trước cho bộ đếm là 17, và dùng done bit của bộ đếm để cài đặt lại bộ đếm này.

4. Lập trình bộ chuyển đổi thumbwheel để thay đổi các thời lượng xác lập trước trên bộ thời chuẩn (TON). Điều này cho phép bạn tăng hoặc giảm tốc độ của chuỗi thứ tự. Lệnh FRD được dùng để chuyển đổi và dịch chuyển giá trị BCD của thumbwheel sang giá trị của bộ thời chuẩn.

Sau khi hoàn tất chương trình, bạn chuyển sang chế độ RUN và chỉ chạy chương trình. Các đèn của bạn sẽ lần lượt sáng/tắt theo chu kỳ. Với bộ chuyển đổi thumbwheel, bạn hãy thay đổi các thời lượng cài đặt trước trên bộ thời chuẩn. Chu kỳ của các đèn sẽ chậm hoặc nhanh dần tùy theo thời lượng đó.

Bài tập lớn 15.0

QUY TRÌNH LƯU TRỮ

MỤC ĐÍCH

Viết chương trình sử dụng lệnh logic số tập tin (FAL) để thực hiện tập tin đối với phần tử và tập tin đối với các phép logic tập tin nhiều từ ngữ.

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Hiểu khái niệm lệnh FAL
- Nghiên cứu các lệnh FSC, FLL, và COP
- Thực hiện các phép logic nhiều từ ngữ.

TÀI LIỆU

Bạn hãy đọc Phụ lục A: Các lệnh tập tin FAL, FSC, FLL, COP.

THẢO LUẬN

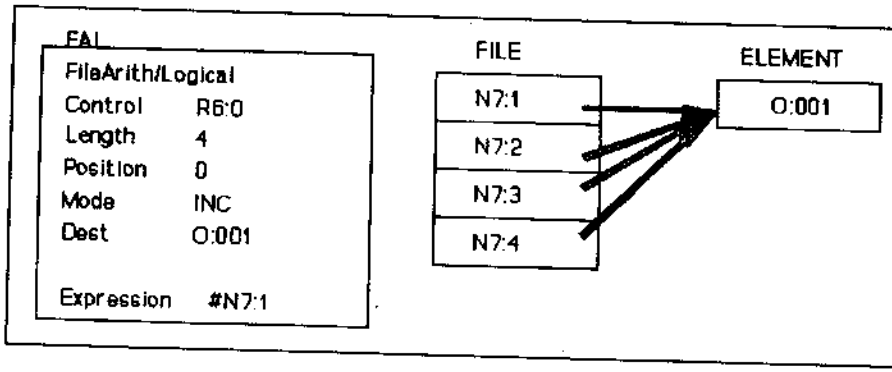
Tập tin

Tập tin là nhóm các từ ngữ bảng dữ liệu liên tiếp được dùng để lưu thông tin. Mỗi phần tử 16-bit trong tập tin có thể được mã hóa với thông tin khác nhau. Thông tin này được dùng để thực hiện các phép toán xác định theo địa chỉ nguồn và các toán tử bạn viết trong biểu thức, liệt kê trong các lệnh.

Trong ví dụ phần tử tập tin (Hình 15.1), mỗi chuyển tiếp từ sai sang đúng trên bậc thang, bộ xử lý sẽ được một phần tử của tập tin số nguyên N7: 1 bắt đầu từ phần tử zero, sẽ lần lượt truyền dữ liệu của từng phần tử vào phần tử đích O:001. Lệnh này viết qua dữ liệu bất kỳ trong phần đích.

Chế độ vận hành

Có ba chế độ vận hành dựa trên tốc độ phần tử dịch chuyển qua từng lần quét.



Hình 15.1. Lệnh Tập Tin – Phần Tử

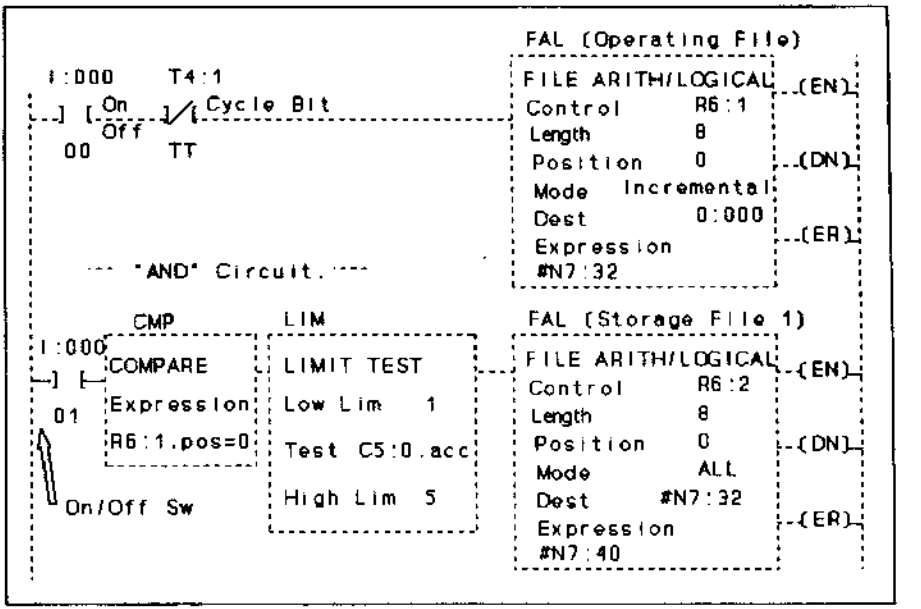
1. Thứ nhất là chế độ ALL. Trong chế độ này, tốc độ/lần quét bằng giá trị chiều dài tập tin. Toàn bộ tập tin được vận hành trong một lần quét. Ví dụ, nếu có 8 phần tử trong tập tin, cả 8 phần tử đều được xử lý trong một lần quét.
2. Trong chế độ NUMERIC, tốc độ/lần quét nhỏ hơn giá trị chiều dài tập tin. Sự xử lý tập tin được phân phối theo số lần quét chương trình. Ví dụ, nếu có tám phần tử trong tập tin và tốc độ/lần quét là hai, mỗi lần quét sẽ xử lý 2 từ ngữ.
3. Chế độ INCREMENTAL, xử lý một phần tử mỗi khi bậc thang chương trình chuyển từ sai sang đúng.

Bài tập

Bài tập này dùng các lệnh FAL để tạo lập hệ thống lưu trữ. Thông tin chứa ba quá trình khác nhau được lưu trong ba lệnh FAL, chúng được lập trình để vận hành theo kiểu tập tin – tập tin. Tập tin lưu trữ chuyển các từ ngữ thông tin đến tập tin xử lý (tập tin – phần tử). Tập tin xử lý cho phép thông tin phân phối đến ngõ ra thực (Hình 15.2).

Bạn hãy chú ý bậc thang thứ nhất trên Hình 15.2. Đây là tập tin xử lý lệnh FAL được lập trình để vận hành theo kiểu lệnh tập tin – phần tử. Lệnh FAL sẽ dịch chuyển 8 từ ngữ của các lệnh bắt đầu ở tập tin #N7:32, mỗi lần tăng một từ ngữ với bộ thời chuẩn, đến từ ngữ lệnh xuất (O:000).

Lệnh FAL thứ hai trên Hình 15.2 được dùng làm tập tin lưu trữ (tập tin – tập tin). Khi bậc thang này là đúng, lệnh FAL truyền 8 từ ngữ từ tập tin #N7:40 đến tập tin #N7:32 trong một lần quét.



Hình 15.2. Các bậc thang tập tin xử lý, quá trình lưu trữ FAL

Bậc thang này chỉ đúng khi: (1) giá trị ACC của bộ đếm trong khoảng 1 đến 5 khi kiểm tra giới hạn. (2) từ ngữ điều khiển FAL vận hành (R6:1) là zero; (3) công tắc ngõ vào (I: 000/01) đúng.

LOGIC CHƯƠNG TRÌNH

1. Công tắc ON/OFF để dừng và khởi động hệ thống.
2. Bộ thời chuẩn TON để lập chuỗi thứ tự và điều khiển quá trình vận hành.
3. Lệnh vận hành FAL dịch chuyển 8 từ ngữ thông tin đến lệnh xuất từ ngữ (tập tin – từ ngữ), điều khiển các đèn báo.
4. Ba lệnh lưu trữ FAL (tập tin – tập tin), mỗi lệnh lưu 8 từ ngữ thông tin khác nhau để vận hành:

   A. Dữ liệu của FAL lưu trữ có thể dịch chuyển qua quá trình xử lý chỉ khi lệnh vận hành FAL ở vị trí zero.

   B. Mỗi lệnh FAL phải chạy trong tập tin xử lý trong ít nhất 5 chu kỳ hoàn chỉnh trước khi có thể thay đổi quá trình.

Bạn hãy quan sát thông số chế độ của từng FAL. Các FAL lưu trữ sử dụng chế độ ALL. Điều này cho phép tập tin hoàn tất dịch chuyển trong một chuyển tiếp từ sai qua đúng.

3. Sau khi nhập các bậc thang tập tin lưu trữ, bạn hãy tải các tập tin lưu trữ của mình. Bạn nhập dữ liệu từ Bảng 15-1 vào các xử lý tập tin lưu trữ. Bạn tải các tập tin này vào chế độ giám sát dữ liệu.

STORAGE FAL #1	Binary Mode				Decimal Mode
N7:40	0001	0001	0001	0001	4369
N7:41	0010	0010	0010	0010	8738
N7:42	0100	0100	0100	0100	17476
N7:43	1000	1000	1000	1000	–30584
N7:44	0001	0001	0001	0001	4369
N7:45	0010	0010	0010	0010	8738
N7:46	0100	0100	0100	0100	17476
N7:47	1000	1000	1000	1000	–30584
STORAGE FAL #2	**Binary Mode**				**Decimal Mode**
N7:48	0011	0011	0011	0011	13107
N7:49	1100	1100	1100	1100	–13108
N7:50	0111	0111	0111	0111	30583
N7:51	1000	1000	1000	1000	–30584
N7:52	1100	1100	1100	1100	–13108
N7:53	0011	0011	0011	0011	13107
N7:54	1110	1110	1110	1110	–4370
N7:55	0001	0001	0001	0001	4369
STORAGE FAL #3	**Binary Mode**				**Decimal Mode**
N7:56	1111	1111	1111	1111	–1
N7:57	0001	0001	0001	0001	4369
N7:58	1111	1111	1111	1111	–1
N7:59	1000	1000	1000	1000	–30584
N7:60	1111	1111	1111	1111	–1
N7:61	1001	1001	1001	1001	–26215
N7:62	1111	1111	1111	1111	–1
N7:63	0110	0110	0110	0110	26214

Bảng 15.1. Dữ liệu tập tin lưu trữ, chế độ nhị phân và thập phân.

4. Sau khi nhập thông tin vào các tập tin, bạn hãy tiếp tục hoàn tất chương trình. Nhập mạch thời chuẩn chu kỳ dựa trên mạch trong

Thí nghiệm 8.3. Các bộ thời chuẩn này sẽ điều khiển tốc độ chu kỳ của tập tin vận hành, cho phép bạn thay đổi tốc độ dịch chuyển thông tin tập tin (#N7:32) đến phần tử xuất (O:000).

5. Để hoàn tất Bài tập bạn cần lập trình bộ đếm thuận được kính hoạt bằng done bit điều khiển của tập tin vận hành (R6:1. DN). Bạn cài đặt lại bộ đếm khi số đếm đạt đến 15. Mục đích của bộ đếm và lệnh kiểm tra giới hạn là thời lượng từng tập tin lưu trữ vận hành theo chu kỳ trong chế độ tự động.

6. Bạn hãy chuyển sang chế độ RUN và thử nghiệm chương trình. Bạn hãy quan sát các đèn, chúng thay đổi chuỗi thứ tự sau 5 chu kỳ hoàn chỉnh của từng quá trình lưu trữ. Khi hoàn tất, bạn hãy lưu trữ chương trình của mình lên đĩa mềm, viết và in báo cáo.

Thí nghiệm 15.1. Quá trình lưu trữ #2

Bạn viết chương trình cải tiến chương trình lưu trữ trong Bài tập 15.0 với các bước sau:

1. Tải và đặt lại tên cho chương trình của Bài tập 15.0, chẳng hạn Exp 15-1

2. Thay các lệnh FAL của ba tập tin lưu trữ bằng lệnh chép (COP) tập tin. Bạn hãy sử dụng cùng các địa chỉ (không thay đổi).

3. Xóa ba lệnh kiểm tra giới hạn.

4. Xóa bộ đếm thuận và lệnh cài đặt lại bộ đếm này.

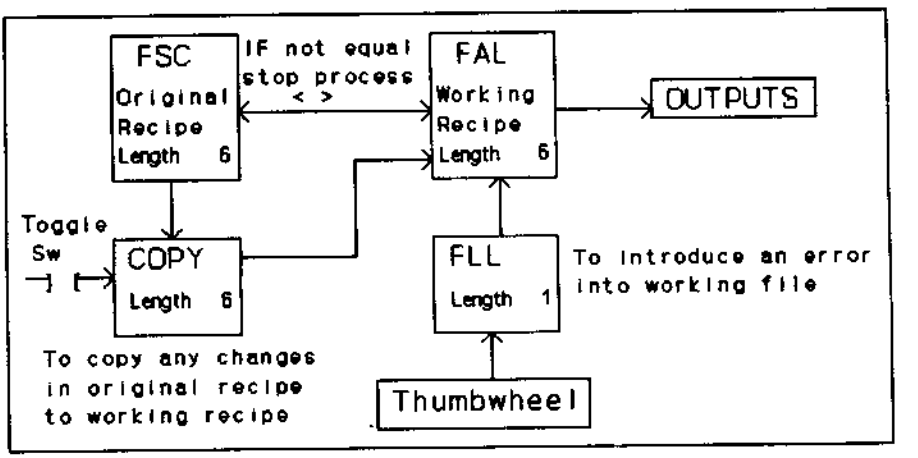
5. Khóa liên động các công tắc toggle để bảo đảm mỗi lần chỉ dịch chuyển một tập tin lưu trữ. Sử dụng các công tắc chạy/dừng Toggle, bạn có thể dịch chuyển bằng tay các tập tin lưu trữ COP đến tập tin vận hành theo yêu cầu.

6. Lưu chương trình của bạn với trên đã đặt ở bước 1 (Exp 15-1).

Bạn có khả năng sao chép (copy) một trong ba quá trình lưu trữ vào quá trình xử lý bằng cách chọn và đóng công tắc tương ứng. Các lệnh so sánh cho phép bạn chỉ sao chép ở vị trí zero. Nếu xóa lệnh so sánh, các tập tin sao chép sẽ kích hoạt ở vị trí bất kỳ.

Thí nghiệm 15.2. CÔNG THỨC BÁNH MÌ HUNGARY

Bạn được yêu cầu viết chương trình cho dây chuyền sản xuất bánh mì Hungary. Vấn đề chính là so sánh biến thiên mẻ bánh giữa công thức gốc (FSC) và công thức thực tế (FAL) trong khi vận hành quy trình Hình 15.4 minh họa lưu đồ kiểm tra công thức mẻ bánh.



Hình 15.4. Lưu đồ công thức mẻ bánh

Chương trình này yêu cầu sử dụng lệnh tìm kiếm/so sánh tập tin (FSC). Lệnh FSC là lệnh xuất, so sánh các giá trị trong các tập tin nguồn. Lệnh này so sánh từng phần tử theo phép toán logic do bạn chọn trong biểu thức khi lệnh tìm được sự so sánh chuyên biệt là đúng, sẽ xác lập bit phát hiện (FD) và bit ngăn chặn (IN). Bit vị trí (POS) cũng được xác lập để ghi lại nơi phát hiện sự so sánh đó.

Các thông số của chương trình bao gồm:

1. Bạn dùng lệnh FAL để lập chuỗi thứ tự công thức thực tế tại địa chỉ xuất (tập tin - phần tử). Công thức bánh đòi hỏi tập tin 6 – phần tử. Điều này tương tự FAL vận hành trong Bài tập 15.0. Bạn hãy tạo lập bộ tập tin mới (N9) trong trạng thái bộ nhớ và xếp các tập tin vào bộ này.

2. Bạn thiết lập so sánh không bằng (nhỏ hơn hoặc lớn hơn, < >) sử dụng lệnh FSC. Lệnh FSC tăng dần gia số đồng thời với FAL qua từng phần tử. Bạn dùng công tắc toggle hoặc bộ thời chuẩn chu kỳ để tạo các bước cho tập tin FAL và FSC. Điều này sẽ so sánh tập

tin nguồn công thức gốc (FSC) với tập tin nguồn công thức thực (FAL). Hai tập tin này phải bằng nhau vào mọi thời điểm.

Bạn hãy kích hoạt đèn cảnh báo và dừng quy trình khi các phần tử tập tin không bằng nhau. Bạn lập trình sự cài đặt lại bằng tay để mở khóa (U) bit ngăn chặn FSC (IN) khi phát hiện điều kiện không bằng nhau.

3. Sử dụng bộ chuyển mạch Thumbwheel và lệnh điền đầy tập tin (FLL) để thay đổi một phần tử trong tập tin công thức thực. Điều này sẽ tạo ra sự khác biệt trong công thức thực khi so sánh với tập tin công thức gốc. Lệnh sao chép tập tin (COP) có thể được dùng để truyền công thức gốc cho tập tin công thức thực tế.

Khi hoàn tất, bạn hãy giám sát quy trình này trên màn hình dữ liệu. Chương trình của bạn sẽ phải tăng dần dữ liệu theo gia số từ các tập tin FAL cho các đèn ở ngõ ra. Các tập tin FSC sẽ phải giám sát thao tác này và dừng hệ thống nếu xảy ra sự không bằng nhau trong tập tin bất kỳ.

Bài tập lớn 16. 0

HỆ THỐNG BĂNG TẢI

MỤC ĐÍCH

Xây dựng và viết chương trình mạch điều khiển và vận hành hệ thống băng tải.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Thực thi các lệnh tăng/giảm một lần.
- Sử dụng các bộ cảm biến quang điện.
- Nâng cao kiến thức lập trình hệ thống băng tải.

TÀI LIỆU

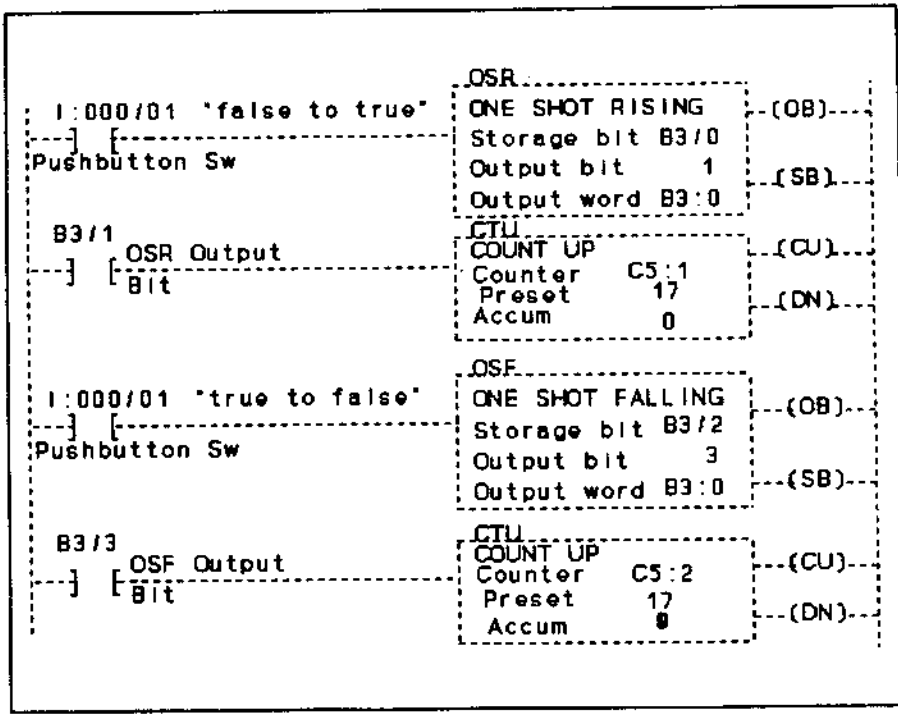
Để biết chi tiết về các lệnh được dùng trong Bài tập này bạn hãy đọc Phụ lục A: Các lệnh điều khiển chương trình, thời chuẩn, đếm, và so sánh, TON, CTU, LIM, MCR, OSR, và OSF.

THẢO LUẬN

Lệnh chuyển đổi biên một lần chỉ thay đổi trạng thái tại điểm chuyên biệt của lệnh nhập chuyển đổi. Lệnh tăng một lần thay đổi trạng thái ở biên dương (biên tăng) của xung nhập. Lệnh giảm một lần thay đổi trạng thái biên âm (biên giảm) của xung nhập. Cả hai chỉ nhạy với các tín hiệu nhập tương ứng tại phần chuyển tiếp của xung nhập. Các thay đổi ở tín hiệu xuất chỉ xảy ra một lần đồng bộ với sự chuyển tiếp của các xung nhập.

Các biến đổi một lần có thể được dùng ở nơi cần xung nhập rõ ràng cho sự vận hành tin cậy. Các hiệu ứng *nảy bật* (tăng, giảm) là đặc biệt quan trọng khi chuyển đổi ngõ vào bộ chuyển mạch cơ khí.

Hình 16.1 minh họa sự khác biệt giữa tăng và giảm được kích hoạt bằng nút nhấn cơ học I:000/01. Khi nhấn nút này, bộ đếm C5:1 sẽ tăng một số đếm. Khi nhả nút nhấn, bộ đếm C5:2 sẽ tăng một số đếm. Sử dụng các lệnh này với các bộ nhập cơ học sẽ bảo đảm vận hành tin cậy hơn.



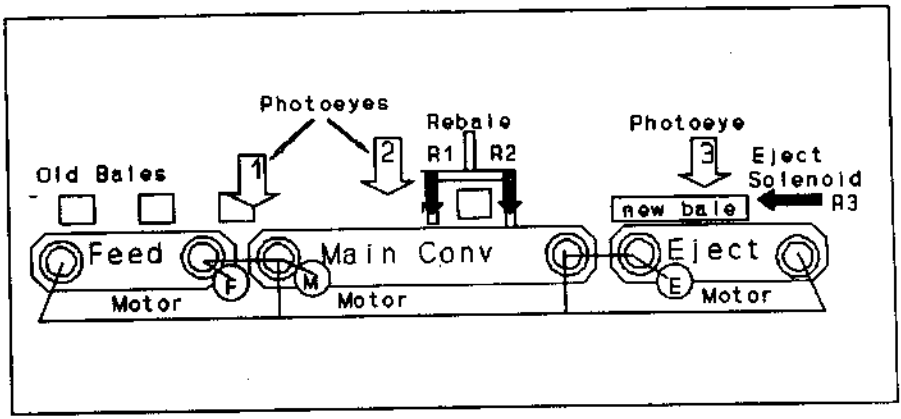
Hình 16.1. Các lệnh tăng và giảm một lần

Bài tập – Băng tải kiện hàng

Bài tập này gồm ba băng tải, băng tải chính dài 15 foot với động cơ 3-pha, 480V, 2 Hp, băng tải nạp 5 foot dùng động cơ 1.5 Hp, và băng tải xuất với động cơ 1.5 Hp.

Chú ý: Các đèn có thể được dùng để mô phỏng băng tải

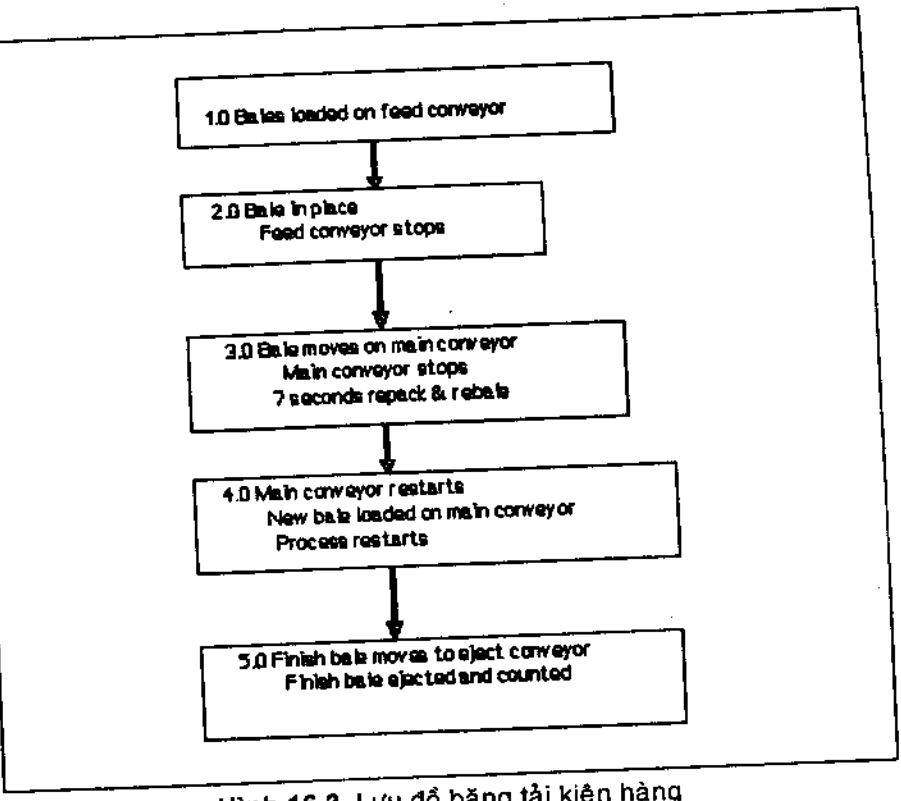
Sự lập trình Bài tập này yêu cầu tinh chỉnh các bộ khởi động của động cơ 3 pha, các bộ cảm biến quang điện và các lệnh PLC để tương tác với hệ thống băng tải. (Hình 16.2). Các kiện hàng cũ được đặt lên băng tải nạp. Băng tải này đưa các kiện hàng cũ đến băng tải chính để xử lý. Cảm biến quang ở phía trước băng tải chính xác định các kiện hàng sẵn sàng chuyển đến băng tải chính. Băng tải chính chuyển kiện hàng đến bộ phận xử lý. Khi hoàn tất, các kiện hàng được đưa đến băng tải xuất để xuất và đếm. Đồng thời, kiện hàng mới chuyển đến băng tải chính để tiếp tục xử lý.



Hình 16.2. Băng tải kiện hàng

LOGIC CHƯƠNG TRÌNH

1. Hình 16.3 minh họa chuỗi thứ tự điều khiển. Các kiện mới được tải lên băng tải nạp. Kiện mới sẽ tải lên mỗi khi có kiện hàng rời khỏi băng tải nạp.
2. Băng tải nạp chuyển các kiện hàng cũ đến tải chính
   A. Cảm biến quang #1 xác định vị trí kiện hàng và sẵn sàng chuyển đến băng tải chính.
   B. Băng tải nạp tải kiện hàng lên băng tải chính khi hoàn tất sự đóng hàng vào kiện trên băng tải chính.
3. Băng tải chính khởi động và chuyển kiện đến nơi đóng kiện và bao gói (R2).
   A. Cảm biến quang #2 được dùng để dừng băng tải ở (R1) và (R2) trong 7 giây: 3 giây để đóng hàng, 3 giây để bao gói, và 1 giây tạm dừng.
   B. Trạm R1 và R2 cách băng tải nạp khoảng 2 phút
4. Băng tải chính khởi động lại sau khi hoàn tất đóng kiện và bao gói.
   A. Kiện hàng mới hoàn thành được chuyển đến băng tải xuất
   B. Băng tải nạp khởi động và kiện hàng khác được chuyển đến băng tải chính để xử lý.

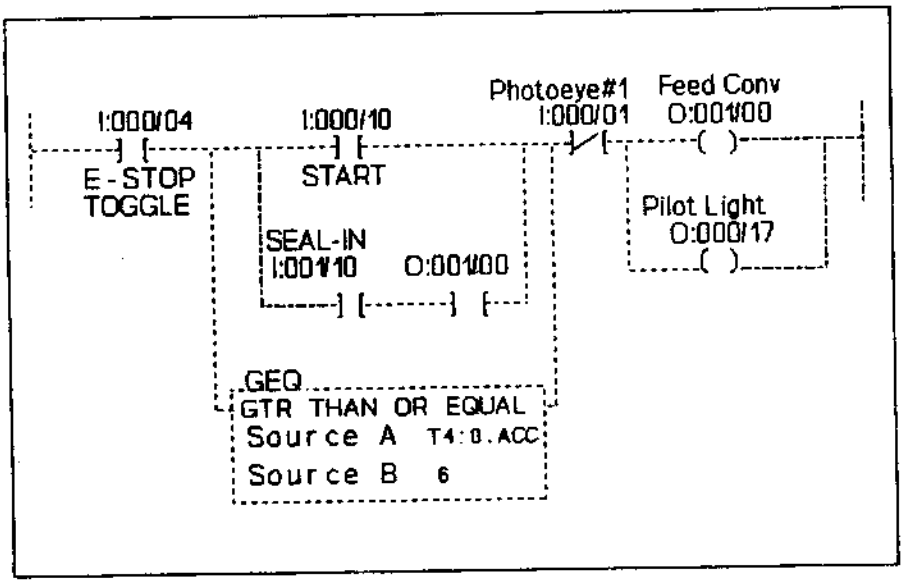


Hình 16.3. Lưu đồ băng tải kiện hàng

5. Cuộn solenoid xuất đưa băng tải hoàn tất đến mấng trượt xuất.
   A. Cảm biến quang #3 xác định kiện hàng đến cuối băng tải sẵn sàng để xuất và đếm.
   B. Băng tải xuất chạy liên tục.

CÁC BƯỚC THỰC HIỆN

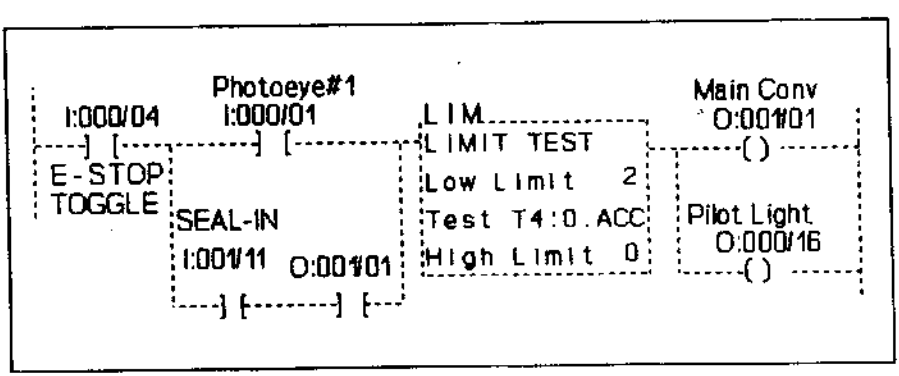
1. Bạn lập trình hệ thống băng tải đóng kiện hàng (Hình 16.2) đã nêu trong logic chương trình, sử dụng công tắc toggle để kích hoạt lệnh MCR trên bậc thang thứ nhất.
2. Bước đầu tiên là lập trình mạch Stop – Start cho băng tải nạp. Mạch này có khả năng khởi động bằng tay và tự động cho băng tải nạp. Bạn hãy quan sát mạch khởi động trên Hình 16.4. Lệnh



Hình 16.4. Bậc thang băng tải nạp

bằng nhau (Equal) được đưa vào mạch này để tái khởi động tự động.

3. Bước kế tiếp là khởi động băng tải chính và dừng băng tải nạp khi kiện hàng đến bộ cảm biến quang #1. Các cảm biến quang điện có thể được mô phỏng bằng cách dùng công tắc toggle (Hình 16.5). Bạn hãy quan sát lệnh kiểm tra giới hạn. Mục đích của lệnh này là dừng băng tải chính khi cảm biến quang #2 phát hiện



Hình 16.5. Bậc thang băng tải chính

kiện hàng đang đến với thời gian trễ 1 giây để kiện hàng đó đi qua cảm biến quang #2.

4. Bạn lập trình và cảm biến quang #2 để điều khiển thời lượng xác lập trước của bộ thời chuẩn đóng (TON) là 7 giây, sử dụng TT bit để đóng bộ thời chuẩn này.

5. Lập trình hai lệnh kiểm tra giới hạn để kiểm tra giá trị tích lũy của bộ thời chuẩn nhằm điều khiển các cuộn solenoid (R1) và (R2). Các lệnh kiểm tra giới hạn này cho phép dễ dàng điều chỉnh thời lượng của hai cuộn solenoid. Bạn xác lập các giới hạn ở R1 và bao gói (R2).

6. Bước cuối cùng trong chương trình là băng tải xuất và cuộc solenoid. Bạn hãy lập trình mạch Stop – Start cho băng tải này chạy liên tục. Sử dụng cùng địa chỉ E-stop như băng tải nạp và băng tải chính. Cảm biến quang #3 kích hoạt bộ đếm, bạn dùng CM bit bộ đếm để kích hoạt cuộn solenoid xuất, xác lập lại bộ đếm bằng tay.

7. Sau khi hoàn tất, bạn hãy kiểm tra và chạy thử chương trình, tuân theo thứ tự đã nêu trong logic chương trình, ghi lại và lưu chương trình của bạn.

8. Cuối cùng bạn viết báo cáo và in chương trình.

ÔN TẬP

Để củng cố và nâng cao kiến thức bạn và hãy thực hành thí nghiệm 16.1

Thí nghiệm 16.1. BĂNG TẢI HỘP

Trong xí nghiệp, các băng tải hộp được dùng phổ biến, các hộp có thể bị kẹt hoặc bị xếp chồng trên băng tải. Nhiệm vụ của bạn là thiết kế và lập trình mạch linh hoạt hệ thống cảnh báo khi điều đó xảy ra. Các thông số vận hành bao gồm:

1. Khi vận hành bình thường, băng tải chuyển từng hộp qua cảm biến quang #1 trong ba giây, bạn hãy dùng mạch Stop – Start để khởi động băng tải.

2. Tình huống 1 — khi không có hộp nào qua cảm biến #1 trong 5 giây trở lên, hệ thống cảnh báo sẽ được kích hoạt. Nếu lâu hơn 10 giây băng tải sẽ dừng lại.

3. Tình huống 2 — Nếu hộp bị kẹt ở phía trước cảm biến #1 hơn 5 giây, hệ thống cảnh báo sẽ kích hoạt, nếu lâu hơn 10 giây, băng tải dừng lại. Để khởi động lại, nút cài đặt lại sẽ mở khóa cho mạch này.

4. Lập trình công tắc giới hạn #1 để đếm số lượng các hộp, bạn hãy dùng lệnh tăng một lần để kích hoạt bộ đếm. Lệnh tăng một lần yêu cầu bộ xử lý PLC5/11 hoặc cao hơn. Nếu không có bộ xử lý, bạn hãy bỏ qua lệnh này.

Bài tập lớn 17.0

DÂY CHUYỀN SƠN PHUN

MỤC ĐÍCH

Viết chương trình bộ ghi dịch chuyển và hệ thống mã hóa để điều khiển hệ thống sơn phun tự động các sản phẩm trên dây chuyền xích tải.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Hiểu về khái niệm bit dịch chuyển trái và phải.
- Sử dụng lệnh dịch chuyển ẩn để mã hóa ngõ ra của dây chuyền.

TÀI LIỆU

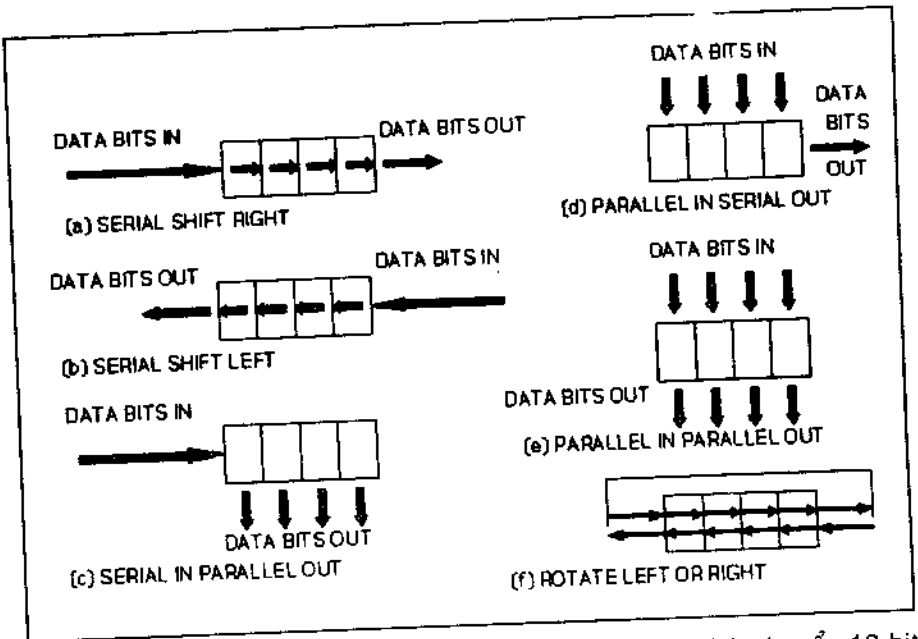
Để hiểu và biết cách sử dụng các lệnh trong bài tấp này bạn hãy đọc Phụ lục A: Bit Modify and Move, Bit Shift, và các lệnh điều khiển chương trình. MVM, BSL, BSR, OSR, OSF, ONS.

THẢO LUẬN

Bộ ghi dịch chuyển là rất quan trọng trong các ứng dụng cần lưu trữ và truyền dữ liệu trong hệ thống điều khiển. Sự khác biệt cơ bản giữa bộ ghi và bộ đếm là bộ ghi không có chuỗi thứ tự chuyên biệt các trạng thái. Nói chung, bộ ghi chỉ được dùng để lưu và dịch chuyển dữ liệu (các số 1 và 0) nhập từ nguồn bên ngoài.

Dung lượng lưu trữ của bộ ghi dịch chuyển là số lượng chữ số 1 và 0 của dữ liệu số có thể chứa trong bộ ghi. Mỗi tầng của bộ ghi dịch chuyển biểu thị một bit dung lượng lưu trữ, do đó số lượng tầng này xác định tổng dung lượng lưu trữ.

Khả năng dịch chuyển của bộ ghi cho phép dịch chuyển dữ liệu từ tầng này qua tầng khác bên trong bộ ghi tùy theo ứng dụng của xung (đồng hồ) dịch chuyển. Hình 17.1 minh họa các kiểu dịch chuyển dữ liệu trong bộ ghi dịch chuyển. Khối chữ nhật biểu thị bộ ghi 4 bit bất kỳ và mũi tên là chiều dịch chuyển dữ liệu. Lệnh dịch chuyển bit sang trái được dùng trong Bài tập này vận hành tương tự dịch chuyển vào nối tiếp ra song song.



Hình 17.1. Dịch chuyển dữ liệu cơ bản sử dụng bộ ghi dịch chuyển 12-bit

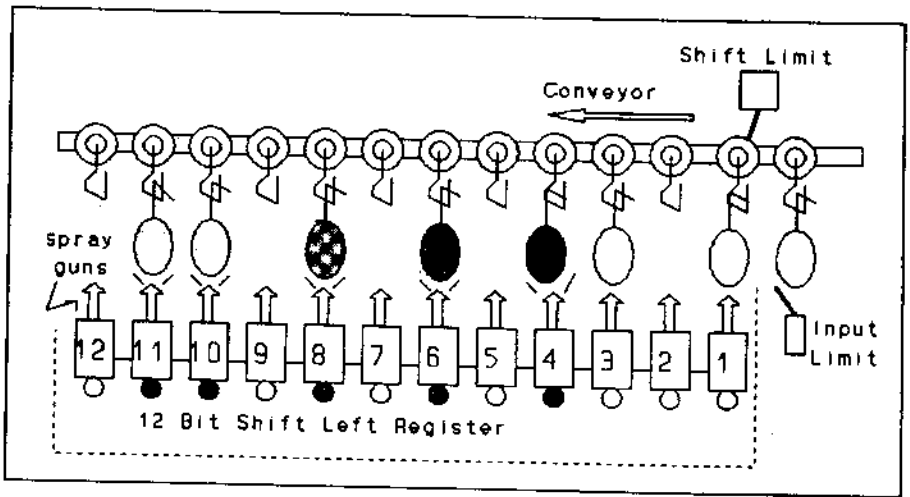
Bài tập

Trong Bài tập này, các đầu phun sơn được điều khiển từ phía ngoài buồng sơn (Hình 17.2). Tín hiệu ghi dịch chuyển được lưu và theo từng móc treo sản phẩm sơn. Khi các móc treo này đến đầu phun tương ứng, cuộn solenoid bên ngoài sẽ kích hoạt đầu phun sơn.

Công tắc giới hạn ngõ vào lắp ở phía buồng sơn sẽ kích hoạt khi sản phẩm ở trên móc treo. Điều này sẽ nhập tín hiệu vào bộ ghi dịch chuyển. Công tắc giới hạn dịch chuyển lắp trên dây chuyền, được kích hoạt theo từng móc treo dù có hay không có sản phẩm trên móc. Công tắc này vận hành theo chiều tiến của chuỗi tín hiệu đi qua bộ ghi dịch chuyển. Tín hiệu này chạy theo sản phẩm trên dây chuyền. Khi tín hiệu bộ ghi dịch chuyển đến tầng thích hợp, sẽ kích hoạt đầu phun để phun sơn lên bề mặt sản phẩm. Đầu phun được chọn theo mã số nhập trong lệnh dịch chuyển ẩn.

LOGIC CHƯƠNG TRÌNH

1. Yêu cầu chính là viết chương trình cho mạch logic sử dụng ghi



Hình 17.2. Hệ thống dây chuyền phun sơn sử dụng bộ ghi dịch chuyển 12-bit

dịch chuyển trái 12-bit nhập để điều khiển 12 đầu phun sơn. Nút nhấn stop-start được dùng để khởi động dây chuyền (xích tải) sơn phun.

2. Công tắc giới hạn ngõ vào được dùng để phát hiện sản phẩm trên dây chuyền khi công tắc này phát hiện sản phẩm, bậc thang sẽ đúng và bit lưu trữ được xác lập (bit dữ liệu vào). Nhiều kiểu cảm biến, chẳng hạn tế bào quang điện hoặc công tắc tiếp xúc, được dùng để tạo ra tín hiệu nhập tương ứng. Trong Bài tập này, nút nhấn được dùng để mô phỏng công tắc giới hạn. Bạn viết chương trình công tắc nhập với lệnh tăng một lần.

3. Công tắc giới hạn dịch chuyển được dùng để định giờ cho lệnh ghi dịch chuyển. Sự kích hoạt bậc thang chứa lệnh dịch chuyển từ sai sang đúng sẽ dịch chuyển bit của bộ ghi. Bạn lập trình công tắc này với lệnh giảm một lần.

4. Lệnh dịch chuyển ẩn được dùng để xác lập hệ thống mã hóa. Mã này cho phép bạn xác định solenoid xuất hoạt động. Các solenoid xuất điều khiển kiểu loại và màu sơn. Bạn dùng mã thập lục phân 4-bit để xác lập đầu phun hoạt động. Các đèn có thể được dùng để mô phỏng solenoid sơn.

5. Bạn lập trình hệ thống đếm các sản phẩm khi ra khỏi bộ ghi dịch chuyển.

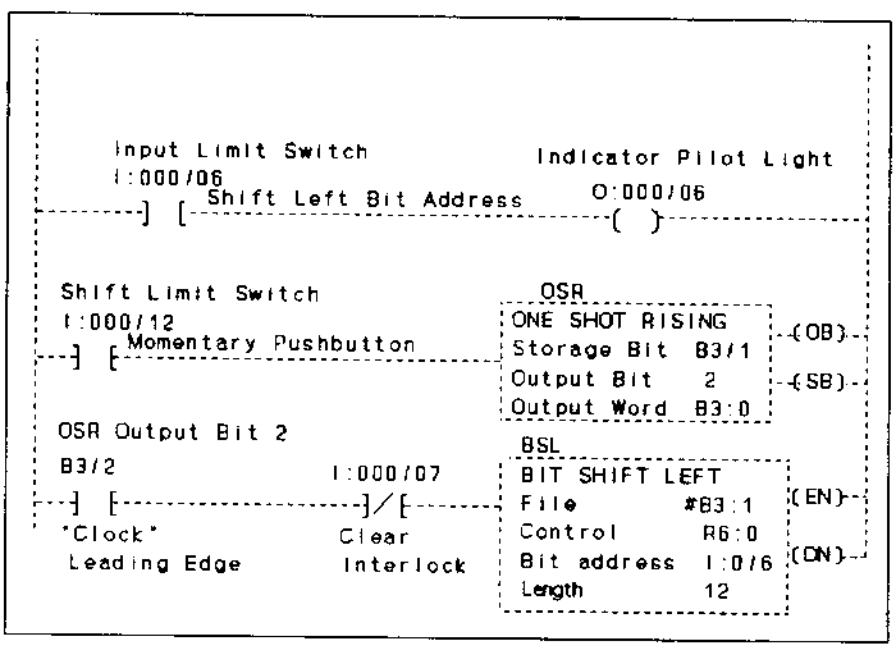
CÁC BƯỚC THỰC HIỆN

1. Bậc thang thứ nhất chứa công tắc toggle kích hoạt lệnh MCR. Trong bậc thang kế tiếp, bạn dùng mạch stop – start để điều khiển bộ khởi động dây chuyền. Bạn hãy dùng địa chỉ công tắc toggle làm địa chỉ stop.

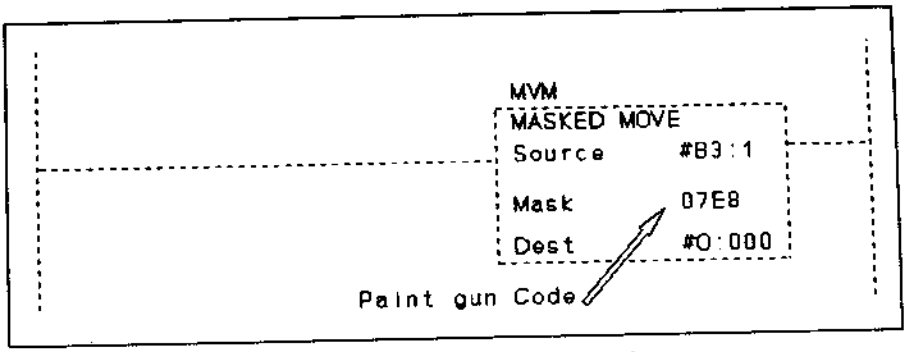
2. Đối với ba bậc thang kế tiếp, bạn hãy nhập chương trình được nêu trên Hình 17.3. Bậc thang thứ nhất chứa công tắc giới hạn ngõ vào kích hoạt đèn báo và bit tải của bộ ghi dịch chuyển. Trong thang kế tiếp, lệnh dịch chuyển trái được định giờ trên dốc lên với bit xuất một lần.

   Chú ý: Các bộ PLC 5/10 và PLC 5/15 không hỗ trợ các lệnh OSR và OSF. Nếu bạn dùng loại PLC này, hãy bỏ qua lệnh dùng một lần và điều khiển lệnh dịch chuyển trực tiếp với công tắc.

3. Kế tiếp, bạn nhập lệnh dịch chuyển ẩn (Hình 17.4). Chú ý địa chỉ ẩn là mã thập lục phân được dùng để điều khiển các bit xuất. Mã này chỉ cho phép các bit xác định từ bộ ghi dịch chuyển đi qua và kích hoạt các bit xuất tương ứng.



Hình 17.3. Các bậc thang lệnh dịch chuyển trái.



Hình 17.4. Lệnh dịch chuyển ẩn.

4. Bạn hoàn tất phần còn lại của chương trình bằng cách bổ sung đếm để đếm các sản phẩm rời khỏi bộ ghi. Số sản phẩm xác lập trước ở bộ đếm là 16. Bạn dùng unload bit (R6: 0.L) và done bit (R6: 0.DD) với phép AND (nối tiếp) điều khiển bộ đếm, dùng done bit để xác lập lại bộ đếm.

5. Cuối cùng, trên thang riêng, bạn hãy nhập lệnh Clear kích hoạt bằng công tắc toggle để khử bộ ghi dịch chuyển.

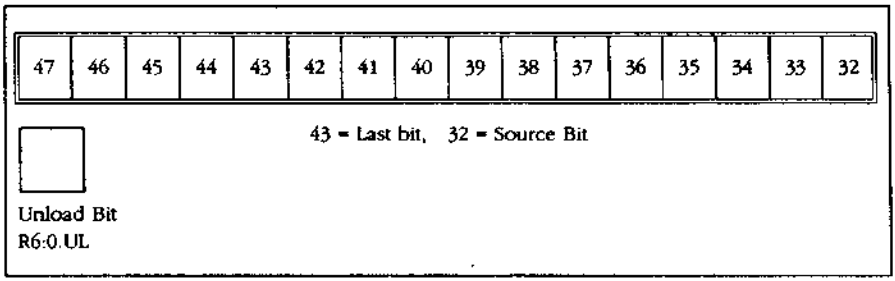
6. Bạn chuyển sang chế độ Run và kiểm tra sự vận hành của bộ ghi dịch chuyển theo các bước dưới đây:

   A. Kích hoạt công tắc giới hạn ngõ vào I: 000/06 để kích hoạt bit nguồn nhập. Đèn báo được dùng để kiểm chứng sự kích hoạt bit này.

   B. Nhấn một lần công tắc giới hạn ghi I:000/12. Điều này sẽ đưa bit nguồn nhập thứ nhất vào tập tin lệnh dịch chuyển trái, đây là số có nghĩa thấp nhất của tập tin (32 - bit). Khi bạn duy trì sự tạo xung, công tắc dịch chuyển tiếp tục đưa bit nhập lần lượt qua các vị trí tiếp đến cuối từ ngữ tập tin. Unload bit sẽ được xác lập khi nhập đến cuối chiều dài của mình (Hình 17.5).

7. Bạn hãy chuyển sang chế độ PROGRAM và chèn thêm mã ẩn. Khử bộ ghi dịch chuyển và lập lại các bước nêu trên. Các đèn ở ngõ ra O:000 sẽ được kích hoạt tùy theo mã ẩn. Bạn hãy so sánh các quan sát của mình với mã ẩn này.

8. Chương trình hoàn tất, để củng cố và nâng cao kiến thức đã học trong Bài tập này, bạn hãy thiết kế và viết chương trình cho các thí nghiệm kế tiếp.



Hình 17.5. Tập tin bit dịch chuyển trái 12-bit.

Thí nghiệm 17.1. SỰ DỊCH CHUYỂN VÒNG

Bạn hãy thiết kế và viết chương trình để vận hành 16 bit trong hệ thống dịch chuyển phải. Lập trình sự dịch chuyển này theo vòng trái tuần hoàn một hoặc nhiều bit trên cơ sở liên tục. Bạn lập trình sự dịch chuyển với các thông số dưới đây.

1. Nút nhấn để định giờ lệnh tăng một lần (OSR). Bạn dùng bit xuất OSR để định giờ cho bộ dịch chuyển phải, tập tin B3:2 lập địa chỉ cho tập tin bộ ghi dịch chuyển.
2. Công tắc toggle để điều khiển ngõ ra. Bạn hãy lập địa chỉ ngõ ra này theo bit nguồn nhập. Lập trình unload bit (R6:1UL) của tập tin điều khiển song song (phép toán OR) với công tắc toggle. Điều này sẽ nạp lại bộ ghi dịch chuyển một cách tự động.
3. Lập trình côntg tắc toggle để khử bộ ghi, bạn hãy dùng lệnh Clear.

Sau khi hoàn thành, bạn có khả năng tải bit trong bộ ghi dịch chuyển và tiếp tục dịch chuyển bit này cho đến khi được nạp lại và tiếp tục dịch chuyển.

Thí nghiệm 17.2.
BỘ GHI DỊCH CHUYỂN PHẢI VÀ TRÁI

Bạn thiết kế và viết chương trình cho mạch điều khiển để vận hành hệ thống điều khiển dịch chuyển trái và phải 16-bit, với các thông số dưới đây:

1. Công tắc toggle để tải lệnh dịch chuyển trái, công tắc toggle thứ hai để tải lệnh dịch chuyển phải. Bạn dùng tập tin B3:3 cho bộ dịch chuyển, nhưng không sử dụng lệnh tăng một lần trong thí nghiệm này.
2. Nút nhấn để dịch chuyển trái và nút nhấn để dịch chuyển phải, bạn chỉ tải một bit cho trong bộ bất kỳ (phải hoặc trái)
3. Lập trình bộ thời chuẩn ngắt (TOF) được kích hoạt bằng một trong các bit trong thanh ghi dịch chuyển.
4. Lập trình bộ thời chuẩn đóng (TON) được kích hoạt bằng bit khác trong bộ ghi dịch chuyển.
5. Lập trình lệnh dịch chuyển để chuyển tập tin B3:3 thành tập tin số nguyên.

Khi hoàn tất, bạn sẽ có khả năng tải bit vào bộ ghi dịch chuyển và thực hiện dịch chuyển bit này sang phải hoặc trái. Bạn hãy quan sát hai bộ thời chuẩn và so sánh sự vận hành của chúng. Bộ thời chuẩn ngắt (TOF) bắt đầu chu kỳ thời chuẩn khi bit dịch chuyển đi qua bằng bit điều khiển bộ thời chuẩn. Bộ thời chuẩn đóng (TON) kích hoạt khi bit dịch chuyển dừng ở vị trí của bit điều khiển bộ thời chuẩn.

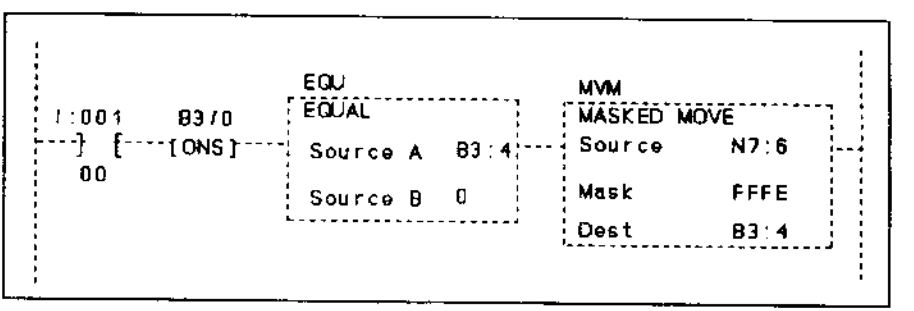
Thí nghiệm 17.3.
DỊCH CHUYỂN VÀO SONG SONG RA NỐI TIẾP

Bạn hãy thiết kế và lập trình mạch điều khiển để vận hành hệ thống điều khiển dịch chuyển trái 16-bit, với các thông số dưới đây:

1. Nhập lệnh dịch chuyển trái 16 bit sử dụng tập tin B3:4. Bit nguồn của bộ ghi dịch chuyển này không được sử dụng. Bạn hãy nhập bit không liên quan cho chương trình này, không dùng lệnh tăng một lần.

2. Lập trình mạch thời chuẩn chu kỳ để định giờ cho bộ ghi dịch chuyển.

3. Sử dụng bộ chuyển đổi thumbwheel để tải tập tin dịch chuyển qua lệnh dịch chuyển ẩn. Bạn dùng công tắc toggle nối tiếp (phép logic AND) với lệnh một lần (ONS) và lệnh bằng nhau (Equal) để điều khiển trên bậc thang dịch chuyển ẩn (Hình 17.6). Mục đích của thang này là tải một số vào tập tin dịch chuyển khi tập tin này bị khử.



Hình 17.6. Bậc thang dịch chuyển ẩn.

4. Lập trình bit ý nghĩa cao nhất (bit cuối) của tập tin dịch chuyển để kích hoạt đèn ở ngõ ra.

Khi hoàn tất, bạn có khả năng tải số 16-bit vào tập tin dịch chuyển với bộ chuyển đổi thumbwheel, sau đó chuyển số này một cách nối tiếp ra khỏi bộ ghi dịch chuyển của bạn.

Bài tập lớn 18. 0

BỘ CHUYỂN TẢI GỖ SÚC

MỤC ĐÍCH

Thực hiện sự chuyển tải đồng bộ với các độ đo thời gian, sử dụng các lệnh so sánh và thời chuẩn đóng.

NỘI DUNG

Sau khi hoàn tất Bài tập này bạn sẽ có khả năng:

- Xây dựng hệ thống chuyển tải gỗ súc với các bộ lấy gỗ
- Nghiên cứu sự dịch chuyển theo thời gian đo.

TÀI LIỆU

Để biết chi tiết các lệnh được dùng trong Bài tập này, bạn hãy đọc Phụ lục A: Các lệnh tập tin và thời chuẩn/đếm, TON,CTU, FAL, và FSC.

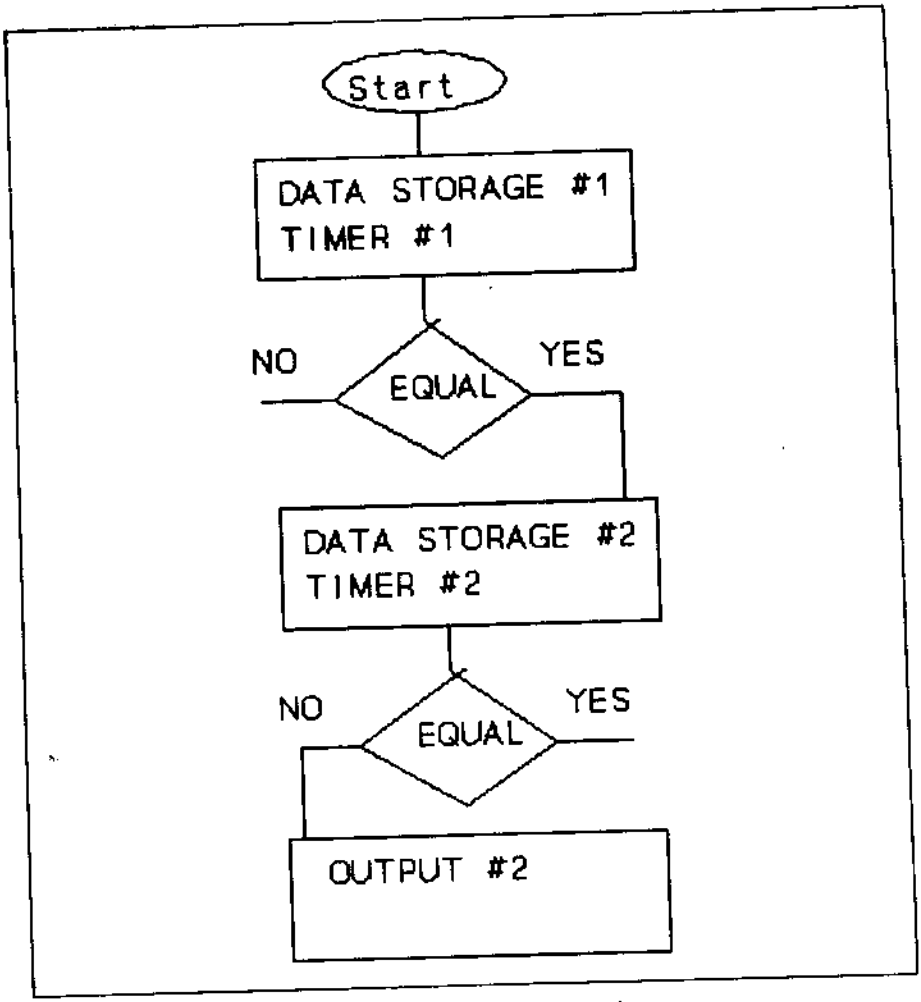
THẢO LUẬN

Lưu trữ và truyền dữ liệu

Nhiều khi trong thiết kế mạch điều khiển bạn cần lưu trữ và truyền (dịch chuyển) dữ liệu. Dữ liệu có thể là từ ngữ hoặc các bit thông tin, giá trị số hoặc các độ đo thời gian. Như bạn đã biết trong các Bài tập 15 và 17, các bit logic hoặc dữ liệu từ ngữ có thể được lưu trong tập tin, sau đó được truyền hoặc dịch chuyển tùy theo logic của chương trình.

Phương pháp thứ hai thường được thực hiện để lưu trữ và truyền dữ liệu là truyền (dịch chuyển) đồng bộ các độ đo thời gian. Điều này yêu cầu sử dụng các bộ thời chuẩn và các lệnh so sánh để chuyển dữ liệu từ bộ thời chuẩn này đến bộ thời chuẩn khác. Hình 18.1 minh họa lưu đồ của quá trình lưu trữ và truyền dữ liệu giữa hai hoặc nhiều bộ thời chuẩn.

1. Bộ thời chuẩn #1 được cấp nguồn khi công tắc Start khởi động để bậc thang là đúng. Bộ thời chuẩn này dùng TT bit để đóng mạch. Trong khi bộ thời chuẩn đang chạy, dữ liệu được lưu trong bộ tích lũy với thời lượng xác định.



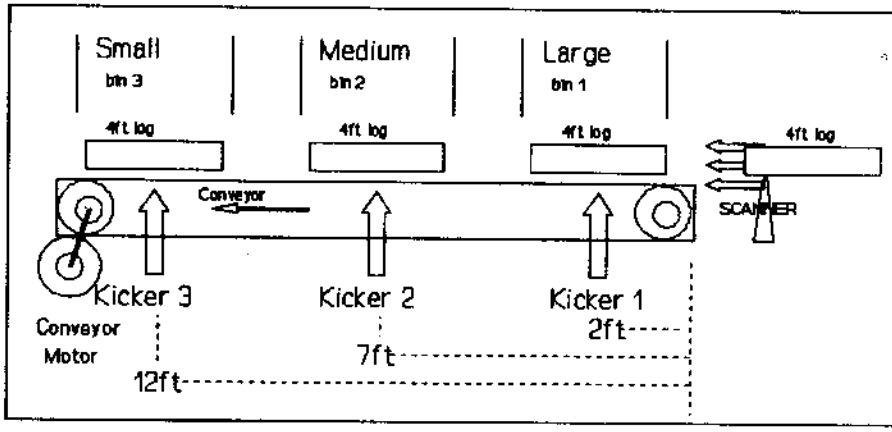
Hình 18.1. Lưu đồ lưu trữ và truyền dữ liệu.

2. Vào thời điểm thích hợp, tùy theo lệnh bằng (Equal), dữ liệu được truyền cho bộ thời chuẩn thứ hai #2. Dữ liệu khi đó được lưu trong giá trị tích lũy của bộ thời chuẩn này. Bộ thời chuẩn #1 có thể được tái kích hoạt để khởi động lại quy trình. Sự truyền (dịch chuyển) được thực thi với các lệnh bằng (Equal). Điều này làm tăng tính linh hoạt để truyền dữ liệu theo các thời lượng khác nhau.

3. Với dữ liệu được truyền cho giá trị tích lũy của thời chuẩn thứ hai có thể kích hoạt ngõ ra hoặc truyền đến bộ thời chuẩn #3 để tiếp tục quy trình. Quy trình này tương tự lệnh dịch chuyển bit. Sự khác biệt là thời lượng trôi qua giữa từng bước dịch chuyển.

Bài tập

Bài tập này yêu cầu viết chương trình cho hệ thống dịch chuyển các súc gỗ dài 4-foot dọc theo băng tải (Hình 18.2). Các súc gỗ được quét để xác định kích cỡ (đường kính). Mạch dịch chuyển (Hình 18.3) xác định vị trí các súc gỗ được lấy ra khỏi băng tải để đưa đến ngăn chứa tương ứng. Các súc gỗ được phân loại theo ba cỡ đường kính: nhỏ, đến 14 inches; trung, 15 - 24 inches; và lớn, bằng hoặc trên 25 inches.



Hình 18.2. Quy trình băng tải gỗ súc.

LOGIC CHƯƠNG TRÌNH

1. Trên bậc thang thứ nhất, bạn nhập công tắc toggle để điều khiển lệnh MCR, sau đó lập trình mạch stop – start để điều khiển bộ khởi động của động cơ điện chạy băng tải.

2. Lập trình bộ cảm biến quang để quét kích cỡ các súc gỗ dịch chuyển trên băng tải. Sắp xếp các bộ cảm biến này theo góc nghiêng đứng, thứ nhất là bộ cảm biến súc gỗ lớn, sau đó trung bình, và cuối dùng và các súc gỗ nhỏ (ba bộ cảm biến quang). Mọi súc gỗ đều dài 4 foot và có kích cỡ phân loại theo đường kính:

A. Nhỏ, đường kính đến 14 inches, bộ cảm biến #1.

B. Trung, đường kính 15-24 inches, bộ cảm biến #2.

C. Lớn, đường kính không dưới 25 inches, bộ cảm biến #3.

3. Bộ khởi động solenoid lấy súc gỗ được dùng cho từng bộ chọn gỗ đưa vào ngăn tương ứng.

   A. Bộ ngắt bằng tay được dùng để vận hành từng bộ chọn lấy gỗ theo kích cỡ.

   B. Bộ thời điểm chuẩn 1- giây được dùng lấy các súc gỗ và khởi động lại bộ chọn lấy gỗ.

4. Mạch nhớ được dùng để xác định kích cỡ súc gỗ trên băng tải. Mỗi lần tối đa ba súc gỗ chạy trên băng tải với các tổ hợp có thể như sau:

   A. Số súc gỗ nhỏ, tối đa bằng 3.

   B. Số súc gỗ trung, tối đa bằng 2.

   C. Số súc gỗ lớn, tối đa bằng 1.

   D. Phối hợp bất kỳ các kích cỡ nêu trên.

5. Chuỗi thứ tự vận hành gồm các bước như sau:

   A. Các cảm biến quang quét các súc gỗ đi vào băng tải.

   B. Băng tải chạy với tốc độ 1 foot/giây.

   C. Các vị trí ngăn chứa cách tâm cửa vào băng tải: lớn, 2 feet; trung 7 feet; và nhỏ, 12 feet; đo theo các tâm từ điểm cửa vào.

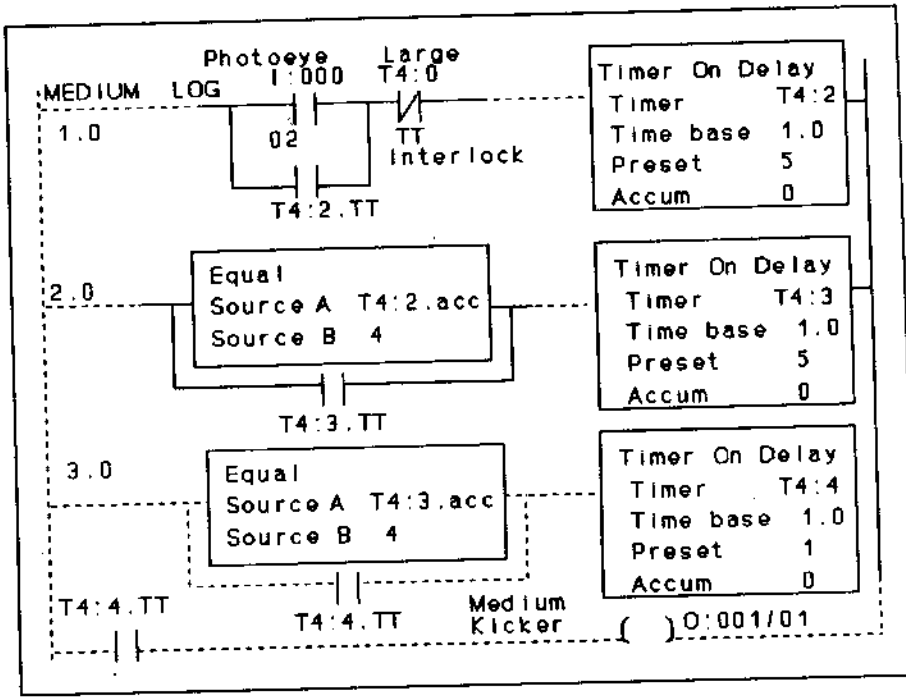
   D. Các súc gỗ cách nhau khoảng 1 foot trên băng tải.

   E. Súc gỗ không chuyển vào ngăn chứa vì lý do bất kỳ sẽ tiếp tục chạy vòng trở lại băng tải để được chọn theo cỡ tương ứng.

CÁC BƯỚC THỰC HIỆN

1. Thiết kế và lập trình hệ thống băng tải phân loại súc gỗ được minh họa trên Hình 18.2. Điều này đòi hỏi hệ thống lưu trữ và dịch chuyển dữ liệu liên tục theo dõi các súc gỗ trên băng tải.

2. Đối với chương trình bậc thang thứ nhất, công tắc toggle được dùng để kích hoạt lệnh MCR. Sau đó, bạn nhập mạch Stop – Start để vận hành bộ khởi động của động cơ chạy băng tải.



Hình 18.3. Các bậc thang lưu trữ và truyền dữ liệu.

3. Bước kế tiếp là nhập chương trình lưu trữ nhớ dữ liệu được minh họa trên Hình 18.3. Đây là mạch cho các súc gỗ cỡ trung. Các đèn báo có thể được dùng để mô phỏng bộ chọn và lấy gỗ.

4. Mạch nhớ trên Hình 18.3 với các chỉnh sửa nhỏ có thể được dùng cho các súc gỗ nhỏ và lớn. Các súc gỗ lớn chỉ cần một bộ thời chuẩn, còn súc gỗ nhỏ cần đến 3 bộ thời chuẩn để liên tục theo dõi các súc gỗ. Bạn lập trình mạch nhớ cho các súc gỗ nhỏ và lớn, sau đó phối hợp chúng với mạch nhớ trên Hình 18.1.

   Cần bảo đảm TT bit của bộ thời chuẩn thứ nhất dùng cho gỗ cỡ trung khóa liên động với cảm biến gỗ nhỏ. Bạn hãy dùng TT bit của bộ thời chuẩn thứ nhất trong mạch nhớ súc gỗ lớn để khóa liên động với các bộ cảm biến súc gỗ nhỏ và trung.

5. Bạn hãy kiểm tra mạch này, bạn có thể phải tinh chỉnh các bộ thời chuẩn để bảo đảm lấy các súc gỗ ở vị trí thích hợp. Bạn hãy thử áp dụng các phối hợp dưới đây:

A. Ba súc gỗ nhỏ liên tiếp trên băng tải.

B. Một nhỏ, một trung, nhỏ, trung.

C. Lớn, trung, trung.

Nếu mạch vận hành chuẩn xác, bạn hãy lưu và ghi lại chương trình.

6. Sau khi hoàn tất phần này của Bài tập 18.0, bạn hãy thực hành với Thí nghiệm 18.1.

Thí nghiệm 18.1. Bộ quét súc gỗ

Hệ thống băng tải và tuyển chọn gỗ được bổ sung với bộ quét súc gỗ bằng chùm tia thay cho các cảm biến quang riêng rẽ. Bộ quét sẽ hiển thị số tỷ lệ với kích cỡ súc gỗ. Bạn sử dụng số này để xác định ngăn chứa gỗ và tuyển chọn súc gỗ tương ứng với các thông số dưới đây:

1. Các khoảng số gồm:

   A. Nhỏ = 100 đến 1440

   B. Trung = 1441 đến 2280

   C. Lớn = 2281 trở lên

2. Các lệnh lập trình:

   A. Bộ chuyển đổi thumbwheel hiển thị các số quét.

   B. Lệnh kiểm tra giới hạn để xác định kích cỡ súc gỗ

   C. Các địa chỉ bit được điều khiển bằng lệnh kiểm tra giới hạn. Bạn dùng các địa chỉ bit này thay cho địa chỉ cảm biến quang trong Bài tập 18.0.

Bài tập lớn 19. 0

BỘ TẠO CHUỖI THỨ TỰ

MỤC ĐÍCH

Viết chương trình để điều khiển sự vận hành máy theo thứ tự, sử dụng các lệnh chuỗi thứ tự nhập và xuất.

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Nghiên cứu khái niệm lệnh chuỗi thứ tự nhập và xuất.
- Sử dụng các lệnh COP để tải quy trình lập chuỗi thứ tự.

TÀI LIỆU

Để biết chi tiết về các lệnh được dùng trong Bài tập này, bạn hãy đọc Phụ lục A: Các lệnh so sánh, tập tin và chuỗi thứ tự. LIM, EQU, FAL, FSC, COP, SQI, SQO.

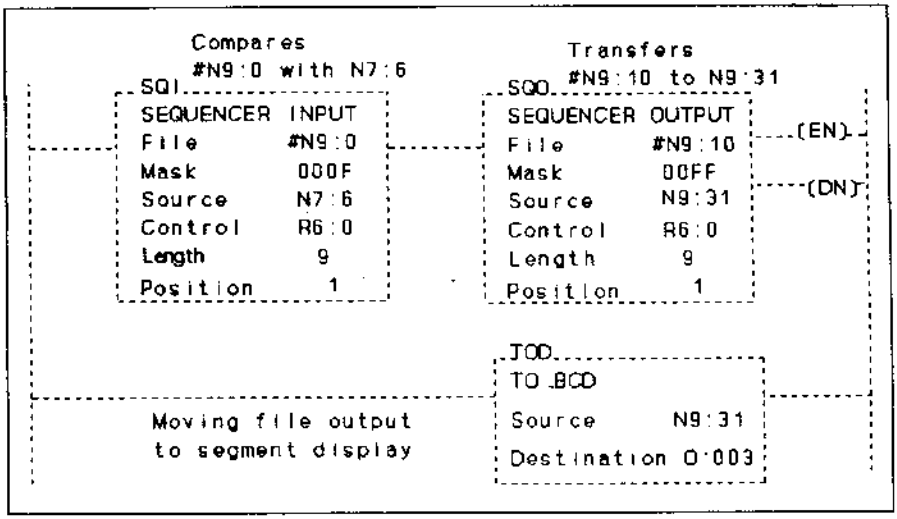
THẢO LUẬN

Chuỗi thứ tự là rất quan trọng trong các ứng dụng với sự vận hành ổn định và lặp lại trong hệ thống điều khiển tự động. Lệnh chuỗi thứ tự được dùng theo cặp để giám sát và điều khiển theo thao tác chuỗi thứ tự (Hình 19.1). Thứ nhất là lệnh nhập thứ tự (SQI). Lệnh này phát hiện thời điểm hoàn tất một bước. SQI so sánh các điều kiện bạn xác lập trong tập tin với các điều kiện thực tế tương ứng (nguồn), tìm kiếm sự tương hợp (tập tin = nguồn) giữa hai nhóm điều kiện đó.

Khi xảy ra sự tương hợp trong lệnh SQI, lệnh thứ hai là lệnh xuất chuỗi thứ tự (SQO) sẽ chuyển sang từ ngữ kế tiếp trong tập tin. SQO sẽ chuyển từ ngữ thông tin kế tiếp này, toàn bộ 16 bit dữ liệu trong tập tin, đến địa chỉ đích (các ngõ xuất thực).

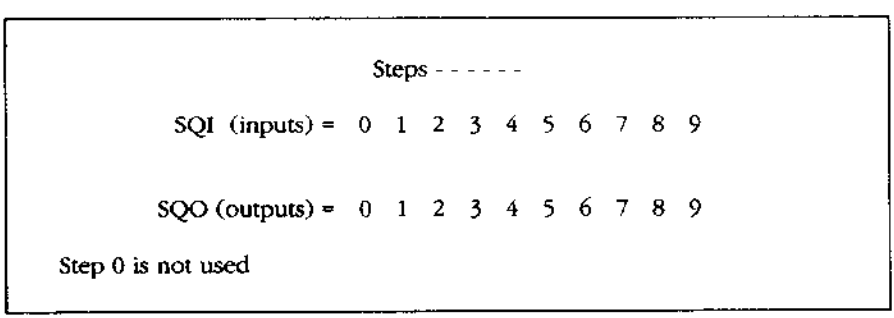
Bài tập

Trong bài tập này, bạn sử dụng bộ chuyển đổi thumbwheel để điều khiển các ngõ nhập của lệnh chuỗi thứ tự nhập. Lệnh nhập chuỗi thứ tự



Hình 19.1. Bậc thang nhập và xuất chuỗi thứ tự.

hoạt động tương tự bộ so sánh. Chỉ khi có sự tương hợp giữa bộ chuyển thumbwheel và tập tin nhập, chuỗi thứ tự mới có thể chuyển sang từ ngữ kế tiếp. Khi bạn tiến hành từng bước qua chuỗi thứ tự nhập, vị trí chuỗi thứ tự xuất luôn luôn có một bước phía trước vị trí nhập. Bạn hãy xem các vị trí nhập/ xuất chuỗi thứ tự trên Hình 19.2. Hình này minh họa sự vận hành chuỗi thứ tự. Chuỗi luôn luôn bắt đầu ở bước 1.



Hình 19.2. Chuỗi thứ tự các bước

Hai tập tin sao chép được dùng để lưu các quy trình thứ tự khác nhau. Điều này tương tự quy trình lưu trữ trong Thí nghiệm 15.1.

LOGIC CHƯƠNG TRÌNH

1. Lệnh MCR để đóng (ON) và ngắt (OFF) mạch hệ thống
2. Lệnh đọc chuyển khối được dùng để tiếp cận các thumbwheel.
3. Lệnh nhập chuỗi thứ tự và lệnh xuất chuỗi thứ tự trên cùng một bậc thang, chuỗi thứ tự các thao tác như sau:

   A. Khi thumbwheel (từ ngữ nguồn) phát hiện sự tương hợp (bằng nhau) trong các bước của tập tin nhập chuỗi thứ tự, sẽ cho phép tập tin xuất chuỗi thứ tự chuyển sang mức từ ngữ kế tiếp. Điều này sẽ chuyển điều kiện của từ ngữ đó đến địa chỉ đích tương ứng (các đèn báo).

   B. Bạn lập trình hệ thống sử dụng hai lệnh COP để lưu hai quy trình thứ tự. Các quy trình này có thể được dùng để thay đổi thông tin tập tin đồng thời trong cả tập tin nhập và tập tin xuất chuỗi thứ tự.

CÁC BƯỚC THỰC HIỆN

1. Thiết kế và viết chương trình vận hành máy theo chuỗi thứ tự sử dụng các lệnh nhập và xuất theo thứ tự. Bước thứ nhất là lập trình công tắc toggle để kích hoạt lệnh MCR.
2. Kế tiếp, lập trình lệnh đọc truyền khối để tiếp cận các công tắc thumbwheel.
3. Nhập chương trình nhập/ xuất chuỗi thứ tự được nên trên Hình 19.1. Bạn hãy dùng các địa chỉ nhập/ xuất thích hợp, sau đó lập trình lệnh TOD để hiển thị kết quả của tập tin xuất thứ tự trên màn hình 7-đoạn.
4. Bước kế tiếp là nhập lệnh sao chép tập tin (COP) để lưu trữ quy trình riêng rẽ. Quy trình này gồm tất cả các tập tin nhập và xuất chuỗi thứ tự. Bạn dùng công tắc toggle để kích hoạt sự truyền một trong các tập tin này cho các tập tin nhập khác và xuất chuỗi thứ tự. Bạn hãy sao (copy) bậc thang này với lệnh (COP) tập tin thứ hai cho quy trình khác và dùng các địa chỉ tương ứng. Mục đích là truyền thông tin mới cho các tập tin nhập/ xuất theo chuỗi thứ tự.
5. Sau khi nhập xong chương trình, bạn hãy chuyển sang chế độ giám sát dữ liệu và chọn chế độ thập phân. Bạn hãy nhập các số trên Bảng 19.1 vào các lệnh COP của quy trình 1 và 2.

Bảng 19.1. Nhập /xuất chuỗi thứ tự.

Địa chỉ	0	1	2	3	4	5	6	7	8	9
Nhập					**Các xác lập Thumbwheel**					
N9:00	0	0	0	0	0	0	0	0	0	0
Xuất					**Các đèn ngõ ra**					
N9:20	0	0	0	0	0	0	0	0	0	0
Quy trình A					**File COP #1**					
N9:40	0	1	2	3	4	5	6	7	8	9
N9:50	0	74	32	92	56	255	256	12	99	36
Quy trình B					**File COP #2**					
N9:60	0	1	2	3	4	5	6	7	8	9
N9:70	0	24	13	255	256	17	18	19	20	21

6. Bạn cần bảo đảm mọi công tắc nhập trong chương trình đều mở, sau đó chuyển sang chế độ RUN. Trong chế độ giám sát dữ liệu, bạn hãy kiểm chứng sự tải thông tin từ các quy trình vào bộ tạo thứ tự. Bạn hãy đóng công tắc toggle điều khiển quy trình A để tải vào các bộ nhập/ xuất thứ tự. Điều này sẽ tải đồng thời chuỗi thứ tự nhập và xuất với Quy trình A.

7. Bạn lần lượt qua các bước của chuỗi thứ tự. Hãy quan sát khi bạn thử các số khác nhau trên bộ chuyển đổi thumbwheel, các đèn ngõ ra sẽ theo thứ tự chỉ khi các tín hiệu nhập của bộ thumbwheel tương hợp với tín hiệu nhập của tập tin. Bạn hãy kiểm tra bằng cách thực hiện các bước sau:

   A. Xác lập bộ chuyển đổi thumbwheel cho từng tổ hợp số nhập được nêu trong chuỗi thứ tự. Các số này được so sánh với các xác lập số nhập chuyển đổi thumbwheel, khi phát hiện sự tương hợp, sẽ lần lượt bước qua chuỗi thứ tự xuất, mỗi lần một từ ngữ.

   B. Bạn hãy quan sát các đèn ngõ ra lần lượt sáng theo chuỗi thứ tự và tương hợp với tập tin xuất khi bạn kích hoạt bộ chuyển đổi thumbwheel. Bạn hãy tiếp tục kích hoạt bộ thumbwheel từ một đến chín. Khi thumbwheel đạt đến số chín, bộ xuất thứ tự sẽ quay trở về bước # 1.

Bạn hãy quan sát số 256. Khi bạn đạt đến số này, số zero sẽ xuất hiện trên màn hình. Điều này là do số 256 trong hệ nhị phân thuộc byte cao hơn, vì vậy số đó bị che khuất và không xuất hiện.

C. Bạn dừng chuỗi thứ tự và chuyển sang giám sát dữ liệu, sau đó hãy đóng công tắc toggle (I: 000/05) để sao chép hai quy trình trong chuỗi tập tin thứ tự. Tập tin thứ tự khi đó tương tự quy trình thứ hai.

D. Bạn hãy lặp lại các bước A và B nêu trên với quy trình thứ hai trong chuỗi thứ tự nhập/ xuất. Khi hoàn tất, bạn hãy lưu chương trình.

8. Bạn đã nhận thấy lệnh nhập chuỗi thứ tự có thể được dùng để so sánh tập hợp xác định các tín hiệu nhập với các tín hiệu nhập thực tế. Lệnh nhập chuỗi thứ tự về cơ bản là bộ so sánh. Lệnh này so sánh các cài đặt trên thông tin trong tập tin đến ngõ xuất thực, do đó đóng vai trò truyền thông tin.

ÔN TẬP

Để củng cố và nâng cao kiến thức đã học trong Bài tập này, bạn hãy thực hành thí nghiệp kèm theo, sau đó viết và in báo cáo.

Thí nghiệm 19.1. HỆ THỐNG RỬA XE HƠI

Bạn hãy viết chương trình cho hệ thống rửa xe hơi, với chuỗi thứ tự vận hành như sau:

Nhập – xà bông\nước – cọ rửa – nước rửa – lau khô – sấy – hoàn tất

Giới hạn 1___5 giây___20 giây___30 giây___40 giây___50 giây___ 60 giây.

1. Bạn hãy dùng lệnh nhập thứ tự và lệnh xuất thứ tự, giới hạn 1 sẽ khởi động dây chuyền để chuyển xe hơi về phía trước. Dây chuyền sẽ chạy theo thời lượng được xác định trước (60 giây) cho từng xe đi vào hệ thống rửa xe, mỗi lần chỉ rửa một xe.

2. Xe đi dọc theo dây chuyền và quá trình rửa được kích hoạt bằng lệnh xuất chuỗi thứ tự.

3. Lệnh nhập chuỗi thứ tự có thể được dùng để so sánh thời lượng của thứ tự rửa xe được liệt kê ở phần trên với bộ thời chuẩn đóng (TON).

Bài tập lớn 20. 0

LƯU ĐỒ CHỨC NĂNG THEO CHUỖI THỨ TỰ

MỤC ĐÍCH

Nghiên cứu khái niệm lập trình module theo lưu đồ chức năng thứ tự (SFC)

NỘI DUNG

Sau khi hoàn tất Bài tập này, bạn sẽ có khả năng:

- Thiết kế và lập trình lưu đồ chức năng thứ tự.
- Sử dụng Bài tập hiện có và chia Bài tập thành các khối logic chỉ chứa một ý tưởng hoặc một tập hợp các thao tác

TÀI LIỆU

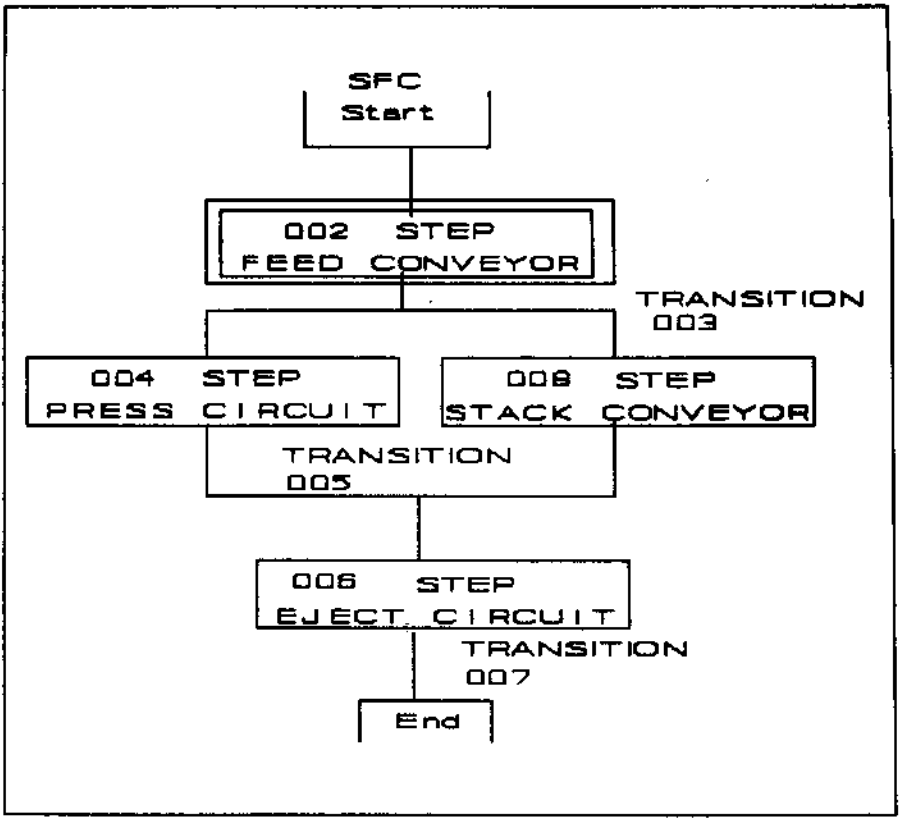
Để biết chi tiết các lệnh được dùng trong Bài tập này, bạn hãy tham khảo Phụ lục A: Nhập cặp bước/ chuyển tiếp, tạo lập sự lựa chọn và các nhánh đồng thời.

THẢO LUẬN

Lưu đồ chức năng thứ tự (SFC) cho phép bạn tiến hành viết chương trình cấu trúc theo chiều thuận từ trên xuống. Lập trình cấu trúc là khái niệm quan trọng tác động đến sự thiết kế và tổ chức phần mềm.

Thay vì xây dựng chương trình bậc thang dài dòng, bạn có thể chia chương trình thành nhiều bước, bộ xử lý mỗi lần có thể chạy một bước. Từng bước liên hệ với tập tin chương trình chứa logic bậc thang cần thiết để hoàn tất các tác vụ điều khiển. Kiểu lập trình này cho phép bạn chuyên biệt thứ tự để bộ xử lý chạy các tập tin chương trình. SFC báo cho bộ xử lý chạy tập tin chương trình nào và vào thời điểm nào (Hình 20.1).

Khi bạn dùng SFC, bộ xử lý chỉ quét bước hoạt động trong SFC. Tất cả bước khác đều không được quét, do đó giảm thời gian quét chương



Hình 20.1. Lưu đồ SFC

trình. Khi bộ xử lý giám sát SFC, bộ xử lý thực thi tập tin chương trình liên quan với các bước hoạt động. Đầu cuối lập trình sẽ làm sáng các bước hoạt động của SFC, tăng tính thuận tiện để giám sát và xử lý sự cố trong logic bậc thang.

Bài tập

Mục đích của bài tập này là nghiên cứu khái niệm lập trình theo lưu đồ chức năng thứ tự. Bạn sẽ thiết kế lưu đồ chức năng thứ tự giảm để minh họa các khái niệm cơ bản trong lập trình SFC. Các hàm logic được dùng trong chuỗi thứ tự tương tự mạch điện bạn đã gặp trong Bài tập 12.0. Logic bậc thang chương trình được chỉnh sửa để phù hợp với chương trình SFC.

LOGIC CHƯƠNG TRÌNH

1. Thiết kế tiến trình logic theo thứ tự thuận, liên hệ trực tiếp với mạch điều khiển trong Bài tập 12.0.

2. Phân chia Bài tập 12.0 thành 4 bước thứ tự. Bắt đầu từ phần băng tải nạp, đây là bước thứ nhất, sau đó bạn chia mạch logic còn lại thành các bước logic. Bước kế tiếp là mạch máy ép. Bạn tiếp tục quá trình này cho đến khi mỗi bước chỉ chứa một tập hợp ý tưởng, hoặc một nhóm trong quy trình. Phần cuộn solenoid khí nén (xuất) là bước logic kế tiếp trong chuỗi thứ tự. Cuối cùng băng tải xếp chồng là phần kết thúc (Hình 20.1).

CÁC BƯỚC THỰC HIỆN

1. Bước thứ nhất là xây dựng lưu đồ SFC được nêu trên Hình 20.1. Bạn hãy đặt con trỏ trên tập tin #1 và chuyển sang tập tin giám sát (F8) để xây dựng lưu đồ chức năng thứ tự. Chuỗi thứ tự các bước lập trình như sau:

   (F3) Append STP/TRN (gắn kết bước / chuyển tiếp).

   (F6) Tạo tập tin #2.

   (F10) Đổi tên tập tin, FEED.

   (F6) Tạo tập tin #3.

   (F10) Đổi tên tập tin, FEEDONE.

   (F1) Chọn bước – con trỏ trên tập tin #2.

   (F2) Chọn chuyển tiếp – con trỏ trên tập tin #3.

Enter. Bạn hãy dùng nhánh đồng thời và lặp lại quá trình nêu trên cho bước #4, các chuyển tiếp #5 và #6.

2. Kế tiếp là xem lại sự vận hành của chương trình mạch máy ép trong Bài tập 12.0. Bạn hãy quan sát cách thức phân chia phép logic của mạch máy ép thành 4 phần riêng biệt. Bốn phần logic bao gồm:

   A. Mạch băng tải nạp, được đặt trong bước 2, chuyển tiếp 3.

   B. Mạch máy ép, bước 4, chuyển tiếp 5.

   C. Mạch xuất, bước 6, chuyển tiếp 7.

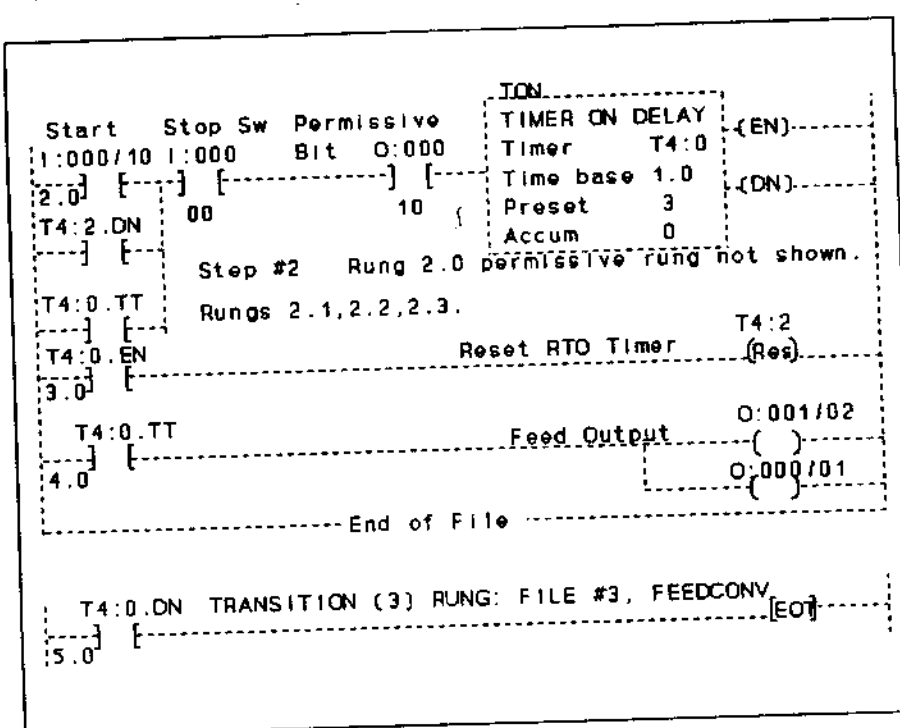
   D. Mạch băng tải xếp chồng, bước 8, tập tin 8.

Bạn hãy xem các Hình từ 20.2 đến 20.5

Mỗi bước và tập tin chuyển tiếp đều lưu logic bậc thang riêng. Logic bậc thang của tập tin bước điều khiển sự vận hành liên quan đến bước đó. Logic bậc thang của tập tin chuyển tiếp điều khiển tiến độ của bước kế tiếp.

Bằng cách sử dụng các tính năng biên tập, bạn hãy chọn và sao chép bốn phần riêng rẽ này vào tập tin dán (paste) sau đó hãy chép (copy) chúng vào lưu đồ chức năng thứ tự tương ứng.

3. Khi hoàn tất chương trình SFC, bạn hãy chuyển sang sơ đồ SFC và kiểm tra mạch này. Bạn chuyển sang chế độ RUN và nhấn nút Start. Chương trình của bạn sẽ chạy lần lượt qua từng bước với thứ tự như sau: 3 giây trong bước 2, 6 giây trong các bước 4 và 8, và 1 giây trong bước 7. Sau chuyển tiếp 7, chuỗi thứ tự sẽ lặp lại. Bạn hãy lưu chương trình này lên đĩa.



Hình 20.2. Bước #2, chuyển tiếp #3 trên FSC

```
                                        TON
                                        TIMER ON DELAY
                Press Timer             Timer      T4:1   (EN)
6.0                                     Time base  1.0
                                        Preset     6      (DN)
                                        Accum      0

        STEP (4) RUNGS:    FILE #4, PRESS
        2.0,3.0,4.0

  T4:1.TT                       Press Solenoid        O:000/02
---] [-------------------------------------------------------( )---
7.0

-----------------------End of File-----------------------------

  T4:1.DN  TRANSITION (5) RUNG: FILE #5, PRESSONE
---] [-------------------------------------------------------[EOT]---
8.0
```

Hình 20.3. Bước # 4, chuyển tiếp #5 trên FSC

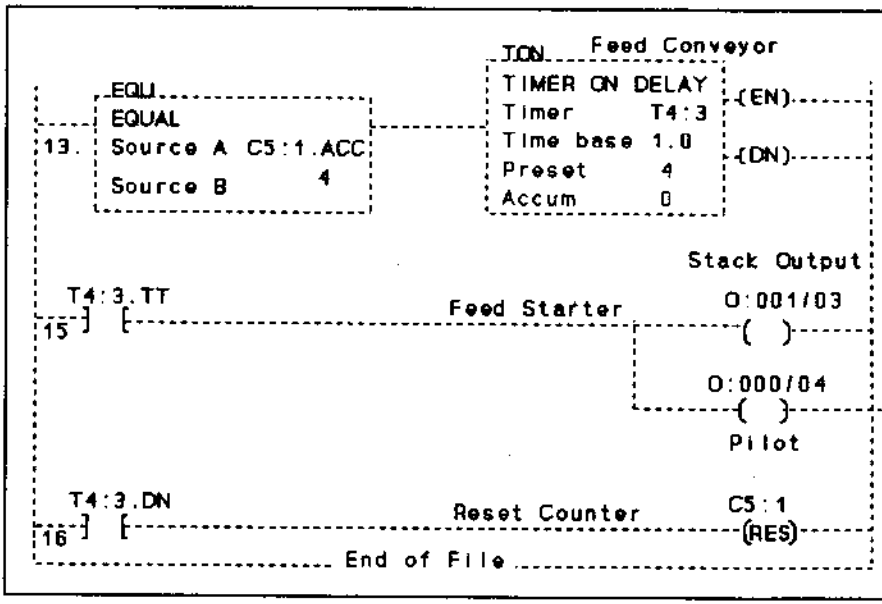
```
                                        RTO
                                        RETENTIVE TIMER
                Eject Timer             Timer      T4:2   (EN)
9.0                                     Time base  1.0
                                        Preset     1      (DN)
  STEP (6) FILE #6                      Accum      0
  Rungs 9.0,10,11

  T4:2.DN     C5:1.DN                   CTU
---] [---------]/[-------------------   COUNT UP   C5:1   (CU)
10.0                                    Preset     4
                                        Accum      3      (DN)

                                                     Eject Output
  T4:2.TT                                            O:000/03
---] [-------------------------------------------------------( )---
11.0
-------------------- End of File -------------------------------

  T4:2.DN  TRANSITION (7) RUNG: FILE #7,EJECTDONE
---] [-------------------------------------------------------[EOT]---
12.0
```

Hình 20.4. Bước #6, chuyển tiếp #7 trên FSC



Hình 20.5. Bước #8, tập tin #8 trên FSC.

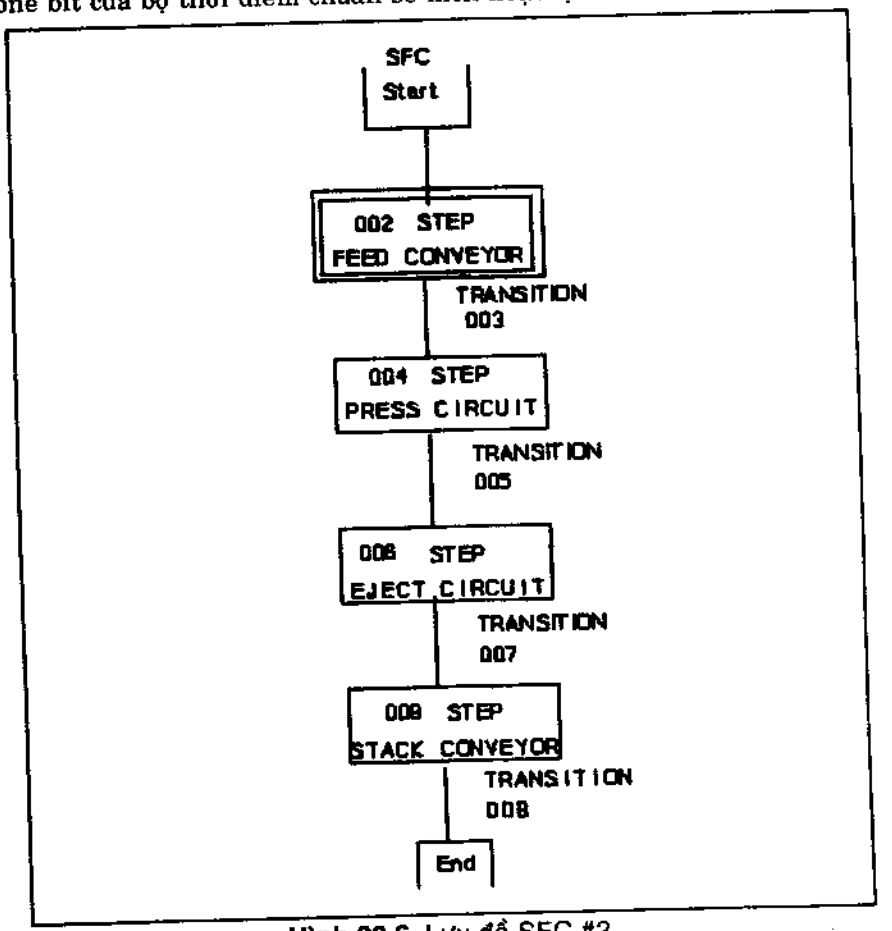
ÔN TẬP

Sau khi hoàn tất phần đầu của Bài tập 20, để củng cố và nâng cao kiến thức, bạn hãy thực hành Thí nghiệm 20.1

Thí nghiệm 20.1. SFC #2

Từ chương trình SFC trong Bài tập 20.0 bạn hãy chỉnh sửa để vận hành với lưu đồ chức năng thứ tự #2 được nêu trên Hình 20.6. Chương trình này tương tự Bài tập 20.0, nhưng có SFC thay đổi.

Phần cần bổ sung là bước và chuyển tiếp đối với băng tải xếp chồng. Băng tải này sẽ được chỉnh sửa để vận hành theo chuỗi thứ tự. Bạn hãy dùng lệnh nhỏ hơn 4 (< 4) cùng với done bit của bộ thời chuẩn để kích hoạt lệnh EOT của băng tải xếp chồng. Lệnh này sẽ kích hoạt EOT khi trên băng tải có chồng ít hơn 4 bản mạch. Khi số bản mạch bằng 4 (=4), done bit của bộ thời điểm chuẩn sẽ kích hoạt lệnh EOT.



Hình 20.6. Lưu đồ SFC #2.

Bài tập lớn 21. 0

CHUYỂN ĐỔI DIGITAL – ANALOG

MỤC ĐÍCH

Nghiên cứu module chuyển đổi digital – analog để điều khiểu bộ truyền động AC.

NỘI DUNG

Sau khi hoàn tất Bài học này, bạn sẽ có khả năng:

- Lập cấu hình module chuyển đổi digital – analog (DAC).
- Thực hiện bộ truyền động AC để điều khiển tốc độ động cơ 3 pha.
- Sử dụng lệnh đọc và viết truyền khối.

TÀI LIỆU

Để biết chi tiết các lệnh được dùng trong Bài tập này, bạn hãy đọc Phụ lục A, Các lệnh truyền khối BTR, BTW.

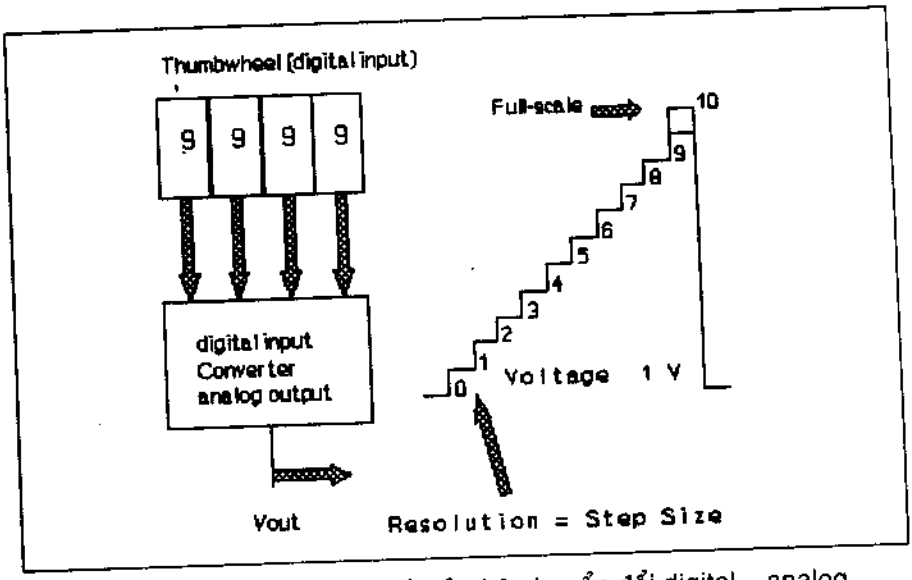
THẢO LUẬN

Khi hệ thống digital, chẳng hạn bộ điều khiển lập trình PLC, được dùng để điều khiển hoặc giám sát quy trình thực tế, bạn cần giải quyết sự khác biệt giữa bản chất digital của PLC và bản chất analog của các biến quy trình. Bộ chuyển đổi dgital – analog giải quyết sự khác biệt này bằng cách chuyển đổi tín hiệu nhập digital thành biến xuất analog (0–10 V hoặc 4-20 mA).

Độ phân giải của bộ chuyển đổi digital – analog (DAC) được định nghĩa là sự thay đổi nhỏ nhất có thể xảy ra trong tín hiệu xuất analog là kết quả của sự thay đổi tín hiệu nhập digital (Hình 21.1). Độ phân giải luôn luôn bằng bit ý nghĩa cuối và được coi là kích cỡ bước. Đây là giá trị điện áp xuất thay đổi khi giá trị nhập digital thay đổi từ một bước sang bước kế tiếp.

Bài tập

Trong Bài tập này, bạn sẽ sử dụng thumbwheel để hiển thị mã nhị



Hình 21.1. Dạng sáng xuất của bộ chuyển đổi digital – analog

phân 4-bit. Mỗi số nhị phân đại diện cho mã nhập digital chuyên biệt các giá trị rời rạc trong một khoảng. Mã nhập digtal này được nhập vào chuyển đổi digital – analog (DAC). Bên trong, bộ DAC so sánh mã này với cấu hình của mình và tạo ra tín hiệu analog xuất tương ứng. Bạn sử dụng tín hiệu xuất đó để điều khiển bộ truyền động dòng điện xoay chiều.

LOGIC CHƯƠNG TRÌNH

1. Mục đích của Bài tập này là viết chương trình cho mạch xuất tín hiệu analog từ 0 đến 10 volt và vận hành từ bộ chuyển đổi thumbwheel.
2. Nhập lệnh khối chuyển tiếp đọc hướng đến module đa công để đọc bộ chuyển đổi thumbwheel.
3. Lập trình các lệnh khối chuyển tiếp đọc và viết kích hoạt module chuyển đổi digital – analog (Hình 21.2).
   A. Lập cấu hình độ phân giải 4 kênh trong module DAC. Các kênh 1 và 2 là ngõ nhập từ 0 đến 9, các kênh 3 và 4 là từ 0 đến 9999.
   B. Chọn ngõ ra từ 0 đến 10 V ở module DAC cung cấp cho bộ

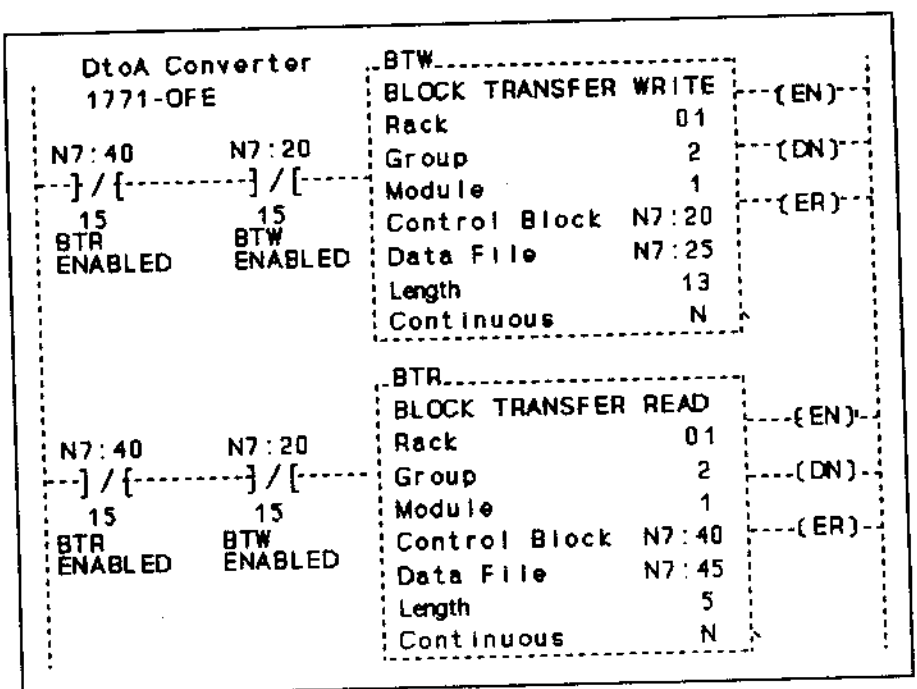
truyền động AC bằng cách đổi công tắc chọn tốc độ trên bộ này sang chế độ tự động.

4. Lập trình hai lệnh dịch chuyển, chuyển dữ liệu từ thumbwheel đến lệnh khối chuyển tiếp khối của module DAC (ví dụ N7:6 to N7:25).

5. Lập trình 8 đèn ở ngõ ra, 4 đèn vận hành với chữ số thứ nhất của bộ thumbwheel (kênh 1). Bốn đèn vận hành với chữ số thứ hai của thumbwheel (kênh 2). Các đèn sẽ hiển thị ngõ ra BCD (nhị phân – thập phân) của bộ chuyển đổi thumbwheel.

CÁC BƯỚC THỰC HIỆN

1. Trên bậc thang thứ nhất, bạn nhập lệnh MCP để điều khiển chương trình.
2. Lập trình lệnh khối chuyển tiếp đọc cho bộ chuyển đổi thumbwheel.
3. Nhập các lệnh khối chuyển tiếp đọc và viết cho module digital – analog. (Hình 21.2).



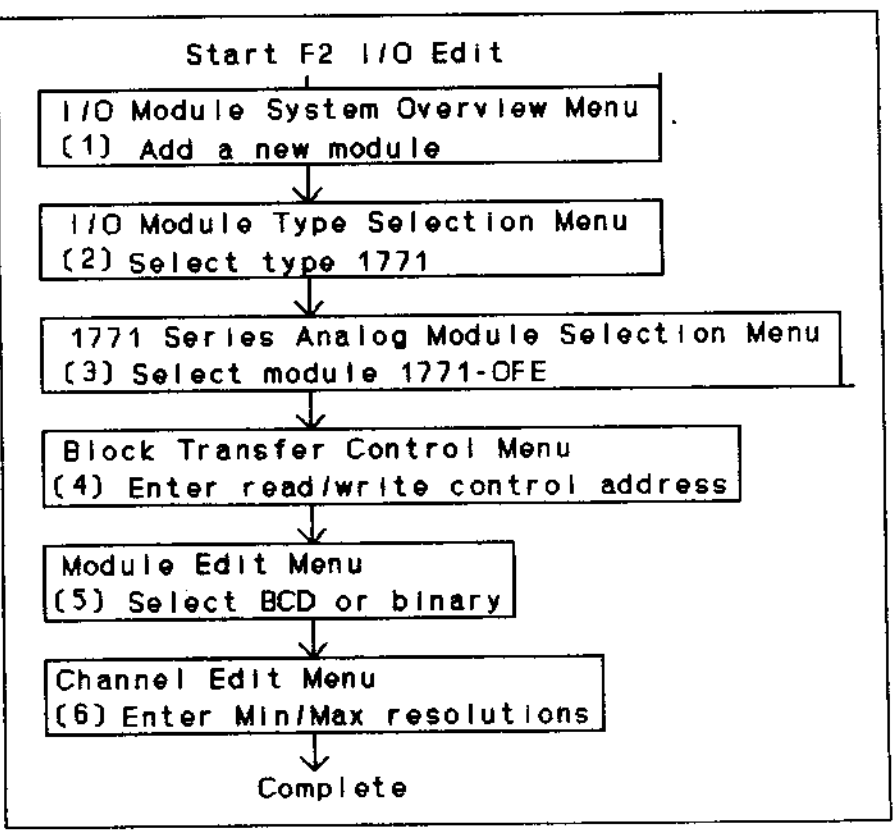
Hình 21.2. Khối chuyển tiếp đọc/viết của bộ chuyển đổi digital – analog.

4. Nhập hai lệnh dịch chuyển song song. Một lệnh sẽ chuyển dữ liệu của bộ thumbwheel thứ nhất cho lệnh viết chuyển tiếp khối (N7:9 to N7:25). Lệnh chuyển dịch thứ hai sẽ chuyển dữ liệu của bộ thumbwheel thứ hai (N7:8 to N7:26)

5. Để hoàn tất chương trình, bạn nhập 4 bậc thang, mỗi bậc đều có lệnh XIC, để điều khiển 8 đèn. Lập địa chỉ các lệnh XIC theo chữ số có ý nghĩa thấp nhất của từng thumbwheel (ví dụ N7:9.0)

6. Sau khi nhập chương trình, bạn hãy đặt con trỏ lên lệnh chuyển khối của module DAC và đến I/O edit (F2). Điều này sẽ đưa bạn đến các menu cấu hình gồm nhiều module khác nhau. Bạn hãy tuân theo lưu đồ được minh họa trên Hình 21.3.

7. (A) Menu thứ nhất là System Overview Menu. Ở đây, bạn có thể



Hình 21.3. Các bước lập cấu hình module.

chọn (F6) và bổ sung module mới. (B) Điều này sẽ đưa bạn đến menu thứ hai, chọn kiểu module I/O. Bạn hãy chọn (F10) module 1771. (C) Kế tiếp là menu 1771 Series Analog Module Selection. Module được dùng trong Bài tập này là D/A 1771 OFE/A module (D) Sau khi chọn module, bạn được nhắc nhập địa chỉ của các lệnh chuyển khối trong Block Transfer Control Menu. Sau khi nhập các địa chỉ, bạn hãy chấp nhận (OK) và chuyển sang Module Edit Menu. (E) Trong Module Edit Menu, bạn chọn tùy chọn BCD cho Bài tập này. Đối với bộ chuyển đổi digital – analog IFE, bạn hãy chọn binary and differential connection.

8. Bạn hãy chọn Channel Edit Menu. Ở đây bạn lập cấu hình độ phân giải của module. Mudule này có thể được lập cấu hình từ tối thiểu là 0 đến tối đa là 9999. Bạn hãy xác lập các kênh 1 và 2 với tối thiểu là 0 và tối đa là 9. Xác lập các kênh 3 và 4 từ 0 đến 9999.

   Bạn chuyển sang chế độ Run và chấp nhận dữ liệu trong chế độ này. Bạn chuyển các tín hiệu nhập digital của thunbwheel vào module DAC. Bạn có thể quan sát các thay đổi này trong Channel Menu.

9. Bạn hãy dùng VOM kế để đo điện áp của DAC trên cả 4 kênh. Số đo điện áp trên module này tỷ lệ với các số trên thumbwheel. Ví dụ, trên các kênh 1 và 2, số 6 trên thumbwheel sẽ là 6,67 V trên module và 67% trong Channel Edit Menu.

   Bạn hãy thay đổi dữ liệu của thumbwheel và tiếp tục đo điện áp trên các kênh 1 và 3. Bạn hãy nhập các giá trị điện áp đo được vào Bảng dữ liệu 21.1.

   Chú ý: Bạn hãy chuyển sang bước 10 nếu có bộ truyền động AC. Nếu không có bộ này, bạn hãy chuyển đến bước 12.

10. Bạn hãy kiểm tra để bảo đảm tuân thủ mọi quy định về an toàn và vận hành trước khi sử dụng bộ truyền động AC tần số biến thiên.

    Bạn kích hoạt công tắc chọn của bộ truyền động theo chế độ vận hành bằng tay. Điều này cho phép bạn vận hành bộ truyền động với các điều khiển tại chỗ. Bạn nên xem kỹ sổ tay hướng dẫn của nhà sản xuất về bộ truyền động AC và kiểm tra các thông số của bộ này.

    Chú ý: Điều chỉnh các thông số và công tắc chọn tốc độ theo Bài tập 5.

Bảng 21.1. Bảng dữ liệu

THUMBWHEEL	CHANNEL 1	FREQUENCY	THUMBWHEEL	CHANNEL 3
0001			1111	
0002			2222	
0003			3333	
0004			4444	
0005			5555	
0006			6666	
0007			7777	
0008			8888	
0009			9999	

11. Khi hoàn tất lập trình trên bộ truyền động, bạn hãy kiểm tra các thông số bằng cách vận hành bộ này theo chế độ điều khiển bằng tay. Điều này cho phép bạn sử dụng điều khiển tại chỗ để vận hành bộ truyền động AC và kiểm tra các thông số.

Khi bộ truyền động vận hành tốt trong các chế độ điều khiển bằng tay, bạn hãy dừng bộ truyền động và chuyển sang chế độ tự động với công tắc chọn tốc độ. Trong chế độ tự động, bạn có thể vận hành bộ truyền động với DAC.

Bạn dùng bộ chuyển đổi thunbwheel để thay đổi điện áp xuất của DAC. Bạn có khả năng thay đổi tần số xuất và thay đổi tốc độ động cơ 3 pha theo thông tin trên bộ chuyển đổi thumbwheel.

Bạn hãy hoàn tất các bước sau đây:

A. Tuân theo dữ liệu thumbwheel bạn đã nhập trong Bảng dữ liệu 21.1.

B. Kiểm tra tần số ra của bộ truyền động, hiển thị trên bảng hiển thị chương trình của bộ truyền động.

C. Bạn hãy nhập các số đo dữ liệu tần số vào Bảng dữ liệu 21.1, so sánh tần số này với dữ liệu thumbwheel. Tần số ra sẽ biến

thiên theo dữ liệu bộ chuyển đổi thumbwheel. Giá trị tần số ra phụ thuộc giá trị bạn đã nhập theo thông số tần số cực đại của bộ truyền động AC.

D. Bạn hãy lưu và in chương trình.

Nếu bộ truyền động AC không vận hành, bạn hãy kiểm tra như sau:

A. Đo tín hiệu ra của DAC để xem điện áp ra có phù hợp với khoảng giá trị 0 - 10 V hay không.

B. Kiểm tra lại cấu hình DAC.

C. Kiểm tra địa chỉ trong chương trình.

D. Kiểm tra lại các đặc tính kỹ thuật của bộ truyền động và công tắc chọn tốc độ.

E. Nếu vấn đề vẫn không được giải quyết, bạn hãy kiểm tra lại chương trình.

12. Khi không có bộ truyền động AC, bạn hãy thực hiện các bước sau:

A. Lập cấu hình các kênh 1 và 2 như phần trên, lập cấu hình các kênh 3 và 4 của module DAC từ 0 đến 9999.

B. Lập trình mạch bộ đếm Up/Dn với công tắc toggle chọn bộ đếm lên hoặc xuống, sử dụng mạch bộ đếm chu kỳ để tăng số gia của bộ đếm một cách tự động.

C. Lập trình lệnh dịch chuyển để chuyển giá trị tích lũy của bộ đếm vào DAC, và dùng điều này để cung cấp tín hiệu nhập cho kênh 3.

D. Lập trình lệnh kiểm tra giới hạn để kiểm tra tín hiệu nhập digital ở kênh 3. Xác lập giới hạn dưới là 1800 và giới hạn trên là 3600. Bạn kích hoạt đèn báo với lệnh kiểm tra giới hạn này. Mục đích là đếm tăng ở bộ đếm cho đến khi giá trị kiểm tra của lệnh kiểm tra giới hạn trong khoảng giới hạn. Điều này sẽ kích hoạt đèn ở ngõ ra.

E. Đối với kênh 4, bạn hãy chuyển địa chỉ bộ chuyển đổi thumbwheel đến ngõ vào của kênh này. Bạn hãy vận hành tín hiệu nhập digital của kênh 4 sử dụng giá trị toàn phần của bộ chuyển đổi thumbwheel.

F. Dùng volt kế đo điện áp ra ở cả bốn kênh khi bạn thay đổi các tín hiệu nhập với bộ đếm hoặc bộ chuyển đổi thumbwheel. Bạn hãy ghi các kết quả vào Bảng 21.2, Bảng dữ liệu.

Bảng 21.2. Bảng dữ liệu

Kênh	Giá trị nhập	Giá trị kênh	Điện áp ra

Bài tập lớn 22.0

CHUYỂN ĐỔI ANALOG – DIGITAL

MỤC ĐÍCH

Nghiên cứu công dụng của chiết áp kế trong mạch điều khiển module chuyển đổi analog-digital (ADC)

NỘI DUNG

Sau khi hoàn tất Bài Tập này, bạn sẽ có khả năng.

- Lập cấu hình module chuyển đổi analog – digital
- Nghiên cứu sự vận hành của bộ chuyển đổi analog – digital.

TÀI LIỆU

Để biết chi tiết các lệnh được dùng trong Bài Tập này, bạn hãy đọc Phụ lục A: các lệnh chuyển khối: BTR, BTW.

THẢO LUẬN

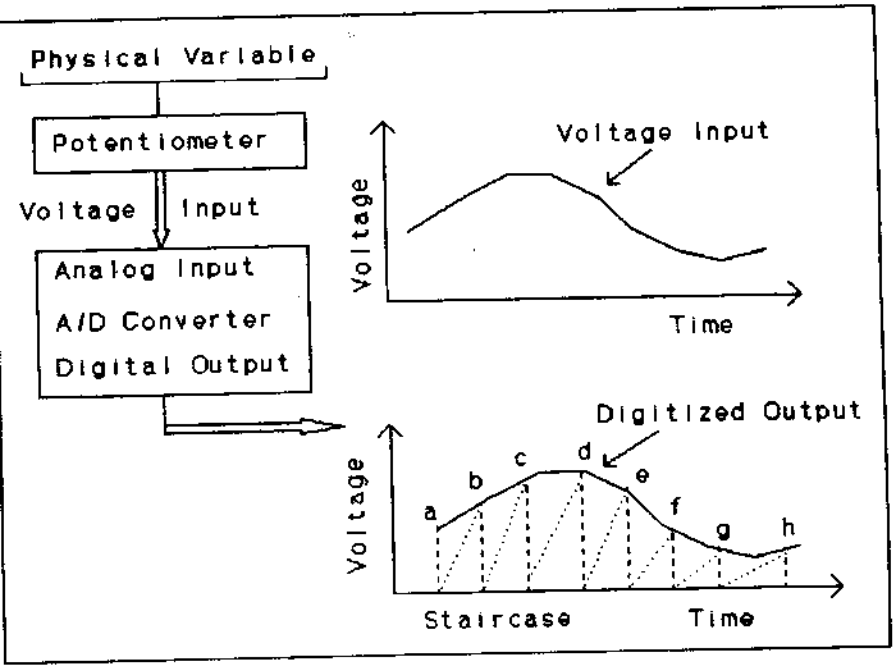
Analog – digital

Bộ chuyển đổi analog – digital (ADC) lấy điện áp nhập analog và sau khoảng thời gian xác định, sẽ tạo ra mã xuất digital tương ứng tín hiệu nhập analog. Hình 22.1 minh họa cách thức ADC liên tục thực hiện các chuyển đổi để số hóa tín hiệu analog ở các điểm a, b, c, d, ..., bằng cách liên tục lấy mẫu biên độ ở nhiều điểm trong chu kỳ tín hiệu. Kết quả là chuỗi các bit dựa trên các điểm của tín hiệu analog gốc.

Chiết áp kế biến thiên với điện áp xuất analog sẽ tạo ra điện áp biến thiên theo tín hiệu nhập analog của bộ chuyển đổi analog-digital. Bộ ADC chuyển tín hiệu nhập analog thành tín hiệu xuất digital. Tín hiệu xuất này chứa số lượng bit biểu thị giá trị của tín hiệu nhập analog.

Bài tập

Mục đích của Bài tập này là nghiên cứu sự vận hành của module chuyển đổi analog-digital (ADC). Để minh họa điều này, bạn sử dụng



Hình 22.1. Tín hiệu nhập analog – tín hiệu xuất digital.

chiết áp để biểu thị điện áp nhập biến thiên. Điện áp nhập analog được đưa vào bộ chuyển đổi analog-digital (ADC). Bên trong, bộ ADC so sánh tín hiệu này với cấu hình của mình và tạo ra tín hiệu xuất digital tương ứng. Bạn sử dụng tín hiệu xuất digital để điều khiển quá trình analog.

LOGIC CHƯƠNG TRÌNH

1. Lập cấu hình module ADC theo các bước dưới đây:

   A. Lập cấu hình kênh 1 với độ phân giải từ 0 đến 212, chuyển địa chỉ kênh xuất digital đến các đèn ngõ ra.

   B. Lập cấu hình kênh 2 với độ phân giải từ 0 đến 512 chuyển địa chỉ kênh xuất digital đến tập tin bit B3:0

   C. Lập cấu hình kênh 3 với độ phân giải từ 0 đến 1024, chuyển địa chỉ kênh xuất digital đến màn hiển thị 7-đoạn

   D. Lập cấu hình kênh 4 với độ phân giải từ 0 đến 1040, chuyển địa chỉ kênh xuất digital đến lệnh kiểm tra giới hạn.

E. Lập cấu hình kênh còn lại từ 0 đến 9999.

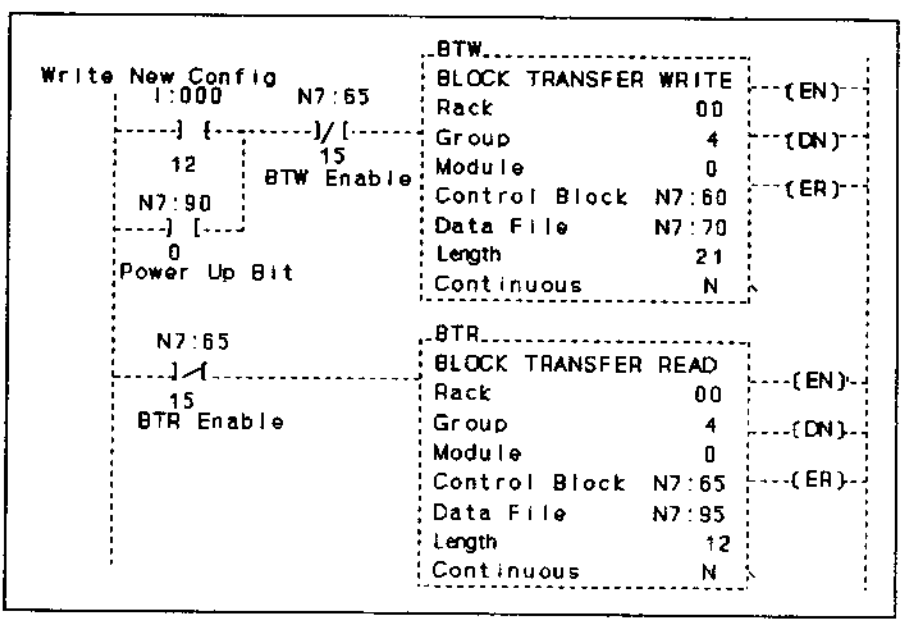
2. Bạn hãy lập trình hai lệnh Move (dịch chuyển), một cho các kênh thứ nhất và thứ hai để điều khiển các đèn và tập tin nhị phân. Lập trình lệnh TOD để đưa kênh 3 đến màn hiển thị 7 đoạn. Đối với kênh 4 bạn dùng lệnh kiểm tra giới hạn để kiểm tra kết quả của kênh này.

CÁC BƯỚC THỰC HIỆN

1. Trên hai bậc thang đầu, bạn lập trình các lệnh chuyển khối đọc và viết (Hình 22.2). Nhập các địa chỉ thích hợp cho ADC.

2. Bạn lập trình các hai lệnh Move (dịch chuyển) song song. Một lệnh sẽ chuyển dữ liệu của chiết áp thứ nhất đến từ ngữ xuất (N7; 99 to O:000), lệnh thứ hai chuyển dữ liệu của chiết áp thứ hai đến tập tin bit (N7:100 to B3:3).

3. Bậc thang kế tiếp, bạn lập trình lệnh TOD để chuyển dữ liệu của chiết áp thứ ba đến màn hình 7-đoạn (N7:101 to O:003).



Hình 22.2. Các lệnh chuyển khối đọc /viết dùng cho ADC.

4. Bạn dùng lệnh kiểm tra giới hạn để kiểm tra kết quả của chiết áp thứ tư, xác lập các giới hạn là 500 và 1500. Đến đây, bạn hoàn tất chương trình, có thể chuyển sang lập cấu hình bộ chuyển đổi ADC.

5. Kế tiếp, bạn đặt con trỏ trên lệnh chuyển khối analog- digital và đến I/O Edit (F2). Bạn hãy tuân theo lưu đồ trên Hình 21.3 của Bài tập 21.0 để lập cấu hình cho module này. Cấu hình của bộ chuyển đổi 1771-IFE tương tự cấu hình 1771-OFE. Bạn hãy dùng chế độ Differential and Binary trên menu. Sự biến thiên các chiết áp sẽ thay đổi điện áp nhập ở từng kênh. Bạn đo điện áp vào trên 4 kênh của bộ ADC sử dụng volt kế digital.

Bạn sẽ có khả năng đo các điện áp nhập trên module này tỷ lệ với các giá trị phân giải. Ví dụ, điện áp 5V trên ngõ vào kênh 1 sẽ có số đo 212 trên Module Edit Menu. Đây là độ phân giải 100% (cực đại) của kênh 1 (N7:99).

Bạn hãy điều chỉnh các chiết áp đo điện áp trên các kênh. Bạn nhập kết quả đo và tính số phần trăm vào Bảng Dữ Liệu 22.1

6. Bạn hãy quan sát dữ liệu trên Bảng 22.1. Dữ liệu số tăng với độ phân giải cực đại của cùng số phần trăm tín hiệu xuất. Giá trị cực đại càng cao, dữ liệu digital càng cao với cùng số phần trăm tín hiệu xuất.

7. Bạn cũng có thể quan sát và kiểm tra chương trình 25% - 75% giá trị của kênh 4. Bạn có khả năng đọc kết quả trên màn hình 7-đoạn hiển thị giá trị BCD của kênh 3.

8. Nếu mọi việc đều chuẩn, bạn hãy lưu và in chương trình.

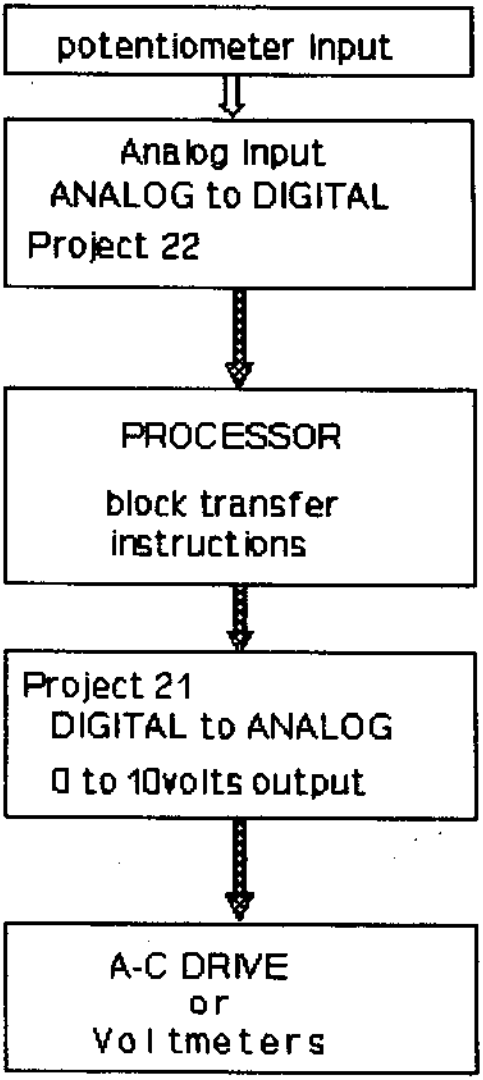
Bảng 22.1. Bảng dữ liệu.

Điện áp vào	Dữ liệu kênh	Độ phân giải %
2V	N7:99	
3V		
4V		
5V		
2V	N7:100	
3V		
4V		
5V		
2V	N7:101	
3V		
4V		
5V		
2V	N7:102	
3V		
4V		
5V		

Thí nghiệm 22.1. SỰ CHUYỂN ĐỔI A/D VÀ D/A

Mục đích của thí nghiệm này là kết hợp Bài tập 21.0 và 22.0 vào một quy trình vận hành chung (Hình 22.3). Ưu điểm cơ bản là sử dụng các tín hiệu nhập (chiết áp kế) của bộ ADC để điều khiển các tín hiệu xuất DAC (bộ truyền động AC hoặc volt kế). Bạn hãy tạo tập tin xử lý riêng và nhập các lệnh chuyển khối đọc /viết của DAC trong Bài tập 21.0 và các lệnh chuyển khối của ADC trong Bài tập 22.0. Lập trình 4 lệnh Move và sử dụng chúng để chuyển tín hiệu xuất digital của ADC thành tín hiệu nhập digital của DAC. Ví dụ, chuyển N7:99 của bộ chuyển đổi analog nhập đến N7:25 của module xuất analog.

Khi bạn điều chỉnh các chiết áp, tốc độ của bộ truyền động AC sẽ thay đổi. Nếu sử dụng VOM kế, bạn sẽ có khả năng thay đổi điện áp ra của các lệnh xuất DAC.



Hình 22.3. Sơ đồ quy trình analog – digital và digital – analog.

Bài tập lớn 23.0

PID

MỤC ĐÍCH

Nghiên cứu lệnh PID điều khiển biến quá trình tại điểm xác lập mong muốn.

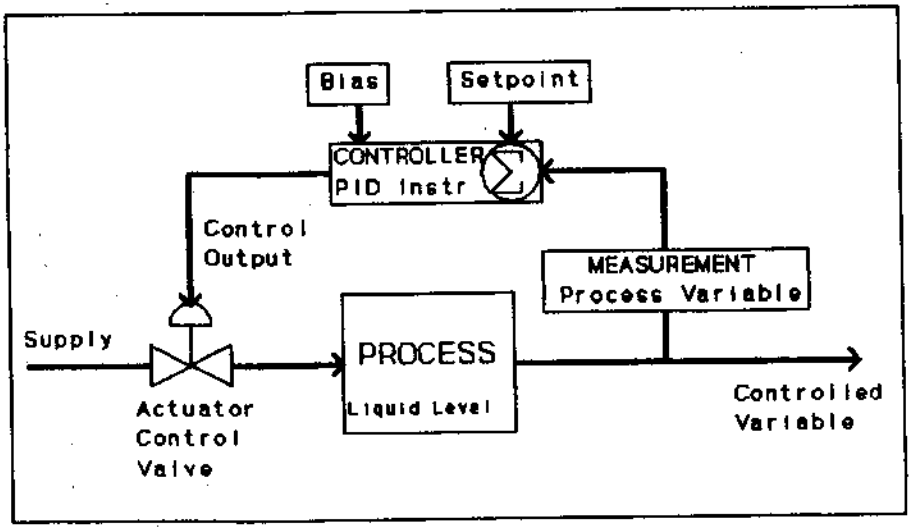
NỘI DUNG

Sau khi hoàn tất Bài Tập này, bạn sẽ có khả năng:

- Lập cấu hình lệnh PID.
- Lập cấu hình module digital – analog.
- Sử dụng lệnh chuyển khối đọc và viết.

TÀI LIỆU

Để biết chi tiết về các lệnh được dùng trong Bài tập này, bạn hãy đọc phụ lục A, Các lệnh PID và chuyển khối. PID, BTR, BTW.



Hình 23.1. Vòng điều khiển hồi tiếp 1

THẢO LUẬN

Sơ đồ khối điều khiển hồi tiếp đơn giản trên Hình 23.1 minh họa 5 phần tử chính trong hầu hết các mạch hồi tiếp.

1. *Đo đạc.* Cần đo dữ liệu để biết giá trị hiện hành của biến được điều khiển từ sơ đồ. Các biến gồm lưu lượng, áp suất, nhiệt độ ... với bộ điều khiển lập trình PID (tỷ lệ – tích phân – vi phân), dữ liệu đo này được coi là biến quá trình.

2. *Điểm xác lập.* Giá trị nhập vào bộ điều khiển để so sánh với tín hiệu đo, xác định giá trị này cao hơn hay thấp hơn giá trị của điểm xác lập.

3. *Bộ tác động.* Bộ này điều chỉnh sự cung cấp năng lượng hoặc vật liệu cho quá trình, có thể là van điều khiển hoặc bộ truyền động tần số, ...

4. *Quá trình.* Các quá trình công nghiệp.

5. *Bộ điều khiển.* Sử dụng hiệu số giữa tín hiệu xác lập trước và tín hiệu đo tạo ra tín hiệu xuất cung cấp cho bộ tác động. Đối với sự điều khiển quá trình cụ thể, sự thay đổi tín hiệu ra bộ điều khiển phải theo chiều ngược với sự thay đổi tín hiệu đo. Với ΓLC, đây là chức năng của lệnh PID.

Mục đích của sơ đồ khối trên Hình 23.1 là duy trì mực chất lỏng xác định trong bình chứa. Khi mực chất lỏng càng cao, lưu lượng càng giảm. Tương tự, khi mực chất lỏng giảm, bộ điều khiển kích hoạt bộ tác động và tăng lưu lượng đi vào bình. Bộ điều khiển hoạt động nhằm duy trì sự cân bằng theo tỷ suất ổn định, khôi phục sự cân bằng giữa cung và cầu.

Bài tập

Trong Bài tập này, bạn nghiên cứu lệnh PID và sự hoạt động của lệnh này trong vòng điều khiển lưu lượng đơn giản theo tỷ lệ. Kỹ thuật được sử dụng ở đây là duy trì logic của mạch theo lượng tối thiểu để tập trung vào các chức năng của lệnh PID. Do đó, để biểu thị chuyển động của bộ tác động cuối (van điều khiển) bạn sẽ sử dụng volt kế analog. Bộ chuyển đổi thumbwheel được dùng để biểu thị giá trị biến quá trình (giá trị đo). Bộ thumbwheel thứ hai được dùng để nhập dữ liệu điểm xác lập. Hai bộ chuyển đổi này cho phép bạn thay đổi bằng tay biến quá trình và giá trị điểm xác lập. Bạn có thể quan sát các thông số PID và tác động của chúng đối với chuyển dịch của volt kế analog.

LOGIC CHƯƠNG TRÌNH

1. Module digital – analog được dùng để tạo ra điện áp xuất điều khiển biến. Bạn lập trình các lệnh chuyển khối đọc và viết hướng đến module digital – analog (Bài tập 21.0).

   A. Lập cấu hình độ phân giải module D/A với BCD từ 0 đến 4095.

   B. Nối ngõ ra kênh 1 từ 0 đến 10V xuất của module xuất analog đến volt kế.

2. Lập trình lệnh PID sử dụng thumbwheel làm ngõ nhập biến quá trình và volt kế là ngõ xuất biến điều khiển.

3. Lập trình lệnh FRD và lệnh Move để chuyển thông tin thumbwheel thành thông số điểm xác lập của lệnh PID. Lập trình bộ chuyển đổi thumbwheel thứ hai. Sử dụng lệnh FRD để chuyển dữ liệu thumbwheel này thành tín hiệu nhập biến quá trình của lệnh PID.

4. Tín hiệu xuất biến quá trình là = gain 3 setfoint – biến quá trình + thiên áp

CÁC BƯỚC THỰC HIỆN

1. Viết chương trình sử dụng lệnh PID để điều khiển quá trình tỷ lệ đơn giản. Bạn nhập chương trình được nêu trên Hình 23.2.

2. Bước kế tiếp là lập cấu hình module D/A. bạn xác lập độ phân giải cực đại của kênh 1 và kênh 2 là 4095 (xem Bài Tập 20.0 về các bước lập cấu hình module).

3. Lập cấu hình module PID bằng cách đặt con trỏ trên lệnh PID và nhấn data monitor (F8). Nhập thông tin được nên trên các Hình 23.3 và 23.4.

   Chú ý: Bạn hãy xem phần hướng dẫn lệnh PID (Phụ lục A) để biết đặc tính kỹ thuật của các thông số PID.

4. Bạn hãy quan sát và nhập dữ liệu các thông số dưới đây, như được nêu trong menu cấu hình điều khiển PD10.0 trên Hình 23.3.

   A. Điểm xác lập – Xác lập bộ chuyển đổi thumbwheel thứ nhất (N7:6) là 50.

   B. Biến quá trình – Xác lập dữ liệu thumbwheel thứ hai (N7:7) là 50.

   C. Sai số – Hiển thị hiệu số giữa điểm xác lập và biến quá trình, với điểm xác lập là 50 và biến quá trình cũng 50, sai số là zero.

```
Rung 2:0
FIGURE 23.2, PROJECT 23. PID

THIS RUNG IS FOR THE 1771-OFE DIGITAL/ANALOG OUTPUT MODULE.
|      N7:40          N7:20          + BTW ------------------------------ +        |
+ ---- ] / [ ---------] / [ ----------+ BLOCK TRANSFER WRITE              + -(EN)-+
|       15             15             | Rack                          00 |        |
|                                     | Group                          4 + -(DN)  |
|                                     | Module                         1 |        |
|                                     | Control Block                N7:20 + -(ER)  |
|                                     | Data File                    N7:25 |        |
|                                     | Length                        13 |        |
|                                     | Continuous                     N |        |
|                                     + ------------------------------- +        |

Rung 2:1
THIS RUNG IS FOR THE 1771-OFE DIGITAL/ANALOG OUTPUT MODULE.
|      N7:40          N7:20          + BTW ------------------------------ +        |
+ ---- ] / [ ---------] / [ ----------+ BLOCK TRANSFER READ               + -(EN)-+
|       15             15             | Rack                          00 |        |
|                                     | Group                          4 + -(DN)  |
|                                     | Module                         1 |        |
|                                     | Control Block                N7:40 + -(ER)  |
|                                     | Data File                    N7:45 |        |
|                                     | Length                         5 |        |
|                                     | Continuous                     N |        |
|                                     + ------------------------------- +        |

Rung 2:2
THIS RUNG IS FOR THE PID INSTRUCTION
PROCESS VARIABLE IS THE MEASURED VALUE OF THE THMBWHEEL SW.
CONTROL VARIABLE IS THE CALCULATED OUTPUT VALUE SENT TO CHANNEL 2, MULTIMETER.
|      1:000                          + PID ------------------------------ +        |
+ ---- ] / [ ---------------------------+ PID                            + -  - +
|       00                            | Control Block                PD10:0 |        |
|                                     | Process variable              N9:0 |        |
|                                     | Tieback                          0 |        |
|                                     + ------------------------------- +        |

Rung 2:3
THIS RUNG CONVERTS THE BCD OUTPUT OF THE THUMBWHEEL TO BINARY AND MOVES IT
TO A DESTINATION FILE N9:1.
THUMBWHEEL N7:6 = SETPOINT VALUE OF PID INSTRUCTION.
|                                     + FRD ------------------------------ +        |
+ ------------------------------------+ FROM BCD                       + -  - +
|                                     | Source                        N7:6 |        |
|                                     |                                 80 |        |
|                                     | Destination                   N9:1 |        |
|                                     |                                 50 |        |
|                                     + ------------------------------- +        |
```

Hình 23.2. Chương trình PID

```
Rung 2:4
THIS RUNG MOVES THE CONVERTED BINARY VALUE OF N9:1 TO THE SETPOINT OF THE
PID INSTRUCTION.
!                                   + MOV ------------------------------ +     !
+ ------------------------------------ + MOVE                            +  -  +
!                                   ! Source                    N9:1    !     !
!                                   !                             50    !     !
!                                   ! Destination           PD10:0.SP   !     !
!                                   !                         50.00000  !     !
!                                   + ------------------------------ +     !
Rung 2:5
THIS RUNG CONVERTS THE BCD VALUE OF THE THUMBWHEEL ADDRESS (N7:7) AND MOVES
IT TO THE PROCESS VARIABLE OF THE PID INSTRUCTION.
THUMBWHEEL N7:7 = PROCESS VARIABLE, (MEASURED VALUE)
!                                   + FRD ------------------------------ +     !
+ ------------------------------------ + FROM BCD                        +  -  +
!                                   ! Source                     N7:7    !     !
!                                   !                              80    !     !
!                                   ! Destination               N9:0    !     !
!                                   !                              50    !     !
!                                   + ------------------------------ +     !
Rung 2:6
THIS RUNG IS TO ACTIVATE THE THUMBWHEEL SWITCHES.
!                                   + BTR ------------------------------ +     !
+ ------------------------------------ + BLOCK TRANSFER READ            + -(EN)-+
!                                   ! Rack                        00    !     !
!                                   ! Group                        2  + -(DN)   !
!                                   ! Module                       1    !     !
!                                   ! Control block              N7:0  + -(ER)   !
!                                   ! Data file                  N7:5    !     !
!                                   ! Length                       0    !     !
!                                   ! Continuous                   Y    !     !
!                                   + ------------------------------ +     !
```

Hình 23.2. Chương trình PID (tt)

D. Bias – Bạn nhập giá trị Bias là 50%.

E. Độ lợi – Nhập độ lợi là 1.

F. Thời gian xác lập lại – Nhập số cực đại 3.04e+38 để loại bỏ sự điều khiển tích phân.

G. Engineering Unit Maximum (Đơn vị kỹ thuật cực đại) - Bạn nhập 100

H. Xuất – độ lợi 3 (SP–PV) + Bias. Phần xuất sẽ hiển thị kết quả của phương trình PID. Giá trị xuất điều khiển sẽ giữa 0 và 100%. Ví dụ, độ lợi 1 là 3 (SP50% - PV 50%) + Bias 50% = xuất 50%. Đồng hồ ngõ ra sẽ chỉ 50% trên thang đo, tương ứng 5 V.

Setpoint:	50.00000	Proportional gain (Kc):	1.000000
Process Var.:	50.00000	Reset Time (Ti) [mins/repeat]:	3.400e+38
Error:	0.000000	Derivative Rate (Td) [mins]:	0.000000
Output %:	50.00000		
Mode:	AUTO	Deadband:	0.000000
PV Alarm:	HIGH	Output Bias %:	50.00000
Deviation Alarm:	POS		
Output Limiting:	NONE	Tieback %:	0.000000
SP Out of Range:	NO	Set Output %:	50.00000
Error within Deadband:	NO		
PID Initialized:	YES		
A/M Station Mode:	AUTO		
Software A/M Mode:	AUTO		
Status Enable (EN):	1		

Hình 23.3. PID Monitor Menu

PID Equation:	DEPENDENT	Engineering Unit Maximum:	100.0000
Derivative of:	PV	Engineering Unit Minimum:	0.000000
Control Action:	SP-PV		
PV Tracking:	NO	Input Range Maximum:	100.0000
		Input Range Minimum:	0.000000
Update Time (secs):	0.5000000	Output Limit High %:	100.0000
		Output Limit Low %:	0.000000
Cascaded Loop:	NO	PV Alarm High:	0.000000
Cascaded Type:	—	PV Alarm Low:	0.000000
Master to This Slave:	—	PV Alarm Deadband:	0.000000
		(+) Deviation Alarm:	0.000000
		(–) Deviation Alarm:	0.000000
		Deviation Alarm Deadband:	0.000000

Hình 23.4. PID Configuration Menu.

Bạn hãy thực nghiệm với các giá trị khác nhau của điểm xác lập và biến quá trình. Bạn quan sát giá trị xuất và so sánh chúng với số đo trên đồng hồ.

5. Bạn hãy dùng các bộ chuyển đổi thumbwheel để nhập các thông số của biến quá trình và điểm xác lập, hoàn tất các Bảng dữ liệu

23.1 và 23.2. Sử dụng ví dụ nêu trên, bạn hãy tính toán và ghi lại từng giá trị xuất của điện áp điều khiển. Sau đó kiểm chứng kết quả với giá trị PID và đo theo đồng hồ đo (volt kế).

Các xác lập dùng cho Bảng 23.1

Độ lợi = 1, Bias = 50%, Đơn vị kỹ thuật cực đại = 100.

Setpoint	Process Variable	Bias %	Output %	Meter Output

Bảng 23.1. Bảng dữ liệu.

Các xác lập dùng cho Bảng 23.2

Độ lợi = 0.5, Bias = 50%, Đơn vị kỹ thuật cực đại = 1000.

Setpoint	Process Variable	Bias %	Output %	Meter Output

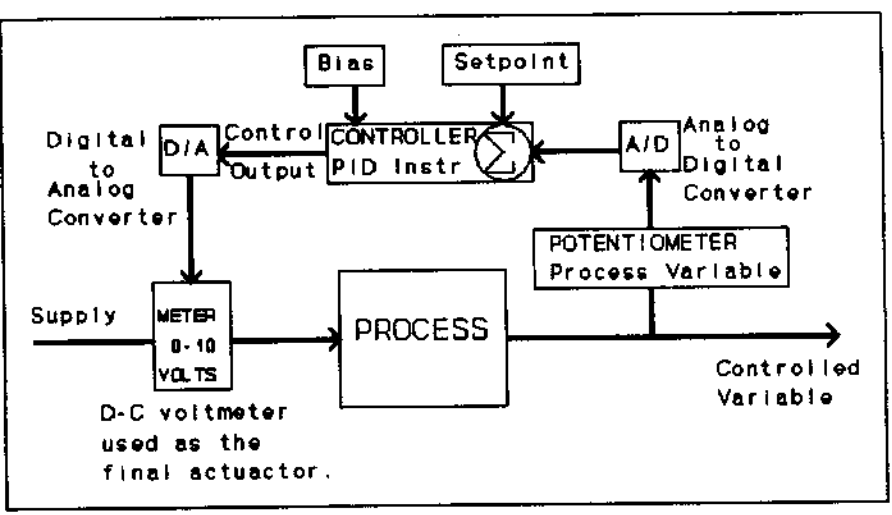
Bảng 23.2. Bảng dữ liệu.

6. Sau khi hoàn tất, bạn lưu và in chương trình. Để củng cố và nâng cao kiến thức, bạn hãy thực hành Thí nghiệm 23.1.

Thí nghiệm 23.1. VÒNG ĐIỀU KHIỂN # 2

Đối với Thí nghiệm này, bạn lập trình vòng điều khiển PID thứ hai, gồm cả điều khiển tỷ lệ và điều khiển tích phân (Hình 23.5). Bạn lập trình theo các thông số dưới đây:

1. Khối điều khiển lệnh PID là PD10:1, biến quá trình N7:100, biến điều khiển N7:25
2. Lập trình bộ chuyển đổi nhập analog (xem Bài tập 22.0), xác lập độ phân giải cực đại là 4095.
3. Lập trình bộ chuyển đổi xuất analog, sử dụng lại lệnh khối trong Bài tập 23.0.
4. Bạn xác lập các thông số sau: Điểm xác lập = 50; Bias = 50%; thời gian cài đặt lại (tích phân) = 25; đơn vị kỹ thuật cực đại = 100; thời gian cập nhật = 1.



Hình 23.5. Vòng điều khiển hồi tiếp 2

Bài tập lớn 24.0

VĂN BẢN CẤU TRÚC

MỤC ĐÍCH

Nghiên cứu công dụng của lập trình văn bản cấu trúc

NỘI DUNG

Sau khi hoàn tất Bài Tập này, bạn sẽ có khả năng:

- Chuyển chương trình logic bậc thang cơ bản thành chương trình văn bản cấu trúc.
- Nghiên cứu công dụng của bộ xử lý từ ngữ để triển khai văn bản cấu trúc và lập tài liệu.

THẢO LUẬN

Phương pháp lập trình đã được trình bày trong các Bài tập trước là kỹ thuật logic bậc thang sắp xếp các lệnh liên quan. Phương pháp lập trình các lệnh logic thứ hai là *văn bản cấu trúc*. Chương trình văn bản cấu trúc là chuỗi các câu lệnh và dấu nhắc, bạn có thể dùng để thực hiện cùng các chức năng logic như logic bậc thang. Mỗi chức năng logic được diễn tả trên một dòng theo hoạt động của dòng đó với sự sử dụng các câu lệnh hoặc dấu nhắc.

Văn bản cấu trúc có thể được coi là công cụ lập trình, giúp bạn làm việc với hiệu suất và năng suất cao hơn, do đó bạn cần nắm vững loại công cụ này.

Bài tập

Bài tập này hướng dẫn cách chuyển đổi chương trình logic bậc thang thành văn bản cấu trúc. Bộ xử lý (phần mềm) WordPerfect sẽ được dùng để lập trình và ghi lại chương trình văn bản cấu trúc. Chương trình trong Bài tập này là mạch logic bậc thang đơn giản (Hình 24.1). Vấn đề chính là viết chương trình văn bản cấu trúc để mô phỏng chương trình logic bậc thang.

LOGIC CHƯƠNG TRÌNH

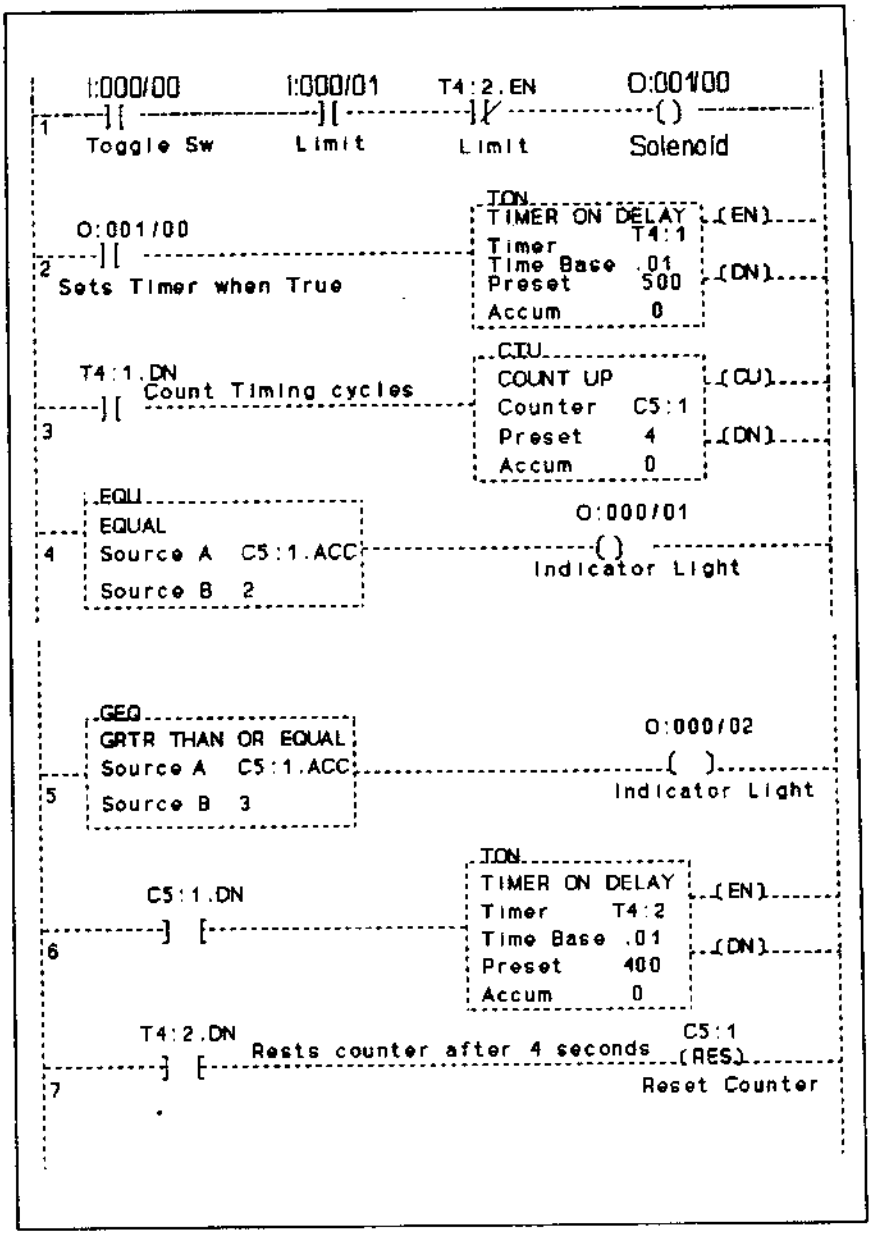
Bước đầu tiên, bạn hãy xem chương trình trên Hình 24.1. Để kiểm nghiệm sự vận hành của mạch này, bạn có thể lập trình theo logic bậc thang trước. Các bước vận hành của mạch này như sau:

1. Chương trình có công tắc toggle kích hoạt ngõ ra solenoid. Khi cuộn dây ngõ ra bộ khởi động được cấp điện, TON # 1 được kích hoạt và bắt đầu chu kỳ thời chuẩn.
2. Lệnh CTU đếm số chu kỳ thời chuẩn của TON#1. khi CTU đạt đến 4, sẽ kích hoạt TON#2 với done bit.
3. Bộ thời chuẩn thứ hai (TON#2) ngắt điện ngõ ra bộ khởi động và cài đặt lại CTU với done bit. Bit kích hoạt bộ thời chuẩn này (EN) giới hạn sự kích hoạt cuộn solenoid.
4. Hai lệnh so sánh, một lệnh kích hoạt đèn ngõ ra khi bằng 2, lệnh thứ hai kích hoạt đèn ngõ ra thứ hai khi bằng hoặc lớn hơn 3.

CÁC BƯỚC THỰC HIỆN

1. Hình 24.2 minh họa chương trình văn bản cấu trúc với các câu lệnh thay cho logic bậc thang trên Hình 24.1. Bước thứ nhất là tạo tập tin cấu trúc. Sau đó chuyển sang giám sát trực tiếp tập tin văn bản cấu trúc và nhập chương trình này trực tiếp vào menu tập tin văn bản cấu trúc.

   Khi đó, phần mềm lập trình văn bản cấu trúc chưa cho phép bạn nhập tài liệu theo định dạng văn bản cấu trúc. Do đó, khi nhập chương trình trực tiếp theo định dạng này, bạn chưa thể nhập tài liệu mô tả. Bên cạnh các câu lệnh là các dấu lược với vạch đỏ hoặc câu lệnh REM để ngăn cách chúng với văn bản cấu trúc. Bạn chỉ nhập văn bản cấu trúc được nêu trên Hình 24.2.
2. Bước kế tiếp, bạn cho chạy thử chương trình, kích hoạt công tắc toggle (ON) và quan sát sự vận hành của từng dòng văn bản. Bạn sẽ thấy các dòng văn bản bộ thời chuẩn và bộ đếm. Văn bản cấu trúc thực hiện cùng các chức năng như logic bậc thang trên Hình 24.1
3. Để đưa tài liệu vào chương trình bạn phải tạo tập tin nguồn (sao chép) của chương trình văn bản cấu trúc. Tập tin nguồn là sao chép nguyên văn tập tin văn bản cấu trúc có chứa tài liệu mô tả có thể xem trên bản in hoặc trên màn hình với tập tin STX.



Hình 24.1. Chương trình logic bậc thang để chuyển đổi thành chương trình văn bản cấu trúc.

```
REM Comments separated with apostrophe ' or REM statement.
'Sol Output. toggle Sw Limit Sw Limit Timer
O:001/00 := I:000/00 AND I:00/01 AND !T4:2.EN;
O:001/00 TON (T4:1, 0.01, 500, 0); 'Starter output activating Timer #1.
T4:1.DN CTU (C5:1, 4, 0); 'Timer #1 done bit activating counter-up.
O:000/01 := C5:1.ACC = 2; ' Output 01 "true" when ctu acc value is equal 2.
O:000/02 := C5:1.ACC >= 3; ' Output bit 02 "true" when ctu acc value >= 3.
C5:1.DN TON (T4:2, 0.01, 400, 0); ' Timer #2 is set by ctu done bit.
T4:2.DN RES (C5:1); ' Counter is reset with timer #2 done bit.
```

Hình 24.2. Chương trình văn bản cấu trúc với các tài liệu.

4. Để tạo tập tin nguồn, bạn hãy chuyển sang tính năng biên tập tiên tiến (F7). Bạn hãy chọn các bậc thang của tập tin văn bản cấu trúc (F1), copy (F3), sau đó copy vào vào đĩa của bạn (F7), bạn hãy đặt tên tập tin là TEST. Điều này sẽ sao chép chương trình theo tập tin STX với tên TEST.STX.

5: Bạn có thể dán (paste) tập tin TEST. STX trở lại từ đĩa và nhập các chú thích (tài liệu) vào định dạng đó hoặc dùng chương trình xử lý từ ngữ, chẳng hạn WordPerfect. Trong WordPerfect, bạn hãy truy xuất tập tin văn bản cấu trúc (TEST.STX), bạn có thể nhập số lượng chú thích tùy ý. Trong định dạng này bạn còn có thể in tập tin của mình.

   Chú ý: Để lưu chương trình, bạn hãy dùng định dạng ASC II TEXT File (CTRL F5 Text Out trong WordPerfect 5.1) Lưu chương trình theo định dạng này sẽ cho phép bạn sao chép trực tiếp vào định dạng văn bản cấu trúc. Để nhập lại tập tin này theo định dạng văn bản cấu trúc, bạn sử dụng các tính năng biên tập tiên tiến được dán từ đĩa (F.8).

6. Bạn sẽ có khả năng xem tập tin STX với các tài liệu trong định dạng STX. Bạn cũng có thể thực hiện các thay đổi tài liệu trong tập tin này. Nhưng, nếu bạn chấp nhận chúng theo tập tin văn bản cấu trúc, bạn hãy nhấn (F8), bất kể các thay đổi.

7. Sau khi hoàn tất, bạn hãy kiểm tra sự vận hành của chương trình và chỉnh sửa lại, nếu cần thiết.

ÔN TẬP

Để củng cố và nâng cao kiến thức, bạn hãy thực hành với Thí nghiệm 24.1.

Thí nghiệm 24.1. VĂN BẢN CẤU TRÚC 2

Bạn hãy xem lại logic bậc thang của Bài tập 10.0, các Hình 10.1 và 10.2. Bạn hãy chuyển chương trình logic bậc thang này thành văn bản cấu trúc. Bạn hãy xây dựng tập tin nguồn với các chú thích, tài liệu, ... và in ra.

PHỤ LỤC A

Các Lệnh Thông Dụng Trong Lập Trình PLC

Các lệnh relay

Các lệnh bộ thời chuẩn

Các lệnh bộ đếm

Các lệnh so sánh

Các lệnh tính toán

Các lệnh logic

Các lệnh chuyển đổi

Các lệnh dịch chuyển và chỉnh sửa bit

Các lệnh tập tin

Các lệnh chẩn đoán

Các lệnh bộ ghi dịch chuyển.

Các lệnh chuỗi thứ tự

Các lệnh điều khiển chương trình

Các lệnh thông báo và điều khiển quá trình.

Các lệnh truyền khối

Các lệnh ASC II

Các lệnh relay

Lệnh		Hoạt động
I:012 —] [— 07	Examine On XIC	Kiểm tra bit bảng dữ liệu I:012/07, tương ứng đầu dây 7 của module nhập thuộc nhánh I/O số 1, nhóm I/O số 2. Nếu bit bảng dữ liệu này là (1), lệnh là đúng.
I:012 —]/[— 07	Examine Off XIO	Kiểm tra bit bảng dữ liệu I:012/07, tương ứng đầu dây 7 của module nhập thuộc nhánh I/O số 1, nhóm I/O số 2. Nếu bit bảng dữ liệu này là (0), lệnh là đúng.
OI:013 —( )— 01	Output Energize OTE	Nếu các điều kiện nhập trước lệnh xuất này trên cùng bậc thang là đúng, xác lập (1) bit O:013/01, tương ứng đầu dây 1 của module xuất thuộc nhánh I/O số 1, nhóm I/O số 3.
O:013 —(L)— 01	Output Latch OTL	Nếu các điều kiện nhập trước lệnh xuất này trên cùng bậc thang là đúng, xác lập (1) bit O:013/01, tương ứng đầu dây 1 của module xuất thuộc nhánh I/O số 1, nhóm I/O số 3. Bit bảng dữ liệu này giữ nguyên xác lập cho đến khi lệnh OTU xác lập lại bit.
O:013 —(U)— 01	Output Unlatch OTU	Nếu các điều kiện nhập trước lệnh xuất này trên cùng bậc thang là đúng, xác lập lại (0) bit O:013/01, tương ứng đầu dây 1 của module xuất thuộc nhánh I/O số 1, nhóm I/O số 3. Điều này là cần thiết để xác lập lại bit đã khóa.
01 —(IIN)—	Immediate Input IIN	Lệnh này cập nhật từ ngữ của các bit nhập – hình ảnh trước sự cập nhật nhập – hình ảnh bình thường kế tiếp. Đối với thiết bị tại chỗ, sự quét chương trình bị gián đoạn khi quét các ngõ nhập của nhóm I/O có địa chỉ; đối với thiết bị xa, sự quét chương trình bị gián đoạn để cập nhật hình ảnh nhập với các trạng thái mới nhất được phát hiện trong bộ đệm I/O xa.
01 —(IOT)—	Immediate Output IOT	Lệnh này cập nhật từ ngữ của các bit xuất – hình ảnh trước khi cập nhật xuất – hình ảnh bình thường kế tiếp. Đối với thiết bị tại chỗ, sự quét chương trình bị gián đoạn khi quét các ngõ xuất của nhóm I/O có địa chỉ; đối với thiết bị xa, sự quét chương trình bị gián đoạn để cập nhật bộ đệm I/O xa với các trạng thái mới nhất được phát hiện trong ảnh xuất.

Các lệnh bộ thời chuẩn

Lệnh	Hoạt động

TON

TIMER ON DELAY
Timer T4:1
Time Base 1.0
Preset 15
Accum 0

Timer On Delay
TON

Status Bits:
EN - Enable
TT - Timer Timing
DN - Done

Nếu các điều kiện nhập là đúng, bộ thời chuẩn T4:1 bắt đầu tăng dần số gia theo các khoảng 1-giây. Khi giá trị tích lũy lớn hơn hoặc bằng giá trị cài đặt trước (15), bộ này dừng lại và xác lập done bit thời chuẩn.

Điều kiện thang	EN 15	TT 14	DN 13	Giá trị tích lũy	Trạng thái TON
Sai	0	0	0	0	Xác lập lại
Đúng	1	1	0	Tăng	Thời chuẩn
Đúng	1	0	1	>= cài đặt trước	Done

TOF

TIMER OFF DELAY
Timer T4:1
Time Base .01
Preset 180
Accum 0

Timer Off Delay
TOF

Status Bits:
EN - Enable
TT - Timer Timing
DN - Done

Nếu các điều kiện nhập là sai, bộ thời chuẩn T4:1 bắt đầu tăng dần số gia theo các khoảng 10 ms khi bậc thang duy trì trạng thái sai. Khi giá trị tích lũy lớn hơn hoặc bằng giá trị cài đặt trước (180), bộ này dừng lại và xác lập done bit thời chuẩn.

Điều kiện thang	EN 15	TT 14	DN 13	Giá trị tích lũy	Trạng thái TOF
Đúng	1	0	1	0	Xác lập lại
Sai	0	1	1	Tăng	Thời chuẩn
Sai	0	0	0	>= cài đặt trước	Done

RTO

RETENTIVE TIMER ON
Timer T4:10
Time Base 1.0
Preset 10
Accum 0

Retentive Timer On
RTO

Status Bits:
EN - Enable
TT - Timer Timing
DN - Done

Nếu các điều kiện nhập là đúng, bộ thời chuẩn T4:10 bắt đầu tăng dần số gia theo các khoảng 1–giây trong khi thang là đúng. Khi thang chuyển sang trạng thái sai, bộ thời chuẩn dừng lại. Nếu thang trở lại trạng thái đúng, bộ thời chuẩn tiếp tục hoạt động. Khi giá trị tích lũy lớn hơn hoặc bằng giá trị cài đặt trước (10), bộ này dừng lại và xác lập done bit thời chuẩn.

Điều kiện thang	EN 15	TT 14	DN 13	Giá trị tích lũy	Trạng thái RTO
Sai	0	0	0	0	Xác lập lại
Đúng	1	1	0	Tăng	Thời chuẩn
Sai	0	0	0	Duy trì	Khử kích hoạt
Đúng	1	0	1	>= cài đặt trước	Done

T4:1
—(RES)—

Timer Reset
RES

Nếu các điều kiện nhập là đúng, bộ thời chuẩn T4: 1 xác lập lại. Lệnh này xác lập lại các bộ thời chuẩn và bộ đếm, và các khối điều khiển. Điều này là cần thiết để xác lập lại giá trị tích lũy của RTO.

Các lệnh bộ đếm

Lệnh	Hoạt động

CTU

COUNT UP	
Counter	C5:1
Preset	10
Accum	0

Count Up
CTU

Nếu các điều kiện nhập là đúng, bộ đếm C5:1 bắt đầu đếm, tăng 1 mỗi khi bậc thang chuyển từ sai sang đúng. Khi giá tích lũy lớn hơn hoặc bằng giá trị cài đặt trước (10), bộ đếm xác lập lại done bit.

Status Bits:
CU–Count Up
CD–Count Down
DN–Count Down one
OV–Overflow
UN–Underflow

Điều kiện thang	CU 15	DN 13	OV 12	Giá trị tích lũy	Trạng thái CTU
Sai	0	0	0	0	Reset
Đúng toggle	1	0	0	Tăng 1	Đếm
Đúng	1	1	0	>= preset	Done
Đúng	1	1	1	>32767	Quá mức

CTD

COUNT DOWN	
Counter	C5:1
Preset	10
Accum	35

Count Down
CTD

Nếu các điều kiện nhập là đúng, bộ thời chuẩn C5:1 bắt đầu đếm, giảm 1 mỗi khi bậc thang chuyển từ sai sang đúng. Khi giá trị tích lũy nhỏ hơn giá trị cài đặt trước (10), bộ đếm xác lập lại done bit.

Status Bits:
CU–Count Up
CD–Count Down
DN–Count Down one
OV–Overflow
UN–Underflow

Điều kiện thang	CD 14	DN 13	UN 11	Giá trị tích lũy	Trạng thái CTD
Sai	0	0	0	0	Reset
Sai	0	1	0	>= preset	Tải trước
Đúng toggle	1	1	0	Giảm 1	Đếm
Đúng	1	0	0	< preset	Done
Đúng	1	0	1	< -32768	Dưới mức

Các lệnh so sánh

Lệnh	Hoạt động

```
LIM
LIMIT TEST (CIRC)
Low limit      N7:10
                   3
Test           N7:15
                   4
High limit     N7:20
                  22
```

Limit Test
LIM

Nếu giá trị kiểm tra (N7:15) > = giới hạn dưới (N7:10) và < = giới hạn trên (N7: 20), lệnh là đúng.

Giới hạn dưới	Kiểm tra	Giới hạn trên	Giới hạn
0	0	10	T
-5	5	10	T
5	11	10	F
10	0	0	T
10	5	-5	F
10	11	5	T

```
MEQ
MASKED EQUAL
Source         D9:5
                0000
Mask           D9:6
                0000
Compare        D9:10
                0000
```

Mask Compare Equal
MEQ

Bộ xử lý lấy giá trị từ nguồn (D9:5) và cho đi qua giá trị thông qua Mask (D9:6). Sau đó bộ xử lý so sánh kết quả với giá trị so sánh (D9:10). Nếu kết quả bằng giá trị so sánh, lệnh là đúng

Nguồn	Mask	So sánh	MEQ
0008	0008	0009	F
0008	0001	0001	F
0087	000F	0007	T
0087	00F0	0007	F

```
CMP
COMPARE
Expression
N7:5 = N7:10
```

Compare
CMP

Nếu biểu thức là đúng, lệnh là đúng. Lệnh CMP có thể thực hiện các so sánh: bằng (=), nhỏ hơn (<), nhỏ hơn hoặc bằng (<=), lớn hơn (>), lớn hơn hoặc bằng (>=), không bằng (<>), và các biểu thức phức tạp (đến 80 ký tự).

```
xxx
xxxxxxxxxxxx
Source A       N7:5
                   3
Source B       N7:10
                   1
```

Nguồn A	Nguồn B	EQU	GEQ	GRT	LEQ	LES	NEQ
10	10	T	T	F	T	F	F
5	6	F	F	F	T	T	T
21	20	F	T	T	F	F	T
-30	-31	F	T	T	F	F	T
-15	-14	F	F	F	T	T	T

Bằng EQU	Nếu giá trị trong nguồn A (N7:5) = giá trị trong nguồn B (N7:10), lệnh này là đúng.
Lớn hơn hoặc bằng GEQ	Nếu giá trị trong nguồn A (N7:5) > = giá trị trong nguồn B (N7:10), lệnh này là đúng.
Lớn hơn GRT	Nếu giá trị trong nguồn A (N7:5) > giá trị trong nguồn B (N7:10), lệnh này là đúng.
Nhỏ hơn hoặc bằng LEQ	Nếu giá trị trong nguồn A (N7:5) <= giá trị trong nguồn B (N7:10), lệnh này là đúng.
Nhỏ hơn LES	Nếu giá trị trong nguồn A (N7:5) < giá trị trong nguồn B (N7:10), lệnh này là đúng.
Không bằng NEQ	Nếu giá trị trong nguồn A (N7:5) <> giá trị trong nguồn B (N7:10), lệnh này là đúng.

Các lệnh tính toán

Lệnh		Hoạt động
CPT COMPUTE Dest N7:3 3 Expression N7:4 – (N7:6 * N7:10)	Compute CPT	Nếu các điều kiện nhập là đúng sẽ tính giá trị biến thức N7:4 – (N7:6* N7:10) và lưu kết quả vào đích Destination (N7:3). Lệnh CPT có thể thực hiện các phép toán: cộng (+), trừ (-), nhân (*), chia (I), chuyển đổi từ BCD (FRD), chuyển đổi sang BCD (TOD), căn bậc hai (SQR), các phép logic AND, OR, NOT, XOR, âm (-), loại bỏ (0), và dịch chuyển, nâng X lên lũy thừa Y (**), radian (RAD), độ (DEG), logarit (LOG), log tự nhiên (LN), sine (SIN), cosine (COS), tang (TAN), arcsine (ASN), arccosine (ACS), arctang (ATN), và các biểu thức phức (đến 80 ký tự).
ACS ARCCOSINE Source F8:19 0.7853982 Dest F8:20 0.6674572	Arc cosine ACS	Nếu điều kiện nhập là đúng, sẽ tính arccosine của giá trị trong F8:19 và lưu kết quả vào F8:20 Trạng thái bit: C — luôn luôn xác lập lại V — xác lập, nếu xảy ra sự quá mức; nếu không, sẽ xác lập lại Z — xác lập, nếu kết quả là zero; nếu không; sẽ xác lập lại S — luôn luôn xác lập lại
ADD ADD Source A N7:3 3 Source B N7:4 1 Dest N7:12 4	Addition ADD	Khi điều kiện nhập là đúng, sẽ cộng giá trị trong nguồn A (N7: 3) với giá trị trong nguồn B(N7:4) và lưu kết quả ở đích (N7:12) Trạng thái bit: C — xác lập, nếu có nhớ; nếu không, sẽ xác lập lại. V — xác lập, nếu có quá mức; nếu không, sẽ xác lập lại. Z — xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại. S — xác lập, nếu kết quả là số âm; nếu không, sẽ xác lập lại.

Lệnh	Hoạt động

ASN

ARCSINE

Source	F8:17
	0.7853982
Dest	F8:18
	0.9033391

Arc sine
ASN

Khi điều kiện nhập là đúng, sẽ tính arcsine của giá trị trong F8:17 và lưu kết quả vào F8:18

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại

ATN

ARCTANGENT

Source	F8:21
	0.7853982
Dest	F8:22
	0.6657737

Arc tangent
ATN

Khi điều kiện nhập là đúng, sẽ tính arctang của giá trị trong F8:21 và lưu kết quả vào F8:22

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

AVE

AVERAGE FILE

File	#N7:1
Dest	N7:0
Control	R6:0
Length	4
Position	0

Average
AVE

Status Bits:
EN - Enable
DN - Done bit
ER - Error Bit

Khi các điều kiện nhập là chuyển từ sai sang đúng, sẽ cộng N7:1; N7:2, N7:3, và N7:4; sau đó chia tổng cho 4 và lưu kết quả vào N7:0

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

Lệnh		Hoạt động
CLR CLR Dest D9:34 0000	Clear CLR	Khi các điều kiện nhập là đúng, sẽ xác tập tin thập phân 9, từ ngữ 34 (xác lập theo zero)

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	luôn luôn xác lập lại
Z	luôn luôn xác lập lại
S	luôn luôn xác lập lại

COS COSINE Source F8:13 0.7853982 Dest F8:14 0.7071068	Cosine COS	Khi các điều kiện nhập là đúng, sẽ tính cosine của giá trị trong F8:13 và lưu kết quả vào F8:14

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

Lệnh	Hoạt động

```
DIV
DIVIDE
Source A      N7:3
                 3
Source B      N7:4
                 1
Dest          N7:12
                 3
```

Division
DIV

Khi các điều kiện nhập là đúng, chia giá trị trong nguồn A (N7:3) cho giá trị trong nguồn B(N7:4), và lưu kết quả vào đích (N7:12).

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu chia cho zero hoặc quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

```
LN
NATURAL LOG
Source        N7:0
                 5
Dest          F8:20
           1.609438
```

Natural log
LN

Khi các điều kiện nhập là đúng, sẽ tính logarit tự nhiên của giá trị trong N7:0 và lưu kết quả vào F8:20

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại.
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại.

```
LOG
LOG BASE 10
Source        N7:2
                 5
Dest          F8:3
          0.6989700
```

Khi các điều kiện nhập là đúng, sẽ tính logarit cơ số 10 của giá trị trong N7:2 và lưu kết quả vào F8:3.

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; Nếu không, sẽ xác lập lại

Lệnh	Hoạt động

MUL
MULTIPLY
Source A N7:3 3
Source B N7:4 1
Dest N7:12 3

Multiply MUL

Khi các điều kiện nhập là đúng, sẽ nhân giá trị trong nguồn A (N7:3) với giá trị trong nguồn B (N7:4) và lưu kết quả vào đích (N7:12).

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

STD
STANDARD DEVIATION
File #N7:1
Dest N7:0
Control R6:0
Length 4
Position 0

Standard Deviation STD

Status Bits:
EN - Enable
DN - Done Bit
ER - Error Bit

Khi các điều kiện nhập chuyển từ sai sang đúng, các phần tử trong N7: 1; N7: 2, N7: 3 và N7:4 được dùng để tính độ lệch tiêu chuẩn của các giá trị đó và lưu kết quả vào đích (N7:0).

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

NEG
NEGATE
Source N7:3 3
Dest N7:12 -3

Negate NEG

Khi các điều kiện nhập là đúng, sẽ lấy dấu ngược với nguồn (N7: 3) và lưu kết quả vào đích (N7:12). Lệnh này chuyển các giá trị dương thành âm, và các giá trị âm thành dương.

Trạng thái bit	Hoạt động
C	xác lập, nếu có nhớ; nếu không, sẽ xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại

Lệnh		Hoạt động

SIN

```
SIN
SINE
Source      F8:11
            0.7853982
Dest        F8:12
            0.7071068
```

Sine
SIN

Khi điều kiện nhập là đúng, sẽ tính sine của giá trị trong F8:11 và lưu kết quả vào F8:12.

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

SQR

```
SQR
SQUARE ROOT
Source      N7:3
            25
Dest        N7:12
            5
```

Square Root
SQR

Khi điều kiện nhập là đúng, sẽ tính căn bậc hai của nguồn (N7:3) và lưu kết quả vào đích (N7:12).

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức xảy ra khi điểm trôi chuyển đổi theo số nguyên, nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

SRT

```
SRT
SORT
File        #N7:1
Control     R6:0
Length      4
Position    0
```

Sort
SRT

Khi điều kiện nhập chuyển từ sai sang đúng, các giá trị trong N7:1, N7::2, N7:3 và N7: 4 được phân loại theo thứ tự tăng.

Lệnh	Hoạt động

SUB
SUBTRACT
Source A N7:3 3
Source B N7:4 1
Dest N7:12 2

Subtract
SUB

Khi điều kiện nhập là đúng, giá trị trong nguồn A (N7:3) trừ giá trị trong nguồn B (N7:4) và lưu kết quả vào đích (N7:12).

Trạng thái bit	Hoạt động
C	xác lập, nếu có nhớ; nếu không, sẽ xác lập lại.
V	xác lập, nếu dưới giới hạn; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; Nếu không, sẽ xác lập lại

TAN
TANGENT
Source F8:15 0.7853982
Dest F8:16 1.000000

Tangent
TAN

Khi điều kiện nhập là đúng, sẽ tính tang của giá trị trong F8:15 và lưu kết quả vào F8:16

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

XPY
X TO POWER OF Y
Source A N7:4 5
Source B N7:5 2
Dest N7:6 25

X to the power of Y
XPY

Khi điều kiện nhập là đúng, sẽ lấy giá trị trong N7:4 nâng lên lũy thừa theo giá trị trong N7:5, và lưu kết quả vào N7:6.

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

Các lệnh logic

Lệnh	Hoạt động

AND

AND	
BITWISE AND	
Source A	D9:3 3F37
Source B	D9:4 00FF
Dest	D9:5 0037

AND

Khi điều kiện nhập là đúng, bộ xử lý tính phép toán AND (bit theo bit) giữa nguồn A (D9:3) và nguồn B (D9:4), lưu kết quả vào đích (D9:5). Bảng chân trị của phép toán AND là:

Nguồn A	Nguồn B	Kết quả
0	0	0
1	0	0
0	1	0
1	1	1

NOT

NOT	
NOT	
Source A	D9:3 00FF
Dest	D9:5 FF00

NOT Operation

Khi điều kiện nhập là đúng, bộ xử lý thực hiện phép toán NOT (bit theo bit) trên nguồn (D9:3) và lưu kết quả vào đích (D9: 5). Bảng chân trị của phép toán NOT là:

Nguồn	Đích
0	1
1	0

OR

OR	
BITWISE INCLUSIVE OR	
Source A	D9:3 3F37
Source B	D9:4 00FF
Dest	D9:5 3FFF

OR

Khi điều kiện nhập là đúng, bộ xử lý tính phép toán OR (bit theo bit) giữa nguồn A (D9:3) và nguồn B (D9:4), lưu kết quả vào đích (D9:5). Bảng chân trị của phép toán OR là:

Nguồn A	Nguồn B	Kết quả
0	0	0
1	0	1
0	1	1
1	1	1

XOR

XOR	
BITWISE EXCLUSIVE OR	
Source A	D9:3 3F37
Source B	D9:4 3F37
Dest	D9:5 0000

Exclusive OR XOR

Khi điều khiển nhập là đúng, bộ xử lý tính phép toán XOR (bit theo bit) giữa nguồn A (D9:3) và nguồn B (D9:4), lưu kết quả vào đích (D9:5). Bảng chân trị của phép toán XOR là:

Nguồn A	Nguồn B	Kết quả
0	0	0
1	0	1
0	1	1
1	1	0

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại
S	xác lập, nếu bit ý nghĩa cao nhất (bit 15 đối với thập phân, bit 17 đối với bát phân) xác lập là (1); nếu không, sẽ xác lập lại.

Các lệnh chuyển đổi

Lệnh		Hoạt động
FRD FROM BCD Source D9:3 0037 Dest N7:12 37	Convert from BCD FRD	Khi các điều kiện nhập là đúng, sẽ đổi giá trị trong nguồn (D9:3) thành giá trị nguyên và lưu kết quả vào đích (N7:12). Nguồn phải trong khoảng 0-9999 (BCD)

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	luôn luôn xác lập lại.
Z	xác lập, nếu giá trị đích là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

TOD TO BCD Source N7:3 44 Dest D9:5 0044	Convert to BCD TOD	Khi các điều kiện nhập là đúng, sẽ đổi giá trị trong nguồn (N7:3) sang định dạng BCD và lưu kết quả vào đích (D9:5)

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu giá trị nguồn là âm hoặc lớn hơn 9999 (nghĩa là bên ngoài khoảng 0-9999).
Z	xác lập, nếu giá trị đích là zero; nếu không, sẽ xác lập lại.
S	luôn luôn xác lập lại.

DEG RADIANS TO DEGREE Source F8:7 0.7853982 Dest F8:8 45	Convert to Degrees DEG	Đổi radian (giá trị trong nguồn A) sang độ và lưu kết quả vào đích (nguồn nhân với 180/π)

Trạng thái bit	Hoạt động
	luôn luôn xác lập lại
	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại.

Lệnh	Hoạt động

```
RAD
DEGREES TO RADIAN
Source      N7:9
            45
Dest        F8:10
            0.7853982
```

Convert to Radians
RAD

Đổi độ (giá trị trong nguồn A) sang radian và lưu kết quả vào đích (nguồn nhân với π/180).

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại.

Các lệnh dịch chuyển và chỉnh sửa bit

Lệnh	Hoạt động

```
MOV
MOVE
Source      N7:3
            0
Dest        N7:12
            0
```

Move
MOV

Khi các điều kiện nhập là đúng, sẽ dịch chuyển bản sao giá trị trong nguồn (N7: 3) đến đích (N7:12), viết đè lên giá trị gốc ở đích.

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	xác lập, nếu có quá mức trong khi chuyển đổi điểm trôi – số nguyên; nếu không, sẽ xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm nếu không, sẽ xác lập lại.

```
MVM
MASKED MOVE
Source      D9:3
            478F
Mask        D9:5
            00FF
Dest        D9:12
            008F
```

Masked Move
MVM

Khi các điều kiện nhập là đúng, bộ xử lý cho phép giá trị trong nguồn (D9:3) đi qua Mask (D9:5) và lưu kết quả vào đích (D9:12), viết đè lên giá trị gốc ở đích.

Trạng thái bit	Hoạt động
C	luôn luôn xác lập lại
V	luôn luôn xác lập lại.
Z	xác lập, nếu kết quả là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu kết quả là âm; nếu không, sẽ xác lập lại.

Lệnh		Hoạt động
BTD BIT FIELD DISTRIB Source N7:3 0 Source bit 3 Dest N7:4 0 Dest bit 10 Length 6	Bit Distribute BTD	Khi các điều kiện nhập là đúng, bộ xử lý sao chép số lượng bit được chuyên biệt theo Length, bắt đầu với bit nguồn (3) của nguồn (N7:3) và đặt các giá trị đó vào đích (N7:4), bắt đầu từ bit đích (10).

Các lệnh tập tin

Lệnh		Hoạt động
FAL FILE ARITH/LOGICAL Control R6:1 Length 8 Position 0 Mode ALL Dest #N15:10 Expression #N14:0 – 256	File Arithmetic and Logic FAL Status Bits: EN - Enable DN - Done Bit ER - Error Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý đọc 8 phần tử của N14:0 và trừ 256 (hằng) khỏi từng phần tử. Trong ví dụ này, kết quả được lưu trong 8 phần tử bắt đầu từ N15:10. Phần tử điều khiển R6:1 điều khiển thao tác. Mode (chế độ) xác định bộ xử lý thực hiện biểu thức với tất cả các phần tử trong các tập tin (ALL) trong một lần quét chương trình, một phần tử trong các tập tin (INC) một lần quét, hay số phần tử chuyên biệt (NUM) trong một lần quét. Lệnh FAL có thể thực hiện các phép toán: cộng (+), trừ (-), nhân (*), chia (I), chuyển đổi từ BCD (FRD), chuyển đổi sang BCD (TOD), căn bậc hai (SQR), phép logic AND, OR, NOT, XOR, âm (-), khử bỏ (0), dịch chuyển, và các lệnh toán khác.
FSC FILE SEARCH/COMPARE Control R9:0 Length 90 Position 0 Mode 10 Expression #B4:0 <> #B5:0	File Search and Compare FSC Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý thực hiện sự so sánh không bằng đối với 10 phần tử giữa các tập tin B4:0 và B5:0. Mode (chế độ) xác định bộ xử lý thực hiện biểu thức với tất cả các phần tử trong các tập tin (ALL) trong một lần quét chương trình, một phần tử trong các tập tin (INC) một lần quét, hay số lượng phần tử chuyên biệt (NUM) trong một lần quét. Phần tử điều khiển R9:0 điều khiển thao tác này. Khi các phần tử nguồn tương ứng là không bằng (phần tử B4:4 và B5:4 trong ví dụ này), bộ xử lý dừng tìm kiếm, xác lập các bit found.FD và inhibit.IN để chương trình bậc thang chuyển sang thao tác mới. Để tiếp tục sự so sánh tìm kiếm bạn phải xác lập lại .IN bit.

Lệnh		Hoạt động
COP COPY FILE Source #N7:0 Dest #N12:0 Length 5	File Copy COP	Khi các điều kiện nhập là đúng, bộ xử lý sao chép nội dung của tập tin nguồn (N7) vào tập tin đích (N12). Tập tin nguồn giữ nguyên, không thay đổi. Lệnh COP sao chép số lượng phần tử từ nguồn theo sự xác định từ Length.
FLL FILL FILE Source N10:6 Dest #N12:0 Length 5	File Fill FLL	Khi các điều kiện nhập là đúng, bộ xử lý sao chép giá trị trong nguồn (N10:6) cho các phần tử ở đích (N12). Lệnh FLL chỉ điền số lượng phần tử trong đích theo sự xác định từ Length.

Các lệnh chẩn đoán

Lệnh		Hoạt động
FBC FILE BIT COMPARE Source #I:031 Reference #B3:1 Result #N7:0 Cmp Control R6:4 Length 48 Position 0 Result Control R6:5 Length 10 Position 0	File Bite Compare FBC Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý so sánh số bit được chuyên biệt trong Cmp Control Length (48) của tập tin nguồn (# I:031) với số bit trong tập tin quy chiếu (# B3:1). Bộ xử lý lưu kết quả (các số bit không tương hợp) trong tập tin kết quả (# N7:0). File R6:4 điều khiển sự so sánh và file R6:5 điều khiển tập tin chứa kết quả. Tập tin chứa kết quả có thể chứa đến 10 sự không tương hợp (số được xác định trong Length) giữa các tập tin được so sánh.
DDT DIAGNOSTIC DETECT Source #I:030 Reference #B3:1 Result #N10:0 Cmp Control R6:0 Length 20 Position 0 Result Control R6:1 Length 5 Position 0	Diagnostic Detect DDT Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý so sánh số bit được chuyên biệt trong Cmp Control Length (20) của tập tin nguồn (# I:030) với số bit trong tập tin quy chiếu (# B3:1). Bộ xử lý lưu kết quả (số bit không tương hợp) trong tập tin kết quả (#N10:0). Phần tử điều khiển R6:0 điều khiển sự so sánh và phần tử điều khiển R6:1 điều khiển tập tin chứa kết quả (#N10:0). Tập tin chứa kết quả có thể chứa đến 5 sự không tương hợp (số được xác định trong Length) giữa các tập tin được so sánh. Bộ xử lý sao chép các bit nguồn vào tập tin quy chiếu cho lần so sánh kế tiếp. Sự khác biệt giữa lệnh DDT và lệnh FBC là mỗi khi lệnh DDT tìm được sự không tương hợp, bộ xử lý thay đổi bit quy chiếu để tương hợp với bit nguồn. Bạn có thể dùng lệnh DDT để cập nhật tập tin quy chiếu, phản ánh sự thay đổi các điều kiện máy hoặc vận hành.

Lệnh		Hoạt động
DTR DATA TRANSITION Source I:002 Mask 0FFF Reference N63:11	Data Transition DTR	Lệnh DTR so sánh các bit nguồn (I:002) thông qua Mask (0FFF) với các bit trong quy chiếu (N63:11). Khi nguồn mask khác với quy chiếu, lệnh là đúng chỉ một lần quét. Các bit nguồn được viết vào địa chỉ quy chiếu cho lần so sánh kế tiếp. Khi nguồn và quy chiếu là như nhau, lệnh duy trì trạng thái sai.

Các lệnh bộ ghi dịch chuyển

Lệnh		Hoạt động
BSL BIT SHIFT LEFT File #B3:1 Control R6:53 Bit Address I:022/12 Length 5	Bit Shift Left BSL Status Bits: EN - Enable DN - Done Bit ER - Error Bit UL - Unload Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, lệnh BSL dịch chuyển số bit đã chuyên biệt theo Length (5) trong tập tin (B3), bắt đầu từ bit 16 (B3:1/0 = B3/16) sang trái theo một vị trí bit. Bit nguồn (I:022/12) dịch chuyển vào vị trí bit thứ nhất, B3:1/0 (B3/16). Bit thứ năm, B3:1/4 (B3/20), dịch chuyển vào UL bit của cấu trúc điều khiển (R6:53).
BSR BIT SHIFT RIGHT File #B3:2 Control R6:54 Bit Address I:023/06 Length 3	Bit Shift Right BSR Status Bits: EN - Enable DN - Done Bit ER - Error Bit UL - Unload Bit	Nếu các điều kiện nhập chuyển từ sai sang đúng, lệnh BSR dịch chuyển số bit chuyên biệt theo Length (3) trong tập tin (B3), bắt đầu với B3:2/0 (=B3/32), sang phải theo một vị trí bit. Bit nguồn (I:023/06) chuyển vào vị trí bit thứ ba B3/34. Bit thứ nhất (B3/32) chuyển vào UL bit của phần tử điều khiển (R6:54).
FFL FIFO LOAD Source N60:1 FIFO #N60:3 Control R6:51 Length 64 Position 0	FIFO Load FFL Status Bits: EN - Enable Load DN - Done Bit EM - Empty Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý tải N60:1 vào phần tử khả dụng kế tiếp trong tập tin FIFO, #N60:3, do R6:51 chỉ đến. Mỗi khi bậc thang chuyển từ sai sang đúng, bộ xử lý tải một phần tử. Khi tập tin FIFO đầy (64 từ ngữ được tải), DN bit được xác lập.
FFU FIFO UNLOAD FIFO #N60:3 Dest N60:2 Control R6:51 Length 64 Position 0	FIFO Unload FFU Status Bits: EU - Enable Unload DN - Done Bit EM - Empty Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý dỡ tải một phần tử từ N60:3 và đưa vào N60:2. Mỗi khi bậc thang chuyển từ sai sang đúng, bộ xử lý dỡ tải một giá trị khác. Mọi dữ liệu trong tập tin #60:3 được dịch chuyển một vị trí hướng đến N60:3. Khi tập tin này rỗng, EM bit được xác lập.

Lệnh		Hoạt động
LFL LIFO LOAD Source N70:1 LIFO #N70:3 Control R6:61 Length 64 Position 0	LIFO Load LFL Status Bits: EN - Enable Load DN - Done Bit EM - Empty Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý tải N70:1 vào phần tử khả dụng kế tiếp trong tập tin LIFO #N70:3, do R6:61 chỉ đến. Mỗi khi bậc thang chuyển từ sai sang đúng, bộ xử lý tải một phần tử. Khi tập tin LIFO đầy (64 từ ngữ được tải), DN bit được xác lập.
LFU LIFO UNLOAD LIFO #N70:3 Dest N70:2 Control R6:61 Length 64 Position 0	LIFO Unload LFU Status Bits: EN - Enable Load EU - Enable Unload DN - Done Bit EM - Empty Bit	Khi các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý dỡ tải phần tử cuối cùng từ #N70:3 và đưa vào N70:2. Mỗi khi bậc thang chuyển từ sai sang đúng, bộ xử lý dỡ tải một phần tử. Khi tập tin LIFO rỗng, EM bit được xác lập.

Các lệnh chuỗi thứ tự

Lệnh		Hoạt động
SQI SEQUENCER INPUT File #N7:11 Mask FFF0 Source I:031 Control R6:21 Length 4 Position 0	Sequencer Input SQI	Lệnh SQI so sánh dữ liệu hình ảnh nhập từ nguồn (#I:031) vào Mask (FFF0) với dữ liệu quy chiếu (# N7:11) để xem hai tập tin này có bằng nhau không. Thao tác được điều khiển bằng thông tin trong tập tin điều khiển R6:21 chỉ định tương hợp với các bit quy chiếu tương ứng, lệnh bậc thang chuyển sang trạng thái đúng.
SQL SEQUENCER LOAD File #N7:20 Source I:002 Control R6:22 Length 5 Position 0	Sequencer Load SQL Status Bits: EN - Enable DN - Done Bit ER - Error Bit	Lệnh SQL tải dữ liệu vào tập tin chuỗi thứ tự (#N7:20) từ nguồn (I:002) bằng cách lần lượt qua số lượng các phần tử do Length (5) của nguồn (I:002) xác định, bắt đầu từ vị trí (0). Thao tác được điều khiển bằng thông tin trong tập tin điều khiển R6:22. Khi bậc thang chuyển từ sai sang đúng, lệnh SQL chuyển sang bước kế tiếp trong tập tin chuỗi thứ tự và tải dữ liệu vào theo từng lần quét nếu bậc thang vẫn duy trì trạng thái đúng.
SQO SEQUENCER OUTPUT File #N7:1 Mask 0F0F Dest O:014 Control R6:20 Length 4 Position 0	Sequencer Output SQO Status Bits: EN - Enable DN - Done Bit ER - Error Bit	Khi bậc thang chuyển từ sai sang đúng, lệnh SQO chuyển qua bước kế tiếp trong tập tin chuỗi thứ tự (# N7:1). Dữ liệu trong tập tin này được truyền qua Mask (0F0F) đến đích (O:014) mỗi lần quét nếu bậc thang vẫn duy trì trạng thái đúng.

Các lệnh điều khiển chương trình

Lệnh		Hoạt động
——(MCR)——	Master Control Reset MCR	Nếu điều kiện nhập là đúng, chương trình quét các bậc thang giữa các bậc thang lệnh MCR và xử lý các ngõ ra một cách bình thường. Nếu điều kiện nhập là sai, mọi ngõ ra không ngăn chặn giữa bậc thang lệnh MCR đều được xác lập lại.
10 ——(JMP)——	Jump JMP	Nếu điều kiện nhập là đúng, bộ xử lý bỏ qua các bậc thang bằng cách chuyển đến bậc thang do nhãn (10) chỉ định.
10 ——[LBL]——	Label LBL	Khi bộ xử lý đọc lệnh JMP tương ứng nhãn 10, bộ xử lý nhảy (chuyển) đến bậc thang chứa nhãn đó và bắt đầu xử lý. **Quan trọng:** Phải là lệnh thứ nhất trên bậc thang.
FOR FOR Label Number 0 Index N7:0 Initial Value 0 Terminal Value 10 Step Size 1	FOR Loop FOR	Bộ xử lý thực thi các bậc thang giữa lệnh FOR và NXT lặp lại trong một lần quét chương trình, cho đến khi đạt được giá trị cuối (10) hoặc lệnh BRK chấm dứt sớm thao tác. Kích cỡ bước là khoảng số gia của vòng lặp.
NXT NEXT Label Number 0	Next NXT	Lệnh NXT đưa bộ xử lý trở về lệnh FOR tương ứng, xác định bằng số nhãn ghi trong lệnh FOR. Lệnh NXT phải được lập trình trên bậc thang không có điều kiện, là bậc thang cuối được lặp lại trong vòng FOR – NEXT.
——[BRK]——	Break BRK	Khi các điều kiện nhập là đúng, lệnh BRK kết thúc sớm vòng lặp FOR – NEXT.
JSR JUMP TO SUBROUTINE Program File 90 Input par N16:23 Input par N16:24 Input par 231 Return par N19:11 Return par N19:12	Jump to Subroutine JSR	Nếu các điều kiện nhập là đúng, bộ xử lý bắt đầu chạy chương trình con Program File (90). Bộ xử lý đi qua các thông số nhập (N16:23, N16:24, 231) đến chương trình con và đi qua các thông số trả về (N19:11, N19:12) để quay lại chương trình chính, nơi bộ xử lý gặp lệnh JSR.
SBR SUBROUTINE Input par N43:0 Input par N43:1 Input par N43:2	Subroutine SBR	Lệnh SBR là lệnh đầu tiên trong tập tin chương trình con. Lệnh này xác định các thông số nhập (N43:0, N43:1, N43:2) bộ xử lý nhận từ lệnh JSR. Bạn không cần dùng lệnh SBR nếu không đi qua các thông số nhập để đến chương trình con.
RET RETURN () Return par N43:3 Return par N43:4	Return RET	Lệnh RET kết thúc chương trình con và lưu các thông số trả về (N43:3, N43:4) để quay lại lệnh JSR trong chương trình chính.

Lệnh		Hoạt động
—[AFI]—	Always False AFI	Lệnh AFI khử hoạt động của bậc thang (bậc thang luôn luôn sai).
—(TND)—	Temporary End TND	Nếu các điều kiện nhập là đúng, lệnh TND dừng bộ xử lý, ngăn chặn bộ này quét phần chương trình còn lại (tạm thời chấm dứt chương trình).
B3 —[ONS]— 110	One Shot ONS	Nếu các điều kiện nhập trước lệnh ONS trên cùng một bậc thang chuyển từ sai sang đúng, lệnh ONS tạo điều kiện cho bậc thang sao cho ngõ ra là đúng trong một lần quét. Bậc thang là sai trong các lần quét kế tiếp.
OSF ONE SHOT FALLING Storage Bit B3/0 Output Bit 15 Output Word N7:0	One Shot Falling OSF Status Bits: OB - Output Bit ① SB - Storage Bit ①	Lệnh OSF cho phép sự kiện xảy ra một lần. Bạn dùng lệnh OSF mỗi khi sự kiện khởi động dựa trên sự thay đổi trạng thái của bậc thang từ đúng qua sai, không dựa vào kết quả trạng thái của bậc thang. Bit xuất (N7:0/15) được xác lập là (1) cho một lần quét chương trình khi bậc thang chuyển từ đúng qua sai.
OSR ONE SHOT RISING Storage Bit B3/0 Output Bit 15 Output Word N7:0	One Shot Rising OSR Status Bits: OB - Output Bit ① SB - Storage Bit ①	Lệnh OSR cho phép sự kiện xảy ra một lần. Bạn dùng lệnh OSR mỗi khi sự kiện khởi động dựa trên sự thay đổi trạng thái của bậc thang từ sai sang đúng, không dựa vào kết quả trạng thái của bậc thang. Bit xuất (N7:0/15) được xác lập là (1) cho một lần quét chương trình khi bậc thang chuyển từ sai sang đúng.
SFR SFC Reset Prog File Number 3 Restart Step At	SFC Reset SFR	Lệnh SFR xác lập lại logic trong lưu đồ chức năng chuỗi thứ tự. Khi lệnh SFR là đúng, bộ xử lý thực hiện quét cuối/quét sau trên mọi bước hoạt động và thao tác trong tập tin được chọn, sau đó xác lập lại logic trong SFC trên lần quét chương trình kế tiếp. Lưu đồ duy trì trạng thái xác lập lại này cho đến khi lệnh SFR chuyển sang trạng thái sai.
—(EOT)—	End of Transition EOT	Lệnh EOT là lệnh cuối cùng trong tập tin chuyển tiếp. Nếu bạn không dùng lệnh EOT, bộ xử lý luôn luôn coi sự chuyển tiếp là đúng.
—(UID)—	User Interrupt Disable UID	Lệnh UID tạm thời khử hoạt động chương trình bậc thang truyền động bằng sự ngắt (chẳng hạn STI, PII) từ sự ngắt chương trình đang thực thi.
—(UIE)—	User Interrupt Enable UIE	Lệnh UIE tái cho phép chương trình bậc thang truyền động bằng sự ngắt có thể dừng chương trình bậc thang đang thực thi.

① Các bit này chỉ nhằm mục đích minh họa, không phải là địa chỉ logic của chúng

Các lệnh thông báo và điều khiển quá trình

Lệnh		Hoạt động
PID PID Control Block PD10:0 Proc Variable N15:13 Tieback N15:14 Control Output N20:21	Proportional, Integral, and Derivative PID Status Bits: EN - Enable DN - Done Bit (for N control blocks only)	Khối điều khiển (PD10:0) chứa thông tin lệnh cho PID. Lệnh PID nhận biến quá trình từ N15:13 và gởi tín hiệu ra của PID đến N20:21. Sự hồi tiếp được lưu trong N15:14 điều khiển trạm điều khiển bằng tay. Bạn còn có thể sử dụng các khối điều khiển N. Nếu bạn dùng khối điều khiển PD, ở đây sẽ không có done bit. Ngoài ra, các điều kiện nhập bậc thang cần phải đúng.
MSG SEND/RECEIVE MESSAGE Control Block MG7:10	Message MSG	Nếu các điều kiện nhập chuyển từ sai sang đúng, dữ liệu được truyền tương ứng các thông số lệnh bạn đã xác lập khi nhập lệnh thông báo. Khối điều khiển (MG7:10) chứa trạng thái và các thông số lệnh. Bạn có thể sử dụng các khối điều khiển N. Đối với MSG liên tục, cần lập điều kiện cho bậc thang là đúng chỉ một lần quét.

Bit #	Status Bits
15	EN - Enable
14	ST - Start Bit
13	DN - Done Bit
12	ER - Error Bit
11	CO - Continuous
10	EW - Enabled-Waiting
9	NR - No Response
8	TO - Time Out Bit

Các lệnh truyền khối

Khối điều khiển số nguyên (N)

Từ ngữ	Mô tả
0	các bit trạng thái
1	số đếm từ ngữ yêu cầu
2	số đếm từ ngữ được truyền
3	số tập tin
4	số phần tử

Khối điều khiển truyền khối (BT)

Từ ngữ nhắc	Mô tả
.EN thru .RW	các bit trạng thái
.RLEN	chiều dài yêu cầu
.DLEN	chiều dài từ ngữ được truyền/mã lỗi
.FILE	số tập tin
.ELEM	số phần tử
.RGS	nhánh/nhóm/rãnh

Word 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EN	ST	DN	ER	CO	EW	NR	TO	RW	**	rack	**	**	group	**	slot

Lệnh	Hoạt động
BTR BLOCK TRANSFER READ Rack 1 Group 0 Module 0 Control Block N10:100 Data File N10:110 Length 40 Continuous Y Block Transfer Read BTR	Nếu các điều kiện nhập chuyển từ sai sang đúng, truyền khối đọc được khởi động cho I/O module trên nhánh 1, nhóm O, module O. Khối điều khiển (BT10:100–6-word file) chứa trạng thái truyền. Tập tin dữ liệu (N10:110) là nơi lưu trữ dữ liệu đọc từ module. Chiều dài BT (40) xác định số từ ngữ được truyền. Truyền khối không liên tục được xếp theo hàng và chỉ chạy một lần trên sự chuyển tiếp bậc thang sai sang đúng; truyền khối liên tục được sắp xếp theo hàng lặp lại nhiều lần. Bạn có thể sử dụng kiểu dữ liệu N (số nguyên) cho khối điều khiển.

Các bộ xử lý PLC-5/30, -5/40 -5/40E, -5/40L -5/60, -5/60L, -5/80, -5/80E		Các bộ xử lý PLC-5/40, -5/40L, -5/60, -5/60L, -5/80, -5/40E, -5/80E		Các bộ xử lý PLC-5/60, -5/60L, -5/80, -5/80E	
S:7 bit #	BT queue full for rack	S:32 bit #	BT queue full for rack	S:34 bit #	BT queue full for rack
08 ①	0	08	10	08	20
09 ①	1	09	11	09	21
10 ①	2	10	12	10	22
11 ①	3	11	13	11	23
12	4	12	14	12	24
13	5	13	15	13	25
14	6	14	16	14	26
15	7	15	17	15	27

① PLC-5/11, -5/20, và 5/20E cũng là các bộ xử lý

Lệnh		Hoạt động
BTW BLOCK TRNSFR WRITE Rack 1 Group 0 Module 0 Control Block BT10:0 Data File N10:10 Length 40 Continuous Y	Block Transfer Write BTW	Nếu các điều kiện nhập chuyển từ sai sang đúng, khối chuyển tiếp ghi được kích hoạt cho I/O module ở nhánh 1, nhóm 0, module 0. Khối điều khiển (BT10:0 – 6 word file) chứa trạng thái truyền. Tập tin dữ liệu chứa dữ liệu để ghi vào module này (N10:10). Chiều dài BT (40) xác định số từ ngữ được truyền. Sự truyền khối không liên tục được xếp theo hàng và chỉ chạy một lần trên chuyển tiếp bậc thang từ sai sang đúng; sự truyền khối liên tục được sắp xếp theo hàng lặp lại nhiều lần. Bạn còn có thể dùng kiểu dữ liệu N (số nguyên) cho khối điều khiển.

Các lệnh ASCII

Các bit trạng thái.

EN - Enable
DN - Done Bit
ER - Error Bit
EM - Empty Bit
EU - Queue
FD - Found Bit

Lệnh		Hoạt động
ABL ASCII TEST FOR LINE Channel 0 Control R6:32 Characters	ASCII Test for Line ABL	Nếu các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý báo cáo số ký tự trong bộ đệm, theo chiều thuận kể cả các ký tự kết thúc dòng và đặt giá trị này vào từ ngữ vị trí của cấu trúc điều khiển (R6:32.POS). Bộ xử lý còn hiển thị giá trị này trong ô ký tự trên màn hiển thị.
ACB ASCII CHARS IN BUFFER Channel 0 Control R6:32 Characters	ASCII Characters in Buffer ACB	Nếu các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý báo cáo tổng số ký tự trong bộ đệm và đặt giá trị này vào từ ngữ vị trí (.POS) của cấu trúc điều khiển. Bộ xử lý còn hiển thị giá trị này trong ô ký tự trên màn hiển thị.
ACI STRING TO INTEGER CONVERSION Source ST38:90 Dest N7:123 75	Convert ASCII String to Integer ACI	Nếu các điều kiện nhập là đúng, bộ xử lý chuyển chuỗi trong ST38:90 thành số nguyên và lưu kết quả trong N7:123

Trạng thái bít	Hoạt động
C	xác lập, nếu nguồn là âm; nếu không, sẽ xác lập lại
V	xác lập, nếu nguồn > = 32,768 hoặc < = -32,708; nếu không, sẽ xác lập lại.
Z	xác lập, nếu nguồn là zero; nếu không, sẽ xác lập lại.
S	xác lập, nếu đích là âm; nếu không, sẽ xác lập lại

Lệnh		Hoạt động
ACN STRING CONCATENATE Source A ST38:90 Source B ST37:91 Dest ST52:76	ASCII String Concatenate ACN	Nếu các điều kiện nhập là đúng, bộ xử lý nối chuỗi trong ST38:90 với chuỗi trong ST37:91 và lưu kết quả vào ST52:76
AEX STRING EXTRACT Source ST38:40 Index 42 Number 10 Dest ST52:75	ASCII String Extract AEX	Nếu các điều kiện nhập là đúng, bộ xử lý trích 10 ký tự bắt đầu từ ký tự thứ 42 của ST38:40 và lưu kết quả vào ST52:75
AIC INTEGER TO STRING CONVERSION Source 876 Dest ST38:42	Convert Integer to ASCII String AIC	Nếu các điều kiện nhập là đúng, bộ xử lý chuyển giá trị 876 thành chuỗi và lưu kết quả vào ST38:42.
AHL ASCII HANDSHAKE LINE Channel 0 AND Mask 0001 OR Mask 0003 Control R6:23 Channel Status	ASCII Handshake Lines AHL	Nếu các điều kiện nhập chuyển từ sai sang đúng, bộ xử lý sử dụng AND Mask và OR Mask để xác định sẽ xác lập hay xác lập lại các dòng DTR (bit 0) và RTS (bit 1) hay để nguyên (không thay đổi). Các bit 0 và 1 của AND Mask làm cho (các) dòng này được xác lập nếu là 1 và để (các) dòng không thay đổi nếu là 0. Bit 0 và 1 của OR Mask làm cho (các) dòng được xác lập nếu là 1 và không thay đổi nếu là 0.
ARD ASCII READ Channel 0 Dest ST52:76 Control R6:32 String Length 50 Characters Read	ASCII Read ARD Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	Nếu các điều kiện nhập chuyển từ sai sang đúng, sẽ đọc 50 ký tự từ bộ đệm và chuyển chúng đến ST52:76. Số lượng ký tự đọc được lưu trong R6:32.POS và hiển thị trong khung Characters Read của phần hiển thị lệnh.
ARL ASCII READ LINE Channel 0 Dest ST50:72 Control R6:30 String Length 18 Characters Read	ASCII Read Line ARL Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	Nếu các điều kiện nhập chuyển từ sai sang đúng, sẽ đọc 18 ký tự (hoặc đến cuối dòng) từ bộ đệm và chuyển chúng đến ST50:72. Số lượng ký tự đọc được lưu vào R6:30.POS và hiển thị trong khung Characters Read của phần hiển thị lệnh.
ASC STRING SEARCH Source ST38:40 Index 35 Search ST52:80 Result 42	ASCII String Search ASC	Nếu các điều kiện nhập là đúng, sẽ tìm kiếm ST52:80 bắt đầu tự ký tự thứ 35 đối với chuỗi được phát hiện trong ST38:40. Trong ví dụ này, chuỗi được phát hiện với chỉ số 42. Nếu không tìm được chuỗi, bit lỗi nhỏ của lệnh ASCII S:17/8 được xác lập và kết quả là zero.

Lệnh		Hoạt động
ASR ASCII STRING COMPARE Source A ST37:42 Source B ST38:90	ASCII String Compare ASR	Nếu chuỗi trong ST37:42 là đồng nhất với chuỗi trong ST38:90, lệnh này là đúng. Chú ý, đây là lệnh nhập. Chiều dài chuỗi không hiệu lực sẽ làm cho bit lỗi nhỏ S:17/8 của lệnh ASCII được xác lập và lệnh này là sai.
AWA ASCII WRITE / Channel Source Control String Length Characters Se	ASCII Write Append AWA Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	Nếu các điều kiện nhập chuyển từ sai sang đúng, sẽ đọc 50 ký tự từ ST52:76, ghi chúng vào kênh 0 và nối hai cấu hình ký tự trong cấu hình kênh (mặc định CR/LF). Số ký tự gởi được lưu trong R6:32.POS và hiển thị trong khung Characters Sent của phần hiển thị lệnh.
AWT ASCII WRITE Channel 0 Source ST37:40 Control R6:23 String Length 40 Characters Sent	ASCII Write AWT Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	Nếu các điều kiện nhập chuyển từ sai sang đúng, sẽ ghi 40 ký tự từ ST37:40 vào kênh 0. Số ký tự gởi được lưu trong R6:23.POS và hiển thị trong khung Characters Sent của phần hiển thị lệnh.

PHỤ LỤC B. CÁC LINH KIỆN VÀ KÝ HIỆU

Bảng 1. Các linh kiện khối tiếp điểm

Ký hiệu	Mạch	Giải thích ❶	Tiêu chuẩn			Mức logic		
			Số catalog			Số catalog		
			Đầu dây ép	Đầu dây cắm ❷		Đầu dây ép	Đầu dây cắm ❷	
	1 NC	T51 và T51E xếp chồng đến 6 khối (6 mạch), trừ khi có quy định khác.	10250 T51	10250 T59		10250 T51E	10250 T59E	
	1 NO	T53 và T53E xếp chồng đến 6 khối (6 mạch), trừ khi có quy định khác.	T53	T60		T53E	T60E	
	NO - NC	T1 và T1E xếp chồng đến 6 khối (12 mạch), trừ khi có quy định khác.	T1	T40		T1E	T40E	
	2 NC	T3 và T3E xếp chồng đến 6 khối (12 mạch), trừ khi có quy định khác.	T3	T42		T3E	T42E	
	2 NO	T2 và T2E xếp chồng đến 6 khối (12 mạch), trừ khi có quy định khác.	T2	T41		T2E	T41E	
		CÁC KHỐI TÍNH NĂNG ĐẶC BIỆT						
	LONC	NC mở trễ. Xếp chồng đến 6 khối (6 mạch), trừ khi có quy định khác	T71	—		T71E	---	
	ECNO-NC	NO đóng sớm và NC tiêu chuẩn. Xếp chồng đến 6 khối, trừ khi có qui định khác.	T47	---		T47E	---	
	ECNO-NO	Một NO đóng sớm và một NO tiêu chuẩn. Không dùng cho công tắc chọn 4-vị trí, các bộ nối đệm hoặc nối cần gạt 2 vị trí hoặc kiểu đẩy – kéo. Xếp chồng đến 4 khối, trừ khi có quy định khác.	T57	---		T57E	—	
	2LONC	Hai tiếp điểm NC mở trễ. T45 xếp chồng đến 6 khối, trừ khi có quy định khác.	T45	---		T45E	---	
	LONC-ECNO	Các tiếp điểm xếp chồng. Không dùng cho công tắc chọn 4–vị trí, các bộ nối đệm hoặc cần gạt 2-vị trí hoặc kiểu đẩy – kéo. Xếp chồng đến 4 khối, trừ khi có quy định khác.	T55	—		T55E	---	
	NC (ASP)	Có thêm lò xo để chống va đập và rung động. T50 xếp chồng đến 6 khối, trừ khi có các quy định khác.	T50	T61		---	---	
	2 NC (ASP)	Có thêm lò xo để chống va đập và rung động. T49 xếp chồng đến 6 khối, trừ khi có các quy định khác.	T49	T91		---	---	
	NO-NC (ASP)	Có thêm lò xo để chống va đập và rung động. T48 xếp chồng đến 6 khối, trừ khi có các quy định khác.	T48	T62		---	---	
	NO-NC (ST)	Khối chuyển tiếp ngắn. Chỉ cần hành trình ngắn để kích hoạt các tiếp điểm. Xếp chồng đến 4 khối trừ khi có quy định khác.	T92	---		---	---	
	2NO-2NC	Bốn mạch xếp trong một khối, định mức cực đại 300V. Không dùng cho công tắc chọn 4–vị trí hoặc nhấn – xoay. Xếp chồng đến 4 khối, trừ khi có quy định khác.	T44	---		---	---	

❶ Đối với các bộ cần gạt, công tắc chọn 4 – vị trí, bộ nhấn – xoay với mã cam 9 và 15, chỉ được phép dùng tối đa 4 khối tiếp điểm. Các bộ đẩy – kéo chỉ dùng tối đa 2 khối tiếp điểm.

❷ Giới hạn một cho mỗi bộ vận hành. Khoảng cách tâm – tâm tối thiểu 2.5 in. Không lắp vào các hộp T-Line tiêu chuẩn.

Các linh kiện bộ nhấn – xoay

MÔ TẢ:

Bộ điều khiển nhấn – xoay kết hợp chức năng của nút nhấn và công tắc chọn. Các tiếp điểm được vận hành bằng tác động phối hợp xoay vành ngoài và nhấn nút.

Tương tự hầu hết các linh kiện T-line, loại này có hai dạng, thứ nhất là dạng linh kiện rời để lắp tại hiện trường, thứ hai là bộ lắp sẵn.

Khi lựa chọn cam và khối tiếp điểm, cần xem xét chức năng với vành ngoài xoay đến vị trí cho trước không ảnh hưởng đến nút nhấn (ký hiệu là N), và đến vị trí đó với nút được nhấn (ký hiệu D). Điều này được thực hiện cho từng vị trí xoay của vành ngoài.

Bộ vận hành và Cam
10250T24111

BỘ VẬN HÀNH VÀ CAM

Màu và kiểu nút	Catalog và số mã		Mã cam chọn từ Bảng 2
	Dọc	Ngang	
	10250	10250	
Đen phẳng	T241	T251	
Đỏ phẳng ❶	T242	T252	
Xanh phẳng	T243	T253	1 đến 18
Đen dài	T261	T271	
Đỏ dài ❶	T262	T272	
Xanh dài	T263	T273	

BẢNG 2. CHỌN CAM VÀ KHỐI TIẾP ĐIỂM CHO CÔNG TẮC 2 - VỊ TRÍ

Kết hợp	Vị trí vành xoay ↖		Vị trí vành xoay ↗		Mã cam 1	Mã cam 2	Mã cam 3	Mã cam 4	Mã cam 5	Mã cam 6	Mã cam 10	Mã cam 11	Mã cam 12	Mã cam 13	Mã cam 14
	Thứ tự mạch ❷														
	N	D	N	D											
1	0	0	0	X	A B NO	A B NO	---	---	A B NO	---	---	A B NO NO	---	---	---
2	0	0	X	0	---	---	---	A B NC NO	A B NC NO	---	---	A B NC NO	A B NC	A or B NO	A B NC
3	0	0	X	X	---	---	---	---	B A NO	A B NO	B A NO	B A NC	---	---	---
4	0	X	0	0	B A NO	B A NC NO	---	---	---	A B NC NO	A B NO	A B NO NC	---	---	B A NO
5	0	X	0	X	A B NO NO	B A NO	---	A B NO	---	---	---	A B NO	B A NO	---	---
6	0	X	X	0	---	---	---	---	---	---	---	---	---	---	A B NC NO
7	0	X	X	X	---	---	A or B NO	B A NO	---	B A NO	A B NO NO	A B NO NO	A B NC NO	---	---
8	X	0	0	0	---	---	A or B NC	B A NC	---	B A NC	A B NC NC	A B NC NO	A B NC NO	---	---
9	X	0	0	X	---	---	---	---	---	---	---	---	---	---	A B NO NC
10	X	0	X	0	A B NC NC	B A NC	---	A B NC	---	---	---	A B NC	B A NC	---	---
11	X	0	X	X	B A NC	A B NO NC	---	---	---	A B NO NC	A B NC	A B NO NO	---	---	B A NC
12	X	X	0	0	---	---	---	---	B A NC	A B NC	B A NC	A B NO NC	---	---	---
13	X	X	0	X	---	---	---	A B NO NC	A B NO NC	---	---	A B NO NC	A B NO	A or B NO	A B NO
14	X	X	X	0	A B NC	A B NC	---	---	A B NC	---	---	A B NO NC	---	---	---

❶ Không dùng cho dừng (Stop) khẩn cấp

❷ N = Nút ở vị trí bình thường; D= Nhấn nút.

Bảng 3. Chọn cam và khối tiếp điểm cho công tắc 3 - vị trí

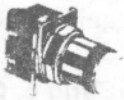
Kết hợp	Vị trí vành xoay ↖	↑	↗	Mã cam 7	Mã cam 8 ❶	Mã cam 9	Mã cam 15	Mã cam 16	Mã cam 17	Mã cam 18
	Thứ tự mạch ❷									
	N D	N D	N D							
1	0 0	0 0	0 X	A B NO NO	A B NC NO	---	B NO ❶	B NO	---	A B NC NO
2	0 0	0 0	X X	---	---	B NO	---	---	A NO	---
3	0 0	0 X	0 0	---	---	A B NO NC ❶	---	---	---	A B NO NO
4	0 0	0 X	0 X	---	---	---	---	---	---	B NO
5	0 0	0 X	X X	---	---	A NO ❶	---	---	---	---
6	0 .0	X X	0 0	---	A B NO NO	---	---	---	---	---
7	0 0	X X	0 X	---	B NO	---	---	---	---	---
8	0 0	X X	X 0	A B NC NO	---	---	---	---	---	---
9	0 0	X X	X X	B NO	---	---	---	---	---	---
10	0 X	0 0	0 0	A B NO NC	A B NO NC	---	A NO ❶	A NO	B NO	A B NO NC
11	0 X	0 0	0 X	A NO	---	---	A B NO NO ❶	A B NO NO	---	---
12	0 X	0 0	X X	---	---	---	---	---	A B NO NO	---
13	0 X	0 X	0 0	---	---	---	---	---	---	A NO
14	0 X	0 X	0 X	---	---	---	---	---	---	A B NO NO
15	0 X	X X	0 0	---	---	---	---	---	---	---
16	0 X	X X	0 X	---	A NO	---	---	---	---	---
					A B NO NO					
17	0 X	X X	X X	A B NO NO	---	---	---	---	---	---
18	X 0	0 0	0 0	A B NC NC	---	---	---	---	---	---
19	X 0	0 0	X X	---	A NC	---	---	---	---	---
20	X 0	0 0	X 0	---	A B NC NC	---	---	---	---	---
21	X 0	X X	0 0	---	---	---	---	---	A B NC NC	---
22	X 0	X X	X X	A B NC NO	A B NC NO	---	---	A NC	B NC	A B NC NO
23	X 0	X X	X 0	A NC	---	---	---	A B NC NC	---	---
24	X 0	X 0	X 0	---	---	---	A B NC NC	---	---	A B NC NC
25	X 0	X 0	X X	---	---	---	A NC ❶	---	---	---
26	X X	0 0	0 0	B NC	---	A NC ❶	---	---	---	A NC
27	X X	0 0	0 X	A B NO NC	---	---	---	---	---	---
28	X X	0 0	X 0	---	B NC	---	---	---	---	---
29	X X	0 0	X X	---	A B NC NO	A B NC NO ❶	---	---	---	---
30	X X	X X	0 0	---	---	B NC	---	---	A NC	---
31	X X	X X	X 0	A B NC NC	A B NO NC	---	---	B NC	---	A B NO NC
32	X X	X 0	X 0	---	---	---	B NC ❶	---	---	B NC
33	X X	X 0	X X	---	---	---	A B NO NC ❶	---	---	A B NC NC

A B NO NC
Mắc nối tiếp

A B NO NC
Mắc song song

Các nối kết này không được thực hiện ở nhà máy. Chúng được giải thích trong bảng lựa chọn theo yêu cầu, nhưng phải được thực hiện ở nơi sử dụng

Bộ vận hành ROTO-PUSH với nút phẳng



Bộ vận hành ROTO-PUSH với nút dài

❶ Giới hạn đến 4 khối tiếp điểm

❷ N = Nút nhấn ở vị trí bình thường
D = Nhấn nút.

Nhập cặp bước/ chuyển tiếp

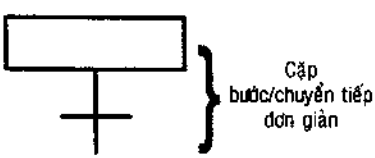
Bước và chuyển tiếp được nhập theo cặp. Bộ biên tập SFC chỉ cho phép bạn nhập các cặp bước và chuyển tiếp đúng. Bạn hãy xem ví dụ dưới đây về các cặp bước/ chuyển tiếp đúng.

Quan trọng: Mọi tập tin chuyển tiếp đều phải có lệnh điều kiện EOT (kết thúc chuyển tiếp). Nếu không có lệnh này, bộ xử lý tự động coi tập tin chuyển tiếp là đúng.

Các ví dụ trong phần này nêu ra các bước cần thực hiện sau khi có tập tin chương trình.

Bảng 4 minh họa phương pháp nhập bước và chuyển tiếp đơn giản. Để nhập các cấu trúc SFC, bạn bắt đầu từ màn hình lựa chọn SFC Edit.

Bảng 4: Nhập Bước/ Chuyển tiếp đơn giản trên bộ xử lý PLC-5

Cặp bước/ chuyển tiếp	Thực hiện
	[F3] - Gắn kết bước/ chuyển tiếp hoặc [F4] – Chèn bước/ chuyển tiếp. Đưa con trỏ đến tập tin chương trình bước. [F1] - Chọn bước Đưa con trỏ đến tập tin chương trình chuyển tiếp. [F2] – Chọn chuyển tiếp [Enter] để trở về lựa chọn SFC Edit [F9] – Cấu hình/xây dựng [F10] – Sơ đồ xây dựng

Bảng 5 trình bày các tác vụ bạn có thể thực hiện trên màn hình Assigning Ladder Logic (gắn kết hoặc chèn bước/ chuyển tiếp).

Bảng 5. Các tác vụ khả dụng trên màn hình Assigning Logic

Tác vụ	Nhấn phím
Chọn số tập tin được làm sáng cho bước hiện hành	[F1] – chọn bước
Chọn số tập tin được làm sáng cho chuyển tiếp hiện hành	[F2] – Chọn chuyển tiếp
Chọn nhiều bước và chuyển tiếp trong cấu trúc hiện hành (nhánh), đưa con trỏ đến bước hoặc chuyển tiếp kế tiếp.	[F3] – Bước/chuyển tiếp kế tiếp
Bỏ tập tin chương trình đang được trỏ vào	[F5] – Bỏ chọn tập tin
Tạo tập tin chương trình	[F6] – Tạo tập tin
Đổi tên tập tin chương trình đang chọn	[F10] – Đổi tên tập tin

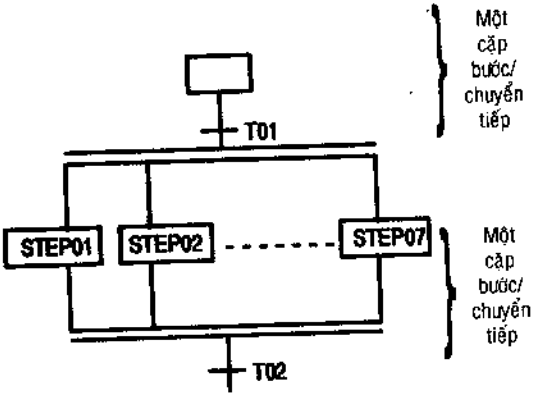
Tạo sự lựa chọn và các nhánh đồng thời

Để tạo các nhánh, bạn hãy xem các bước trong Bảng 6

Bảng 6. Nhập bước/ chuyển tiếp nhánh lựa chọn trên bộ xử lý PLC – 5

Các cặp bước/ chuyển tiếp	Nhấn phím
	[F1] – Chọn nhánh
	[F3] – Gắn kết, hoặc [F4] - chèn Gõ điểm cuối của nhánh (gõ số lớn nhất để gắn kết nhánh ở cuối cấu trúc hiện hành)
	[Enter] Đưa con trỏ đến chân nhánh
Một cặp bước/ chuyển tiếp	[F1] – Kéo dài bên trái, hoặc [F2] – kéo dài bên phải cho mọi chân nhánh
	Đưa con trỏ đến nhánh bước trước.
	[F5] - Chỉnh sửa bước/chuyển tiếp. Đưa con trỏ đến tập tin chương trình bước.
Một cặp bước/ chuyển tiếp	[F1] – Chọn bước
	[F3] – Bước/chuyển tiếp kế tiếp
	Đưa con trỏ đến tập tin chương trình chuyển tiếp.
	[F2] – Chọn chuyển tiếp
	[F3] – Bước/chuyển tiếp kế tiếp
	Xác định các chuyển tiếp còn lại
	[Enter] trở về lựa chọn Selected Branch
	[Enter] trở về lựa chọn SFC Edit
	Đưa con trỏ đến chuyển tiếp dưới, thuộc nhánh bất kỳ
	[F4] – Chèn bước/chuyển tiếp
	Xác định các tập tin chương trình bước và chuyển tiếp
	Bổ sung các bước/ chuyển tiếp cho các đường dẫn còn lại trong nhánh
	[Enter] trở về SFC Edit
	[F9] – SFC config/Build
	[F10] – Build Chart

Bảng 7. Nhập bước/chuyển tiếp nhánh đồng thời trên bộ xử lý PLC –5

Cặp bước/ chuyển tiếp	Nhấn phím
	[F2] – Nhánh đồng thời [F3] – Gắn kết Gõ điểm kết thúc nhánh [Enter] [F1] – Kéo dài bên trái, hoặc [F2] – kéo dài bên phải cho mọi chân nhánh Đưa con trỏ đến bước trên nhánh tận cùng bên trái. [F5] – chỉnh sửa bước/chuyển tiếp Đưa con trỏ đến tập tin chương trình bước [F1] – Chọn bước. [F3] – Bước/chuyển tiếp kế tiếp Xác định các tập tin bước còn lại. [F3] – Bước/ chuyển tiếp kế tiếp, con trỏ ở trên chuyển tiếp này. Đưa con trỏ đến tập tin chương trình chuyển tiếp. [F2] – Chọn chuyển tiếp [Enter] trở về lựa chọn Simultaneous Branch [Enter] trở về lựa chọn SFC Edit [F9] – SFC Config/Build [F10] – Build Chart

MỤC LỤC

HƯỚNG DẪN THIẾT KẾ & LẬP TRÌNH PLC

- Trần Thế San - Nguyễn Ngọc Phương

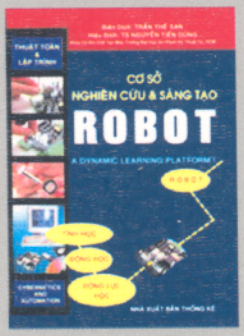
Chịu Trách Nhiệm Xuất Bản.
Giám Đốc : **VÕ VĂN ĐÁNG**
Tổng Biên Tập : **NGUYỄN ĐỨC HÙNG**

Biên Tập : **TRẦM MY**
Trình Bày : **DUY TRẦN**
Vẽ Bìa : **DUY TRẦN**

Liên kết xuất bản

C. TY VĂN HÓA **TRÍ DÂN** - HS. **NGUYỄN TRÃI**
96/7 DUY TÂN - P.15 - Q. PHÚ NHUẬN - TP.HCM
ĐT : 8383669 - 9901846 - Fax : 9971765

In 1000 cuốn khổ (14.5x20,5)cm tại Xưởng In Trung Tâm Hội Chợ Triển Lãm Việt Nam. GPXB số 42-1301/XB-QLXB do Cục Xuất Bản cấp ngày 10/09/2004. In xong & nộp lưu chiểu tháng 06 năm 2005.



Nơi Phát Hành
HIỆU SÁCH NGUYỄN TRÃI
4A - Ng Trãi - Q5 - Tp. HCM
ĐT: 8383669

GIÁ: 25 000đ