

ThS. Châu Chí Đức

TỔNG QUAN về TỰ ĐỘNG HÓA VỚI SIMATIC

**Thành phố Hồ Chí Minh
10-2012**

Lời nói đầu

Bất cứ một họ sản phẩm nào cũng vậy, nếu nắm được tổng quan sẽ giúp người sử dụng chọn lựa một hoặc nhiều mẫu phù hợp với các yêu cầu của mình. Đối với các hệ thống tự động hóa điều này lại càng quan trọng, vì lựa chọn không đúng sẽ gây thiệt hại về kinh tế và khó khăn cho công tác bảo trì sửa chữa cũng như khả năng mở rộng của hệ thống.

Nhằm giúp độc giả yêu thích tự động hóa có một cái nhìn tổng quan về họ SIMATIC của Siemens, tôi xin gửi đến các bạn tập tài liệu này.

Ngoài việc đi sơ lược qua các sản phẩm, tập tài liệu còn giúp các bạn nắm được phần mềm STEP 7, trình soạn thảo chương trình, các ngôn ngữ lập trình, cách thức lập chương trình và chẩn đoán lỗi với họ Simatic.

Tập tài liệu chỉ ở dạng lý thuyết. Để có thể tự học (lý thuyết và thực hành) các bạn nên đọc các quyển mà tác giả dự định sẽ hoàn thành:

1. Kỹ thuật điều khiển lập trình PLC Simatic S7-300/400
2. Mạng truyền thông công nghiệp Profibus
3. Mạng truyền thông công nghiệp AS-Interface
4. và một số tài liệu liên quan đến PLC khác

Vì tài liệu đã thực hiện từ lâu, nên việc cập nhật thông tin mới chưa đầy đủ và còn thiếu sót. Nếu được sự ủng hộ từ độc giả, chắc chắn nó sẽ được chỉnh đốn và bổ sung lại. Rất mong được sự góp ý của các bạn.

Sách chỉ ở dạng Ebook được xuất bản ở dạng file pdf được chia làm 2 phiên bản: một phiên bản được mở và đọc tự do, còn phiên bản còn lại cần phải có password.

Thông tin liên lạc: **ccduduc2006@gmail.com**

Mục lục

1	Dẫn nhập
1.1	Các phần tử của hệ thống tự động.....
1.2	Từ nhiệm vụ tự động hoá đến chương trình hoàn tất.....
1.3	Bộ điều khiển PLC làm việc như thế nào?
1.4	Cách thức đưa một tín hiệu nhị phân từ bộ cảm biến vào chương trình
1.5	Cấu trúc của một dự án simatic.....
2	Bộ điều khiển simatic
2.1	Các phần tử của một trạm simatic.....
2.2	Micro plc simatic S7-200
2.3	Bộ điều khiển simatic S7-300.....
2.4	Simatic S7-400 cho các nhiệm vụ điều khiển có yêu cầu cao ..
2.5	Khả năng dự phòng cao của Simatic.....
2.6	Hệ thống simatic C7
2.7	Máy tính tự động hóa simatic M7
2.8	Xử lý kết nối với các module digital
2.9	Xử lý kết nối với các module analog.....
2.10	Các module chức năng trợ giúp CPU
2.11	Kết nối các module CP đến mạng truyền thông
2.12	“Bộ não” của simatic S7: module CPU
2.13	Bộ xử lý trung tâm với các chức năng đặc biệt
2.14	Simatic PC Based với WinAC
2.15	Các kết nối xử lý phân tán.....
2.16	Simatic DP - ngoại vi phân tán
2.17	DP slaves: kết nối xử lý ở gần máy
2.18	Thiết bị lập trình simatic
3	Step 7: công cụ chuẩn cho Simatic
3.1	Quản lý dữ liệu ở Simatic
3.2	Step 7
3.3	Simatic Manager
3.4	Dự án và thư viện
3.5	Soạn thảo một dự án
3.6	Đặt cấu hình cho một trạm simatic
3.7	Sắp xếp và gán các tham số cho các module
3.8	Địa chỉ các module.....
3.9	Tạo chương trình người dùng
3.10	Soạn thảo ký hiệu
3.11	Trình soạn thảo.....
3.12	Lập trình khối logic.....

3.13	Lập trình một khối dữ liệu
3.14	Lập trình hướng nguồn của các khối
3.15	Trợ giúp tạo chương trình.....
3.16	Chẩn đoán hệ thống
3.17	Nạp chương trình người dùng vào CPU
3.18	Chẩn đoán trong khi kiểm tra chương trình.....
3.19	Giám sát, điều khiển và cưỡng bức các biến.....
3.20	Trạng thái chương trình
3.21	Kiểm tra chương trình người dùng ở offline với PLCsim.....
3.22	Điều khiển PID với phần mềm simatic
3.23	Lập hồ sơ kết nối dây với Docpro
3.24	Kết nối mạng điện thoại với Telesevice.....
4	Ngôn ngữ lập trình.....
4.1	Ngôn ngữ lập trình cơ bản LAD, FBD và STL.....
4.2	Các phép toán nhị phân.....
4.3	Các phép toán số.....
4.4	Điều khiển theo lưu đồ chương trình
4.5	Ladder Logic LAD
4.6	Function Block Diagram FBD.....
4.7	Statement List STL
4.8	Ngôn ngữ điều khiển có cấu trúc SCL
4.9	Biểu đồ chức năng liên tục CFC
4.10	Điều khiển trình tự GRAPH.....
4.11	Điều khiển đồ hình trạng thái HiGRAPH
5	Chương trình người dùng
5.1	Khối tổ chức và cấp ưu tiên
5.2	Các kiểu xử lý chương trình
5.3	Chương trình khởi động
5.4	Reset bộ nhớ
5.5	Chương trình chính
5.6	Thông tin khởi động.....
5.7	Các chức năng CPU
5.8	Bộ đệm.....
5.9	Thời gian chu kỳ, thời gian đáp ứng
5.10	Xử lý ngắt (tổng quan)
5.11	Ngắt cứng.....
5.12	Ngắt chu kỳ.....
5.13	Ngắt thời gian trong ngày
5.14	Ngắt trễ thời gian.....
5.15	Ngắt đa xử lý.....
5.16	Xử lý lỗi
5.17	Xử lý các sự cố ngắt.....

5.18	Khối người dùng (tổng quan)
5.19	Cấu trúc của một khối
5.20	Gọi khối và các tham số khối
5.21	Dữ liệu cục bộ tạm thời (temporary local data).....
5.22	Dữ liệu cục bộ tĩnh (static local data)
5.23	Multi-instances, local instance.....
5.24	Timer của simatic.....
5.25	Counter của simatic
5.26	Vùng địa chỉ toàn cục.....
5.27	Địa chỉ dữ liệu toàn cục.....
5.28	Địa chỉ tuyệt đối và địa chỉ ký hiệu
5.29	Địa chỉ gián tiếp
5.30	Các kiểu dữ liệu-tổng quan
5.31	Các kiểu dữ liệu cơ bản
5.32	Các kiểu dữ liệu tổng hợp.....
5.33	Các kiểu tham số
5.34	Kiểu dữ liệu người dùng định nghĩa (UDT)
6	Truyền thông.....
6.1	Các mạng ở Simatic.....
6.2	Các lợi ích truyền thông
6.3	Đặt cấu hình network
6.4	Đặt cấu hình các kết nối.....
6.5	Đặt cấu hình các ngoại phân bố.....
6.6	Các địa chỉ trong hệ thống DP Master.....
6.7	Các chức năng DP đặc biệt
6.8	Truyền thông dữ liệu toàn cục.....
6.9	Truyền thông SFC.....
6.10	Truyền thông SFB.....
7	Bảng điều khiển và giám sát xử lý.....
7.1	Đặt cấu hình Simatic HMI.....
7.2	Bảng điều khiển nút nhấn PP7 và PP17.....
7.3	Text display TD 17
7.4	Các bảng điều khiển text OP3, OP7 và OP17
7.5	Các bảng điều khiển đồ họa OP27 và OP37
7.6	Touch Panels TP27 và TP37
7.7	Hệ MP270 và OP37/pro chạy trên windows
7.8	Thực hiện tối ưu các thao tác với Operator Interfaces.....
7.9	Kết nối với các trạm Simatic S7
7.10	Đặt cấu hình thiết bị HMI với Simatic Protool
7.11	Chẩn đoán xử lý dùng Simatic ProAgent.....
7.12	Quan sát, điều khiển vận hành dùng Simatic WinCC
7.13	Chẩn đoán xử lý chương trình người dùng bằng S7-Pdiag.....

1 Dẫn nhập

1.1 Các phần tử trong hệ thống tự động SIMATIC

Hệ thống tự động SIMATIC là một chuỗi các phần tử kết hợp với nhau với những phương pháp đồng nhất về cấu hình quản lý dữ liệu và truyền tải dữ liệu.

Thiết bị điều khiển lập trình **SIMATIC S7** (PLC) là dạng cơ bản của hệ thống tự động. Có ba loại : s7-200 là dạng cực nhỏ; s7-300 là dạng nhỏ sử dụng cho các ứng dụng nhỏ và trung bình và s7-400 là bộ điều khiển đáp ứng được các yêu cầu điều khiển cao nhất. Các thiết bị PLC chủ yếu bao gồm: một CPU (Bộ điều khiển trung tâm) và các module xuất nhập. CPU chứa chương trình của người dùng được viết bằng ngôn ngữ dùng cho PLC, và các module xuất nhập để kết nối với các máy móc hoặc các thiết bị cần điều khiển.

Máy tính cho hệ thống tự động **SIMATIC M7** là loại máy tính tương thích với hệ AT, ở máy tính này hệ thống SIMATIC được cung cấp chính là phần mềm. CPU M7 và các module ứng dụng (SM) có thể dùng với s7-300 và s7-400 nối vào các máy móc hay các thiết bị được điều khiển bởi các module xuất nhập của S7 và module mở rộng đặc biệt M7.

Thiết bị hoàn chỉnh **SIMATIC C7** được thiết kế cho một hệ thống điều khiển máy móc và thể hiện tính chất gọn nhẹ. Nhân viên điều hành kiểm soát và điều khiển máy móc từ màn hình điều khiển và các phím. Các thiết bị điều khiển bên trong sẽ kiểm soát thiết bị thông qua các cổng xuất nhập hoặc các cổng xuất nhập mở rộng.

Thiết bị điều khiển sử dụng phần mềm **SIMATIC WINAC** được cài đặt trên một máy tính tiêu chuẩn hoặc ở dưới dạng phần mềm dùng riêng cho PLC hoặc dưới dạng các card cắm vào PC. Chúng cung cấp các giao tiếp ứng dụng cho máy vi tính và cho phép chúng ta duy trì trạng thái trực tuyến trên máy.

Hệ thống phân tán các ngõ vào ra **SIMATIC DP** cho phép chúng ta lắp đặt các module xuất nhập PLC cách xa hệ thống điều khiển trung tâm với số lượng dây tối thiểu bằng phương tiện mạng

PROFIBUS. Chương trình của người dùng xử lý các module xuất nhập theo cùng một cách giống như các module xuất nhập ở trung tâm.

Hệ giao tiếp người và máy **SIMATIC HMI** cho phép hiển thị từ văn bản cho đến khả năng điều hành bằng hệ thống đồ họa, sản phẩm phần mềm này cung cấp tất cả các tiện nghi cần thiết cho bạn để điều hành và giám sát máy móc hay qui trình sản xuất. Phần mềm chỉ ra các tình trạng của thiết bị máy móc cùng với các sự kiện và các thông báo lỗi, quản lý các phương án xử lý sự cố, đo đạc các giá trị, cung cấp dịch vụ hỗ trợ việc tìm lỗi, bảo hành và sửa chữa.

Hệ thống mạng **SIMATIC NET** sẽ kết nối tất cả các trạm của SIMATIC lại với nhau và đảm bảo việc kết nối thông suốt trên một đường dây cáp. Chúng có thể trao đổi dữ liệu với nhau hoặc có thể truyền thông với các trạm trong hệ thống mạng từ thiết bị lập trình trung tâm. Với SIMATIC NET ta có thể nối mạng với các thiết bị không thuộc chủng loại SIMATIC, kể cả từ thiết bị trường trong hệ thống điều khiển cho đến việc quản lý dữ liệu bằng máy vi tính.

Phần mềm **STEP 7** là phần mềm cơ bản về tự động hóa tích hợp toàn diện với cấu hình đồng nhất cả về lập trình, quản lý dữ liệu cũng như truyền dữ liệu. Bạn sử dụng STEP 7 để cấu hình các phần tử của SIMATIC, chỉ định các thông số cho chúng và lập trình. SIMATIC Manager là phần mềm cần thiết trong step 7, nó là công cụ trung tâm để quản lý dữ liệu tự động. Nó lưu trữ tất cả các dữ liệu của một dự án tự động hóa trong một thư mục dự án có cấu trúc phân cấp và ngoài ra còn tích hợp trong thư viện của nó các khối chương trình ứng dụng của các phiên bản phần mềm của các hệ tự động khác trước SIMATIC S7.

Các hoạt động chính với STEP 7 là :

- Cấu hình phần cứng: là cấu hình các module được xếp đặt trên giá, chỉ định những địa chỉ cho chúng và thiết đặt các tính chất cho khối.
- Cấu hình kết nối truyền thông: Là định nghĩa các thành phần kết nối và tính chất của kết nối này.
- Viết chương trình cho người dùng bằng ngôn ngữ lập trình PLC theo dạng bậc thang logic (LAD), sơ đồ khối chức năng (FBD) hoặc là dạng liệt kê câu lệnh (STL) và kiểm tra chương trình bằng cách nối trực tiếp với hệ điều khiển.

Có nhiều gói phần mềm tùy chọn khác nhau để mở rộng những chức năng của công cụ chuẩn của STEP 7. Ví dụ công cụ hỗ trợ thêm ngôn ngữ lập trình và phương pháp lập trình, phần mềm thiết kế giao diện

cho HMI, phần mềm để cấu hình cho việc xử lý truyền thông, chức năng điều khiển có hồi tiếp .v.v.

Hình 1.1 *Các phần tử của hệ thống tự động simatic*

1.2 Từ nhiệm vụ tự động hóa đến chương trình hoàn tất

Khi bạn bắt đầu giải quyết một bài toán về tự động hóa, bạn sẽ phải tự hỏi là phải dùng Loại PLC nào. Nếu các máy móc được điều khiển là một hệ thống nhỏ, không phức tạp thì S7-200 là đạt chưa hay cần đến S7-300? Chúng ta thấy điều khiển một thiết bị bằng S7-400 thì tốt hơn hay là chỉ cần hai thiết bị thuộc chủng loại S7-300? Chúng ta cần một trạm điều khiển với các cổng xuất nhập gọn nhẹ ở trung tâm hay là các cổng xuất nhập phân tán trong hệ thống?

Những nét đại cương sau đây là các bước đưa chúng ta từ nhiệm vụ tự động hóa để đi đến chương trình hoàn tất. Những trường hợp riêng với các thiết bị cô đọng, một vài bước có thể được bỏ qua hoặc là một vài bước khác lại có thể phải thêm vào.

1.2.1 Chọn phần cứng

Có rất nhiều tiêu chí để chọn các kiểu điều khiển. Đối với một ứng dụng điều khiển nhỏ thì tiêu chí chính là số lượng ngõ vào/ra và kích thước của chương trình người dùng. Đối với hệ thống lớn hơn

bạn cần phải tự hỏi liệu thời gian đáp ứng có đủ ngắn không, và liệu rằng bộ nhớ sử dụng có đủ lớn cho khối lượng dữ liệu cần quản lý (phương pháp, lưu trữ) để có thể ước chừng được nguồn tài nguyên được yêu cầu. Bạn cần phải có nhiều kinh nghiệm về các giải pháp tự động đã tích lũy trước đó; không có một quy tắc chung nào cho việc này.

Một hệ thống máy móc để sản xuất có thể sẽ được điều khiển chỉ với một thiết bị PLC. Trong trường hợp này số lượng ngõ vào và ngõ ra cũng như kích thước của bộ nhớ và có thể tốc độ (thời gian đáp ứng) sẽ giúp bạn việc chọn lựa giữa S7-200, S7-300 hay là S7-400. Thế còn máy móc hoạt động như thế nào? (điều này cho phép bạn xác định rằng nên dùng thiết bị điều hành giám sát riêng lẻ của hệ SIMATIC HMI, hay là một hệ thống hoàn chỉnh SIMATIC C7). Đối với một hệ thống mở rộng đến nhiều vị trí khác nhau, thì tính kinh tế sẽ tốt hơn khi sử dụng các cổng xuất nhập loại phân tán hơn là loại tập trung, trong nhiều trường hợp, điều này chẳng những làm giảm việc nối dây, mà còn phân nhóm điều khiển, kiểm soát ở vị trí hoạt động của nó từ đó cắt giảm đáng kể thời gian đáp ứng và giá thành nghiên cứu dự án (nên sử dụng hệ xuất nhập kiểu phân tán SIMATIC DP, có thể sử dụng các loại cổng DP thông minh có chương trình riêng để xử lý tín hiệu riêng cho chúng). Giải pháp tự động hóa phân tán mang đến nhiều lợi ích: các chương trình người dùng cho nhiều phần khác nhau của thiết bị sẽ ngắn hơn, sẽ có thời gian đáp ứng nhanh hơn, và thông thường có thể khởi động độc lập. Thời gian cần thiết để trao đổi dữ liệu với hệ kiểm soát trung tâm trong hệ thống SIMATIC cực kỳ dễ dàng bằng cách sử dụng SIMATIC NET và các chức năng truyền thông kèm theo.

1.2.2 Chọn ngôn ngữ lập trình

Việc chọn ngôn ngữ lập trình tùy thuộc vào nhiệm vụ. Nếu nó chủ yếu chỉ gồm các tín hiệu số thì ngôn ngữ lập trình LAD và FBD là lý tưởng. Đối với các nhiệm vụ khó khăn hơn đòi hỏi phải duy trì các biến phức tạp và các địa chỉ truy cập gián tiếp, ta có thể dùng ngôn ngữ STL bao gồm ngôn ngữ assembly, ngôn ngữ điều khiển theo cấu trúc SCL là chọn lựa tốt nhất cho những ai đã quen với ngôn ngữ lập trình cấp cao và những ai muốn viết chương trình để xử lý một phần lớn dữ liệu của qui trình.

Có nhiều phương pháp khác nhau để lập trình làm cho việc lập trình dễ dàng hơn: trong đó có S7-GGRAPH để lập trình theo trình tự, ta có thể lập trình theo đồ hình trạng thái với S7-HIGGRAPH, và S7-CFC (lưu đồ hoạt động liên tục) để kết nối các hoạt động đang có lại với nhau. Các khối được tạo sẵn giúp cho bạn viết chương trình được dễ

dàng hơn, ví dụ để viết chương trình điều khiển có hồi tiếp hay là các thông báo về thiết đặt.

1.2.3 Cách tạo một dự án

Tất cả dữ liệu cho giải pháp tự động hóa của bạn được tập trung lại trong một "dự án"(Project). Dự án được tạo ra bằng phần mềm STEP 7. Một dự án là một thư mục phần mềm mà trong đó tất cả dữ liệu được trình bày theo cấu trúc thứ tự nhiều cấp. Thứ bậc kế tiếp về phía dưới của " dự án" là các "trạm". Các trạm này chứa một hay nhiều CPU với một chương trình riêng. Sử dụng menu lệnh để chèn vào các đối tượng mới, mở những đối tượng này và khởi động một cách tự động những công cụ cần thiết để làm việc với chúng.

Ví dụ: Một trạm chứa một đối tượng được gọi là *Hardware*. Bạn nhấp đúp vào đối tượng này để khởi động công cụ *cấu hình phần cứng (hardware configuration)* để cấu hình các phần cứng của trạm, nghĩa là sắp đặt các module trên một giá, và chỉ định các địa chỉ cũng như các thông số cho chúng. Trong SIMATIC, bạn làm tất cả bằng phần mềm; bạn định nghĩa tính chất của các module trong các hộp thoại với sự hỗ trợ của phần trợ giúp trực tiếp.

1.2.4 Viết , tìm lỗi và lưu trữ chương trình

Chương trình người dùng chứa tất cả các chỉ thị và các sắp xếp được lập trình bởi người dùng cho việc xử lý tín hiệu để điều khiển máy móc hay thiết bị. Những nhiệm vụ lớn, phức tạp sẽ dễ giải quyết nếu chúng được chia ra thành các khâu nhỏ có thể quản lý được, các khâu này có thể lập trình dưới dạng các khối (chương trình con). Việc chia các khối có thể theo đối tượng quá trình hay là đối tượng nhiệm vụ. Trong trường hợp đầu, mỗi khối chương trình tương ứng với một phần của máy móc thiết bị hay là của một thiết bị (trộn, băng chuyền, tổ hợp khoan). Trong trường hợp thứ hai chương trình được chia ra theo nhiệm vụ điều khiển, ví dụ chọn tín hiệu, chọn truyền thông, cho cách thức hoạt động trong thực tế, một hỗn hợp của hai loại cấu hình trên thường được dùng.

Các đối tượng dùng để tạo ra chương trình là bảng ký hiệu và khối biên dịch và tùy thuộc vào ngôn ngữ lập trình – đây là chương trình của các tập tin nguồn. Bạn nhấp đúp chuột vào các tập tin của chương trình nguồn hay các khối đối tượng để bắt đầu chương trình soạn thảo và có thể nhập hay thay đổi chương trình của khối. Bạn viết chương trình ở chế độ không nối máy (ở chế độ OFFLINE) và nhớ nó vào đĩa cứng của thiết bị lập trình.

Để khởi động hệ thống, bạn nối thiết bị lập trình với CPU, tải chương trình vào bộ nhớ chương trình của CPU và kiểm tra nó. Bạn có thể giám sát và thay đổi giá trị của các biến và quan sát chu trình thực hiện thế nào. Các chức năng chẩn đoán rộng giúp bạn có thể xác định nơi chôn và nguồn gốc của lỗi rất nhanh chóng. Bạn có thể kiểm soát những vùng riêng lẻ của chương trình khi không nối máy với phần mềm tùy chọn thêm có tên là PLCSIM. Khi bạn khởi động hệ thống thành công, bạn sao chép chương trình vào card nhớ và tạo hồ sơ cho dự án, ví dụ ở dưới dạng là tập tin hướng dẫn mạch điện với phần mềm DOCPRO. Với phần mềm STEP7 bạn có thể lưu trữ toàn bộ dự án dưới dạng tập tin nén.

1.3 Bộ điều khiển PLC

Nói chung khi điều khiển một hệ thống, vấn đề điều khiển được giải quyết bằng cách nối dây hoàn tất các contactor và các relay riêng lẻ theo nhiệm vụ đặt ra. Contactor và relay là hệ kiểm soát điện và điện tử được kết hợp từ các phần tử riêng lẻ được gọi là *hệ điều khiển lập trình cứng (hard-wired programmed controller)*. Thuật ngữ "lập trình" ở đây có nghĩa là nối dây. Mặt khác, hệ *điều khiển logic khả trình (programmable logic controller)* được thành lập bằng những phần tử cơ bản tiêu chuẩn, chức năng điều khiển theo yêu cầu được đưa vào chương trình do người dùng tạo ra đặt vào CPU.

Hệ tự động hóa SIMATIC S7 dựa trên cơ sở là bộ điều khiển logic khả trình trung tâm. Giải pháp cho một thao tác tự động được nhớ trong bộ nhớ chương trình của CPU dưới dạng các chỉ thị của chương trình. CPU đọc lần lượt hết chỉ thị này đến chỉ thị khác, biên dịch chúng và đảm bảo chúng đã được thực hiện.

Quy trình về các chỉ thị lệnh của CPU theo mã máy MC7, dù là viết trực tiếp hay là do biên dịch mà có. Bất kỳ một ngôn ngữ lập trình nào bạn dùng để viết chương trình cho người dùng luôn luôn được biến đổi thành các chỉ thị thích hợp với MC7, dạng liệt kê lệnh STL là ngôn ngữ lập trình gần với mã máy MC7 nhất.

1.3.1 Các phép toán logic sử dụng các tín hiệu nhị phân

Ví dụ một đoạn chương trình: Khi hai ngõ vào I5.2 và I4.7 ở trạng tín hiệu "1" thì ngõ ra Q8.5 cũng được đặt ở trạng thái tín hiệu "1" còn nếu không thì nó được đặt ở "0". Chương trình viết trong MC7 và STL như sau :

```

...
= Q 8.4 // Từ hoạt động logic trước
A I 5.2 // Kiểm tra ngõ vào I 5.2 ("lần kiểm đầu tiên"). Trạng thái
tín hiệu này trở thành kết quả của hoạt động logic
(RLO).

A I 4.7 // Kiểm tra ngõ vào I 4.7, thi hành toán tử AND với nội
dung chứa ở RLO và lưu lại dưới dạng RLO mới

= Q 8.5 // Xuất RLO ra ngõ ra Q 8.5.
A I 5.3 // đến hoạt động logic kế tiếp
...

```

CPU sẽ xử lý chương trình này từng lệnh một theo thời gian. Khi câu lệnh A I5.2 được xử lý, module ngõ vào tương ứng chọn cảm biến ở địa chỉ đã quy định, ngõ vào I5.2. CPU kiểm tra trạng thái của tín hiệu của cảm biến đã được lựa chọn ở trên. Nếu đây là câu lệnh logic đầu tiên thì sau chỉ thị điều khiển, CPU ngay lập tức nhớ trạng thái của địa chỉ đã kiểm tra vào RLO (kết quả của phép toán logic) mà không cần thực hiện phép toán logic trên nó. Khi xử lý đến lệnh kế tiếp, CPU sẽ thực hiện phép toán logic cần thiết trên kết quả của việc kiểm tra và kết quả của phép toán logic trước đó đã được nhớ trong RLO. Trong ví dụ, nó xử lý lệnh xuất ra ngõ ra bằng cách đưa kết quả lưu trữ của phép toán logic ở RLO ra địa chỉ ngõ ra (Q8.5).

Khi kết quả “cũ” của phép toán logic đã được lưu trữ, chỉ thị kiểm tra kế tiếp bắt đầu với một phép toán logic mới. Ở đây kết quả của kiểm tra đầu tiên trở thành kết quả mới của phép toán logic và được chứa vào RLO.

Hình 1.2 Cách thức PLC xử lý một câu lệnh bit

1.3.2 Xử lý chương trình chu kỳ.

CPU của PLC xử lý chương trình của người dùng một cách liên tục. Chương trình thực hiện liên tục ngay cả không có tác động nào cần thiết ở bên ngoài, ví dụ nếu thiết bị máy móc ở bên ngoài đang ở trạng thái dừng. Điều này khiến cho việc lập trình dễ hơn. Ví dụ, bạn có thể viết chương trình dạng bậc thang giống như là bạn đã vẽ sơ đồ logic dạng bậc thang cho các relay hay bạn có thể viết chương trình theo sơ đồ chức năng như bạn đã từng nối các linh kiện điện tử. Một cách tổng quát, PLC tương tự như bộ kiểm soát bằng relay hay ngắt điện tử: nhiều phép toán logic trong chương trình có tác dụng “theo dạng song song” cùng một lúc. Tại sao thực hiện được như vậy?

Khi nguồn được bật lên, CPU xử lý hệ điều hành, và chương trình khởi động nếu có. Rồi nó quay trở về chương trình chính xử lý những phép toán logic cho lần đầu tiên. CPU kiểm tra các tín hiệu ngõ vào, thực hiện những phép toán logic trên chúng và set hay là reset các ngõ ra.

Khi CPU thực hiện chương trình chính đến phần cuối cùng, nó tự động thực hiện xử lý chương trình trở lại. Nó lại kiểm tra ngõ vào, thực hiện các phép toán logic trên chúng và lại set hay reset ngõ ra. Bằng phương pháp này trạng thái của các ngõ ra đã được điều chỉnh để thích hợp với sự thay đổi trạng thái của các ngõ vào ở những quãng thời gian rất ngắn (đáp ứng thời gian tùy thuộc vào độ dài chương trình chính).

Kiểu chương trình xử lý theo chu kỳ tuần hoàn thể này là kiểu phổ biến của các bộ PLC. Đối với các ứng dụng

Hình 1.3 Xử lý chương trình chu kỳ ở PLC

đặc biệt, chu kỳ của chương trình đang xử lý có thể được ngắt bởi các ngắt phần cứng hay là các ngắt theo thời gian trong ngày (ví dụ như các ngắt cố định).

1.4 Cách thức đưa một tín hiệu nhị phân từ bộ cảm biến vào chương trình

Nhằm mục đích để làm được công việc của nó, CPU trong PLC cần phải liên kết với thiết bị máy móc hay thiết bị mà nó sẽ điều khiển. Mỗi liên kết này được thực hiện bởi các module ngõ vào và ngõ ra, các module này được nối với cảm biến hay các phần tử tác động tức là các máy móc hay là các thiết bị.

1.4.1 Kết nối với PLC, địa chỉ các module.

Khi bạn nối dây cho máy móc hoặc thiết bị, có nghĩa là bạn đã quyết định mỗi một tín hiệu được nối vào PLC ở đâu. Một tín hiệu ngõ vào, ví dụ là tín hiệu từ tiếp điểm S0 có nghĩa là “khởi động động cơ”, được nối vào một đầu nối của một số module ngõ vào. Đầu nối này có một “địa chỉ” (ví dụ byte 5, bit 2). Địa chỉ của một module có thể là cố định theo rãnh mà ở đó module được lắp đặt vào giá hoặc bạn có thể thiết đặt nó với công cụ cấu hình phần cứng của STEP 7. Địa chỉ của các module được xếp đặt ở dạng byte (là những nhóm gồm 8 bit các ký hiệu nhị phân) địa chỉ khởi đầu của module là địa chỉ thấp nhất của khối. Nếu ví dụ đây là 0, và module có thêm một byte nữa, địa chỉ của byte kế tiếp của module sẽ tự động là 1, và cứ thế tiếp tục. Các bit địa chỉ ở mỗi byte được đánh số từ 0 đến 7.

1.4.2 Địa chỉ tuyệt đối và địa chỉ ký hiệu.

CPU thi hành chương trình của người dùng (chương trình chính) một cách tuần tự khi chạy đến phần cuối của chương trình, nó bắt đầu lại từ đầu. Mỗi khi nó xử lý một chương trình, trước tiên CPU tự động sao chép tín hiệu từ các module ngõ vào vào bảng bộ đệm ngõ vào, một vùng trong bộ nhớ hệ thống (system memory) của CPU. như vậy tín hiệu có thể được định vị dưới địa chỉ của nó trong bộ đệm, ví dụ ngõ vào I5.2. Hình thức diễn tả “I5.2” là địa chỉ tuyệt đối. Nếu có năng lượng ở ngõ vào này thì I5.2 có trạng thái tín hiệu “1”.

Bạn có thể cho ngõ vào một tên, ngay khi bạn thiết lập cấu hình phần cứng hoặc sau này. Điều này được thực hiện trong bảng ký hiệu bằng cách nhập vào các ký tự hoặc số ký hiệu có thể mang tải ý nghĩa của tín hiệu ngõ vào ở địa chỉ tuyệt đối, ví dụ “mở máy động cơ”. Câu văn diễn tả “mở máy động cơ” là địa chỉ ký hiệu bạn có thể chỉ định

những tên cho tất cả các tín hiệu trong bảng ký hiệu và dùng những tên này để định địa chỉ tín hiệu cho rõ ràng hơn trong chương trình người dùng.

Hình 1.4 Địa chỉ khối, địa chỉ tuyệt đối và địa chỉ ký hiệu

1.4.3 Kiểm soát ngõ ra của máy móc hay thiết bị.

Các ngõ ra cũng được xử lý theo cùng cách thức. Chúng có địa chỉ trong vùng nhớ của CPU được gọi là “bảng bộ đệm ngõ ra”. Trong chương trình, một ngõ ra thì được set hay reset – theo bộ đệm - tùy thuộc vào các điều kiện xác định. Ở phần kết thúc của chương trình, CPU chép bảng bộ đệm ngõ ra của quy trình vào các module ngõ ra và điều khiển các đầu mối liên hệ với các phần tử tác động đã nối vào nó, ví dụ một contactor, một bóng đèn hay một thiết bị điện tử.



Hình 1.5 Một hệ tư đồng tượng trưng bằng các đối tượng "logic" trong dự án SIMATIC

1.4.4 Ghi địa chỉ cho các khối.

Bạn cũng có thể ghi địa chỉ ngõ vào, ngõ ra cho các module một cách trực tiếp trong chương trình. Trong trường hợp này, đơn vị nhỏ nhất bạn có thể đặt địa chỉ là 1 byte. Điều này gây ra nhiều khó khăn cho việc kiểm tra hay đặt một tín hiệu đơn giản, nghĩa là chỉ là một bit đơn. Đối với một chương trình người dùng nó không có gì khác nhau giữa việc bạn nối một cảm biến và một thành phần tác động vào các module đã được lắp đặt lên giá trung tâm, hay là phân bố các module ngõ vào và ngõ ra đã được cài đặt trong một dự án và nối vào bộ điều khiển trung tâm bằng bus nối dữ liệu. Bạn đánh địa chỉ ngõ vào và ngõ ra theo cùng một cách, nghĩa là bạn kiểm soát cảm biến trong bảng bộ

đệm ngõ vào và điều khiển các thành phần tác động bằng cách set hay reset các ngõ ra ở bảng bộ đệm ngõ ra.

1.5 Cấu trúc của một dự án SIMATIC

Khi bạn đặt cấu hình một hệ thống tự động với STEP 7, bạn tượng trưng những đối tượng “thực” với các đối tượng “logic” trong một dự án của SIMATIC. Thứ tự ưu tiên của đối tượng đã được đưa vào chuẩn trong công việc cấu hình thực cho phần cứng, nghĩa là dự án chứa một hay nhiều module PLC (*các trạm*) mỗi trạm chứa một CPU, và mỗi CPU này chứa một chương trình của người dùng được tạo ở bảng ký hiệu, các tập tin nguồn và mã biên dịch (*các khối*). Màn hình dưới đây (hình 1.5) là cửa sổ của một dự án trong phần mềm quản lý SIMATIC. Về phía bên trái bạn có thể thấy cấu trúc của dự án bằng các thư mục của nó. Các thư mục này phải chứa những thư mục khác và những đối tượng ở về phía bên phải. Một trạm, là một thư mục mà nó có thể chứa những thư mục khác hay các đối tượng. Nếu nhấp đúp vào đối tượng ở khung màn hình bên phải, thì công cụ để soạn thảo đối tượng đó sẽ mở. Ví dụ, nhấp đúp vào một hệ thống mạng để bắt đầu định cấu hình cho hệ thống mạng, nhấp đúp vào một module (ở trong thư mục các khối) thì bắt đầu soạn thảo chương trình.

2 Bộ điều khiển SIMATIC

Bộ điều khiển logic khả trình kiểu SIMATIC là cốt lõi của một hệ thống tự động. Nó giúp ta kiểm soát các sản phẩm máy móc, các cơ sở sản xuất hay các quy trình công nghiệp. Chúng ta phân biệt các “họ” điều khiển khác nhau theo phạm vi thực hiện và lĩnh vực áp dụng :

- Các bộ điều khiển thuộc họ **SIMATIC S7** là quả tim của phần cứng SIMATIC. Chúng có 3 phiên bản :
 - SIMATIC S7-200, hệ PLC vi nhỏ và gọn.
 - SIMATIC S7-300, hệ PLC xếp theo module và có tính năng thực hiện trung bình.
 - SIMATIC S7-400, hệ PLC xếp theo module có tính năng thực hiện cao cấp.

Trạm S7-200 bao gồm một module cơ bản và có thể mở rộng với nhiều module thêm vào. Trong các trạm S7-300/400, module nguồn cung cấp, module CPU và các module xuất nhập được lắp đặt trên cùng một giá. Cấu hình có tính chất trung tâm này có thể được mở rộng thêm với các giá mở rộng để cài đặt thêm các module ngõ vào và ra. Các module mở rộng có thể được lắp đặt ở xa, nghĩa là nó có thể được đặt ở một khoảng cách xa với giá trung tâm. Đối với SIMATIC S7-200 ta sử dụng ngôn ngữ lập trình Step7 Micro/win. Phần mềm STEP 7 với các ngôn ngữ lập trình khác nhau được cung cấp cho các thiết bị điều khiển của các hệ SIMATIC S7-300/400.

- **SIMATIC C7** là một chuỗi của các đơn vị hoàn chỉnh được kết hợp vào các module điều khiển của S7-300 và một bảng điều khiển.

SIMATIC C7 là một trạm với những ngõ vào và ra được tích hợp. Chúng có thể được mở rộng bằng cách xếp đặt ở phần trung tâm hoặc bằng cách thêm các module vào/ra trong hệ S7-300. Bạn dùng công cụ lập trình tiêu chuẩn Step7 để soạn chương trình cho các hệ điều khiển của SIMATIC C7.

- **SIMATIC M7** là những hệ thống máy tính được thiết kế cho hệ S7-300 hay S7-400, các module của các chuỗi SIMATIC M7 là CPU và các module chức năng FM có thể xem như tương thích với hệ máy vi tính đời AT. Bạn lập trình cho các module này với ngôn ngữ lập trình cấp cao như là C++ hay là Visual Basic. Việc nối vào các

quy trình công nghệ hay các dự án được thực hiện bằng các module của S7-300/400 hay là các module mở rộng M7.

- **SIMATIC WinAC** là một gói chương trình có thể chạy với các máy vi tính tiêu chuẩn bằng cách sử dụng Window NT. Hệ điều khiển có thể đặt vào trong phần mềm một cách hoàn hảo và còn gọi là Soft-PLC. Và cũng cần có một card để kết nối với các dự án hoặc các quá trình người ta dùng hệ PROFIBUS-DP.
- **SIMATIC DP** là các module đã được cài đặt cùng với các thiết bị máy móc liên hệ và nối vào trạm chủ của hệ thống mạng PROFIBUS-DP. Có nhiều CPU của SIMATIC cung cấp một chế độ giao tiếp mà có thể nối một cách dễ dàng với các cổng phân phối vào ra. Như thế bạn có thể dùng các giao tiếp theo tiêu chuẩn này để nối với các thiết bị thuộc thể hệ thành phần thứ 3 vào hệ điều khiển SIMATIC.

Các trạm SIMATIC truyền thông với nhau qua các hệ thống mạng. *SIMATIC NET* nhận biết MPI, PROFIBUS và mạng Industrial Ethernet là các mạng phụ thuộc. Bạn cũng có thể nối với một thiết bị đơn lẻ, chẳng hạn như máy in bằng kết nối điểm – điểm. Nhân viên điều hành kiểm soát và giám sát các thiết bị trong quy trình của hệ SIMATIC HMI cũng nối được vào các mạng nội bộ này.

2.1 Các thành phần của trạm SIMATIC

Một PLC hoàn chỉnh gồm có tất cả các module xuất/nhập được gọi là một “trạm”. Trạm S7-300 hay S7-400 gồm có ít nhất là một giá lắp với nguồn cung cấp và module CPU. Các module xuất/nhập cho phép liên kết với máy móc hay thiết bị. Các thành phần của một trạm SIMATIC được mô tả như dưới đây :

- **Giá lắp ráp.**

Kết nối các module riêng lẻ lại với nhau, S7-300 sử dụng các thanh ray đơn giản với độ dài của nó được xác định bằng số các module. S7-400 có giá bằng nhôm với độ dài cố định và các bus nối dữ liệu nằm ở phía sau để nối các khối lại với nhau.

Nguồn cung cấp (PS) cung cấp nguồn điện áp cho các module của PLC; điện áp ngõ vào có thể ở các mức 120VAC, 230VAC hay 24VDC.

- **Bộ xử lý trung tâm (CPU: Central Processing Unit).**

Nơi lưu trữ và xử lý chương trình của người dùng; Nó chỉ định thông số cho các module; duy trì sự giao tiếp với các thiết bị lập trình, các module và các trạm thêm vào qua hệ thống bus MPI. Ngoài ra nó có

thể được trang bị thêm với phần giao tiếp DP để sử dụng làm DP-Master hay DP-slave.

➤ *Module tín hiệu (SM: Signal module)*

Là module làm tương thích giữa tín hiệu từ bên ngoài phù hợp với các mức tín hiệu của module xử lý, hay biến đổi tín hiệu từ các module xử lý thành tín hiệu phù hợp với các đối tượng điều khiển (ví

dụ: relay, van tuyến tính...). Các module tín hiệu có hai loại: vào/ra số (DI: Digital Input, DO: Digital Output) và vào/ra analog (AI: Analog Input, AO: Analog Output), đồng thời nó cũng có thể được sử dụng để nối các cảm biến và các phần tử tác động đã đặt ở các vùng nguy hiểm như vùng một và hai. (Xem các tài liệu về *safety bus* hay *safety PLC* để nắm thêm thông tin)

➤ *Module chức năng (FM: Function module).*

Duy trì các qui trình phức tạp hay các qui trình có mang tính chất thời gian một cách độc lập đối với các module CPU ví dụ: đếm, điều khiển vị trí và điều khiển vòng kín.

➤ *Xử lý truyền thông (CP: Communication Processor).*

Nối trạm SIMATIC với các mạng nội bộ, chẳng hạn như mạng Industrial Ethernet, mạng PROFIBUS-FMS, mạng AS-Interface hay Point-to-Point...

Thành phần lắp ở đằng sau giá được chia thành 2 phần: P-Bus (Bus ngoại vi) cho việc trao đổi nhanh các tín hiệu ngõ vào và ngõ ra, và K-BUS (Bus truyền thông) để trao đổi một số lượng lớn dữ liệu. Việc kết nối giữa các bus truyền thông và CPU hay các mạch giao tiếp điểm-điểm khiến cho các module chức năng và module xử lý truyền thông với các thành phần giao tiếp truyền thông có thể truyền tải tín hiệu đến hệ thống bus MPI. Do đó người ta có thể lập trình các module này thông qua các giao tiếp của CPU bằng thiết bị lập trình.

Các module xuất/nhập phân tán được nối vào mạng SIMATIC là một phần của mạng. Một trạm SIMATIC có thể có nhiều thành phần DP để truyền thông với nhau, việc truyền thông này qua một hoặc nhiều mạng PROFIBUS.

2.2 Micro PLC SIMATIC S7-200

SIMATIC S7-200 là hệ PLC siêu nhỏ, gọn dùng để thay thế cho các relay và các hệ thống điều khiển bằng contactor. SIMATIC S7-200 có thể đảm đương nhiệm vụ một mình hay là làm việc trong một hệ thống với các thiết bị kiểm soát khác. Có rất nhiều module được mở rộng để nối với máy móc hay với một hệ thống khác. Phần mềm cho nó là Step7 Micro.

2.2.1 Các module cơ bản kiểu gọn nhẹ và các module mở rộng.

Có nhiều loại module cơ bản khác nhau với các chức năng khác nhau. Các module cơ bản gồm: CPU, các ngõ vào và ngõ ra sử dụng

điện áp 24VDC hay 100VAC/ 120VAC cho đến 230VAC và relay được lắp ráp cực gọn. Số lượng và kiểu ngõ vào, ngõ ra tùy thuộc vào loại CPU được dùng. Phần các ngõ vào cho bộ đếm tốc độ cao tăng cường tính thời gian thực cho các module. Các chiết áp các điện trở được ghép tích hợp cho phép điều chỉnh các giá trị số đơn giản với một cây vặn vít, không cần lập trình.

Bạn có thể gia tăng số lượng ngõ vào và ngõ ra bằng cách thêm vào các module mở rộng. Có đầy đủ các module ngõ vào ngõ ra của hai loại digital và analog. Các module mở rộng được gá vừa vặn lên các thanh ray (theo tiêu chuẩn DIN) sát bên module cơ bản và được nối với nhau bằng các bus nối.

2.2.2 Điều khiển hoạt động và giám sát quá trình với TD200.

Màn hình hiển thị văn bản TD200 được thiết kế đặc biệt để dùng với bộ điều khiển lập trình S7-200. Nó có 8 chức năng mẫu chốt có thể lập trình được nhằm thực hiện những nhiệm vụ điều khiển. Màn hình LCD với 2 dòng mỗi dòng 20 ký tự để hiển thị thông điệp có thể chấp nhận dung nạp đến 4 biến. Sử dụng giao tiếp PPI để kết nối vào CPU.

2.2.3 Khả năng truyền thông giống như một trạm “lớn” .

Tất cả các module xử lý trung tâm của S7-200 được trang bị cổng giao tiếp PPI (Point to Point Interface). Đây là cổng giao tiếp để nạp chương trình vào CPU. Ngoài ra giao tiếp PPI cũng cho phép liên kết nhiều module xử lý trung tâm S7-200 (có thể đến 31 CPU) và nối với thiết bị lập trình. Dữ liệu trao đổi trên cơ sở nguyên tắc chủ tớ. Tốc độ truyền thông từ 9,6kbit/s đến 19,2 Kbit/s.

Giao tiếp PPI cũng có thể dùng trong các giao thức truyền thông hướng đối tượng là bit mà bạn có thể tự tạo riêng bằng cách sử dụng các chức năng truyền thông có thể lập trình được của chương trình người dùng. Theo phương pháp này người ta có thể nối bất kỳ đầu nối nào có giao tiếp theo kiểu RS 232. Thí dụ nối với một thiết bị đọc mã vạch sử dụng chuẩn ASCII.

Nhờ tính chất tích hợp trong giao tiếp, một vài bộ xử lý trung tâm S7-200 cũng có thể được nối vào hệ thống mạng MPI để sử dụng như là thiết bị phục vụ, do đó việc trao đổi dữ liệu với các trạm SIMATIC khác có thể thực hiện được.

Khối truyền thông CP242-2 cho phép nối đến 31 module phân tán (mạng Asi) với một trạm S7-200 và như thế có thể làm gia tăng số lượng ngõ vào và ngõ ra. Để thực hiện giao tiếp với các hệ thống khác

qua mạng Profibus DP ta sử dụng module CP248-2 (đóng vai trò là DP slave).

2.3 Bộ điều khiển SIMATIC S7-300.

2.3.1 Giá trung tâm

Giá trung tâm của module điều khiển lập trình S7-300 chứa CPU và 8 module ngoại vi I/O, module chức năng, và module truyền thông. CPU cần nguồn nuôi 24 VDC. Bạn có thể lắp đặt nguồn cung cấp này trên các thanh ray theo chuẩn DIN ở bên trái CPU. Các module có thể kết nối với nhau bằng các bus I/O và bus truyền thông. Các bus liên kết các module. Điều này có nghĩa, chiều dài của bus luôn phụ thuộc vào số lượng module, và không bao giờ vượt quá module sau cùng.

Các rãnh của module được đánh số thứ tự liên tiếp nhau: rãnh 1 dành cho nguồn điện (cho dù không được lắp đặt cũng để trống); rãnh 2 cho CPU; rãnh 3 cho các module giao tiếp IM. Rãnh từ 4 đến 11 dành cho module I/O, các module chức năng, và các module truyền thông lắp đặt kế tiếp nhau.

2.3.2 Giá mở rộng

Nếu lắp đặt ở một giá đỡ không thỏa mãn yêu cầu, từ CPU 314 trở đi bạn có thể:

- Lắp đặt thành 2 giá đỡ (với IM 365, cho khoảng cách tối đa giữa các giá là 1 m).
- Hoặc sắp đặt tối đa thành 4 giá đỡ (với IM 360 và IM 361, cho khoảng cách tối đa giữa các giá là 10 m).

Các module giao tiếp IM được lắp đặt giữa CPU và module I/O đầu tiên. Giống như giá trung tâm, giá mở rộng là những thanh ray theo chuẩn DIN. Module IM để liên lạc với giá trung tâm, được đặt ở rãnh 3. Module IM để truyền thông CPU với các module khác được

đặt ở rãnh 3. Các rãnh từ 4 đến 11 dành riêng cho các module khác lắp đặt kế tiếp nhau.



2.3.3 Các phân đoạn bus nội bộ của S7-300.

Module ứng dụng FM 356 của họ M7-300 của hệ vi tính tự động có cách lắp đặt đặc biệt. Module FM 356 "phân chia" bus nằm ở mặt sau của giá và kiểm soát các module còn lại vừa lắp đặt sau để tạo "phân đoạn bus nội bộ". Nhưng trong cách sắp xếp này, không được phép nhiều hơn tám module I/O, kể cả module FM 356, được lắp trên một giá.

2.3.4 SIMATIC lắp đặt bên ngoài.

SIMATIC S7-300 cung cấp nhiều module đặc biệt để sử dụng trong môi trường có nhiệt độ cao. Những module này hỗ trợ dải nhiệt độ từ -25°C đến $+60^{\circ}\text{C}$., chịu được rung động theo chuẩn IEC 68 phần 2-6. Nó chịu được ẩm độ, và giá lạnh theo IAC 721-3-3, lớp 3 K5, và phù hợp với yêu cầu lắp đặt của xe tải (có chứng nhận của ngành đường sắt theo chuẩn EN 50155). Còn tất cả các đặc tính khác tương tự như các thiết bị tiêu chuẩn.

2.4 SIMATIC S7-400 cho các nhiệm vụ điều khiển có yêu cầu cao

2.4.1 Giá trung tâm

Các giá ở S7-400 có thể là giá trung tâm (CR, central rack), giá mở rộng (ER, extention rack) hay cho cả hai (UR, universal rack). Chúng gồm thanh nhôm đã được định hình, mỗi thanh có chiều dài cố

định và được cung cấp với bus nằm ở mặt sau giá và các đầu cắm bus đầy đủ.

Giá trung tâm ở S7-400 có 18 hay 19 chỗ cắm(UR1, UR2) với bề rộng lắp vào cố định, ở đây cũng có đặt các chỗ cắm cho nguồn cung cấp và CPU, cũng có thể có hai hoặc nhiều hơn một module cho nguồn và CPU. Thông thường sự sắp xếp các các module được sắp xếp từ bên trái qua, đầu tiên là nguồn cung cấp, sau đó là CPU và các module ngoại vi. Bên phải của giá cắm các module kết nối cho các giá mở rộng. Bus nằm ở mặt sau giá bao gồm P-Bus song song và K-Bus nối tiếp, nối tất cả các chỗ cắm lại với nhau.

Với giá CR2, bạn có thể thực hiện tách hai CPU trong một giá với nguồn cung cấp chung. Cả hai CPU có thể trao đổi dữ liệu với nhau qua K-Bus, tuy nhiên P-Bus thì tách riêng cho các module ngoại vi riêng của chúng. Phần bên trái có 10 chỗ cắm, bên phải 8.

Giá UR2-H bao gồm 2 phần với 9 chỗ cắm. Mặc dù vậy nó cũng có thể được đặt vào các giá trung tâm và giá mở rộng ở các trạm S7-400 tiêu chuẩn hay các trạm S7-400H có khả năng dự phòng. Mỗi phần cần một nguồn cung cấp; P-Bus và K-Bus thì được tách riêng.

Giá mở rộng

Nếu chỗ cho các module ngoại vi ở giá trung tâm không đủ hay muốn lắp thêm các module ở cách xa, thì bạn đưa vào trạm một hoặc nhiều giá mở rộng. Có thể nối đến cực đại là 21 giá mở rộng với một giá trung tâm. Để phân biệt số của các giá bạn điều chỉnh công tắc địa chỉ ở module IM nhận (IM receive). IM nhận chỉ được phép gắn ở tận cùng bên phải của giá mở rộng.



Với việc nối IM 460-1 và IM 461-1, giá mở rộng có thể được sắp xếp cách xa giá trung tâm đến 1,5m, với nguồn cung cấp được truyền là 5V. Cũng như vậy trong phạm vi gần đến 3m có thể hoạt động được đến 4 giá mở rộng qua IM 360-0 và 361-0. Ở phạm vi xa, với các module IM 360-3 và IM 361-3 có thể kết nối đến 4 giá mở rộng để điều khiển ở xa đến 100m.

Giá mở rộng ER1 và ER2 với 18 cũng như 9 chỗ cắm cho các module tín hiệu “đơn giản” mà không bị cắt bởi các ngắt xử lý, không phải được cung cấp nguồn DC 24 VÀ qua P-Bus, không cần thiết điện áp backup và không nối với K-Bus. K-Bus thì tồn tại ở giá UR1 và UR2, nếu chúng được sử dụng hoặc là giá trung tâm hoặc là giá mở rộng với số từ 1 đến 6.

Kết nối từ các module SIMATIC S5

Với module giao tiếp IM 463-2, bạn có thể nối thiết bị mở rộng SIMATIC S5 (EG 183U, EG 185U, EG 186U cũng như ER 701-2 và ER 701-3) với một trạm S7-400 và thiết bị mở rộng này một lần nữa. Ở thiết bị mở rộng S5, module giao tiếp IM 314 đảm nhiệm kết nối. Bạn có thể thực hiện tất cả các module analog và digital cho phép ở thiết bị mở rộng đã cho. Ở một trạm S7-400 có thể nối đến cực đại 16 thiết bị mở rộng S5 phân tán.

Chế độ làm việc tăng cường với chế độ đa xử lý

Bạn có thể chuyển đổi trạm S7-400 thành một bộ điều khiển đa xử lý bằng cách chèn nhiều CPU vào (tối đa là 4). CPU được chọn xác định cho kiểu hoạt động này cũng được gọi là “đa xử lý”. Ngay khi bạn cắm nhiều hơn một CPU, PLC sẽ tự động đặt ở chế độ hoạt động đa xử lý. Tất cả các CPU có cùng chế độ hoạt động. Điều này có nghĩa chúng được khởi động cùng với nhau và tất cả chúng cũng đi vào trạng thái STOP nếu một CPU bị hỏng. Mỗi CPU xử lý chương trình người dùng riêng của nó độc lập với các CPU khác.

Mỗi module I/O được gán một bộ xử lý xác định. Bao gồm địa chỉ của nó và các ngắt. Tất cả các module I/O được đặt trong cùng phần P-Bus. Điều này có nghĩa bạn phải cho chúng các địa chỉ khác nhau mặc dù chúng thuộc các CPU khác nhau. Điều này cũng tương tự cho các ngoại vi phân tán. Mặc dù mỗi CPU có thể hoạt động với một hoặc nhiều hệ DP master độc lập với các CPU khác, nhưng mỗi module phân tán và trung tâm phải có địa chỉ độc nhất của nó (địa chỉ P-Bus) trong toàn bộ hệ thống.

2.5 Khả năng dự phòng của SIMATIC

Đối với các quá trình chậm: Phần mềm redundance với các phần tử tiêu chuẩn

Với các phần tử chuẩn SIMATIC S7-300/400, có thể cho phép bạn tạo ra một hệ thống dự phòng (redundance) dựa vào phần mềm redundance, mà trong đó trạm chủ điều khiển quá trình và trong trường hợp nó bị hỏng thì trạm dự phòng đảm nhận vai trò điều khiển. Khả năng dự phòng thông qua phần mềm redundance thích hợp với các quá trình chậm, vì việc chuyển mạch ở các trạm dự phòng có thể cần nhiều giây tùy theo cấu trúc của thiết bị tự động. Trong thời gian này tín hiệu của quá trình bị “đóng băng”. Sau đó trạm dự phòng tiếp tục làm việc với các dữ liệu sau cùng của trạm chủ.

Cả hai CPU tham gia trao đổi các dữ liệu tức thời qua một mạng con, ở nơi kết nối truyền thông cũng có thể được sử dụng. Một kênh phụ của module vào/ra được thực hiện với ngoại vi phân tán (ET 200M với module giao tiếp IM 153-3 cho PROFIBUS-DP có dạng redundance). Bên cạnh đó cả hai CPU cũng có thể điều khiển cho các ngoại vi không có redundance, cho dù là tập trung hay phân tán.

Trong thiết kế có gói phần mềm “Software Redundant”.

Khả năng dự phòng qua chế độ Hot Stand-By: SIMATIC S7-400H

SIMATIC S7-400H là một hệ tự động hoá có khả năng dự phòng với cấu tạo dự phòng (redundancy) cho lắp thêm từ hai CPU 417-4H với

module đồng bộ cho sự hiệu chuẩn dữ liệu trên cáp quang. Cả hai thiết bị làm việc ở “hot stand-by”: trong trường hợp lỗi, thiết bị nguyên vẹn đảm nhận thông qua chuyển mạch tự động, không có xung đột với sự xử lý của chương trình người dùng.

Hệ thống-H có thể được xây dựng từ hai giá trung tâm riêng biệt UR1 cũng như UR2 hay từ một giá UR2-H. Bên cạnh đó luôn luôn để sẵn nguồn cung cấp và module CPU, bạn có thể nối thiết bị ngoại vi với khả năng sử dụng bình thường hay dự phòng:

1. Dạng một kênh một phía

Module ngoại vi hiện có đơn giản và được nối ở một trong hai hệ thống thành phần: tập trung hay phân tán. Trạng thái tín hiệu cho phép thực hiện qua kết nối bổ sung. Chương trình người dùng trong cả hai hệ thống thì giống nhau. Nếu một hệ thống không thực hiện, thì thiết bị ngoại vi kết nối với chúng không thể thực hiện được nữa.

2. Dạng một kênh chuyển mạch

Module ngoại vi thì tồn tại đơn giản là ngoại vi phân tán ET 200M. Cấu tạo ngoại vi phải là đối xứng; Mỗi hệ thống thành phần có sự truy xuất ngoại vi qua module giao tiếp IM 153-3 với giao tiếp DP redundance. Hệ -H chỉ sử dụng tức thời một giao tiếp; Qua kết nối redundance cho phép thực hiện trạng thái tín hiệu ở cả hai hệ thống thành phần. Chương trình người dùng thì giống nhau ở cả hai hệ thống thành phần.

3. Dạng một phía redundance

Module ngoại vi được cắm đôi ở trong hai hệ thống bus, trong một giá trung tâm. Chức năng bổ sung được thực hiện trong chương trình người dùng.

4. Dạng chuyển mạch redundance

Module I/O được cắm đôi ở hai trạm ET 200M với bus nằm ở mặt sau giá tích cực. Module giao tiếp bổ sung IM 153-3 nối các trạm ET 200M đến cả hai hệ thống con. Chức năng redundance được thực hiện trong chương trình người dùng.

Bạn cũng có thể thực hiện một sự trao đổi redundance dữ liệu bằng cách lập lại trạng thái trước đó qua Industrial Ethernet hay mạng PROFIBUS. Ngoài ra để kết nối tích cực, bạn có thể lắp đặt đến 3 kết nối redundance mà hệ điều hành có thể sử dụng nếu kết nối cơ bản bị hỏng. Kết nối có khả năng dự phòng được lập trình như các kết nối chuẩn.

Hãy sử dụng gói phần mềm “S7-400H để cài đặt hệ thống redundance.



2.6 SIMATIC C7

SIMATIC C7 là hệ thống hoàn chỉnh nó đáp ứng cả những tiêu chuẩn cao nhất của yêu cầu điều khiển. Chúng được coi như là hệ thống điều khiển máy móc hoàn chỉnh, kết hợp với bảng điều khiển và bộ lập trình SIMATIC S7-300 có các ngõ vào/ra tích hợp trong một khối. Phía trước của bảng điều khiển có cấp bảo vệ IP 65 và phía sau có cấp bảo vệ IP 20. Khi cần các trạm SIMATIC C7 dễ dàng được mở rộng bằng các module I/O. Cũng giống như tất cả các trạm SIMATIC,

SIMATIC C7 cũng được trang bị cổng giao tiếp MPI để kết nối với thiết bị lập trình và trao đổi dữ liệu với các trạm khác.

Bảng điều khiển và bộ điều khiển lập trình

Phụ thuộc vào loại trạm C7 sử dụng, bảng điều khiển (OP) gồm một hiển thị LCD hiển thị 2 hàng x 40 ký tự, 4 hàng x 20 ký tự, 8 hàng x 40 ký tự. Trạm C7 626 có một vùng hiển thị với 320 x 240 chấm hiển thị. Tất cả đều có một bàn phím dạng màng với 14 phím mềm và 10 phím chức năng. Nếu bạn muốn tổ hợp trọn bộ một trạm SIMATIC C7 vào trong một trang thiết bị đã cài đặt, bạn có thể xác định yêu cầu của mình, xem xét tới logo, màu sắc và nhãn của công ty.

Hệ PLC tương thích với bộ vi xử lý trung tâm S7 - 300. Tùy thuộc vào dạng hệ thống C7, vùng nhớ làm việc cho chương trình S7-300 có dung lượng từ 32 KB tới 128 KB và vùng nhớ chương trình từ 48KB đến 8MB.

Mở rộng trạm SIMATIC

C7-621 / C7 – 621 Asi là các trạm nhỏ nhất của hệ thống SIMATIC C7 và có thể mở rộng tới 4 module S7 – 300 sử dụng module giao tiếp IM 621. Các trạm S7 khác có thể mở rộng tới 3 giá đỡ (rack) module S7 – 300, mỗi giá đỡ chứa 8 module. Module IM 360 dùng cho giao tiếp đã được tích hợp sẵn.

Module Input /Output C7 cho phép tiết kiệm khoảng trống khi mở rộng hệ thống. Nó được bọc kim loại, có thể lắp đặt phía sau trạm S7 và được nối bằng cáp bus (không dùng cho C7-621 / C7-621 Asi). Module mở rộng này cung cấp 16 ngõ vào Digital, 16 ngõ ra Digital, 4 ngõ vào analog, 4 ngõ ra analog và 4 ngõ vào vạn năng (tùy chọn).

Các khả năng truyền thông

Các module C7-621 ASI có một giao tiếp AS chủ (AS interface Master) cho phép kết nối tới 31 giao tiếp AS tớ (AS interface Slaves). Các module C7-633/634/626 có một giao tiếp PROFIBUS DP. Tất cả các trạm C7 đều được trang bị giao tiếp MPI.

Bạn có thể mở rộng các hệ thống C7 bằng các module S7-300, cho phép sử dụng C7 với module truyền thông CP. Bằng cách này bạn có thể kết nối hệ thống tới Industrial Ethernet hay mạng PROFIBUS, AS-i và point to point. SIMATIC C7 đem lại các dịch vụ truyền thông dữ liệu toàn cục (Global Data Communication). Truyền thông SFC nhờ giao tiếp MPI và truyền thông SFB. Các phần mềm có thể nạp vào trong hệ thống cho phép kết nối tới các trạm S5 và hệ thống thứ 3. Việc truyền dữ liệu có thể theo chu kỳ hay theo sự kiện điều khiển.

Tạo cấu hình, lập trình với STEP 7, ProTool và WinCC Flexible

Bạn tạo cấu hình một trạm SIMATIC với phần mềm STEP 7 hay STEP 7 Mini. Trong *SIMATIC Manager* tạo một project hoặc mở một *Project* định nghĩa cho *SIMATIC 300 Station*, sau đó nhấp đúp chuột vào *Hardware* để khởi động công cụ cấu hình phần cứng *HW Config*. Trong *Hardware catalog* chọn 1 trạm C7 và kéo thả vào cửa sổ cấu hình trạm. Ở đây hiển thị 1 giá đỡ (rack) module với CPU và các ngõ vào/ra. Bạn có thể mở rộng trạm bằng các module input/output từ họ S7, thêm vào các giá đỡ mở rộng hoặc sử dụng giao tiếp DP nếu có để kết nối CPU tới mạng PROFIBUS.

Đối với S7-300, bạn lập trình theo một trong các ngôn ngữ STEP 7 chuẩn như LAD (Ladder Logic), FBD (Function Block Diagram) hay STL (Statement List).

Sử dụng công cụ ProTool hoặc WinCC Flexible để tạo cấu hình bảng điều khiển, cấu hình phần mềm cho các thiết bị HMI. Trong

SIMATIC Manager chọn một *SIMATIC OP* với *S7 Project*, chọn trạm C7 tương ứng trong ProTool và chọn hiển thị. Sau đó bạn nạp cấu hình này vào trạm C7.

Bạn có thể sử dụng mật khẩu với nhiều cấp bảo vệ, cho phép các quyền khác nhau đối với người vận hành. Bằng cách này bạn có thể cho phép hay không cho phép quyền truy cập, phụ thuộc vào người vận hành nào được truy cập. Password có thể được thay đổi trong mục *password Proceeing*.

2.7 SIMATIC M7

Họ SIMATIC M7 bao gồm các module máy tính tương thích AT của SIMATIC S7 300/400. Các module này mở rộng phạm vi chức năng của các bộ điều khiển SIMATIC bằng một phần mềm mở rộng để tổ hợp các sản phẩm phần mềm chuẩn. SIMATIC M7 hiện tại có thể hoạt động như một CPU hay một module chức năng (FM). Nó có thể được lắp đặt trong trạm SIMATIC S7 để thay thế S7 CPU hay hoạt động như một module M7-FM để tăng cường tính năng bằng các module máy tính mở rộng. Tất cả các module M7 đều có giao tiếp MPI để kết nối trực tiếp đến các trạm SIMATIC khác.

SIMATIC M7 nâng cao hiệu suất máy tính

Một trạm M7-300 là một trạm S7-300, trong đó các CPU S7-300 được thay thế bằng M7-CPU. Một M7-CPU hoạt động tương tự như CPU họ S7, điều khiển các module S7-300 lắp đặt tối đa trên 4 giá đỡ, mỗi giá đỡ có 8 module. Một module chức năng M7-300 có thể “chia” bus và quản lý việc điều khiển các module S7-300 theo các bus cục bộ. Tuy nhiên không được lắp quá 8 module trên mỗi rack.

Một số đặc tính kỹ thuật của M7-300: vi xử lý 80486 DX2, tần số xung đồng hồ 50 MHz, bộ nhớ chính 8 MB, bộ nhớ chương trình hay dữ liệu ở dạng card nhớ tới 16MB.

Trạm M7-400 của M7-CPU có thể truy xuất trực tiếp tới tất cả các module S7-400. Tuy nhiên không thể thực hiện đa xử lý bằng các bộ xử lý trung tâm S7 hoặc M7. Nếu bạn sử dụng các máy tính lớn trong trạm S7-400 của mình, thì bạn nên sử dụng module chức năng M7-400, xin nhớ rằng công suất tiêu tán lớn cần phải lắp thêm quạt giải nhiệt.

Một số đặc tính kỹ thuật của M7-400: vi xử lý pentium 75 hoặc 120MHz, bộ nhớ chính 16 MB, bộ nhớ chương trình hay dữ liệu ở dạng card nhớ tới 16 MB. Bộ xử lý M7-488 có thể quản lý card nhớ EPROM MEM 478, ổ đĩa cứng có thể boot.

Ổ đĩa cứng và mềm của S7-400

Các module mở rộng chỉ có thể được sử dụng để kết nối với bộ xử lý trung tâm M7 hay module chức năng. Bạn lắp các module này vào bên phải của bộ xử lý trung tâm M7 hay module chức năng.

Các module mở rộng M7 có các module giao tiếp phụ cho phép kết nối tới các thiết bị như: màn hình VGA, bàn phím, chuột và máy in. Các module giao tiếp cho PROFIBUS và Industrial Ethernet kết nối hệ thống với mạng truyền thông dữ liệu. Họ M7 có các module ngoại vi với các ngõ vào/ra digital, analog làm tăng phạm vi chức năng của trạm S7.

Module lưu trữ chính của M7 gồm một đĩa cứng 520 MB có thời gian truy xuất 14 ms và một ổ đĩa mềm 3.5". Bộ chuyển đổi module AT cho M7-400 có thể kèm theo một module PC có chiều dài cực đại là 164 mm.

Tạo cấu hình bằng SIMATIC Manager

Tạo cấu hình cho một trạm M7 tương tự như cho một trạm S7. Trong SIMATIC Manager tạo *SIMATIC 300 Station* hoặc *SIMATIC 400 Station*. Nhấp đúp chuột vào *Hardware* để khởi động công cụ cấu hình phần cứng *HW Config*, chọn *rack* từ *catalog module*. Bây giờ bạn có thể cài đặt một M7 CPU thay vì một S7 CPU, hoặc sử dụng một module chức năng kết nối với S7 CPU.

SIMATIC Manager tự động tạo ra một thư mục M7 với một bảng ký hiệu và một thư mục con *Blocks*. Nhập thông số mong muốn của trạm S7 cho bảng ký hiệu. M7-ProC/C++ chuyển đổi các ký hiệu thành các biến C để các tín hiệu xử lý trong chương trình C được xác định bằng tên ký hiệu.

Nếu bạn đã cài đặt phần mềm CFC, bạn sẽ thấy thư mục *Drawing* với lược đồ CFC, bạn có thể chèn nó bên dưới thư mục M7.

Phần mềm cho SIMATIC M7

SIMATIC M7 sử dụng hệ điều hành M7-RMOS32 real-time multiasking. Multiasking có nghĩa bạn có thể chia nhiệm vụ điều khiển ra làm nhiều nhiệm vụ con và các nhiệm vụ con này được xử lý theo các cấp ưu tiên khác nhau. Giao tiếp M7-API (giao tiếp lập trình ứng dụng) tích hợp hệ điều hành RMOS32 trong môi trường SIMATIC.

Hệ điều hành có thể sử dụng MS-DOS 6.22 hoặc không sử dụng. Nếu sử dụng thì MS-DOS hoạt động như một "nhiệm vụ" dưới quyền RMOS32 điều khiển màn hình, bàn phím, các ổ đĩa và cung cấp các tập tin hệ thống có thể truy cập được bằng hệ điều hành RMOS32. MS-DOS không được phép truy xuất các tập tin khi các nhiệm vụ real-time đang được thực hiện. Nếu không sử dụng MS-DOS thì phần cứng

M7 chỉ được điều khiển bằng RMOS32, RMOS32 sử dụng hệ thống quản lý tập tin HSFS (High Speed File System) cho tất cả các ổ đĩa.



2.8 Qui trình kết nối với các module digital

Các module Digital là các bộ chuyển đổi các tín hiệu nhị phân. CPU của trạm SIMATIC nhận các thông tin về chế độ hoạt động của quá trình xử lý nhờ các module ngõ vào Digital, các giao tiếp trong qui trình xử lý nhờ các module ngõ ra Digital. Các tín hiệu số giữa backplane bus và việc xử lý được cách ly bằng các cảm biến quang điện (Optocoupler).

Các module Digital có 1, 2 hay 4 byte tương ứng với 8, 16 hay 32 tín hiệu. Các module digital thường được đặt địa chỉ trong vùng nhớ đệm để trạng thái tín hiệu có thể được xử lý ở dạng bit. Các module Digital tiên tiến có khả năng cung cấp các thông tin chẩn đoán về trạng thái tín hiệu.

Các module ngõ vào Digital

Các module ngõ vào digital chuyển đổi tín hiệu điện áp bên ngoài 24VDC hay 120/230VAC sang tín hiệu số. Để module hoạt động chuẩn xác thì cảm biến kết nối phải được chọn có điện áp nằm trong phạm vi cho phép và phải có khả năng cung cấp dòng điện ngõ vào khi trạng

thái tín hiệu ngõ vào là “1”. Các tín hiệu ngõ vào phải được lọc. Điều này có nghĩa là cần triệt nhiễu trên đường dây cấp nguồn và làm giảm xung điện áp lan truyền. Quá trình lọc làm trì hoãn các tín hiệu ngõ vào. Đối với các module digital có ngõ vào kích xử lý ngắt thì có thể làm giảm thời gian trì hoãn (delay) tín hiệu ngõ vào. Tuy nhiên khi làm giảm thời gian trì hoãn thì đồng thời cũng làm giảm sự chống nhiễu. Nên luôn có sự suy tính giữa độ chống nhiễu cao (thời gian trì hoãn dài) với việc nhận tín hiệu nhanh (thời gian trì hoãn ngắn).

Các module ngõ ra digital

Để có thể giao tiếp được trong qui trình xử lý, CPU cần đến các bộ chuyển đổi bên trong thành các mức tín hiệu điện áp và dòng điện sử dụng cho hệ thống hoặc xử lý. Các module ngõ ra digital có một bộ nhớ dữ liệu lưu trữ dữ liệu nhận được và sau đó chuyển các dữ liệu này tới một bộ khuếch đại. Bộ khuếch đại cung cấp công suất chuyển mạch yêu cầu. Các bộ khuếch đại điện áp một chiều được trang bị bảo vệ ngắn mạch bằng mạch điện tử, còn các bộ khuếch đại điện áp xoay chiều được bảo vệ ngắn mạch bằng cầu chì.

Khi chọn lựa một module ngõ ra digital cần xem xét tới khả năng chuyển mạch, toàn bộ tải cho phép và dòng điện rò cho phép. Dòng điện rò này tại mức “0” của tín hiệu không được xuống thấp hơn giá trị cho phép, nếu không thì thiết bị sẽ không đáp ứng với tín hiệu Stop.

Trong các chế độ hoạt động STOP, HALT và trong suốt thời gian khởi động thực hiện chương trình một tín hiệu vô hiệu hóa ngõ ra làm vô hiệu hóa tất cả các module ngõ ra digital. Ở trạng thái này chúng không đưa ra điện áp mà chỉ đưa ra một giá trị thay thế xác định hoặc duy trì giá trị sau cùng.

Các module digital có bảo vệ chống nổ

Các module có bảo vệ chống nổ của SIMATIC “là thiết bị điện tích hợp” có bảo vệ “an toàn” (theo chuẩn DIN EN 50020). Chúng được chấp nhận sử dụng để kết nối các thiết bị điện an toàn bên trong ở vùng 1 và 2. Các module này được lắp đặt bên ngoài vùng nguy hiểm. Điện áp tải 24 VDC được dẫn nhờ cáp LK 393. Các module chống nổ được sử dụng trong các trạm S7-M7/300 hay trong các I/O của trạm ET 200M. Sử dụng STEP 7 để thiết lập các module này.

2.9 Qui trình kết nối với các module analog

Các module analog là các bộ chuyển đổi tín hiệu cho xử lý tín hiệu analog. Các module ngõ vào analog chuyển đổi các tín hiệu analog từ quá trình xử lý hay hệ thống thành các tín hiệu digital để CPU của trạm SIMATIC có thể xử lý. Các module ngõ ra analog chuyển đổi tín hiệu

digital từ trạm SIMATIC thành tín hiệu analog để xử lý, chẳng hạn như giá trị điểm đặt của các cơ cấu chấp hành. Mỗi một đại lượng analog như giá trị đo hay giá trị điểm đặt chiếm một “kênh” trên module. Các module analog hiện có 4, 8, hay 16 kênh tương ứng với 8, 16 hay 32 byte. Một giá trị analog được số hoá như một số nguyên 16 bit (dữ liệu dạng INT). Các module analog tiên tiến có khả năng cung cấp thông tin về tình trạng của module hay tính toán giá trị giới hạn.

Các module analog thường được đặt địa chỉ ngoài vùng đệm, đặc biệt khi chúng được đọc hoặc ghi trực tiếp. Đây là trường hợp đặc biệt, chẳng hạn trong các mạch điều khiển vòng kín có chu kỳ xử lý độc lập với chương trình chính.

Các module ngõ vào analog

Các module ngõ vào analog sử dụng phương pháp tích hợp để chuyển đổi các đại lượng nhận được từ quá trình xử lý (điện áp, dòng điện, điện trở) thành các giá trị số. Phụ thuộc vào sự triệt nhiễu điện áp (đối với tần số 400/60/50/10 Hz), việc chuyển đổi cần 2,5/20/20 và 100 miligiây. Phạm vi chuyển đổi rộng (9/12/12/15 bit + dấu). Đặt phạm vi dòng điện và điện áp bằng các công tắc. Đối với các giá trị đặt cần có độ chính xác cao hơn bạn sử dụng công cụ STEP 7 HW Config.

Các module ngõ ra analog

Các module ngõ ra analog chuyển đổi các giá trị digital thành tín hiệu điện áp và dòng điện khác nhau. Các tín hiệu bên trong và bên ngoài được cách ly với nhau về điện. Các giá trị digital nhận được từ CPU được lưu trữ trong vùng nhớ dữ liệu trong module, từ đây chúng được chuyển tới bộ chuyển đổi để chuyển đổi các giá trị này thành các đại lượng analog trong thời gian 0,8/1,5 mili giây và sau đó chúng được chuyển tới địa chỉ ngõ ra.

Các module analog có bảo vệ chống nổ

Các module có bảo vệ chống nổ của SIMATIC “là thiết bị điện tích hợp” có bảo vệ “an toàn bên trong” ([EE x ib] IIC theo chuẩn DIN EN 50020). Chúng được chấp nhận sử dụng để kết nối các thiết bị điện an toàn bên trong ở vùng 1 và 2. Các module này được lắp đặt bên ngoài vùng nguy hiểm. Điện áp tải 24 VDC được dẫn nhờ cáp LK 393. Các module chống nổ được sử dụng trong các trạm S7- M7/300 hay trong phân bố I/O của trạm ET 200M. Sử dụng STEP 7 để thiết lập các module này.

Các modul analog SIMATIC S7-HART I loại bảo vệ “an toàn bên trong” và có thể sử dụng trong I/O của các trạm ET 200M. Các modul

này làm thay công việc của một HART master cho phép 2 thiết bị HART kết nối với nhau. Truyền thông dữ liệu thực hiện nhờ thủ tục HART version 5.4.

2.10 Các module chức năng trợ giúp CPU

Các module chức năng (Function module) là các modul tiền xử lý tín hiệu “thông minh”. Chúng khống chế và xử lý các tín hiệu nhận được từ hệ thống mà không dùng đến tài nguyên của CPU và đưa các thông tin này trở lại quá trình xử lý hay đưa tới các giao tiếp bên trong của CPU. Chúng làm thay các công việc mà thường CPU không thể thực hiện được theo tốc độ yêu cầu như đếm xung, điều khiển vị trí hay điều khiển truyền động và như vậy tài nguyên của CPU được dùng xử lý các công việc điều khiển quan trọng khác.

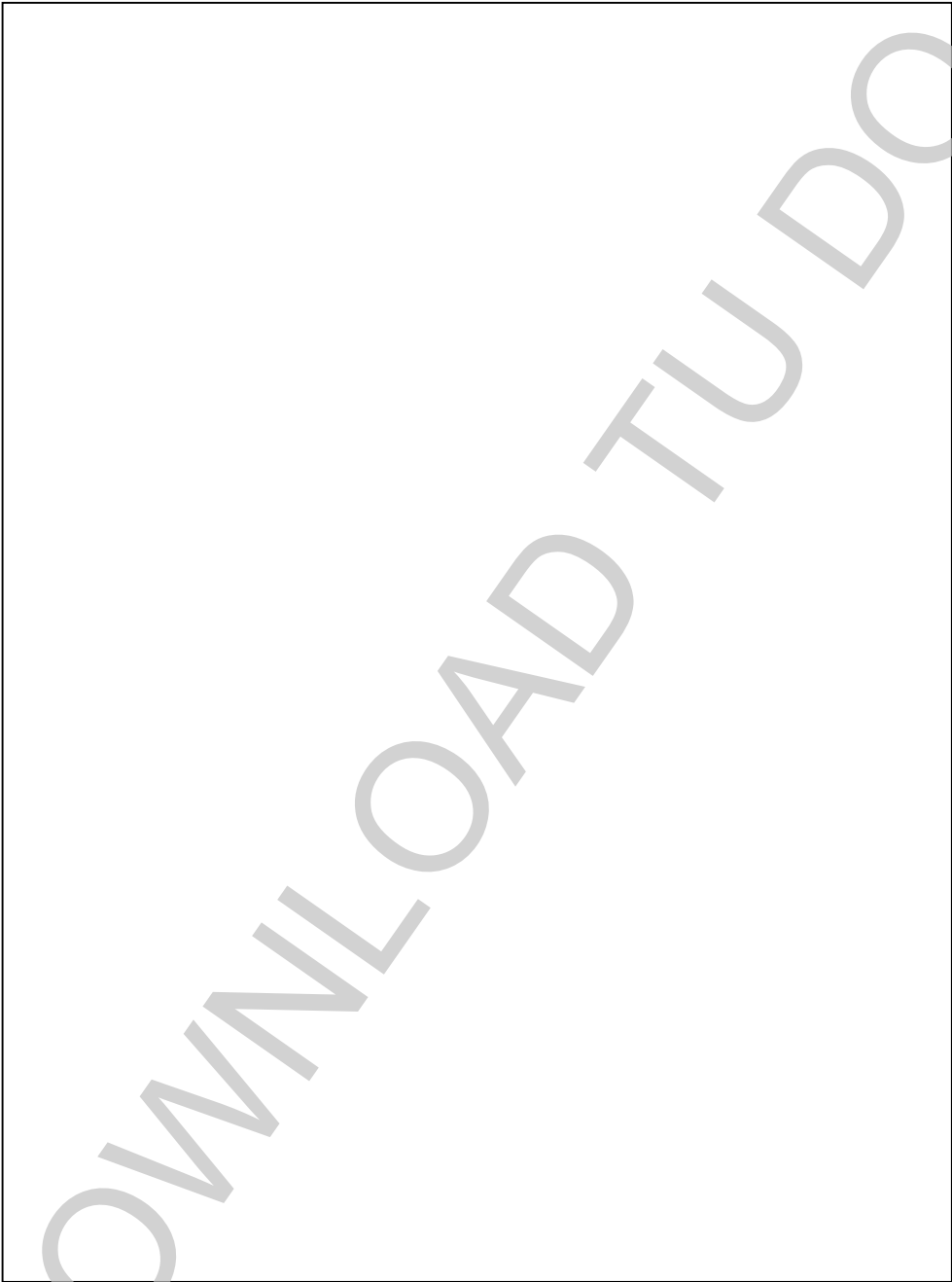
Bạn sử dụng công cụ HW Config của STEP 7 để cài đặt các thông số yêu cầu cho các module chức năng. Các khối chuẩn cần thiết cho chương trình và các đối thoại chuẩn dùng cho cài đặt thông số là các phần của phân phối dữ liệu và được tích hợp tự động trong SIMATIC Manager khi bạn cài đặt các module chức năng, kèm theo các đối thoại chẩn đoán dùng cho chạy thử chương trình.



2.11 Các module CP (Communiation Processor) kết nối tới mạng truyền thông

Các bộ xử lý truyền thông CP thực hiện thay các công việc truyền thông của CPU. Chúng kết nối tới mạng, chịu trách nhiệm thiết lập kết nối và truyền dữ liệu trên mạng, chúng cung cấp các dịch vụ truyền thông cần thiết cho CPU và chương trình người dùng.

Các modul CP dùng cho PROFIBUS và Industrial Ethernet được thiết lập từ phần mềm tùy chọn NCM (một phần của STEP 7) cung cấp các màn hình cài đặt thông số và các màn hình chẩn đoán cho việc kiểm tra và thực hiện chương trình. Tất cả các CPU khác khi cài đặt được trang bị các màn hình cần thiết tích hợp trong “Hardware Configuration”.



2.12 Bộ não SIMATIC–S7: các CPU module

Các module CPU thuộc họ SIMATIC–S7 bao gồm bộ xử lý trung tâm giám sát hệ điều hành chương trình người dùng, cài đặt thông số module, và truyền dữ liệu. Tất cả các module CPU được trang bị giao tiếp MPI để kết nối đến thiết bị lập trình, hệ thống giao tiếp vận hành hay nối tới các trạm SIMATIC khác.

Hệ điều hành điều khiển chế độ hoạt động

Hệ điều hành quản lý tất cả các chức năng của CPU. Nó điều khiển trình tự khởi động, khởi tạo và thực hiện chương trình chính, cập nhật vùng nhớ đệm, ghi nhận thông báo và thông báo lỗi, khởi tạo và xử lý theo các cấp ưu tiên. Nó cũng giám sát tất cả các hoạt động và chỉ báo lỗi bằng cách ghi chúng vào vùng đệm chẩn đoán và kích các LED ở bảng điều khiển sáng lên. Phía trên bảng điều khiển có một công tắc có khoá cho phép bạn khởi động việc thực hiện chương trình của mình.

Công tắc có khoá này có 4 vị trí là: STOP, RUN, RUN-P, và MRES. Vị trí STOP là dừng chương trình người dùng, RUN và RUN-P thực hiện chương trình, khi công tắc ở vị trí RUN chương trình người dùng chỉ có thể được đọc, ở vị trí RUN-P chương trình có thể được đọc và chỉnh sửa. Khi công tắc ở vị trí MRES sẽ reset toàn bộ và reset nguội (cold reset).

Bộ xử lý trung tâm thi hành chương trình người dùng

Bộ xử lý trung tâm của các module CPU S7-300 (ngoại trừ CPU 318) là một bộ vi điều khiển chuyển đổi mã MC7 của chương trình người dùng sang ngôn ngữ máy. Các module CPU 312IFM xử lý 1000 lệnh trong vòng 0,7 miligiây, CPU 316 xử lý 1000 lệnh trong vòng 0,3 miligiây. Module CPU 318 và các module CPU dùng cho S7-400 có thể xử lý trực tiếp chương trình người dùng bằng một mạch ASIC. Các module CPU 412 xử lý 1000 lệnh trong vòng 0,08 miligiây và đối với CPU 416 là 0,08 miligiây.

Trong các trạm S7-300, bộ xử lý trung tâm xử lý việc truyền thông với các bộ lập trình hoặc trao đổi dữ liệu với các trạm khác. Các trạm S7-400 có riêng một bộ xử lý truyền thông dùng cho mục đích này.



Vùng nhớ chương trình, vùng nhớ làm việc và vùng nhớ chính

CPU lưu trữ dữ liệu người dùng trong 2 vùng nhớ: vùng nhớ làm việc và vùng nhớ chính. Vùng nhớ làm việc là vùng nhớ chứa toàn bộ chương trình người dùng bao gồm cả cấu hình dữ liệu, hoặc ở dạng module tích hợp hoặc card nhớ dạng cắm. Vùng nhớ chính là một vùng nhớ RAM chứa các mã chương trình theo trình tự và dữ liệu người dùng sử dụng.

Thiết bị lập trình truyền chương trình người dùng bao gồm cả cấu hình dữ liệu tới vùng nhớ chương trình. Khi khởi động, hệ điều hành sẽ biên dịch cấu hình dữ liệu, cài đặt thông số CPU và thông số của các module I/O. Mã chương trình và dữ liệu người dùng sử dụng được sao chép vào vùng nhớ chính. Khi chương trình người dùng được nhập vào thiết bị lập trình, các khối được gọi từ vùng nhớ làm việc và được cập nhật bằng các giá trị hiện hành của địa chỉ dữ liệu từ vùng nhớ chính.

Card nhớ mở rộng vùng nhớ chương trình người dùng

Các card nhớ hiện có là RAM và flash EPROM; các card RAM được sử dụng chủ yếu để mở rộng vùng nhớ làm việc (ví dụ ở CPU các trạm S7-400), nó cho phép chỉnh sửa chương trình khi ở online. Khi tháo các card RAM ra thì dữ liệu sẽ bị mất.

Bạn có thể sử dụng 1 flash EPROM để duy trì chương trình bao gồm cấu hình dữ liệu và các thông số module trong trường hợp bị mất

nguồn mà không cần pin dự phòng. Để thực hiện điều này, nạp chương trình ở chế độ offline vào card nhớ flash EPROM được cài đặt trong thiết bị lập trình. Bạn cũng có thể nạp chương trình ở online dùng card nhớ được cài đặt trên module CPU nếu CPU được trang bị card nhớ cho mục đích này.

Các CPU họ S7-300 (trừ CPU 318) có một vùng nhớ RAM đủ lớn để lưu trữ toàn bộ chương trình người dùng. Vùng nhớ RAM của các CPU S7-400 và CPU 318 có thể dùng cho các chương trình nhỏ hay dùng để thay đổi các khối đơn lẻ. Nếu độ lớn chương trình vượt quá dung lượng của vùng nhớ chương trình thì bạn cần sử dụng một card RAM để kiểm tra chương trình.



Pin backup giúp lưu dữ liệu vùng nhớ

Pin backup được đặt trong các module CPU (S7-300) hoặc ở module nguồn cung cấp (CPU S7-400). Điện áp pin được CPU giám sát liên tục, khi pin yếu sẽ có chỉ báo và có thể đánh giá từ chương trình người dùng. Pin có thời hạn duy trì trong vòng một năm.

Vùng nhớ chương trình RAM cần có một pin backup để lưu trữ dữ liệu chương trình. Các card nhớ EEPROM và dạng flash EPROM không cần pin backup.

Các bộ điều khiển S7-300 lưu trữ các giá trị hiện hành của vùng nhớ làm việc (các khối dữ liệu) và vùng nhớ hệ thống (các bit nhớ, timer, counter) trong vùng nhớ non-volatile(hay còn gọi là retentive memory), bằng cách này dữ liệu được duy trì mà không cần pin backup thậm chí trong trường hợp mất nguồn.

Vùng nhớ hệ thống chứa các biến

Vùng nhớ hệ thống chứa địa chỉ (các biến) được truy xuất bằng chương trình người dùng. Các địa chỉ được nhóm lại thành các vùng địa chỉ với một số lượng địa chỉ xác định cho mỗi CPU. Bộ nhớ hệ thống của CPU bao gồm các vùng địa chỉ sau:

Inputs (I)

Vùng nhớ đệm của các module ngõ vào digital

Outputs (O)

Vùng nhớ đệm của các module ngõ ra digital

Các bit nhớ (M)

Vùng nhớ có thể đặt địa chỉ từ bất kỳ vị trí nào trong chương trình người dùng

Chức năng thời gian (T)

Các timer dùng trì hoãn và giám sát thời gian

Chức năng đếm (C)

Các bộ đếm dùng cho đếm lên và đếm xuống

Dữ liệu cục bộ tạm thời (L)

Các vùng đệm trung gian dùng cho xử lý khối. Các dữ liệu cục bộ tạm thời được lưu trữ trong ngăn xếp L (ngăn xếp dữ liệu cục bộ), nó bị CPU chiếm dụng trong khi thực hiện chương trình. Vùng nhớ hệ thống cũng có các buffer dùng truyền thông và thông báo hệ thống (vùng đệm chẩn đoán). Đối với CPU S7-400 người sử dụng có thể xác định độ lớn của vùng đệm dữ liệu, vùng đệm vào ra và ngăn xếp L.

2.13 Các bộ xử lý trung tâm với chức năng đặc biệt

Bộ xử lý trung tâm CPU 312IFM và 314IFM

CPU 312IFM và CPU314IFM được trang bị các ngõ vào và ra tích hợp có thể đặt địa chỉ giống như các ngõ I/O chuẩn từ chương trình người dùng.

Chúng cũng có thể được sử dụng như các ngõ vào báo hiệu (alarm) hay các ngõ vào dùng cho các chức năng đặc biệt của hệ điều hành. Khi cài đặt các thông số CPU bằng STEP 7, bạn kích hoạt *một* chức năng đặc biệt và dùng khối hàm hệ thống SFB để chèn chức năng này vào trong chương trình của mình.

➤ *Bộ đếm lên/xuống SFB 29HS_COUNT*

Bộ đếm dùng 2 ngõ vào dùng cho đếm lên/xuống tới 10 kHz, một ngõ chọn hướng đếm và một ngõ điều khiển. Giá trị đếm hiện hành được đọc dưới dạng số nguyên kép (double integer) và được truyền tới 2 bộ so sánh. Nhiều chức năng được kích hoạt khi bộ so sánh đạt tới giá trị đặt trước như : set ngõ ra digital, tạo xử lý ngắt, reset một bộ đếm, set một bộ so sánh.

➤ *Đồng hồ đo tần số SFB 30 FREQ_MES*

Một trong số các ngõ vào được sử dụng để đo tần số tới 10 kHz theo hai phương pháp: đếm cạnh tín hiệu trong thời gian 0,1 giây, 1 giây, và 10 giây (phù hợp với tần số cao), và phương pháp đo khoảng cách giữa hai cạnh tín hiệu sau 1ms, 2 ms, và 4 ms (phù hợp với tần số thấp). Tần số được đọc dưới dạng giá trị một số nguyên kép (double integer) và có thể giám sát giá trị giới hạn bằng 2 bộ so sánh.

➤ *Bộ đếm lên/xuống A/B SFB 38 HSC_A_B (chỉ có ở CPU 314IFM)*

Hai bộ đếm riêng biệt, mỗi bộ sử dụng 2 ngõ vào digital dùng cho đếm lên/xuống tới 10 kHz. Giá trị đếm hiện hành được đọc dưới dạng giá trị một số nguyên kép (double integer) và được truyền tới 1 bộ so sánh. Bộ so sánh có thể set ngõ ra digital, tạo xử lý ngắt, reset một bộ đếm, set một bộ so sánh.

➤ *Điều khiển vị trí SFB 39 POS (chỉ có ở CPU 314IFM)*

Giá trị thực được lấy từ các cảm biến bất đồng bộ 24 V tại tần số tới 10 kHz và được đồng bộ hoá bằng cách sử dụng các ngõ vào digital hay phân mềm. Bốn ngõ ra digital, điều khiển điều tốc (tốc độ, hướng) nhanh/từ từ, hay 1 ngõ ra analog với 2 ngõ ra digital tùy chọn điều khiển 1 bộ biến tần.

Bộ xử lý trung tâm CPU 614

Được thiết kế chuyên dùng trong công nghiệp xử lý chất dẻo, CPU 614 có thể dùng để thay thế CPU S7-300. CPU 614 cho phép bạn lập trình bằng ngôn ngữ lập trình C, ngoài ngôn ngữ lập trình PLC. Các đặc tính kỹ thuật

của nó tương tự như CPU 314. Vùng nhớ chính có dung lượng 128 KB. Vùng nhớ chương trình có dung lượng 192 KB và có thể mở rộng tới 512 KB bằng cách lắp đặt thêm một card EPROM.

Gọi một khối mã assembly SFB 63 AB_CALL

Gọi một khối mã assembly viết bằng ngôn ngữ lập trình Assembler hoặc C từ chương trình người dùng STEP 7 (chỉ có ở CPU 614).

2.14 SIMATIC PC Based với WinAC

SIMATIC Win AC (Windows Automation Center) là một bộ điều khiển họ SIMATIC chạy trên các máy PC chuẩn. Nó hoạt động dựa trên hệ điều hành Windows NT. Nó có hai phiên bản: Slot-PLC và Software-PLC.

Các giao tiếp chuẩn với Win AC

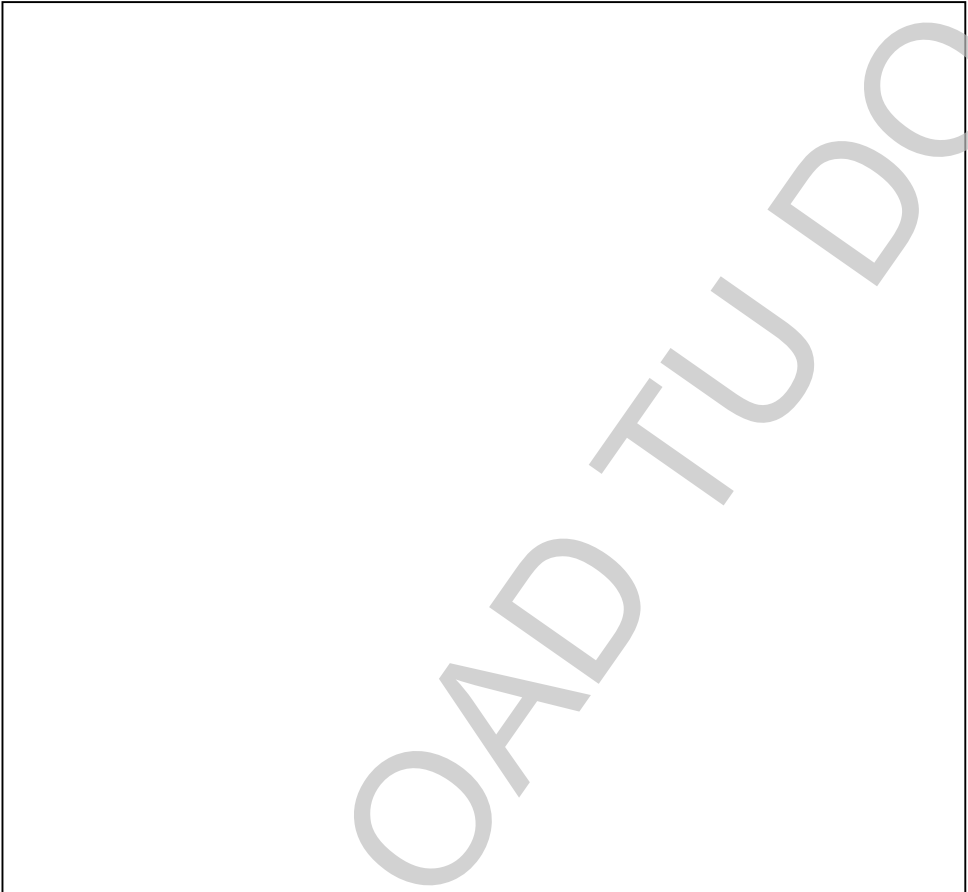
SIMATIC WinAC tạo sự kết nối giữa phần mềm tự động điều khiển và phần mềm tiêu chuẩn. SIMATIC Controller được gọi là Control Engine trong *WinAC computing*, Control Engine có thể là một “software- PLC” (WinLC, Windows Logic Controller) hay là một “Slot PLC” (ví dụ CPU 416 dạng module PC).

Việc trao đổi dữ liệu bằng Control Engine sử dụng công nghệ Microsoft OLE (Object Linking and Embedding), với công nghệ này bạn có thể sử dụng SIMATIC Controls đã có hoặc sử dụng ActiveX và OLE Controls (OLE Custom Controls) của mình. WinAC cũng có một *OPC Server* (OLE dùng cho điều khiển xử lý) cho phép các ứng dụng OPC Client khác có thể truy cập Control Engine. WinAC Tool Manager giúp bạn quản lý các ứng dụng chuẩn có trao đổi dữ liệu với Control Engine, như Visual Basic hay tính toán các bảng tính trong Microsoft Excell.

WinAC giúp phân bố các ứng dụng trên nhiều máy tính khác nhau. Những máy tính này được liên kết với nhau thành một mạng nhờ DCOM (Distributed Component Object Model) của Microsoft. Chẳng hạn bạn có thể sử dụng WinAC trên một PC khác hay lập trình cho Control Engine bằng STEP 7.

Điều khiển dùng WinLC

Với SIMATIC WinLC (Windows Logic Controller) bạn có thể sử dụng PC của mình như một bộ điều khiển SIMATIC. Bạn sử dụng các công cụ chuẩn của SIMATIC Manager để tạo cấu hình, cài đặt thông số và lập trình WinLC, việc kết nối tới máy hay hệ thống cần điều khiển được thực hiện qua PROFIBUS dùng một card CP 5412.



bị lập trình để lập trình, tạo cấu hình cho bộ điều khiển giống như S7-CPU nhờ giao tiếp MPI.

Nếu bạn đã cài đặt STEP 7 vào PC có WinAC và muốn đặt địa chỉ trực tiếp cho bộ điều khiển thì bạn phải thiết lập kết nối giữa 2 phần mềm bằng cách: gọi chương trình cài đặt giao tiếp PG/PC trong Windows Control Panel, bên dưới Access Point of Application cài đặt giao tiếp “S7 ONLINE (STEP7)“, ví dụ “CPU 416-2DP ISA Card“.

Dùng SIMATIC Manager để tạo một trạm có Slot-PLC. Trong một Project chèn một *SIMATIC 400 Station* và mở mục *Hardware*, chọn bộ điều khiển “WinAC ...“ từ “*SIMATIC Based Control*“ trong *Hardware Cataloge*. Các bước kế tiếp và viết chương trình tương tự như S7-CPU.

Hình 2.12 Tự động với WinAC

2.15 Các kết nối xử lý phân tán

PROFIBUS –DP

PROFIBUS-DP (*Distributed Peripherals*) cung cấp một giao tiếp chuẩn cho việc trao đổi dữ liệu xử lý nhĩ phân giữa một “module giao tiếp” đã cài đặt trong bộ điều khiển PLC và các thiết bị liên quan. Module giao tiếp hoạt động như một DP-master và các thiết bị liên quan hoạt động như DP-slave. Distributed Peripherals là các module được nối tới một module DP-master nhờ mạng truyền thông PROFIBUS-DP. PROFIBUS-DP là một mạng truyền thông đã chuẩn hoá theo chuẩn châu Âu EN 50170 cho kết nối DP-slave chuẩn.

PROFIBUS-PA

PROFIBUS-PA (*Process Automation*) là một mạng truyền thông dùng cho công nghiệp chế biến và cũng có thể sử dụng trong các vùng nguy hiểm như zone 1 trong công nghiệp hoá học hoặc sử dụng trong công nghiệp chế biến thực phẩm.

Hiện có 2 thành phần dùng để nối PROFIBUS-DP và PROFIBUS-PA:

- Khớp nối DP/PA (PA Coupler): được sử dụng khi PROFIBUS-DP có thể hoạt động được tại tốc độ truyền 45,45kBit/s
- Liên kết DP/PA (PA Link): chuyển đổi tốc độ truyền của PROFIBUS-DP sang tốc độ truyền của PROFIBUS-PA

SIMATIC PDM (*Process Device Manager*) là một công cụ độc lập dùng cho đặt thông số, thực hiện nhiệm vụ và chẩn đoán cho các thiết bị liên quan có PROFIBUS-PA hoặc tính năng HART. Bộ chuyển đổi HART (*Highway Address Remote Transducer*) được thiết lập nhờ thiết bị DDL (*Device Description Language*).

Giao tiếp cảm biến/cơ cấu chấp hành (AS-interface)

Giao tiếp cảm biến/cơ cấu chấp hành (AS Interface) là một hệ thống mạng dùng cho tất cả các cảm biến và cơ cấu chấp hành. Một AS Interface chủ có thể điều khiển tới 31 AS Interface tớ (slave) nhờ một cáp giao tiếp 2 dây. Dây cáp truyền tín hiệu điều khiển và cung cấp điện áp. Các AS Interface tớ có thể là các cơ cấu chấp hành dạng bus, các cảm biến hoặc các module AS Interface cho phép kết nối tới 8 cơ cấu chấp hành (“thông thường”) nhĩ phân hay cảm biến. Một đoạn AS Interface có chiều dài tối đa 100 mét. Nếu sử dụng bộ lặp (các AS Interface tớ và các bộ nguồn cung cấp

ở hai phía) hay bộ mở rộng (các AS Interface và bộ nguồn cung cấp chỉ có ở phía cách xa AS Interface chủ) thì cáp có thể kéo dài tới 2x100 mét tạo ra chiều dài tối đa 300 mét.

Kết nối giao tiếp nối tiếp

Liên kết PROFIBUS-DP/RS 232C hoạt động như một bộ chuyển đổi giữa một giao tiếp RS 232C (V.25) với PROFIBUS-DP. Sử dụng kết nối DP/RS thì các thiết bị có giao tiếp RS 232C được kết nối trực tiếp tới PROFIBUS-DP. Nó cung cấp thủ tục 3964 R và free ASCII protocol.

2.16 SIMATIC-DP- ngoại vi phân tán

Hệ thống PROFIBUS-DP master

Một DP chủ (DP master) và các DP tớ (DP Slaves) liên kết với nhau tạo ra một hệ thống DP chủ. Một phần mạng có thể kết hợp tối đa 32 trạm và toàn bộ mạng có thể liên kết tối đa 127 trạm. Số lượng tối đa các DP tớ có trong một hệ thống DP chủ phụ thuộc vào loại DP chủ được sử dụng. Một trạm SIMATIC có thể có nhiều hệ thống chủ. Bạn có thể kết nối các thiết bị lập trình, các bảng điều khiển, cũng như các SIMATIC S5-DP tớ hay các DP tớ chuẩn của các hãng sản xuất khác tới PROFIBUS-DP network.

Các DP chủ điều khiển I/O

DP chủ là một trạm trong mạng PROFIBUS và truyền thông với các DP tớ của nó bằng truyền dữ liệu theo chu kỳ. Các module sau đây có thể sử dụng như các DP chủ:

- CPU có giao tiếp DP chủ hoặc module giao tiếp dạng cắm (ví dụ: CPU 315-2DP, CPU 417)
- Module giao tiếp liên kết với CPU (ví dụ IM 467)
- Module CP liên kết với CPU (ví dụ CP 342-5, CP 443-5)

Các DP chủ có thể là các Master cấp 1 dùng cho trao đổi dữ liệu ở chế độ xử lý online và là các DP chủ cấp 2 dùng cho các công việc dịch vụ và chẩn đoán khi được thực hiện bằng thiết bị lập trình.

2.17 DP Slave _ kết nối xử lý ở gần máy

Các DP tớ (slaves) là các nút trạm thụ động trong mạng truyền thông PROFIBUS. Trong các trạm SIMATIC S7 chúng ta phân biệt các loại DP tớ sau:

<i>Compact DP slaves:</i>	Giống như các module đơn lẻ
<i>Modular DP Slaves:</i>	Gồm nhiều module
<i>Intelligent DP Slaves:</i>	Chứa chương trình người dùng điều khiển các module của trạm slave

Họ ET 200 cung cấp các thiết bị I/O cho SIMATIC DP. Các version khác hiện có là: compact DP slave ET 200 và ET 200B, module DP slave ET 200M và ET 200X. ET 200X giống như Intelligent DP slave.

Hình 2.13 Các thành phần trong hệ thống master DP.**ET 200L: nhỏ gọn như postcard**

ET 200L là một trạm I/O nhỏ gọn có cấp bảo vệ IP 20. Nó chủ yếu được sử dụng trong các ứng dụng tự động có không gian chật hẹp. Tốc độ truyền tối đa trên mạng PROFIBUS là 1,5 Mbit/s. ET 200L có thể được trang bị các module input/output và hiện có 3 dạng sau:

- ET 200L: kiểu này không thể mở rộng
- ET 200L-SC: có thể mở rộng bằng SIMATIC SMART
- ET 200L IM-SC: có thể mở rộng dùng SIMATIC SMART và các module analog Input trong cấu hình hỗn hợp.

ET 200B: nhỏ gọn, độ sâu lắp đặt thấp

ET 200B là một trạm I/O nhỏ gọn có cấp bảo vệ IP 20. Nó chủ yếu được sử dụng trong các ứng dụng ở những nơi chỉ cần ít ngõ vào ra hoặc nơi có độ sâu lắp đặt thấp. Tốc độ truyền tối đa trên mạng PROFIBUS là 12Mbit/s. ET 200B có thể có các module input/output analog và digital.

ET 200M: mở rộng bằng module S7-300

ET 200M là một trạm modular I/O có cấp bảo vệ IP 20. Nó kết hợp tới 8 module S7-300. Tốc độ truyền tối đa là 12 Mbit/s. Trong kết nối với một S7-400 DP chủ thì trạm có thể được thiết lập để đóng vai trò các module bus. Điều này cho phép thay thế các module trong quá trình hoạt động. ET 200M cũng có thể được dùng trong mạng con PROFIBUS (subnet).

ET 200S: modular và cải tiến

ET 200S là một trạm modular I/O cải tiến được trang bị các module analog I/O, các module xử lý, các bộ cấp nguồn cho tải, các module nguồn. Có cấp bảo vệ IP 20 và có thể thay thế trong quá trình hoạt động.

ET 200X: lắp đặt phân tán trong vùng lân cận máy

ET 200X là một trạm modular I/O có cấp bảo vệ IP 65/IP 67, tốc độ truyền tối đa 12 Mbit/s. Nó đặc biệt phù hợp cho việc lắp đặt trong vùng lân cận máy mà không cần tủ điều khiển. Một module cơ bản hiện có ngõ I/O digital hoặc có tính năng PLC, thiết lập kết nối tới mạng PROFIBUS. Có thể kết nối tới 7 module mở rộng ở các dạng sau vào module cơ bản:

- Digital input/output module
- Analog input/output module

- Module dùng kết nối tới giao tiếp AS (như giao tiếp AS chủ)
- Module khí nén có 4/2 lỗ van
- Giao tiếp khí nén kèm theo một van FESTO CPV (tối đa một giao tiếp cho mỗi trạm)
- Các bộ cấp nguồn cho tải (tối đa 6 bộ cho mỗi trạm), đường dây điện tử/ cơ điện hoặc các bộ khởi động đảo chiều để điều khiển tất cả các tải 3 pha (max 5,5 kW, 400V AC)

Việc lắp đặt các module mở rộng được thực hiện dễ dàng nhờ các bộ nối khớp kín các đường dây tín hiệu và điện áp phụ cho các ngõ vào và ra. Có thể thay thế từng module riêng lẻ mà không cần tháo rời trạm.

ET 200X hoạt động như một modular DP slave

Nếu được trang bị một module BM 141 hoặc BM 142 thì trạm ET 200X hoạt động giống như một DP tớ : tất cả các module mở rộng được đặt địa chỉ bằng DP master. Ngoài ra các module BM 141 được trang bị 8 ngõ vào digital 24V DC và module BM 142 có 4 ngõ ra digital 24V DC 2A.

ET 200X hoạt động như một Intelligent DP slave

Một trạm ET 200X có module BM 147/CPU có thể đóng vai trò như một intelligent DP slave. Địa chỉ của các module mở rộng có thể xác định bằng các slot lắp đặt và chỉ có thể cài đặt trực tiếp bằng CPU tớ (trong trường hợp này là BM 147/CPU). Vùng truyền (tương thích cấu trúc và dữ liệu có thể điều chỉnh) tới DP chủ cung cấp 32 byte cho các ngõ vào và 32 byte cho các ngõ ra.

Nhờ có tính năng PLC, BM 147/CPU có thể xử lý trước các tín hiệu trong trạm ET 200X hoặc điều khiển trạm và do đó giảm bớt tải cho trạm SIMATIC trung tâm. Module được lập trình bằng các ngôn ngữ của STEP 7. Phạm vi chức năng là một phần chức năng S7-300 (không có các khối hàm, các khối dữ liệu tức thời)

BM 147/CPU có vùng nhớ chương trình RAM/flash EPROM 12 KB; không có sẵn card nhớ. Nó được lập trình và cấu hình thông qua mạng PROFIBUS tức là bằng thiết bị lập trình của trạm chủ.

2.18 Thiết bị lập trình SIMATIC

Các thiết bị lập trình SIMATIC là các công cụ hoàn hảo dùng để lập trình và cấu hình cho các bộ điều khiển PLC SIMATIC, chúng được coi là

các máy tính đầy đủ chức năng được thiết kế sử dụng trong công nghiệp, luôn được thay đổi phù hợp với sự phát triển mới nhất của công nghệ máy tính (các đặc tính kỹ thuật được đưa ra năm 1999).

Phần mềm cần thiết bao gồm Windows 98 (Windows NT), STEP 7, STEP 7 Micro/Win32 và STEP 5/ST (hiện có 5 ngôn ngữ) được cài đặt sẵn trong tất cả các thiết bị lập trình. Tất cả các thiết bị lập trình đều có thể sử dụng để lập trình cho card nhớ SIMATIC. Ngoài các giao tiếp PC chuẩn thì mỗi thiết bị lập trình được trang bị một giao tiếp MPI/DP để kết nối tới giao tiếp MPI của CPU hoặc mạng con PROFIBUS (subnet), kèm theo một cáp dài 5 mét.

Thiết bị lập trình **PG 720 PII** là một máy tính AT tương thích cầm tay. Nó có pin do đó không cần phải nối tới nguồn nuôi. Nó đặc biệt phù hợp trong bảo trì tại chỗ hay công việc dịch vụ. Khi sử dụng trong văn phòng, bạn kết nối nó tới nguồn cung cấp bằng một bộ nguồn cung cấp riêng. Khả năng mở rộng: 2 PCMCIA slot dạng II hoặc 1 PCMCIA slot dạng III.

Thiết bị lập trình **PG 740 PIII** là một máy tính AT tương thích xách tay có các khả năng mở rộng sau: 1 PCI/ISA và 1 ISA slot (170 mm), 1 PCMCIA slot dạng II.

Thiết bị lập trình **RI 45 PIII Tower** là một desktop (dạng hình tháp) sử dụng trong văn phòng, nó có các khả năng mở rộng sau: 5 PCI slot dài (3 ISA slot dài), 3 ISA slot ngắn, vùng trống cho một ổ đĩa 3.5" và một ổ đĩa 5.25" hoặc hai 3.5" và một ổ 3.5" bên trong.

3 STEP 7: công cụ chuẩn cho SIMATIC

STEP 7 là công cụ chuẩn cho SIMATIC. STEP 7 sử dụng hệ điều hành Microsoft Windows 95/98, Microsoft Windows NT, Microsoft Windows XP.

STEP 7

STEP 7 là phiên bản hoàn chỉnh cho các bộ điều khiển PLC: SIMATIC S7-300/400, SIMATIC M7-300/400 và SIMATIC C7. Công cụ chủ yếu là *SIMATIC Manager*, nó quản lý tất cả các dữ liệu và tất cả các công cụ cần thiết để xử lý các dữ liệu này. Bằng STEP 7 bạn thiết lập phần cứng của bộ điều khiển SIMATIC, chọn địa chỉ module, cài đặt thông số module, thiết lập kết nối với mạng. STEP 7 hiện có các ngôn ngữ lập trình là Ladder Logic (LAD), Function Block Diagram (FBD) và STL (Statement List) dùng để viết chương trình cho bộ điều khiển.

STEP 7 Mini

STEP 7 Mini sử dụng phù hợp cho các ứng dụng đơn lẻ với SIMATIC S7-300 và SIMATIC C7. Sử dụng STEP 7 Mini bạn có thể thiết lập cấu hình phần cứng cho SIMATIC S7-300 và SIMATIC C7 bao gồm cả ngoại vi. Ngôn ngữ lập trình dạng LAD, FBD, STL được trợ giúp hoàn chỉnh. Chương trình người dùng được tạo ra bằng *STEP 7 Mini* có thể được xử lý thêm bằng *STEP 7 Basic*. *STEP 7 Mini* không được cung cấp các phần mềm tùy chọn và truyền thông tới các trạm SIMATIC khác.

STEP 7 Micro

STEP 7 Micro là phần mềm lập trình cho SIMATIC S7-200. Bằng STEP 7 Micro bạn tạo chương trình để cho S7-200 CPU xử lý dưới dạng bảng câu lệnh Statement List hay dạng lược đồ giống Ladder logic. Chương trình người dùng chỉ bao gồm một khối đơn và có thể chứa các chương trình con. Chức năng của S7-200 bao gồm các phép toán nhị phân, chức năng định thời, đếm, các lệnh số học 16 Bit và 32 Bit, các lệnh số học có dấu, chức năng so sánh, các bộ chuyển đổi số, điều khiển PID, xử lý bảng biểu, truyền dữ liệu. STEP 7 Micro cũng có thể cấu hình cho hiển thị text TD 200.

Có 2 phiên bản *STEP 7 Micro*, *STEP 7 Micro/Dos* cần có hệ điều hành MS-DOS 5.0 và có thể được sử dụng trong các PC với các chức năng hạn chế. STEP 7 Micro/DOS cũng có thể chạy trong cửa sổ DOS dưới Windows

3.1/3.11 hoặc Windows 95/98. STEP 7 Micro/Win được thiết kế chạy trên hệ điều hành Windows của Microsoft. Có 1 phiên bản 16 bit có thể chạy trên Win3.1/3.11 và phiên bản 32 bit có thể được tích hợp trong công cụ chuẩn STEP 7.

3.1 Quản lý dữ liệu ở SIMATIC -PLC

Phần mềm STEP 7 là một công cụ mạnh giải quyết tất cả các công việc điều khiển của bạn. Với STEP7 bạn không những viết chương trình là các lệnh, chức năng chương trình mà bộ điều khiển xử lý sau đó, mà bạn còn có thể tạo ra và soạn thảo tất cả các dữ liệu tự động. Nói cách khác việc thiết lập dữ liệu cho phần cứng, cài đặt thông số cho các module và dữ liệu kết nối để truyền thông đều được định nghĩa và tổ chức trong STEP 7.

Dữ liệu tự động này nằm trong bộ điều khiển PLC ở các vùng nhớ khác nhau. Trước hết cần phải có thiết bị lập trình PG, thường được hiểu là “*Configuration Unit*” hay “*Engineering Tool*”. Nó cũng có thể là một PC chuẩn sử dụng STEP 7. Tất cả các dữ liệu tự động được lưu trữ trong “*thư mục dự án*” trong ổ đĩa cứng. Thư mục dự án này là một cấu trúc phần mềm mà bạn không thể nhìn thấy thực sự trừ phi bạn xem trong Windows Explorer. Bạn soạn thảo dữ liệu trong thư mục dự án bằng *SIMATIC Manager* là giao tiếp người dùng của STEP 7. *SIMATIC Manager* hiển thị dữ liệu tự động trong các cửa sổ Windows.

Dữ liệu tự động trong đĩa cứng gọi là *offline data*. Khi STEP 7 đã “*dịch*” dữ liệu tự động nó có thể được nạp vào PLC. Thường thì bộ xử lý trung tâm CPU xử lý dữ liệu và phân phối tới các module khác khi có yêu cầu. Dữ liệu được nạp vào trong vùng nhớ người dùng của CPU là dữ liệu *offline data*. Dữ liệu *online* và *offline* phải luôn luôn được đồng bộ hóa để không xảy ra lỗi cấu hình.

Hãy tìm hiểu sơ bộ về phần chương trình người dùng, một “*block*” mà chúng ta thấy ở cửa sổ *offline* trong *SIMATIC Manager* nằm trong quản lý dữ liệu hay nói cách khác là trong đĩa cứng. Bây giờ bạn có thể chèn một block mới và viết chương trình. Khi khởi động trình soạn thảo khối được truyền tới vùng nhớ chính của thiết bị lập trình PG và được lưu trữ tại đó trong suốt quá trình soạn thảo. Chỉ khi nào bạn nhấp vào menu mục *FILE* → *SAVE* thì khối mới được lưu vào đĩa cứng. Bạn có thể lập trình ở chế độ *offline* có nghĩa là không kết nối với CPU.

Sau khi đã soạn thảo xong, bạn phải nạp vào trong bộ điều khiển PLC, thường gọi là “PLC” và có thể thử chương trình. Có 2 cách để nạp khối: kết nối giao tiếp MPI vào thiết bị lập trình và PLC, sau đó chuyển CPU sang chế độ “online“, sau đó nạp khối vào trong bộ nhớ người dùng của CPU bằng SIMATIC Manager. Cách thứ 2 là một khối hay toàn bộ chương trình đã được kiểm tra và cuối cùng là nạp vào CPU: nạp chương trình vào một card nhớ FEPROM và sau đó chèn card này vào CPU, mà không cần phải chuyển thiết bị lập trình sang online.

Hình 3.1 Quản lý dữ liệu trong bộ điều khiển PLC SIMATIC

Khi đã nối với CPU bạn xem nội dung vùng nhớ người dùng của CPU trong cửa sổ offline của SIMATIC Manager. Bây giờ bạn có thể soạn thảo khối từ vùng nhớ này của CPU trong chế độ online. Trình soạn thảo hay

SIMATIC Manager truyền khối vào trong vùng nhớ chính của thiết bị lập trình PG, ở đó bạn có thể chỉnh sửa khối. Chọn *PLC* → *DOWNLOAD* để ghi khối đã soạn thảo vào trong CPU mà không cần khối đã được chấp nhận trong quản lý dữ liệu của thiết bị lập trình PG.

Cách này chỉ nên dùng nếu bạn không muốn lưu khối ở offline, chẳng hạn nó được tạo ra chỉ để thử chương trình. Nói chung, quản lý dữ liệu *offline* trên đĩa cứng và quản lý quản lý dữ liệu online ở CPU có thể không khác nhau. Một mặt thì dữ liệu được lưu trữ trong thư mục dự án cũng được dùng để soạn thảo dữ liệu *online* (ví dụ các ký hiệu, chú thích), mặt khác tất cả các dữ liệu cần được hoàn chỉnh và không thay đổi sau khi chương trình đã được thử.

Đề nghị: Nếu bạn đang để CPU ở chế độ online, bạn lưu mỗi khối mà bạn đang soạn thảo bằng trình soạn thảo vào đĩa cứng cũng như vào CPU, bất kể khối đó là một “online block” hay “offline block”.

Các dữ liệu tự động khác cũng được giải quyết theo cách tương tự.

3.2 STEP 7

STEP 7 là công cụ tự động chính của SIMATIC. Nó bao gồm các ứng dụng cho cấu hình phần cứng, kết nối mạng và cho người dùng viết chương trình. Dữ liệu được sắp xếp dưới dạng “dự án” có chung quản lý dữ liệu. Chức năng của STEP 7 có thể mở rộng bằng nhiều phần mềm tùy chọn.

STEP 7 có 5 ngôn ngữ (Đức, Anh, Pháp, Ý và Tây Ban Nha). Với mỗi ngôn ngữ cần phải có khoảng 150 Mbyte trong đĩa cứng. Ngoài ra một tập tin swap-out chiếm 128 Mbyte trong vùng nhớ làm việc (work memory). Bạn nên dành 50 Mbyte cho dữ liệu chương trình của mình, và cho mỗi phần mềm thêm vào 10 tới 20 Mbyte. Vùng nhớ cần thiết có thể tăng lên do sử dụng các phép tính đặc biệt, chẳng hạn như copy một dự án (project).

STEP 7 là ứng dụng 32 bit, sử dụng hệ điều hành Microsoft Windows 95/98 hoặc Microsoft Windows NT hay Microsoft Windows XP. STEP 7 V5.x cho phép “nhiều người sử dụng” (multi-user capable), có nghĩa là bạn có thể soạn thảo một dự án nằm trong trạm trung tâm từ nhiều trạm khác nhau một cách đồng thời. Bạn điều chỉnh việc cài đặt trong *Windows Control Panel* bằng chương trình “SIMATIC Workstation”.

Bản quyền

Để sử dụng STEP 7 bạn cần phải có bản quyền (cho phép sử dụng) có trong một đĩa mềm. Sau khi cài đặt xong STEP 7, bạn cần phải đăng ký. Nếu đăng ký chưa có sẵn trong đĩa cứng thì sau đó bạn phải đăng ký.

Nếu bạn mất bản quyền do lỗi đĩa cứng thì bạn có thể sử dụng bản quyền hạn chế trong trường hợp khẩn cấp (*Limited-time emergency*) cho đến khi có bản quyền mới. Bản quyền hạn chế này nằm trong đĩa mềm bản quyền mà bạn đã có.

Online help

SIMATIC Manager cung cấp cho bạn các thông tin trong quá trình viết chương trình mà không cần sổ tay hướng dẫn sử dụng. Bạn sử dụng *menu help* để chọn đề mục. Ví dụ khi bạn nhìn vào mục *getting started* trong *online help* bạn sẽ thấy tổng quát về làm việc với *SIMATIC Manager*.

Chọn *HELP* → *CONTENTS* để khởi động trợ giúp chủ yếu của STEP 7 từ mỗi ứng dụng, mục này chứa các trợ giúp cơ bản. *HELP* → *CONTENTS-SENSITIVE HELP F1* đưa ra các trợ giúp phụ thuộc vào ngữ cảnh, có nghĩa là khi nhấn phím F1 sẽ cho biết các thông tin về mục được chọn bằng chuột hoặc thông tin về thông báo lỗi.

Trong thanh công cụ có mục có một mũi tên và một dấu hỏi, nếu nhấp vào mục này thì con trỏ chuột chuyển thành dấu hỏi, dùng trỏ chuột “trợ giúp” này bạn nhấp vào một mục trên màn hình, chẳng hạn một biểu tượng hay menu lệnh bạn sẽ nhận được các trợ giúp tương ứng.

3.3 SIMATIC Manager

SIMATIC Manager là công cụ chính của STEP 7, sau khi lắp đặt bạn sẽ thấy biểu tượng của nó trên cửa sổ Desktop.

Nhấp đúp chuột vào biểu tượng khởi động *SIMATIC Manager*.



Có thể khởi động từ thanh công cụ :

START → *SIMATIC* → *SIMATIC MANAGER*.

Với *SIMATIC Manager* bạn có thể làm việc với các đối tượng (object) trong thế giới STEP 7. Các đối tượng này thể hiện “thực tế” các đối tượng chương trình của bạn. Một dự án (project) chứa toàn bộ chương trình, một trạm (station) tương ứng với một bộ điều khiển PLC. Một dự án có thể chứa nhiều trạm được kết nối với nhau bằng một mạng con MPI. Một CPU được chèn vào trong một trạm, CPU này chứa một chương trình, trong trường hợp

này là một chương trình S7, mặt khác chương trình này cũng là một “thư mục” (folder) dùng cho các mục đích sau này, chẳng hạn như block chứa các khối đã được dịch.

Để bắt đầu viết chương trình bạn mở một dự án (project) đã có sẵn hoặc tạo một dự án mới, dự án mẫu có sẵn trong phần mềm sẽ giúp bạn làm tương tự. Nếu mở một thư mục bằng *FILE* → *OPEN* thì cửa sổ dự án bị chia làm 2 phần: bên trái là cấu trúc thư mục (cấu trúc hình cây), bên phải là nội dung mục được chọn. Nhấp chuột vào *checkbox* có dấu cộng ở phần cửa sổ bên



dự nhấp đúp vào ký hiệu OB1 để khởi tạo công cụ liên quan tới đối tượng, trong ví dụ là trình soạn thảo.

3.4 Thư viện và dự án

Thư viện và dự án nằm phía trên cùng trong cấu trúc hình cây của STEP7 như các “đối tượng chính”.

Dự án (project) dùng để lưu trữ theo thứ tự tất cả các dữ liệu và chương trình cần cho giải quyết mục đích tự động. Các phần quan trọng là:

- Cấu hình dữ liệu cho cấu trúc phần cứng
- Đặt thông số cho các module
- Cấu hình dữ liệu cho mạng truyền thông
- Các chương trình (khối logic, dữ liệu, ký hiệu, tập tin nguồn)

Các đối tượng trong dự án được sắp xếp theo dạng hình cây. Mỗi một dự án là điểm khởi đầu cho việc soạn thảo tất cả các đối tượng chứa bên trong nó. Trang bên cho biết quá trình xử lý các đối tượng này.

Các *thư viện* lưu trữ các phần tử chương trình tái sử dụng. Thư viện cũng có cấu trúc hình cây: chúng có thể chứa các chương trình S7, một thư mục cho chương trình nguồn và một bảng ký hiệu. Trong khi viết chương trình hay một phần chương trình, với khái niệm kết nối online bạn cũng có các tính năng tương tự như khi bạn làm việc trong một dự án. STEP7 có các *thư viện chuẩn* gồm các phần sau:

- Các khối tổ chức (Organization Blocks)
 - Các bảng mẫu về thông tin khởi động trong dữ liệu cục bộ
- Các khối hàm hệ thống (System Function Blocks)
 - Gọi các giao tiếp cho các hàm hệ thống (SFC) và các khối hàm hệ thống (SFB)
- Các khối hàm IEC
 - Các hàm (FC) cho xử lý các biến DT và các biến STRING
- Các khối điều khiển PID
 - Các khối hàm FB cho các bộ điều khiển liên tục và từng bước
- Các khối truyền thông
 - Các hàm FC cho giao tiếp SEND/RECEIVE

Cấu trúc thư mục hình cây

Các đối tượng (object) STEP7 được liên kết với nhau bằng cấu trúc hình cây. Hình trang bên cho thấy các phần quan trọng của cấu trúc hình cây khi bạn làm việc ở offline bằng STEP7 cho các ứng dụng S7. Các đối tượng hiển thị có màu đậm là các thư mục dùng cho các mục đích thêm. Tất cả các đối tượng trong hình đều có ở chế độ offline. Bạn tìm thấy các đối tượng này trong đĩa cứng của thiết bị lập trình PG. Nếu PG của bạn kết nối với CPU ở online (thường là PLC) bạn có thể xem online bằng cách chọn VIEW →

ONLINE . Sau đó bạn sẽ thấy các đối tượng trong thiết bị “đích” nằm trong một cửa sổ phụ và không có các đối tượng gạch chân trong hình.



3.5 Soạn thảo một dự án

Khi soạn thảo một dự án mới, bạn tạo các “thư mục” cho dữ liệu, sau đó tạo ra các dữ liệu. Thường thì bạn tạo một dự án với phần cứng, cấu hình phần cứng và tạo thư mục cho chương trình. Bạn cũng có thể tạo chương trình S7 mà không có phần cứng trực tiếp từ thư mục dự án (*project folder*). Chú ý là việc cài đặt thông số module như thay đổi địa chỉ, cài đặt CPU hoặc cấu hình kết nối chỉ có thể thực hiện được bằng công cụ *HW-Config*.

Bạn phải dùng *SIMATIC Manager* để soạn thảo toàn bộ dự án. Việc tạo, sao chép, xóa, thay đổi tên thư mục hay tập tin bằng Windows Explorer trong phạm vi cấu trúc thư mục của dự án có thể gây hại đến tính hoàn hảo của soạn thảo bằng *SIMATIC Manager*.

Tạo mới hay mở rộng một dự án (project)

STEP7 Wizard giúp bạn tạo một *project* mới. Bạn xác định CPU sử dụng và *Wizard* tạo một project với một S7 Station, CPU đã chọn, thư mục chương trình S7, thư mục tập tin nguồn và thư mục Blocks. Phần này mô tả các bước thực hiện cần thiết, nếu bạn muốn tạo hay mở rộng một project:

1 Mở một project hay tạo mới

Khởi động *SIMATIC Manager* và mở một project đã có bằng *FILE* → *OPEN* hoặc tạo một project mới bằng *FILE* → *NEW*.

2 Chèn một Station mới trong project

Trong *SIMATIC Manager* chọn project và chọn *INSERT* → *STATION* ... để chèn thư mục cho một trạm mới.

Nếu project chứa nhiều trạm thì cũng chèn thư mục cho trạm khác hay đối tượng cho mạng con tương tự như trên.

3 Cấu hình trạm

Khi trạm đã mở bạn sẽ thấy mục *Hardware* ở phía bên phải cửa sổ, mở mục này công cụ *HW-Config* được khởi động và hiển thị một *window station* rộng cùng *Hardware Catalog*. Bây giờ từ *Hardware Catalog*, bạn kéo các racks (giá đỡ) và CPU vào trong cửa sổ trạm (*window station*) dùng cách kéo và thả (*drag and drop*) (tới đây đã đủ cho việc tạo chương trình người dùng, thông tin thêm xem phần “Cấu hình một trạm SIMATIC”). Dịch trạm, lưu trữ, và đóng lại để quay trở về *SIMATIC Manager*.

Ngoài mục *Hardware*, trạm được mở còn cho thấy CPU, đối với CPU thì cùng một lúc *SIMATIC Manager* tạo ra biểu tượng kết nối, một S7

program kèm bảng ký hiệu, thư mục tập tin nguồn và thư mục blocks. Bây giờ cấu trúc của một project đã được thiết lập xong.

4 Xem nội dung của S7 program

Mở CPU bằng cách nhấp vào *checkbox*, mở *S7 program* và thư mục *Blocks*, phía bên phải *SIMATIC Manager* cho thấy các ký hiệu dùng cho dịch cấu hình dữ liệu (dữ liệu hệ thống) và một khối tổ chức rỗng cho chương trình chính (OB 1).

5 Soạn thảo chương trình

Bây giờ bạn đang ở cấp thấp nhất của thư mục hình cây. Trước hết bạn mở OB 1, một cửa sổ với các đặc tính của chương trình xuất hiện, sau đó bạn mở trình soạn thảo (Program Editor) để viết chương trình trong khối tổ chức.

Mỗi lần bạn mở một object thì đồng thời các công cụ liên quan cũng được khởi tạo. Chẳng hạn khi bạn mở một subnet hay *connection* thì công cụ cấu hình mạng Net-Pro được khởi tạo, khi bạn mở Hardware thì *Hardware Config* được khởi tạo; nếu bạn mở symbols thì *Symbol Editor* được khởi tạo và khi bạn mở một khối logic thì đồng thời khởi động trình soạn thảo *Program Editor*.

Tạo một project không có S7 Station

Bạn có thể tạo một chương trình mà không có cấu hình ban đầu của một trạm. Do đó bạn phải tạo lập một thư mục (folder) cho chương trình của mình. Chọn project và dùng *INSERT* → *PROGRAM* → *S7 PROGRAM* để tạo một *S7 program*. *SIMATIC Manager* tạo các đối tượng *symbols*, *source files*, và *Blocks* dưới *S7 program*. *Blocks* chứa khối OB 1 rỗng.

Tạo một thư viện

Bạn có thể tạo một chương trình trong thư viện nếu bạn muốn sử dụng chương trình nhiều lần. Bằng cách này bạn có một chương trình mẫu mà bạn có thể copy toàn bộ hay một phần vào trong chương trình hiện hành của mình. Hãy chú ý là chương trình trong thư viện không thể kết nối online, do đó bạn chỉ có thể thử chương trình trong phạm vi dự án.

Quản lý, sắp xếp lại và nén dự án (project)

SIMATIC Manager quản lý các dự án và thư viện trong danh mục dự án hay thư viện. Dùng *FILE* → *MANAGER*. *SIMATIC Manager* hiển thị tất cả các dự án cùng tên và đường dẫn. Bây giờ bạn có thể xóa các project

trong danh mục mà bạn không muốn hiển thị hoặc chấp nhận các project mới trong danh mục. Các thư viện cũng được quản lý theo cách tương tự.

Khi chọn *FILE* → *REARRANGE...*, SIMATIC Manager sẽ xóa các khoảng trống tạo ra do bạn xóa các đối tượng và tối ưu hoá việc lưu trữ dữ liệu giống như chương trình defragmenting tối ưu hoá việc lưu trữ dữ liệu trong đĩa cứng. Do phải di chuyển dữ liệu nên việc sắp xếp lại tốn một khoảng thời gian.

Bạn có thể nén một project hay một thư viện bằng *FILE* → *ARCHIVE...*, SIMATIC Manager sẽ lưu trữ các object được chọn dưới dạng nén trong một tập tin nén (thư mục dự án hoặc thư mục thư viện với tất cả các thư mục và tập tin cấp thấp hơn). Trong quá trình nén yêu cầu ngưng soạn thảo project hay thư viện và đóng tất cả các cửa sổ.

Thư viện và dự án không thể được xử lý dưới dạng nén. Dùng *FILE* → *RETRIEVE* để giải nén các đối tượng. Các object đã giải nén được tự động chấp nhận trong quản lý dự án hay thư viện.

3.6 Đặt cấu hình cho một trạm SIMATIC

Bạn thiết lập cấu hình phần cứng cho bộ điều khiển PLC bằng công cụ HW Config. Cấu hình được tạo ở offline không cần CPU. Bạn có thể cài đặt địa chỉ và thông số cho module bằng công cụ này. Khi đã cài đặt phần cứng, bạn có thể tạo cấu hình phần cứng trong hay sau khi viết chương trình.

Bắt đầu thiết lập cấu hình phần cứng bằng cách chọn *EDIT* → *OPEN* hoặc nhấp đúp vào mục *Hardware* trong thư mục *SIMATIC 300 / 400 Station* đã mở. Bạn cài đặt cơ bản cho cấu hình phần cứng bằng cách chọn *OPTION* → *CUSTOMIZE*. Sau khi tạo cấu hình để kiểm tra xem phần nhập có lỗi hay không chọn *STATION* → *CONSISTENCY CHECK*. Chọn *STATION* → *SAVE* để lưu trữ bảng cấu hình với tất cả các thông số project vào đĩa cứng. Chọn *STATION* → *SAVE AND COMPILE* để bảng cấu hình được biên dịch (compiled) cùng lúc khi chúng được lưu và dữ liệu đã biên dịch được lưu trữ trong mục *system data* trong thư mục *Blocks* ở offline. Sau khi biên dịch bạn có thể truyền dữ liệu cấu hình vào CPU được kết nối bằng cách chọn *PLC* → *DOWNLOAD*. Mục *system data* trong thư mục *Blocks* ở offline thể hiện dữ liệu cấu hình hiện hành trong CPU, bạn có thể truyền “ngược” chúng vào đĩa cứng bằng cách chọn *PLC* → *UPLOAD*.

Bạn xuất các dữ liệu cấu hình bằng cách chọn *STATION* → *EXPORT*, sau đó STEP7 tạo ra một tập tin có định dạng ASCII chứa dữ liệu cấu hình và các dữ liệu thông số của module



HW Config làm việc với các bảng, mỗi bảng tương trưng cho một rack, một module, hoặc một trạm DP. Một bảng cấu hình hiển thị các slot với các module gắn vào chúng hoặc các đặc tính của module như địa chỉ hay số thứ tự. Nhấp đúp vào hàng module trong bảng cấu hình sẽ mở ra cửa sổ đặc tính của module và cho phép bạn cài đặt các thông số cho module.

Xem modules trong hardware catalog

Bạn có thể hiển thị hay tắt Hardware Catalog bằng cách sử dụng *View Catalog*. Nó chứa các rack, các module hay các module giao tiếp mà STEP7 có thể nhận biết. Bạn có thể tạo Hardware Catalog cho riêng mình với cấu trúc mà bạn chọn bằng cách sử dụng *Options Edit catalog Profiles* chỉ hiển thị các module mà bạn muốn làm việc.

Nếu nhấp đúp vào thanh tiêu đề bạn có thể “gắn” Hardware Catalog vào bên phải của cửa sổ trạm để tránh nó che mất nội dung cửa sổ trạm. Nếu nhấp đúp lần nữa vào thanh tiêu đề thì loại bỏ việc gắn này.

3.7 Sắp xếp và gán các tham số cho module

Sắp xếp module

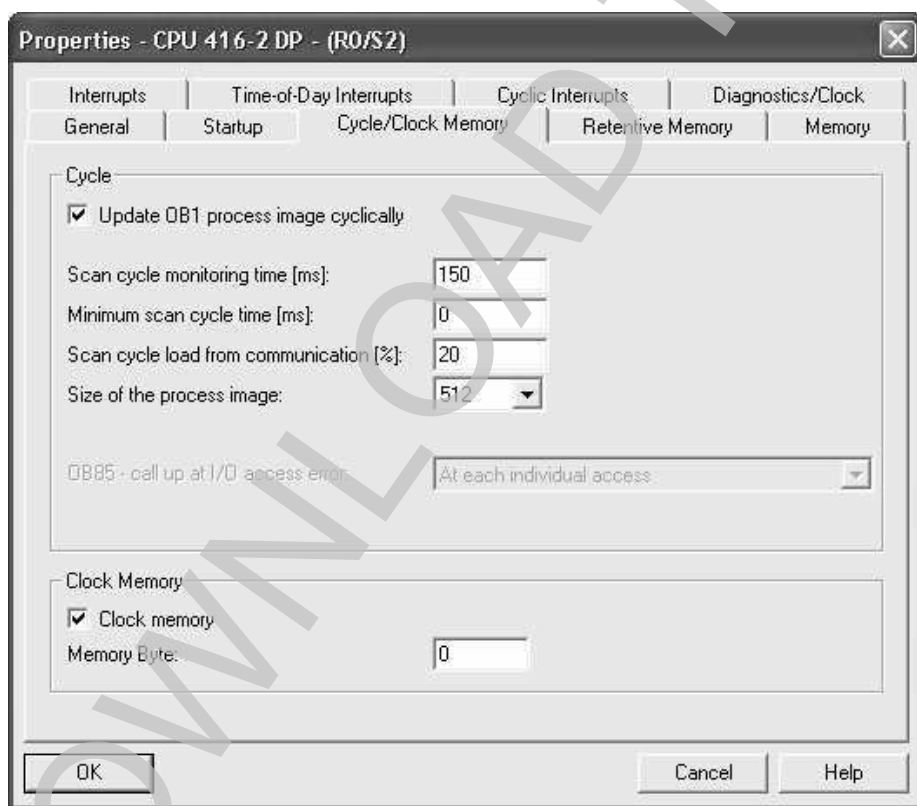
Để sắp xếp các module trong Hardware Catalog, chẳng hạn “SIMATIC 300” và “RACK 300”.

Bạn chọn các ray lắp đặt (theo chuẩn DIN) bằng chuột và “đề chuột” kéo ray vào trong phần phía trên của cửa sổ trạm, sau đó “nhả” ray vào bất kỳ chỗ nào mong muốn (kéo và nhả). Bảng cấu hình trống dùng cho rack trung tâm xuất hiện, bây giờ bạn có thể lấy các module cần thiết từ Hardware Catalog theo cách tương tự và kéo chúng vào các slot dự định. Một biểu tượng no-stopping báo cho bạn biết là không thể kéo module vào slot đã

chọn. Bạn tạo bảng cấu hình cho các rack thêm bằng cách kéo các rack đã chọn từ catalog vào trong cửa sổ trạm.

Cài đặt thông số module

Bạn thiết lập đặc tính cho module bằng cách cài đặt các thông số cho nó. Việc này chỉ cần thiết khi bạn muốn thay đổi các thông số đã được cài đặt trước, yêu cầu trước tiên cho việc cài đặt thông số là bạn phải sắp xếp các module trong một bảng cấu hình. Trong bảng cấu hình bạn nhấp đúp vào module hoặc chọn module và chọn *EDIT* → *OBJECT PROPERTIES*, một hộp thoại xuất hiện với các mục về module cho phép điều chỉnh thông số. Việc cài đặt thông số cho CPU theo cách này làm thay đổi tính chất thi hành của chương trình người dùng.



Hình 3.6 Cài đặt thông số, ví dụ cho một CPU.

3.8 Địa chỉ các module

Khi bạn sắp xếp module thì *HW Config* tự động đưa ra địa chỉ đầu. Bạn có thể thấy địa chỉ này trong bảng cấu hình ở phần dưới của cửa sổ trạm

hoặc trong hộp thoại *Object Properties* của module ở dạng câu hỏi. Đối với S7-400-CPU và S7-CPU-300 có giao tiếp DP bạn có thể thay đổi địa chỉ đầu module. Phạm vi địa chỉ của các module không nên trùng nhau.

Nếu chọn *DEBUG* → *RESUME* thì chương trình được xử lý ở tốc độ bình thường cho tới khi nó gặp điểm breakpoint kế tiếp.

3.21 Kiểm tra chương trình người dùng ở offline với PLCSIM

Phần mềm PLCSIM cho phép bạn thử chương trình ở offline mà không cần dùng thêm phần cứng. PLCSIM mô phỏng một bộ PLC (là một PLC với các modules) trên thiết bị lập trình. CPU này không xác định rõ. Bạn có thể thử chương trình đối với bất cứ CPU nào bằng PLCSIM. Nó không phân biệt khối được viết bằng ngôn ngữ lập trình nào.

Khi bạn đã cài đặt PLCSIM thì nó được tích hợp vào SIMATIC Manager. Khi đã viết xong chương trình hoặc thậm chí một khối bạn muốn thử, bạn khởi động PLCSIM bằng cách chọn menu *OPTIONS* → *SIMULATE MODULES* hoặc nhấp chuột vào biểu tượng tương ứng.

Giao diện của PLCSIM

Giao diện của PLCSIM hiển thị trong một cửa sổ riêng. Trong cài đặt chuẩn cửa sổ này chứa một CPU subwindow hiển thị việc điều khiển và các LED của CPU. Chỉ cần nhấp chuột để chuyển CPU mô phỏng sang chế độ RUN, RUN-P, STOP hoặc thực hiện reset bộ nhớ trên CPU. Chẳng hạn, nếu CPU mô phỏng chuyển sang chế độ STOP khi có lỗi lập trình thì sẽ được chỉ báo bằng đèn LED error và đèn STOP LED.

Chương trình cần điều khiển hiển thị các ngõ vào và ra trong các cửa sổ phụ (subwindows). Mỗi một cửa sổ phụ tượng trưng một bit, byte, word hay doubleword với các định dạng dữ liệu khác nhau. Chẳng hạn, bạn có thể sử dụng các cửa sổ phụ này để mô phỏng trạng thái tín hiệu ngõ vào (nhấp chuột vào chúng) và quan sát các ngõ ra được set và reset.

Bạn xác định các giá trị digital cho đúng với định dạng dữ liệu. Một con trượt cho phép bạn mô phỏng các giá trị thay đổi liên tục chẳng hạn như các giá trị analog. Bạn có thể thay đổi vị trí con trượt bằng cách dùng chuột hoặc các phím mũi tên.

Bạn có thể quản lý và chỉnh sửa các địa chỉ trong CPU trong các subwindows. Có các cửa sổ phụ cho các bit nhớ, timer, counter và các biến chung. Trong một subwindow biến (chứa các input, output, bit nhớ, hoặc biến toàn cục) bạn có thể nhập bất cứ địa chỉ nào bao gồm địa chỉ dữ liệu xác định (ví dụ DB62.DBB 15).

Có thể chọn *OPTIONS* → *ATTACH SYMBOLS* để cài một bảng ký hiệu cho PLCSIM, cho phép bạn xem các tên ký hiệu cho các biến (chọn menu



vào PLC → MODULE INFORMATION trong SIMATIC Manager để đọc vùng đệm chẩn đoán và tìm ra nguyên nhân gây STOP.

Khi bạn thử chương trình dùng bảng biến và chức năng quản lý trạng thái chương trình với chế độ từng bước (single-step) có các breakpoints trong STL, thì CPU mô phỏng bằng PLCSIM cũng xử lý giống như một CPU thật sự.

Các chức năng khác

PLCSIM mô phỏng một CPU có các hàm hệ thống và các khối hàm hệ thống chọn lựa. Nó cung cấp các cấp ưu tiên cao nhất (các khối tổ chức). Tại một thời điểm chỉ mô phỏng một PLC. PLCSIM còn có thể lưu trữ việc sắp xếp các cửa sổ và hiển thị việc sắp xếp này ở lần kế tiếp khi bạn mở nó.

Chương trình PLCSIM *S7ProSim* là một điều khiển ActiveX dùng cho các ứng dụng dùng công nghệ Microsoft OLE/COM. Điều khiển này cho phép bạn xử lý mô phỏng bằng PLCSIM.

3.22 Điều Khiển PID Với Phần Mềm SIMATIC

Có nhiều cách để điều khiển PID bằng hệ thống tự động SIMATIC. Chẳng hạn, các module chức năng FM 355 và FM 455 có thể thực hiện các chức năng điều khiển áp suất, nhiệt độ, dòng chảy độc lập với chương trình người dùng trong CPU. Hệ điều hành của CPU 314IFM bao gồm các bộ điều khiển PID có thể được gọi từ chương trình người dùng như là các khối hàm hệ thống (SFB). Chúng có tính năng giống như các khối hàm của phần mềm Standard PID Control và Modular PID Control được trình bày dưới đây.

Standard PID Control

Phần mềm *Standard PID Control* bao gồm tất cả các khối cần thiết cho điều khiển PID, cùng với các công cụ để cài đặt thông số. Bằng Standard PID Control bạn có thể kết hợp các bộ điều khiển liên tục PID controllers, step controller, pulse controller vào trong chương trình của mình. Các bộ điều khiển có một cấu trúc được định nghĩa trước. Bạn có thể kích hoạt/không kích hoạt từng chức năng đơn lẻ như giám sát mức, độ phẳng, giá trị giới hạn bằng các công tắc phần mềm. Mỗi một vòng điều khiển (control loop) sẽ lưu trữ dữ liệu vào trong các khối dữ liệu tức thời riêng của nó. Các ứng dụng như bộ điều khiển có điểm đặt cố định (fixed-setpoint controller) với ngõ ra không liên tục cho các cơ cấu chấp hành làm cho phần mềm này rất dễ sử dụng.

Sử dụng công cụ cài đặt thông số để: cài đặt thông số dạng biểu đồ trong sơ đồ khối, quản lý vòng điều khiển (máy vẽ đường cong 4 biến, hiển thị các biến điều khiển liên quan trong vòng), tối ưu hoá và vận hành các bộ điều khiển.

Có thể chạy được trên CPU S7-300 (từ CPU 313 trở lên), S7-400 và C7.

Modular PID Control

Phần mềm *Modular PID Control* bao gồm các khối cần cho điều khiển PID và các công cụ khởi tạo. Phần mềm này được thiết kế dùng cho các cấu trúc điều khiển phức tạp bằng cách liên kết các khối. Dưới đây là một số ví dụ ứng dụng minh họa sử dụng Modular PID Control.

Sử dụng công cụ khởi tạo để quản lý các vòng điều khiển (máy vẽ đường cong 4 biến, hiển thị các biến điều khiển liên quan trong vòng), tối ưu hoá và vận hành các bộ điều khiển.

Có thể chạy được trên CPU S7-300 (từ CPU 313 trở lên), S7-400 và C7, hiện có ở dạng thư viện cho M7.

PID Self-Tuner

Phần mềm *PID Self-Tuner* thêm chức năng tự động điều chỉnh cho bộ điều khiển PID đã cài đặt trong hệ thống (điều chỉnh ban đầu và điều chỉnh liên tục). Chức năng này phù hợp cho điều khiển nhiệt độ, mức làm đầy và dòng chảy với đáp ứng truyền ổn định, tín hiệu cho cơ cấu chấp hành unipolar, điều chế số lần trì hoãn và điều chế hệ số xử lý.

Có thể chạy được trên CPU S7-300 (từ CPU 313 trở lên), S7-400 và C7 kết nối với các PID Controllers, Standard PID Control, Modular PID Control, FM 355 và FM 455.

Fuzzy Control ++

Phần mềm *Fuzzy Control ++* gồm có tất cả các khối cần thiết và công cụ cấu hình. Fuzzy Control ++ cho phép bạn trang bị một hệ thống điều khiển chất lượng tốt, tự động hoàn chỉnh cho các quá trình xử lý rất khó mô tả bằng toán học. Trong hệ điều khiển mờ (fuzzy control system) kết hợp kiến thức kinh nghiệm với đặc tuyến của quá trình xử lý được thay thế bằng các quy tắc có tính định tính mô tả quá trình xử lý. Việc này thực hiện qua 2 bước:

- Các giá trị tức thời của tín hiệu ngõ vào và ngõ ra được ký tự hoá bằng các định nghĩa “fuzzy” và định tính hoá như “nhiều“, “ít“, “nóng“, “lạnh“, “nhanh“, “chậm“ ...
- Các kiến thức kinh nghiệm có được từ quá trình xử lý được công thức hoá bằng quy tắc “If-Then“

Quá trình xử lý được thực hiện như sau: giá trị tức thời của ngõ vào được dịch sang thành cấp sự thật (“mờ hoá“), phép toán logic được thực hiện dựa vào cấp sự thật (“If“), cấp sự thật cho ngõ ra được xác định (“Then“ hoặc “phép suy luận“) và giá trị ngõ ra được tính bằng kết quả các chức năng thành phần và tính toán trọng tâm (“xóa mờ hoá“).

Công cụ cấu hình là menu-driven cho phép bạn cài đặt các thông số cho hệ điều khiển mờ (phác thảo qui tắc “If-Then“ và xác định phạm vi giá trị cho các tín hiệu ngõ vào và ra), mô phỏng các tín hiệu ngõ vào dùng máy vẽ đường cong, quản lý hoạt động các bộ điều khiển bằng dạng biểu đồ, hiển thị các đặc tính của hệ điều khiển mờ theo 3 chiều và quản lý tới 4 tín hiệu ngõ vào và ra (máy vẽ đường cong có khả năng lưu trữ dữ liệu vào đĩa cứng để xử lý mở rộng bằng các ứng dụng chuẩn).

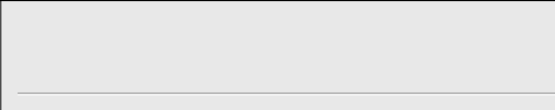
Có thể chạy được trên CPU S7-300 và S7-400 kết nối WinCC; kết hợp với các bộ điều khiển PID thông thường và mạng nơron (neural networks).

NeuroSystems

Phần mềm *neuro Systems* cho các mạng nơron (neural networks) gồm có các khối chuẩn cần cho điều khiển và các công cụ cấu hình. Các hệ Neuro Sytems sử dụng lý tưởng trong điều khiển và mô phỏng các quá trình mà bạn biết rất ít hoặc không biết gì đặc tuyến của chúng. Từ các giá trị đo được hiện có *neuro Systems* sẽ nhận biết các đặc tuyến điển hình của quá trình hoặc các đặc điểm quan trọng của các giá trị đo được, chẳng hạn như nhận biết mẫu.

Các chức năng mà bạn có thể thực hiện bằng công cụ cấu hình là: cài đặt các thông số cho mạng neural Networks (xem xét dữ liệu và xác định phạm vi giá trị), mô phỏng các tín hiệu vào nhờ máy vẽ đường cong, hiển thị 3 chiều của các đặc tính của mạng thần kinh và quản lý các đáp ứng vọt lố của 4 tín hiệu vào ra (máy vẽ đường cong với khả năng lưu trữ dữ liệu vào đĩa cứng để xử lý mở rộng bằng các ứng dụng chuẩn).

Có thể chạy được trên CPU S7-300 (từ CPU 314 trở lên) và S7-400 liên kết với WinCC; kết hợp với Fuzzy Control ++.



Khi việc xử lý in hoàn tất thì chỉ mục tài liệu cũng được in. Trong hình là thể hiện các đối tượng đã in. Bạn có thể xác định trình bày cho chỉ mục tài liệu tương tự như đối tượng cần in.

Hình 3.21 Dùng wizard chọn đối tượng in

OB1 - <offline>

==

Name: CPU 1514-2 DP

Family: 1500 07

Author: Chau Chi Duc

Version: 1.0

Block version: 1

Time stamp code: 10/04/2008 08:41:50 PM

Interface: CP 1543-1 DP, CP 1543-1 DP

Lengths (block/logic/data): 00120 00124 00120

Block	Block Name	Address	Comment
OB1	OB1	0.0	
FC1	FC1	0.0	
FC2	FC2	0.0	
FC3	FC3	0.0	
FC4	FC4	0.0	
FC5	FC5	0.0	
FC6	FC6	0.0	
FC7	FC7	0.0	
FC8	FC8	0.0	
FC9	FC9	0.0	
FC10	FC10	0.0	
FC11	FC11	0.0	
FC12	FC12	0.0	
FC13	FC13	0.0	
FC14	FC14	0.0	
FC15	FC15	0.0	
FC16	FC16	0.0	
FC17	FC17	0.0	
FC18	FC18	0.0	
FC19	FC19	0.0	
FC20	FC20	0.0	
FC21	FC21	0.0	
FC22	FC22	0.0	
FC23	FC23	0.0	
FC24	FC24	0.0	
FC25	FC25	0.0	
FC26	FC26	0.0	
FC27	FC27	0.0	
FC28	FC28	0.0	
FC29	FC29	0.0	
FC30	FC30	0.0	
FC31	FC31	0.0	
FC32	FC32	0.0	
FC33	FC33	0.0	
FC34	FC34	0.0	
FC35	FC35	0.0	
FC36	FC36	0.0	
FC37	FC37	0.0	
FC38	FC38	0.0	
FC39	FC39	0.0	
FC40	FC40	0.0	
FC41	FC41	0.0	
FC42	FC42	0.0	
FC43	FC43	0.0	
FC44	FC44	0.0	
FC45	FC45	0.0	
FC46	FC46	0.0	
FC47	FC47	0.0	
FC48	FC48	0.0	
FC49	FC49	0.0	
FC50	FC50	0.0	
FC51	FC51	0.0	
FC52	FC52	0.0	
FC53	FC53	0.0	
FC54	FC54	0.0	
FC55	FC55	0.0	
FC56	FC56	0.0	
FC57	FC57	0.0	
FC58	FC58	0.0	
FC59	FC59	0.0	
FC60	FC60	0.0	
FC61	FC61	0.0	
FC62	FC62	0.0	
FC63	FC63	0.0	
FC64	FC64	0.0	
FC65	FC65	0.0	
FC66	FC66	0.0	
FC67	FC67	0.0	
FC68	FC68	0.0	
FC69	FC69	0.0	
FC70	FC70	0.0	
FC71	FC71	0.0	
FC72	FC72	0.0	
FC73	FC73	0.0	
FC74	FC74	0.0	
FC75	FC75	0.0	
FC76	FC76	0.0	
FC77	FC77	0.0	
FC78	FC78	0.0	
FC79	FC79	0.0	
FC80	FC80	0.0	
FC81	FC81	0.0	
FC82	FC82	0.0	
FC83	FC83	0.0	
FC84	FC84	0.0	
FC85	FC85	0.0	
FC86	FC86	0.0	
FC87	FC87	0.0	
FC88	FC88	0.0	
FC89	FC89	0.0	
FC90	FC90	0.0	
FC91	FC91	0.0	
FC92	FC92	0.0	
FC93	FC93	0.0	
FC94	FC94	0.0	
FC95	FC95	0.0	
FC96	FC96	0.0	
FC97	FC97	0.0	
FC98	FC98	0.0	
FC99	FC99	0.0	
FC100	FC100	0.0	
FC101	FC101	0.0	
FC102	FC102	0.0	
FC103	FC103	0.0	
FC104	FC104	0.0	
FC105	FC105	0.0	
FC106	FC106	0.0	
FC107	FC107	0.0	
FC108	FC108	0.0	
FC109	FC109	0.0	
FC110	FC110	0.0	
FC111	FC111	0.0	
FC112	FC112	0.0	
FC113	FC113	0.0	
FC114	FC114	0.0	
FC115	FC115	0.0	
FC116	FC116	0.0	
FC117	FC117	0.0	
FC118	FC118	0.0	
FC119	FC119	0.0	
FC120	FC120	0.0	
FC121	FC121	0.0	
FC122	FC122	0.0	
FC123	FC123	0.0	
FC124	FC124	0.0	
FC125	FC125	0.0	
FC126	FC126	0.0	
FC127	FC127	0.0	
FC128	FC128	0.0	
FC129	FC129	0.0	
FC130	FC130	0.0	
FC131	FC131	0.0	
FC132	FC132	0.0	
FC133	FC133	0.0	
FC134	FC134	0.0	
FC135	FC135	0.0	
FC136	FC136	0.0	
FC137	FC137	0.0	
FC138	FC138	0.0	
FC139	FC139	0.0	
FC140	FC140	0.0	
FC141	FC141	0.0	
FC142	FC142	0.0	
FC143	FC143	0.0	
FC144	FC144	0.0	
FC145	FC145	0.0	
FC146	FC146	0.0	
FC147	FC147	0.0	
FC148	FC148	0.0	
FC149	FC149	0.0	
FC150	FC150	0.0	
FC151	FC151	0.0	
FC152	FC152	0.0	
FC153	FC153	0.0	
FC154	FC154	0.0	
FC155	FC155	0.0	
FC156	FC156	0.0	
FC157	FC157	0.0	
FC158	FC158	0.0	
FC159	FC159	0.0	
FC160	FC160	0.0	
FC161	FC161	0.0	
FC162	FC162	0.0	
FC163	FC163	0.0	
FC164	FC164	0.0	
FC165	FC165	0.0	
FC166	FC166	0.0	
FC167	FC167	0.0	
FC168	FC168	0.0	
FC169	FC169	0.0	
FC170	FC170	0.0	
FC171	FC171	0.0	
FC172	FC172	0.0	
FC173	FC173	0.0	
FC174	FC174	0.0	
FC175	FC175	0.0	
FC176	FC176	0.0	
FC177	FC177	0.0	
FC178	FC178	0.0	
FC179	FC179	0.0	
FC180	FC180	0.0	
FC181	FC181	0.0	
FC182	FC182	0.0	
FC183	FC183	0.0	
FC184	FC184	0.0	
FC185	FC185	0.0	
FC186	FC186	0.0	
FC187	FC187	0.0	
FC188	FC188	0.0	
FC189	FC189	0.0	
FC190	FC190	0.0	
FC191	FC191	0.0	
FC192	FC192	0.0	
FC193	FC193	0.0	
FC194	FC194	0.0	
FC195	FC195	0.0	
FC196	FC196	0.0	
FC197	FC197	0.0	
FC198	FC198	0.0	
FC199	FC199	0.0	
FC200	FC200	0.0	
FC201	FC201	0.0	
FC202	FC202	0.0	
FC203	FC203	0.0	
FC204	FC204	0.0	
FC205	FC205	0.0	
FC206	FC206	0.0	
FC207	FC207	0.0	
FC208	FC208	0.0	
FC209	FC209	0.0	
FC210	FC210	0.0	
FC211	FC211	0.0	
FC212	FC212	0.0	
FC213	FC213	0.0	
FC214	FC214	0.0	
FC215	FC215	0.0	
FC216	FC216	0.0	
FC217	FC217	0.0	
FC218	FC218	0.0	
FC219	FC219	0.0	
FC220	FC220	0.0	
FC221	FC221	0.0	
FC222	FC222	0.0	
FC223	FC223	0.0	
FC224	FC224	0.0	
FC225	FC225	0.0	
FC226	FC226	0.0	
FC227	FC227	0.0	
FC228	FC228	0.0	
FC229	FC229	0.0	
FC230	FC230	0.0	
FC231	FC231	0.0	
FC232	FC232	0.0	
FC233	FC233	0.0	
FC234	FC234	0.0	
FC235	FC235	0.0	
FC236	FC236	0.0	
FC237	FC237	0.0	
FC238	FC238	0.0	
FC239	FC239	0.0	
FC240	FC240	0.0	
FC241	FC241	0.0	
FC242	FC242	0.0	
FC243	FC243	0.0	
FC244	FC244	0.0	
FC245	FC245	0.0	
FC246	FC246	0.0	
FC247	FC247	0.0	
FC248	FC248	0.0	
FC249	FC249	0.0	
FC250	FC250	0.0	
FC251	FC251	0.0	
FC252	FC252	0.0	
FC253	FC253	0.0	
FC254	FC254	0.0	
FC255	FC255	0.0	
FC256	FC256	0.0	
FC257	FC257	0.0	
FC258	FC258	0.0	
FC259	FC259	0.0	
FC260	FC260	0.0	
FC261	FC261	0.0	
FC262	FC262	0.0	
FC263	FC263	0.0	
FC264	FC264	0.0	
FC265	FC265	0.0	
FC266	FC266	0.0	
FC267	FC267	0.0	
FC268	FC268	0.0	
FC269	FC269	0.0	
FC270	FC270	0.0	
FC271	FC271	0.0	
FC272	FC272	0.0	
FC273	FC273	0.0	
FC274	FC274	0.0	
FC275	FC275	0.0	
FC276	FC276	0.0	
FC277	FC277	0.0	
FC278	FC278	0.0	
FC279	FC279	0.0	
FC280	FC280	0.0	
FC281	FC281	0.0	
FC282	FC282	0.0	
FC283	FC283	0.0	
FC284	FC284	0.0	
FC285	FC285	0.0	
FC286	FC286	0.0	
FC287	FC287	0.0	
FC288	FC288	0.0	
FC289	FC289	0.0	
FC290	FC290	0.0	
FC291	FC291	0.0	
FC292	FC292	0.0	
FC293	FC293	0.0	
FC294	FC294	0.0	
FC295	FC295	0.0	
FC296	FC296	0.0	
FC297	FC297	0.0	
FC298	FC298	0.0	
FC299	FC299	0.0	
FC300	FC300	0.0	
FC301	FC301	0.0	
FC302	FC302	0.0	
FC303	FC303	0.0	
FC304	FC304	0.0	
FC305	FC305	0.0	
FC306	FC306	0.0	
FC307	FC307	0.0	
FC308	FC308	0.0	
FC309	FC309	0.0	
FC310	FC310	0.0	
FC311	FC311	0.0	
FC312	FC312	0.0	
FC313	FC313	0.0	
FC314	FC314	0.0	
FC315	FC315	0.0	
FC316	FC316	0.0	
FC317	FC317	0.0	
FC318	FC318	0.0	
FC319	FC319	0.0	
FC320	FC320	0.0	
FC321	FC321	0.0	
FC322	FC322	0.0	
FC323	FC323	0.0	
FC324	FC324	0.0	
FC325	FC325	0.0	
FC326	FC326	0.0	
FC327	FC327	0.0	
FC328	FC328	0.0	
FC329	FC329	0.0	
FC330	FC330	0.0	
FC331	FC331	0.0	
FC332	FC332	0.0	
FC333	FC333	0.0	
FC334	FC334	0.0	
FC335	FC335	0.0	
FC336	FC336	0.0	
FC337	FC337	0.0	
FC338	FC338	0.0	
FC339	FC339	0.0	
FC340	FC340	0.0	
FC341	FC341	0.0	
FC342	FC342	0.0	
FC343	FC343	0.0	
FC344	FC344	0.0	
FC345	FC345	0.0	
FC346	FC346	0.0	
FC347	FC347	0.0	
FC348	FC348	0.0	
FC349	FC349	0.0	
FC350	FC350	0.0	
FC351	FC351	0.0	
FC352	FC352	0.0	
FC353	FC353	0.0	
FC354	FC354	0.0	
FC355	FC355	0.0	
FC356	FC356	0.0	
FC357	FC35		

Hình 3.22 ví dụ về trình bày in

Hình trên minh họa ví dụ về trình bày in của OB 1 theo khổ giấy DIN A4 dạng đứng. Bạn có thể chọn khổ giấy A3 thẳng đứng hoặc nằm ngang hoặc theo trình bày của bạn.

3.24 Kết Nối Mạng Điện Thoại Với Teleservice

Phần mềm *TeleService* dùng để kết nối thiết bị lập trình hoặc PC tới SIMATIC PLC thông qua mạng điện thoại, cho phép bạn quản lý, điều khiển, vận hành từ xa thiết bị hoặc nhà máy từ một trung tâm.

TeleService sử dụng Microsoft Window 95/98 hoặc Window NT trên thiết bị lập trình hoặc PC chuẩn. Việc kết nối có thể thực hiện bằng kết nối vật lý hoặc qua cổng COM ảo. Tại PLC cuối cùng cần phải có một bộ chuyển đổi TS cho phép thiết lập kết nối tới MPI hoặc PROFIBUS Subnet. TeleService là một modem độc lập và làm việc được với tất cả các modem tương thích cài đặt trong Windows control Panel. Phụ thuộc vào modem mà bạn có thể sử dụng mạng điện thoại analog hay digital hoặc một radio network.

Sử dụng TeleService như sau:

➤ **Hoạt động tại PG/PC cuối**

Cài đặt TeleService, cài đặt một modem cục bộ trong Windows Control Panel, cài đặt cho giao tiếp với PG/PC, chọn TeleService và đặc tính modem để đạt được liên kết tối ưu.

➤ **Cài đặt tại nhà máy (plant) cuối**

Cài đặt các thông số cho TS adapter

➤ **Sử dụng TeleService**

Tạo một “telephone book“, sử dụng các chức năng của STEP 7 để thiết lập kết nối và công việc bảo trì tại nhà máy, chẳng hạn như phân tích lỗi, chỉnh sửa, nâng cấp và cập nhật.

Sử dụng kết nối trực tiếp để cài đặt thông số cho bộ TS adapter. Bạn cũng có thể sử dụng cổng COM để kết nối trực tiếp TS adapter từ PC tới mạng MPI hoặc PROFIBUS.

SIMATIC PLC cũng có thể khởi tạo thiết lập kết nối. Tại PG/PC cuối bạn cần có phần mềm *PRODAVE MPI* cho phép trao đổi dữ liệu giữa SIMATIC PLC và một chương trình PC chuẩn chẳng hạn MS Excel.



4 Ngôn Ngữ Lập Trình

Bạn sử dụng ngôn ngữ lập trình SIMATIC để viết chương trình. Có nhiều ngôn ngữ lập trình và phương pháp lập trình để bạn chọn phù hợp với nhu cầu của mình. Trọn bộ STEP 7 có 3 ngôn ngữ lập trình (LAD, FBD, STL). Các phần mềm khác hiện có là tùy chọn.

Ngôn ngữ lập trình dạng đồ họa đặc biệt phù hợp cho thể hiện các bit logic trong dạng soạn thảo sơ đồ logic hình thang (Ladder logic LAD) hoặc sơ đồ mạch điện tử (Function Block diagram FBD). Lập trình dạng liệt kê câu lệnh STL cho phép giải quyết các biến phức tạp hay sử dụng đặt địa chỉ gián tiếp.

SCL là ngôn ngữ lập trình dùng cho các chương trình có giải thuật phức tạp và giải quyết một số lượng lớn cấu trúc dữ liệu. Bạn viết các khối chương trình riêng lẻ giống như thực hiện trong các ngôn ngữ lập trình LAD, FBD, STL. Bạn cũng có thể viết toàn bộ chương trình trong SCL.

Sử dụng CFC để liên kết các khối tương tự như vẽ một biểu đồ. Các

Hình 4.1 Ngôn ngữ lập trình S7

khối có thể được soạn thảo từ bất kỳ ngôn ngữ nào (LAD, FBD, STL hoặc SCL), CFC cần có SCL để có thể xử lý được khối.

GRAPH là phương pháp lập trình dùng cho các hệ thống điều khiển trình tự. Bạn có thể dùng LAD hoặc STL để viết các điều kiện chuyển từ bước này sang bước kế tiếp của chương trình. Các nhánh biến đổi hoặc song song mở rộng phạm vi xử lý tuyến tính.

Cuối cùng là phần mềm HiGraph cho phép biểu diễn các vấn đề điều khiển dưới dạng biểu đồ trạng thái. Ở đây điều cần nhấn mạnh không phải là việc xử lý tuyến tính mà là việc thay đổi trình tự trạng thái và các chuyển dịch điển hình của các quá trình bất đồng bộ

4.1 Ngôn Ngữ Lập Trình Cơ Bản LAD, FBD và STL

Công cụ STEP 7 bao gồm một trình soạn thảo dùng cho các ngôn ngữ lập trình Ladder Logic (LAD), Function Block Diagram (FBD) và Statement List (STL). Các ngôn ngữ lập trình này đáp ứng chuẩn IEC 1131-3 (DIN EN 6.1131-3). Bạn sử dụng các ngôn ngữ lập trình này để viết chương trình của mình.

LAD và FBD là các ngôn ngữ lập trình dạng biểu đồ: trong LAD bạn tạo chương trình bằng cách kết nối các tiếp điểm nối tiếp hoặc song song và trong FBD là kết nối các hộp AND và OR. STL là ngôn ngữ lập trình dạng văn bản: mô tả nhiệm vụ điều khiển ở dạng danh mục liệt kê câu lệnh.

Các ngôn ngữ lập trình này không những cho phép bạn làm việc với các trạng thái tín hiệu nhị phân mà còn dễ dàng xử lý các giá trị digital với các định dạng khác nhau ở bất cứ ngôn ngữ nào. Chẳng hạn bạn có thể so sánh số và phân số hoặc thực hiện các phép tính số học cơ bản. Mỗi một ngôn ngữ lập trình cung cấp các chức năng cho phép bạn xây dựng chương trình theo yêu cầu riêng. Có nghĩa là bạn có thể viết toàn bộ chương trình bằng một trong số các ngôn ngữ lập trình.

Bạn có thể dùng các phép toán logic và chức năng Set/Reset để viết chương trình điều khiển cho PLC thực hiện các chức năng của công tắc tơ hoặc điều khiển rơ le hoặc các mạch điện đã nối dây. Đối với các công việc điều khiển phức tạp hơn bạn có thể xử lý các giá trị digital bằng các chức năng so sánh, chức năng toán học và số học, chức năng chuyển đổi và xoay, chức năng nhảy để thay đổi chương trình cho phù hợp với nhu cầu.

Bạn viết chương trình trong thư mục có tên “blocks“. Khối tổ chức (*Organization Blocks*) tạo giao tiếp giữa hệ điều hành và chương trình người

dùng. Khi một sự kiện xảy ra, hệ điều hành CPU sẽ gọi các khối này và đánh dấu điểm bắt đầu của chương trình theo cấp ưu tiên hoặc cấp xử lý. Các khối dạng *Function Block* và *Function* có thể được gọi và được xử lý từ khối tổ chức.



Soạn thảo chương trình trong Ladder Logic (LAD) giống như kết nối một sơ đồ logic relay. Trạng thái tín hiệu nhị phân được thể hiện bằng các *tiếp điểm* nối tiếp hay song song. Kết thúc mỗi một đường mạch là một cuộn dây (coil). Các chức năng phức tạp được thể hiện bằng các *hộp*.

Soạn thảo chương trình trong Ladder Logic (LAD) giống như vẽ một sơ đồ mạch điện tử. Thực hiện phép toán logic bằng kết nối *AND* và *OR function* và thể hiện bằng các *hộp đơn giản*. Các *hộp phức tạp* dùng cho các phép tính có các biến digital.

Soạn thảo chương trình trong Statement List (STL) giống như một bảng liệt kê. Mỗi một hàng là một câu lệnh xác định chức năng cần thực hiện, có khi là một địa chỉ cho biết chức năng nào được thực hiện.

Hình 4.2 Các ngôn ngữ lập trình cơ bản Ladder Logic, Function Block Diagram, Statement List.

Tất cả các ngôn ngữ lập trình đều có chức năng gọi khối. Ngôn ngữ lập trình dạng biểu đồ sử dụng các *hộp* có các ngõ vào và ra để truyền và nhận giá trị từ khối được gọi. STL có lệnh CALL để gọi khối và sử dụng một bảng liệt kê thông số để truyền và nhận giá trị của khối được gọi.

Ngoài ra còn có các ưu điểm khác của khối lập trình như rõ ràng, các phần chương trình có thể cài đặt và chứa thông số. Phương pháp này cho phép bạn tự chọn ngôn ngữ lập trình cho mỗi khối, có nghĩa là bạn không nhất thiết luôn luôn sử dụng một ngôn ngữ lập trình cho toàn bộ chương trình. Chẳng hạn bạn có thể viết các khối cho phép toán logic bằng LAD bởi vì bạn thích sự sắp xếp dạng đồ họa các tiếp điểm và viết các khối xử lý dữ liệu bằng STL bởi vì bạn có thể sử dụng hiệu quả địa chỉ các thanh ghi để đặt địa chỉ dữ liệu gián tiếp trong chương trình. Mỗi một khối có thể gọi nhiều khối được viết bằng một ngôn ngữ khác.

Bạn có thể luôn luôn chuyển đổi khối được viết bằng LAD, FBD sang STL. Ở điều kiện nhất định bạn có thể xem các khối được viết ở dạng LAD sang FBD và ngược lại: các phần chương trình không thể hiển thị dạng biểu đồ sẽ được hiển thị dạng STL. Chương trình soạn thảo bằng LAD, FBD hoặc STL có thể “được biên dịch lại” (decompiled), có nghĩa là bạn có thể luôn luôn tạo ra được một khối từ mã máy, chẳng hạn như khi bạn muốn xem một chương trình người dùng trong CPU mà không có dữ liệu offline.

Trong soạn thảo chương trình dùng LAD và FBD thì trình soạn thảo sẽ kiểm tra tất cả các dữ liệu nhập. Khi bạn lưu một khối, nó được tự động dịch sang mã máy. Chỉ có thể lưu trữ một khối khi nó không có lỗi.

Ở dạng soạn thảo STL bạn có thể tạo ra các tập tin nguồn. Đây là một tập tin văn bản có các ký tự định dạng ASCII. Bạn có thể lưu một chương trình nguồn bất cứ lúc nào thậm chí chưa soạn thảo xong hay chứa lỗi. Bạn cũng có thể viết một chương trình nguồn dùng trình soạn thảo khác và sau đó “nhập” vào chương trình của mình. Để tạo ra mã máy thì chương trình nguồn phải được biên dịch.

Để thử chương trình mà bạn đã viết bằng các ngôn ngữ lập trình trong các khối thì đối với dạng LAD bạn có thể thấy hoạt động của chương trình trên các đoạn mạch, ở dạng FBD bạn sẽ thấy kết quả các phép toán logic hiển thị có màu khác, ở dạng STL cho phép bạn thử chương trình theo từng câu lệnh và quan sát sự thay đổi các giá trị nhị phân và số.

4.2 Các phép toán nhị phân

Các phép toán nhị phân thực hiện các phép toán đại số Boolean dựa trên các biến (signals). Các tín hiệu được xử lý và lưu trữ dạng dữ liệu BOOL. Đây là dạng dữ liệu cho một bit giả sử chỉ có 2 trạng thái là “1” và “0” hoặc “ON” và “OFF”. Một phép toán logic được thực hiện trên các tín hiệu ngõ vào và các ngõ ra (ví dụ được nối tới các contactor, van, đèn) sẽ được set hoặc reset.

Các phép toán Logic

Sơ đồ bên dưới cho thấy kết quả của phép toán logic. Tưởng tượng rằng 2 công tắc tạm thời được nối với các ngõ vào là “Input 1” và “Input 2” và mỗi ngõ ra được nối tới một đèn. Trong 1 *phép toán AND* hoặc *kết nối nối tiếp* thì đèn (“output 1”) sẽ sáng khi cả hai công tắc đều được tác động (đóng điện). Theo ngôn ngữ kỹ thuật điều khiển thì diễn tả như sau: nếu cả 2 trạng thái ngõ vào là “1” thì trạng thái ngõ ra cũng là “1”, phép toán logic là “đúng” (true). Một *phép toán OR* hoặc một *kết nối song song* các tiếp

điểm luôn luôn đúng khi có ít nhất một ngõ vào có trạng thái “1”. Một chức năng AND hoặc OR có thể có hơn 2 ngõ vào, tất cả các ngõ vào sẽ được kiểm tra và sau đó phép toán AND hoặc OR được thực hiện đối với tất cả các ngõ vào.

Phép toán thứ 3 là Exclusive OR, chức năng này hiện có ở dạng FBD và STL. Trong dạng LAD bạn sử dụng các kết nối nối tiếp và song để thực hiện phép toán này. Nếu phép toán Exclusive OR có nhiều ngõ vào thì phép toán là đúng khi số các ngõ vào có trạng thái “1” là số lẻ.

Bạn có thể kết hợp các phép toán nhị phân, chẳng hạn bạn có thể OR ngõ ra của 2 cổng AND để tạo một mạch phức tạp. Trong dạng soạn thảo biểu đồ bạn thực hiện các kết nối cần thiết bằng các sắp xếp dạng hình học, còn trong STL thì sử dụng các dấu ngoặc để sắp xếp thứ tự xử lý.



Phần hình chữ nhật có nghĩa là địa chỉ ngõ ra có trạng thái “1” (ví dụ khi tác động công tắc). Phần còn lại có nghĩa địa chỉ ngõ ra có trạng thái “0”

Phép AND chỉ đúng khi tất cả các ngõ vào có trạng thái “1”

Phép toán OR đúng khi có ít nhất một ngõ vào có trạng thái “1”

Phép Exclusive OR đúng khi 2 ngõ vào có trạng thái khác nhau.

Hình 4.3 Các phép toán logic nhị phân

Phủ định

Tất cả các ngõ vào và ra của một phép toán có thể bị “phủ định“, có nghĩa là trạng thái tín hiệu thay đổi từ “1” sang “0” hoặc “0” sang “1” khi phép toán được thực hiện. Chẳng hạn khi một công việc sẽ được thực hiện bằng tín hiệu “tác động mức 0“, ví dụ khi bạn muốn một ngõ ra có trạng thái “1” khi cả 2 công tắc có trạng thái “0”(không tác động).







Trong lập trình dạng Ladder Logic (LAD), bạn viết chương trình của mình bằng cách sắp xếp các phần tử theo dạng biểu đồ. Các phần tử chủ yếu gồm các tiếp điểm, cuộn dây, hộp cho phép bạn tạo ra một sơ đồ điều khiển rơ le dạng hình thang.

Bạn viết chương trình của mình bằng các “đường mạch” bắt đầu từ bên trái. Chọn một điểm trên đường mạch mà bạn muốn chèn phần tử chương trình, tùy chọn cách chọn phần tử chương trình bằng cách nhấn phím chức năng tương ứng, nhấp chuột vào mục liên quan trong thanh công cụ hoặc chọn phần tử chương trình từ Program Elements catalog. Kết thúc mỗi đoạn mạch bằng một phần tử cuộn dây hoặc hộp. Cũng có thể kết thúc đường mạch bằng một “nhánh hình T” nối tới các phần tử cuộn dây và hộp khác, trước mỗi phần tử này có các phép toán logic.

Trình soạn thảo LAD tạo ra một network dựa trên một “đường mạch chính”. Đây là đường mạch trên cùng bắt đầu việc soạn thảo từ phía bên trái và phải kết thúc bằng một cuộn dây hoặc hộp. Bạn có thể đặt bất cứ phần tử chương trình nào vào đường mạch chính. Đối với các nhánh song song đôi khi có sự giới hạn trong việc chọn các phần tử chương trình. Bạn không được nối “ngắn mạch” một phần tử chương trình LAD bằng một nhánh song song “rỗng” và dòng điện không được phép chạy qua một phần tử chương trình từ phải qua trái (một nhánh song song phải kết thúc trong cùng một nhánh mà nó bắt đầu).

Hầu hết các phần tử chương trình yêu cầu bạn phải nhập địa chỉ (biến). Phương pháp tốt nhất là sắp xếp tất cả các phần tử chương trình trước rồi sau đó mới đặt địa chỉ. Bạn có thể sử dụng địa chỉ tuyệt đối, ví dụ I 1.0 hoặc địa chỉ ký hiệu, ví dụ “Manual_mode” cho phần tử.

Mỗi một network gồm có một “đường mạch” mà một phép toán logic nối tới một cuộn dây hay hộp. Nếu muốn tạo một đường mạch khác, bạn tạo ra một network phía dưới network hiện hành. Bạn có thể cho mỗi network một tên và nhập chú thích chi tiết để giải thích chức năng của network.

Tiếp điểm

Sử dụng tiếp điểm để kiểm tra trạng thái tín hiệu của các địa chỉ, chẳng hạn các ngõ vào. Bằng cách sắp xếp các tiếp điểm nối tiếp hoặc song song bạn có thể thực hiện phép toán logic. Có 2 dạng tiếp điểm: *tiếp điểm không đảo trạng thái* (Normally Open Contacts) và *tiếp điểm đảo trạng thái* (Normally Closed Contacts). Với tiếp điểm không đảo trạng thái thì địa chỉ của nó được kiểm tra đối với trạng thái tín hiệu “1”, tiếp điểm sẽ “đóng”



Các hộp được sử dụng làm phần tử chương trình có các chức năng non-binary. LAD có 2 dạng “hộp chuẩn”: các hộp không có các thông số EN/ENO (ví dụ chức năng Set/Reset, timer, counter và các hộp so sánh) và các hộp có các thông số EN/ENO (ví dụ hộp MOVE, các chức năng toán học và số học, chuyển đổi dạng dữ liệu). Nếu gọi một khối logic chẳng hạn một khối hàm thì trình soạn thảo LAD chỉ hiển thị lệnh gọi ở dạng hộp không có thông số EN/ENO.

Thông số EN/ENO

Sử dụng ngõ vào EN (Enable input) để điều khiển việc xử lý một hộp. Chức năng chỉ được thực hiện khi có “dòng điện chạy vào ngõ vào này”. Nếu khối được xử lý mà không có lỗi thì sẽ “có dòng điện” tại ngõ ra ENO (Enable output). Kết nối các hộp bằng EN và ENO nối tiếp để đảm bảo một hộp chỉ có thể được xử lý khi hộp trước nó được xử lý mà không có lỗi hoặc cho phép bạn sử dụng một tiếp điểm để đóng/ngắt một “đoạn mạch chính” chứa các hộp mắc nối tiếp.

Mạch nối tiếp và song song trong LAD

Một đoạn mạch có thể có một tiếp điểm hoặc chứa nhiều tiếp điểm kết nối với nhau. Hình ví dụ bên dưới cho thấy một mạch nối tiếp có tiếp điểm thường mở (NO) *#Manual_ON* và tiếp điểm thường đóng (NC) *#Automatic_mode*, dòng điện chạy qua đoạn mạch khi *#Manual_ON* có trạng thái tín hiệu “1” và *#Automatic_mode* có trạng thái “0”. Song song với nhánh này là một mạch nối tiếp có tiếp điểm thường mở NO *#Auto_ON* và *#Automatic_mode*. Hai mạch nối tiếp này nối song song với nhau và nối tới ngõ vào Set của hộp Set/Reset.

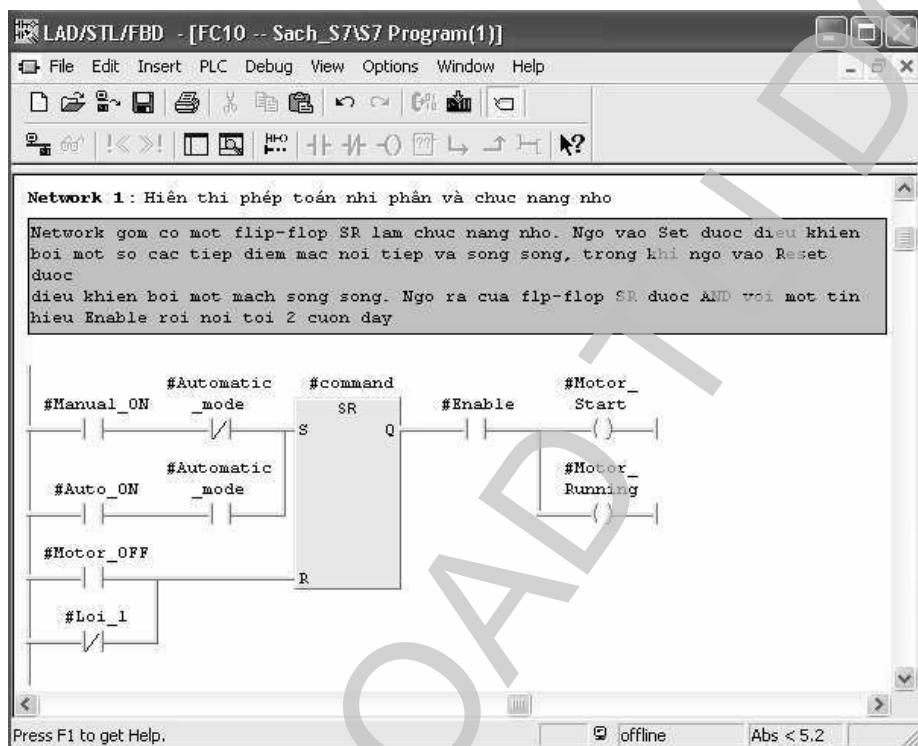
Ngõ vào Reset được nối tới 2 tiếp điểm mắc song song. Chức năng Reset được thực hiện nếu địa chỉ *#Motor_OF* có trạng thái tín hiệu “1” hoặc địa chỉ *#Fault* có trạng thái tín hiệu “0” (tác động mức 0).

Trong một đoạn mạch bạn có thể sử dụng tiếp điểm NOT để phủ định kết quả phép tính logic (cho “dòng điện chạy qua”), có nghĩa là trạng thái tín hiệu “1” trở thành “0” và ngược lại.

Một đoạn mạch luôn phải kết thúc thường là một cuộn dây. Một cuộn dây không có chữ cái hay ký hiệu bên trong cài đặt RLO cho một địa chỉ bit.

Trong ví dụ bên dưới, ngõ ra của hộp Set/Reset được nối tới tiếp điểm NO *#Enable*. Phía sau là một “nhánh hình T”, cho phép bạn điều khiển

nhiều ngõ ra trong một đoạn mạch. Trong ví dụ thì các cuộn dây $\#M_start$ khởi động động cơ và $\#M_running$ dừng hiển thị.



Hình 4.8 Ví dụ về nối nối tiếp và song song trong LAD.

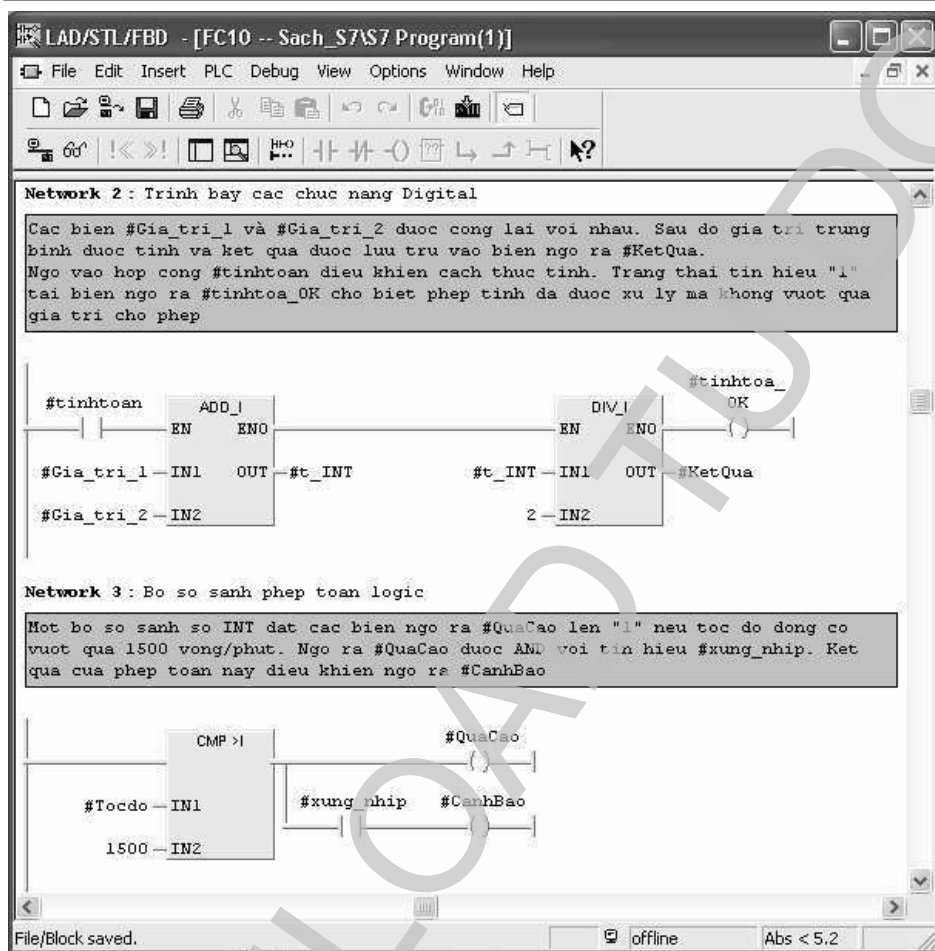
Các chức năng digital trong LAD

Các chức năng digital được biểu diễn ở dạng các hộp. Một hộp digital có thể xử lý một hoặc 2 giá trị ngõ vào và chuyển kết quả tới ngõ ra.

Ngõ vào EN của một hộp quyết định nó có được xử lý hay không: Nếu ngõ vào EN có trạng thái tín hiệu "1" (có dòng điện chạy qua ngõ vào) thì hộp được xử lý. Nếu chức năng được thực hiện mà không có lỗi thì ngõ ra ENO có trạng thái "1". Nếu không hoặc nếu chức năng không được xử lý vì $EN = "0"$ thì ngõ ra ENO có trạng thái "0".

Bạn có thể kết nối ENO và EN để ngõ vào EN của hộp thứ nhất có thể điều khiển xử lý tất cả các hộp phía sau, ngõ ra ENO của hộp cuối cùng sẽ chỉ báo lỗi cho tất cả các hộp.

Bạn chọn các hộp từ Program Elements. Cũng có các hộp "trung tính" (neutral) cho phép bạn tự đặt nhãn theo chức năng mong muốn.



Hình 4.9 ví dụ về chức năng digital trong LAD.

4.6 Function Block Diagram (FBD)

Các phần tử chương trình FBD

Trong soạn thảo Function Block Diagram (FBD) bạn viết chương trình ứng dụng bằng cách kết nối các hộp. FBD cung cấp các hộp chức năng để thực hiện các phép toán logic theo trạng thái tín hiệu, các hộp đơn giản dùng xử lý kết quả của phép toán logic và các hộp phức tạp dùng cho các chức năng non-binary.

Bạn bắt đầu viết chương trình bằng cách chọn phần tử chương trình. Có thể chọn cách chọn phần tử chương trình phía bên trái – bằng cách nhấn các phím chức năng liên quan và chọn menu thích hợp hoặc chọn phần tử từ *Program Element catalog*. Trường hợp đơn giản nhất là kết nối một phép

toán logic tới một ngõ ra và sử dụng kết quả của phép toán logic để set hoặc reset một địa chỉ bit.

Trình soạn thảo FBD tạo ra các network từ trái sang phải, từ trên xuống dưới. Mỗi phép toán logic có một hộp ngõ ra, đơn giản nhất là cài đặt thông số RLO cho một địa chỉ bit. Có thể sử dụng “nhánh hình T” để tạo ra hơn một hộp ngõ ra cho một phép toán logic (“nhiều ngõ ra”). Việc chọn lựa các phần tử chương trình đôi khi bị giới hạn sau nhánh hình T.

Hầu hết các phần tử chương trình yêu cầu bạn phải nhập địa chỉ (biến). Phương pháp tốt nhất là sắp xếp tất cả các phần tử chương trình trước rồi sau đó mới đặt địa chỉ. Bạn có thể sử dụng địa chỉ tuyệt đối, ví dụ I 1.0 hoặc địa chỉ ký hiệu, ví dụ “Manual_mode” cho phần tử.

Mỗi một network chứa một phép toán logic: một phép toán logic gồm các chức năng được liên kết với nhau. Trước khi bắt đầu nhập phép toán logic kế tiếp, cần tạo ra một network phía dưới network hiện hành. Bạn có thể cho mỗi network một tên và nhập chú thích chi tiết để giải thích chức năng của network.

Chức năng nhị phân

Các chức năng nhị phân được sử dụng để kiểm tra trạng thái tín hiệu của các địa chỉ bit, chẳng hạn như các ngõ vào và thực hiện các phép toán nhị phân trên các ngõ vào. Các chức năng hiện có là AND, OR và Exclusive OR (XOR). Tất cả các chức năng trên có thể có hơn 2 ngõ vào. Kết nối các hộp chức năng để tạo ra các phép toán logic phức tạp hơn trong một network.

Nếu chèn một vòng tròn ($\neg$) vào một ký hiệu chức năng thì trạng thái tín hiệu kiểm tra của địa chỉ bị phủ định. Chẳng hạn kết quả kiểm tra trạng thái là “1” khi trạng thái tín hiệu tại địa chỉ là “0”. Bạn cũng có thể đặt một “vòng tròn phủ định” tại ngõ vào của một hộp ngõ ra để cài đặt phủ định kết quả phép toán logic cho một địa chỉ bit.

Bạn có thể kiểm tra trạng thái tín hiệu không chỉ cho các ngõ vào mà còn cho tất cả các địa chỉ bit, bao gồm các bit trạng thái. Chẳng hạn, bạn có thể thực hiện phép toán nhị phân trên kết quả tính toán, như “kết quả = zero”, hoặc trên bit tràn, hoặc sử dụng kết quả phép toán nhị phân để set hay reset cho một ngõ ra.

**Hộp đơn giản**

Các hộp đơn giản tác động các địa chỉ bit như các ngõ ra. Chúng thường chỉ có một ngõ vào có các chữ cái hoặc ký hiệu. Có các hộp đơn giản dùng cho set và reset các địa chỉ bit, nhận biết cạnh tín hiệu, set hoặc reset timer hay các địa chỉ counter, gọi khối không có thông số, thực hiện chức năng nhảy trong chương trình...

Hộp phức tạp

Các hộp phức tạp sử dụng cho các phần tử chương trình có chức năng non-binary. Trong soạn thảo FBD có 2 dạng “hộp chuẩn”(standard boxes): hộp không có thông số EN/ENO (ví dụ chức năng Set/Reset, timer, counter và các hộp so sánh) và hộp có thông số EN/ENO (ví dụ hộp MOVE, chức năng toán học và số học, chuyển đổi dạng số liệu). Nếu gọi một khối logic chẳng hạn khối hàm thì FBD cũng hiển thị khối gọi dạng hộp dạng có các thông số EN/ENO.

Thông số EN/ENO

Sử dụng ngõ vào EN (enable input) để điều khiển việc xử lý một hộp. Chỉ khi nào kết quả của phép toán logic tại ngõ vào bằng “1” thì chức năng mới được thực hiện. Khi hộp được xử lý mà không có lỗi thì trạng thái tín hiệu ngõ ra ENO (enable output) là “1”. Bạn có thể nối các hộp bằng EN/ENO nối tiếp để đảm bảo một hộp chỉ được xử lý nếu hộp trước nó đã được xử lý mà không có lỗi hoặc cho phép sử dụng RLO để kích hoạt/không kích hoạt các hộp nối tiếp.

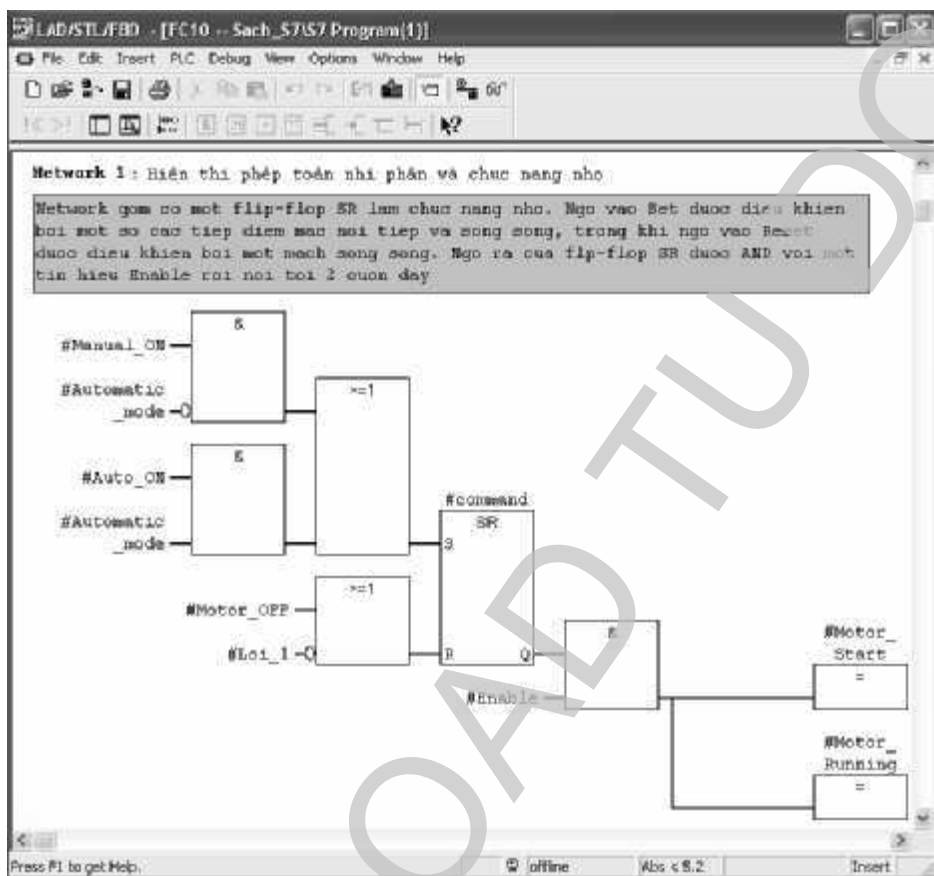
Phép toán logic bit trong FBD

FBD có các chức năng AND, OR, EXOR để thực hiện các phép toán logic dựa trên các trạng thái tín hiệu nhị phân. Bạn nhập các địa chỉ muốn kiểm tra trạng thái tín hiệu tại các ngõ vào của các chức năng này và sử dụng trong phép toán logic.

Trong FBD bạn viết một phép toán logic bit cho một network. Một phép toán logic có thể chỉ gồm một hộp chức năng hoặc có thể có nhiều hộp chức năng được nối với nhau. Trong ví dụ bên dưới hộp AND có ngõ vào trực tiếp *#Manual_ON* và ngõ vào phủ định *#Automatic_mode* được nối tới hộp OR có ngõ vào thứ 2 được nối tới 1 cổng AND. Nếu *#Automatic_mode* có trạng thái tín hiệu “0” thì *#Manual_ON* điều khiển ngõ vào Set của hộp SR, còn khi *#Automatic_mode* có trạng thái tín hiệu “1” thì *#Auto_ON* điều khiển ngõ vào Set của hộp SR.

Tạo phủ định kết quả kiểm tra hoặc kết quả của phép toán logic bằng cách chèn một vòng tròn (o) vào ngõ vào hoặc ngõ ra của một hộp.

Một phép toán logic phải có đầu cuối, thường thì nối tới một ngõ ra hay chức năng Set/Reset. Soạn thảo kiểu nhánh hình T cho phép bạn nối một phép toán logic tới nhiều hộp (đa ngõ ra). Cũng có thể thực hiện các phép toán logic phụ khác sau nhánh T và trước đầu cuối.



Hình 4.11 Ví dụ về chức năng nhị phân trong FBD

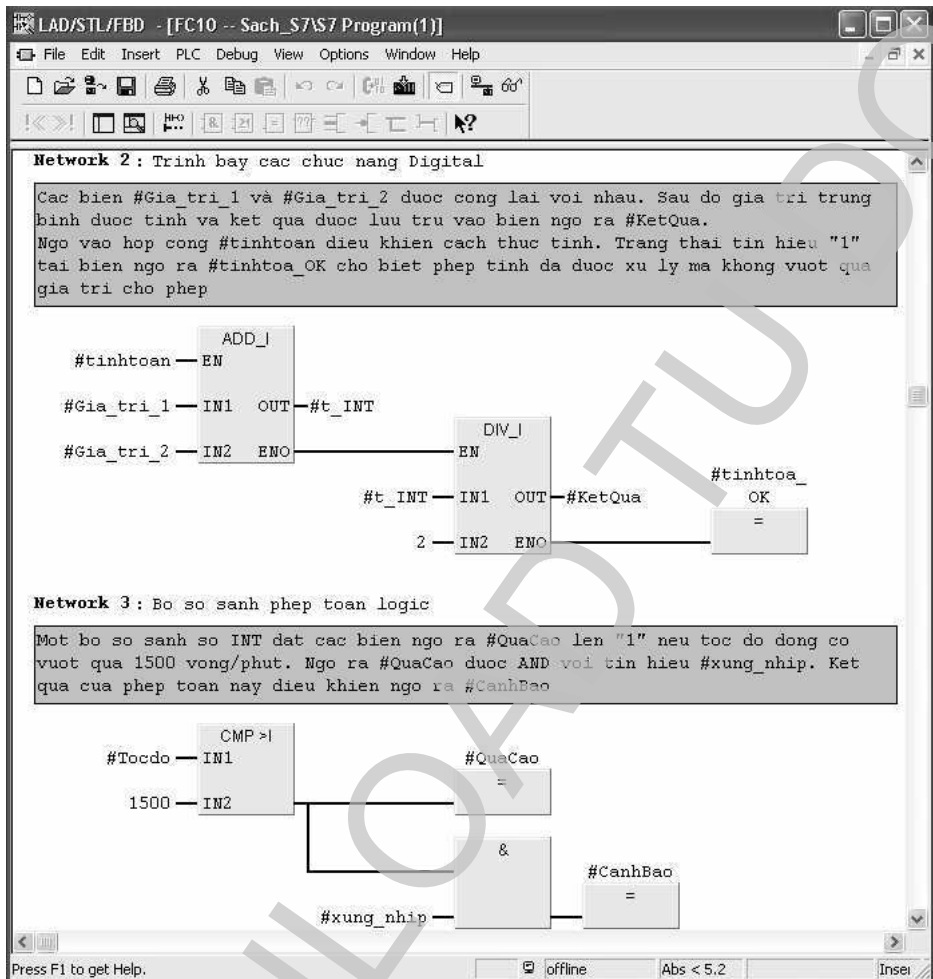
Chức năng số trong FBD

Các chức năng số được biểu diễn ở dạng các hộp. Một hộp số có thể xử lý một hoặc 2 giá trị ngõ vào và chuyển kết quả tới ngõ ra.

Ngõ vào EN của một hộp quyết định chức năng hộp có được xử lý hay không: nếu ngõ vào EN có trạng thái tín hiệu “1” thì hộp được xử lý. Nếu chức năng hộp được xử lý mà không có lỗi thì ngõ ra ENO có trạng thái tín hiệu “1”. Nếu có lỗi hoặc chức năng không được xử lý vì EN = “0” thì ngõ ra có trạng thái tín hiệu “0”.

Bạn có thể liên kết ENO và EN để ngõ vào EN của hộp đầu tiên điều khiển việc xử lý của tất cả các khối phía sau. Ngõ ra ENO của hộp cuối cùng sẽ chỉ báo lỗi cho tất cả các hộp.

Chọn các hộp từ *Program Elements catalog* ; cũng có các hộp “trung tính” cho phép bạn đặt nhãn cho chức năng bạn muốn.



Hình 4.12 Ví dụ về chức năng số trong FBD.

4.7 Statement List (STL)

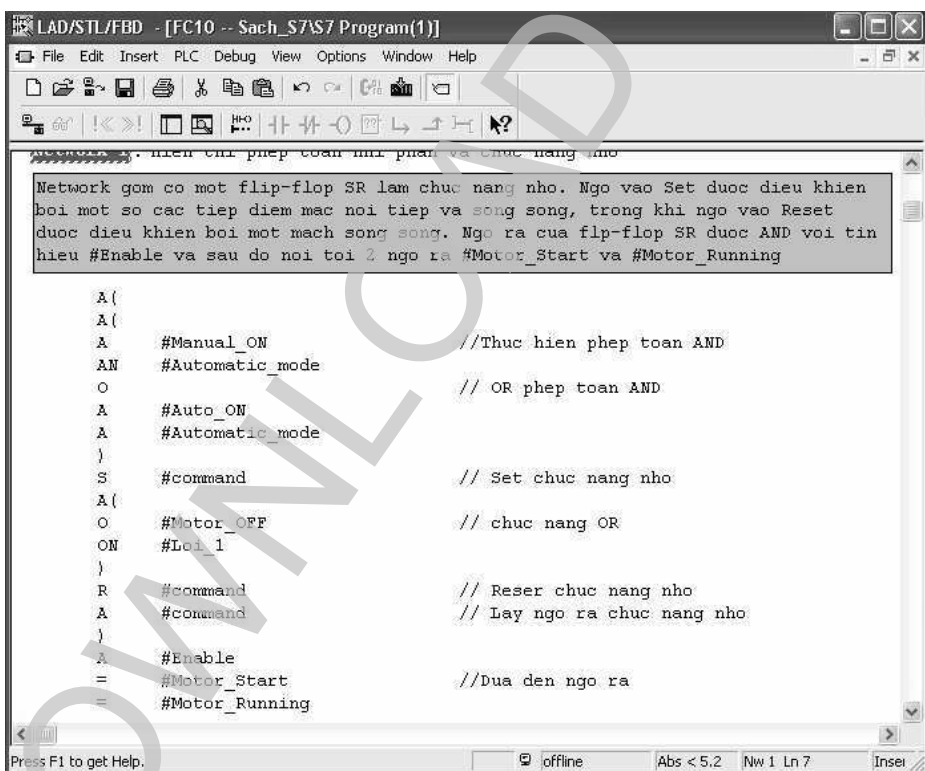
Cấu trúc một lệnh STL

Trong soạn thảo Statement List (STL) bạn viết chương trình ứng dụng dạng chuỗi các câu lệnh. Mỗi một câu lệnh chứa một *lệnh* (instruction) xác định *nhiệm vụ*, phụ thuộc vào dạng câu lệnh mà kèm theo lệnh còn có địa chỉ mà lệnh thực hiện.

Ví dụ trong câu lệnh “A I 1.0” thì lệnh “A” có nghĩa cần kiểm tra xem trạng thái tín hiệu có bằng “1” chưa và lấy kết quả này AND với RLO. Lệnh A tác động tới địa chỉ “I 1.0”. Trong câu lệnh “= Motor_ON” thì lệnh “=” (gán thông số) có nghĩa là RLO được gán cho một địa chỉ là “Motor_ON”.

Phép toán logic bit trong STL

Các phép toán logic bit được thể hiện trong soạn thảo STL bằng các lệnh kiểm tra trạng thái tín hiệu chứa phép toán logic và loại kiểm tra được thực hiện. Ví dụ: lệnh “O” có nghĩa là kiểm tra xem trạng thái tín hiệu bằng “1” chưa và thực hiện phép toán OR ; lệnh ON có nghĩa là kiểm tra xem trạng thái tín hiệu bằng “0” chưa và thực hiện phép toán OR. Khi địa chỉ được kiểm tra xem nó có trạng thái tín hiệu có bằng “1” hay không thì phép toán logic sử dụng trạng thái tín hiệu của nó, còn khi địa chỉ được kiểm tra trạng thái tín hiệu “0” thì phép toán logic sử dụng trạng thái tín hiệu phủ định của nó. Kết quả của phép toán logic được gán cho một địa chỉ bằng lệnh = (gán), S (set), hoặc R (reset). Việc kiểm tra trạng thái tín hiệu kế tiếp bắt đầu một phép toán logic mới. Bạn có thể viết nhiều phép toán logic trên một network và tách chúng ra bằng một hàng trắng.



Hình 4.13 Ví dụ về chức năng nhị phân trong STL.

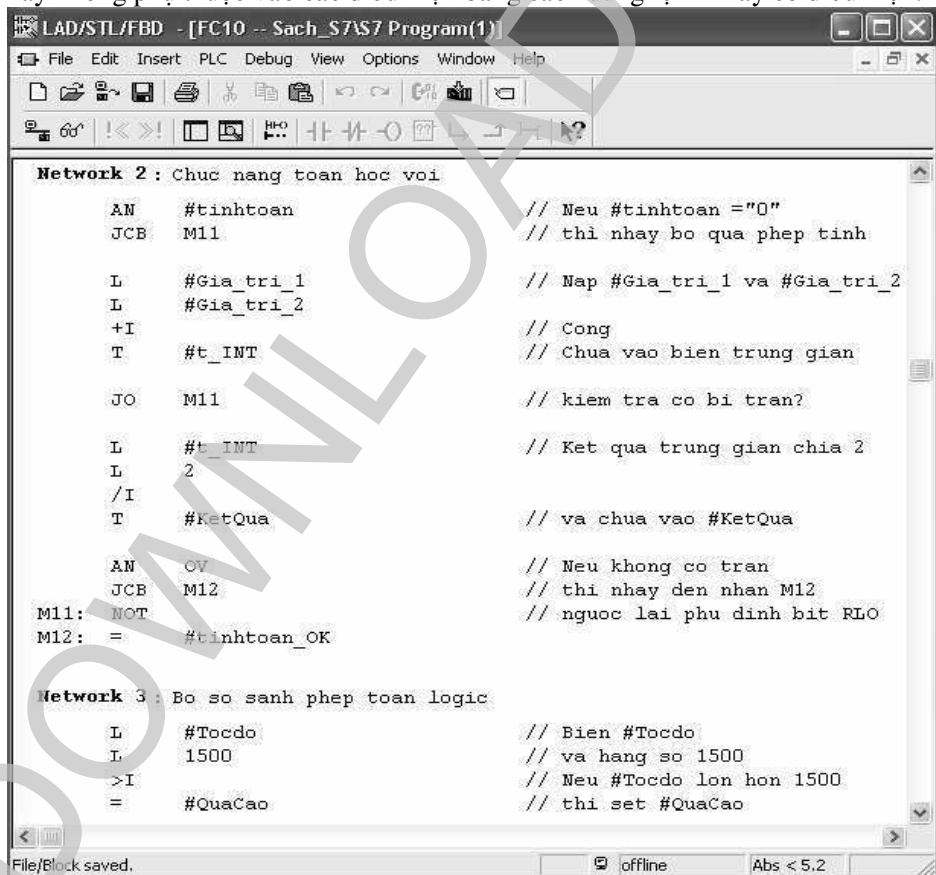
Chức năng số trong STL

Các phép toán số được xử lý trong các thanh ghi tích lũy (Accumulator), chúng là các thanh ghi trong CPU dùng để lưu trữ các giá trị tạm thời. Một

lệnh *Load* nạp một giá trị vào Accumulator 1, lệnh *Load* thứ 2 nạp một giá trị khác vào Accumulator 1 và nội dung trước đó của Accumulator 1 được đẩy vào Accumulator 2. Bây giờ có thể thực hiện phép toán giữa 2 giá trị này, ví dụ phép cộng. CPU lưu trữ kết quả vào Accumulator 1 và kết quả có thể copy tới địa chỉ khác bằng lệnh *Transfer*.

Trong ví dụ dưới thì địa chỉ *#Gia_tri_1* và *#Gia_tri_2* được nạp vào các bộ tích lũy bằng 2 lệnh *load* và được cộng với nhau bằng lệnh *+I*. câu lệnh *T #t_INT* truyền kết quả từ Accumulator 1 và lưu trữ tại địa chỉ *#t_INT*.

Có thể kiểm tra trạng thái tín hiệu của một phép toán số bằng các lệnh kiểm tra hoặc dùng chức năng nhảy trực tiếp. Chẳng hạn, có thể kiểm tra tràn phạm vi bằng lệnh *A OV* hoặc *JO label* (nhảy nhảy). Cũng có thể kiểm tra xem “Result = zero” hoặc “Result =positive”. Các phép toán số luôn được xử lý; tuy nhiên bạn có thể cho phép một phép toán số có được xử lý hay không phụ thuộc vào các điều kiện bằng cách dùng lệnh nhảy có điều kiện.



Hình 4.14 Ví dụ về phép toán số trong STL.

4.8 Ngôn ngữ điều khiển có cấu trúc SCL







chỉ toàn cục, giống như các ngõ vào và các ngõ ra, hay các thông số từ các khối khác. Các khối được kết nối tạo thành chương trình mà các mã có thể xử lý được tạo từ màn hình soạn thảo CFC. Chương trình cũng có thể chứa các nhóm ưu tiên khác nhau (mức độ xử lý chương trình).

Biểu đồ là các khung vẽ trong CFC

Mỗi biểu đồ có thể chứa đến 26 biểu đồ thành phần, mỗi biểu đồ thành phần được thành lập 6 trang. Các trang được sắp xếp thành 2 cột 3 hàng trong màn hình. Không gian làm việc có các thanh lề trên cả 2 cạnh liên kết đến các trang khác hay biểu đồ hoặc các địa chỉ toàn cục.

Một biểu đồ có thể được “cài đặt” trong một biểu đồ khác. Sau đó nó xuất hiện như là một hộp đen với các kết nối được thiết kế và có thể được mở và soạn thảo giống như bất kỳ biểu đồ nào khác. Bạn có thể sử dụng phương pháp này để tạo các cấu trúc theo cấp bậc.

Soạn thảo các biểu đồ CFC

Trong S7 program của SIMATIC Manager, bạn tạo một thư mục biểu đồ có tên Chart bằng cách chọn menu *INSERT* → *S7 SOFTWARE* → *CHART FOLDER* và tạo một biểu đồ mới bằng cách chọn *INSERT* → *S7 SOFTWARE* → *CFC*. Sau đó bạn khởi động màn hình soạn thảo CFC bằng cách nhấp đúp chuột lên biểu đồ mới. Trong Chart View, với việc chọn độ phóng đại đúng bạn sẽ thấy không gian làm việc và các thanh lề. Nếu Catalog khối không được hiển thị, chọn *View* → *CATALOG*.

Catalog khối chứa các khối đang tồn tại (được nhập), các phép toán cơ bản như AND hay các chức năng so sánh, các thư viện khối, và các khối do bạn tạo, các khối không chiếm chỗ (không được biểu diễn) và các biểu đồ tồn tại. Nếu bạn muốn sử dụng CFC để nối các khối tự tạo, thì các khối nhớ trong thư viện dưới ...\\Siemens\\Step7\\S7libs. Các khối có thể được viết trong bất kỳ ngôn ngữ nào.

Bạn có thể lấy các khối từ catalog vào mặt phẳng làm việc. Bây giờ bạn có thể gán tham số cho các ngõ vào và ra của các khối (nhập vào các địa chỉ toàn cục cho chúng) hay nối chúng đến các ngõ vào và ra của các khối khác. Nếu bạn nhận thấy rắc rối trong việc nối các đường thẳng, bạn có thể sử dụng các bộ nối thay thế (nối không được hiển thị như là một đường thẳng). Bạn có thể chọn các đặc trưng khác nhau cho các ngõ vào và ra của khối, cho ví dụ, bạn có thể làm cho chúng không hiển thị trong CFC hay đăng ký chúng để giám sát.



Trong chế độ kiểm tra của màn hình CFC, bạn có thể quan sát giá trị của tất cả sự nối khối được gán các phần tử địa chỉ dữ liệu là các tham số. Trước khi có thể làm điều này, bạn cần phải đăng ký sự kết nối tương thích để quan sát và đặt khoảng thời gian tại những giá trị được cập nhật.

4.10 Điều khiển trình tự GRAPH (sequential Control GRAPH)

S7-GRAPH là một phương pháp lập trình đồ họa cho các hệ thống điều khiển trình tự. Phần mềm này tương thích với ngôn ngữ SFC theo tiêu chuẩn IEC 113-3 (DIN EN 61131-3). Để sử dụng S7-GRAPH bạn phải cần SIMATIC Manager (STEP 7 Basis).

Điều khiển trình tự là gì?

Trong hệ điều khiển trình tự, không giống như hệ điều khiển logic, không gán cố định tín hiệu các ngõ vào đến các ngõ ra mà là kết quả thời gian của nó. Quá trình điều khiển được thực hiện từ cái này đến cái kia được chia thành từng bước riêng. Một bước bao gồm một hay nhiều tác động, ví dụ đóng mạch hay cắt mạch động cơ. Chỉ những tác động của những bước đang xử lý được thực hiện. Bước tiếp theo không thể được xử lý cho đến khi điều kiện chuyển tiếp đạt đến. Sự chuyển tiếp có thể phụ thuộc vào quá trình, ví dụ tín hiệu từ các máy đều được điều khiển hay thiết bị hoặc phụ thuộc vào thời gian, ví dụ kết thúc một thời gian chờ.

Một điều khiển trình tự, hay “bộ trình tự”, bắt đầu với một bước khởi tạo, có thể có nhiều bước khởi tạo trong một bộ trình tự. Trong một bộ trình tự tuyến tính được đi theo bởi các chuyển tiếp luân phiên và các bước.

Cấu trúc của bộ trình tự được lập trình đầu tiên

Một bộ trình tự bao gồm một khối hàm (FB) và một khối dữ liệu (DB). Khối hàm điều khiển bộ trình tự. Khối dữ liệu là các khối thứ cấp cho bộ trình tự FB và chứa cấu trúc của bộ trình tự và dữ liệu liên quan.

Bạn chọn thư mục *Blocks* trong SIMATIC Manager và tạo một FB bằng cách chọn *INSERT* → *S7 PROGRAM* → *FUNCTION BLOCK*. Trong hộp thoại *Properties* bạn chọn ngôn ngữ lập trình là GRAPH. Sau đó nhấp đúp chuột để khởi động phần mềm S7-GRAPH.

S7-GRAPH chèn vào bước đầu tiên và sự chuyển tiếp đầu tiên trong một khối rỗng. Bây giờ bạn tạo cấu trúc bộ trình tự. Bạn có thể thực hiện điều này bằng hai cách: trong chế độ soạn thảo *INSERT* → *DIRECT* bạn chèn một ký hiệu bộ trình tự, ví dụ một cặp bước/chuyển đổi, trong vị trí được chọn; Trong chế độ soạn thảo *INSERT* → *DRAG & DROP* bạn dùng



bước được kích hoạt) và D cho Delay (địa chỉ không được đặt cho đến khi thời gian xác định đạt đến và bị reset khi bước không được kích hoạt).

Bạn có thể lập trình sự chuyển tiếp hoặc là trong Ladder Logic (LAD) hay Function Block Diagram (FBD). Bạn có thể sử dụng các phần tử AND và OR, hoặc liên kết nối tiếp hay song song, và so sánh.

Các chức năng khác điều khiển sự xử lý của chương trình là Interlocks (“Interlocks” ảnh hưởng đến các tác động riêng trong một bước) và giám sát bước ảnh hưởng đến sự chuyển tiếp cho các bước sau.

Gọi bộ trình tự và xử lý

Khi bạn lưu khối hàm bộ trình tự, S7-GRAPH biên dịch nó và tạo ra khối dữ liệu thứ cấp phụ thuộc. Bây giờ bạn phải gọi một cặp FB/DB trong khối được xử lý theo chu kỳ, ví dụ khối tổ chức OB1, để cho phép bộ trình tự được xử lý.

4.11 Điều khiển theo đồ hình trạng thái với HIGRAPH

Phần mềm S7-HIGRAPH là một phương pháp lập trình đồ họa cho các hệ điều khiển đồ hình trạng thái. Nó cần phần mềm cơ sở là SIMATIC Manager (STEP 7).

Hệ điều khiển đồ hình trạng thái là gì?

Hệ điều khiển đồ hình trạng thái mô tả trạng thái của máy và quá trình và các chuyển tiếp có thể có giữa các trạng thái này. Nó là một phần mềm đặc biệt cho các quá trình không trình tự, không đồng bộ. Toàn bộ trạng thái và chuyển tiếp được gọi là đồ hình trạng thái. Nhiều đồ hình trạng thái có thể được kết hợp thành một nhóm đồ hình.

Một đồ hình trạng thái bắt đầu với trạng thái khởi tạo. Nếu một điều kiện cho sự chuyển tiếp ra của trạng thái này được thỏa mãn, trạng thái liên hệ với nó trở nên tích cực. Nhiều chuyển tiếp có thể được gán vào một trạng thái. Nếu tất cả chúng thỏa mãn tại cùng thời điểm, thì sự chuyển tiếp trạng thái được điều khiển với quyền ưu tiên. Tại một thời điểm chỉ có một trạng thái có thể được kích hoạt. Một “chuyển tiếp bất kỳ” đến từ tất cả trạng thái tới một trạng thái đích; một “chuyển tiếp ngược” đi ngược trở lại từ trạng thái hiện hành đến trạng thái trước.

Cấu trúc của đồ hình trạng thái được lập trình đầu tiên

Bạn sử dụng tập tin nguồn để viết một chương trình đồ hình trạng thái. Nó được tạo từ một FC và một DB. Hàm FC điều khiển đồ hình trạng thái; khối dữ liệu chứa các dữ liệu liên quan.

Trong SIMATIC Manager, chọn thư mục Source file và tạo một chương trình nguồn cho S7-HIGRAPH bằng cách chọn menu *INSERT* → *S7 SOFTWARE* → *STATE GRAPH*. Với việc nhấp đúp chuột trên tập tin nguồn sẽ khởi tạo phần mềm S7-Higraph. Một đồ hình trạng thái rỗng chứa trạng thái khởi tạo và sự chuyển tiếp khởi động.

Bạn hãy sử dụng *INSERT* → *STATE* để định vị các ký hiệu trạng thái trong mặt phẳng vẽ. Sau đó bạn liên kết các trạng thái với các chuyển tiếp (*INSERT* → *TRANSITION*). Một chuyển tiếp bất kỳ đến từ bất kỳ điểm nào trong mặt phẳng vẽ tới trạng thái đích và sự chuyển tiếp ngược từ một trạng thái trong mặt phẳng vẽ.

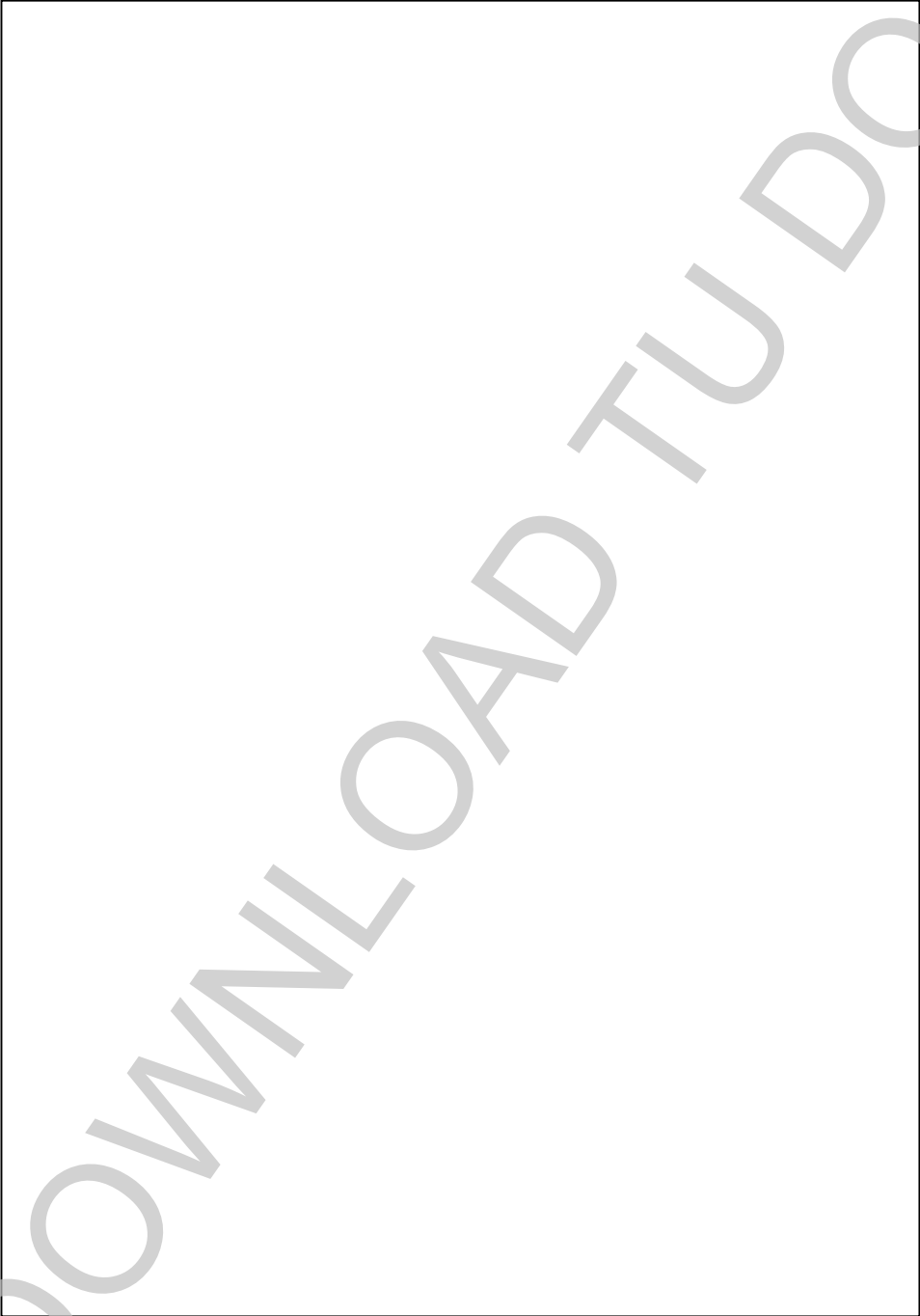
Lập trình các câu lệnh cho các trạng thái và các chuyển tiếp

Chọn *VIEW* → *VARIABLES* để hiển thị cửa sổ thông báo các tham số khối và các biến cục bộ của khối hàm. Nếu bạn muốn sử dụng đồ họa trở lại thì chỉ sử dụng các biến này (sử dụng các thứ cấp của đồ hình trạng thái giống như cho các khối hàm).

Câu lệnh cho các trạng thái và chuyển tiếp thì giống như các câu lệnh STL trong tập tin nguồn. Bạn có thể lập trình bất kỳ câu lệnh nào bạn thích. Mỗi câu lệnh kết thúc với một dấu chấm phẩy. Việc xử lý các câu lệnh của một trạng thái hay một chuyển tiếp luôn luôn bắt đầu với *RLO* = “1”. Các câu lệnh bạn nhập vào được hiển thị trên mặt phẳng vẽ của các trạng thái hay sự chuyển tiếp liên quan. Bạn có thể định vị tất cả các phần tử ở bất kỳ chỗ nào trên mặt phẳng vẽ.

Khi đánh dấu một trạng thái và chọn *VIEW* → *INSTRUCTION*, thì cửa sổ đôi xuất hiện với các kiểu tác động và câu lệnh. Các kiểu tác động xác định các tác động được xử lý như thế nào trong trạng thái kích hoạt. *Entry Actions* được xử lý khi trạng thái được kích hoạt và *Exit actions* khi không được kích hoạt. Khi trạng thái là kích hoạt thì *Preceding Cyclic Actions* được xử lý trước khi sự chuyển tiếp được kiểm tra và *Cyclic Actions* sau khi sự chuyển tiếp đã được kiểm tra.

Khi đánh dấu một chuyển tiếp và chọn menu *VIEW* → *INTRUCTIONS* thì các điều kiện lập trình phải đạt đến trước khi sự chuyển đổi trạng thái xảy ra và nhập vào *Actions*, được xử lý ngay lập tức khi sự chuyển tiếp được kích hoạt.



5 Chương trình người dùng

Bộ điều khiển SIMATIC với các module vào/ra chưa thể điều khiển máy hay thiết bị được. CPU yêu cầu một chương trình xử lý theo từng bước, xử lý các câu lệnh đã nhớ trong chương trình và giải quyết nhiệm vụ điều khiển. Chương trình này được gọi là “chương trình người dùng”. Ngôn ngữ lập trình của phần mềm lập trình STEP 7 được sử dụng để viết chương trình.

Chương trình người dùng được xử lý ở nhiều cách. Sau khi bật nguồn, CPU xử lý chương trình khởi động, sau đó là chương trình chính (luôn luôn lặp lại liên tục), chương trình chính này có thể bị ngắt bởi các sự cố ngắt hay lỗi với chương trình liên quan. Mỗi ngắt có cấp độ ưu tiên riêng.

Chương trình người dùng thường được chia thành nhiều phần riêng. Các phần chương trình này được gọi là *blocks* (khối). Để các khối chương trình này được thực hiện, nó phải được đưa vào trong các khối tổ chức theo cấu trúc chương trình người dùng. Khối chương trình chính là OB1. Các khối chương trình được đưa vào khối này (OB1) gọi là các khối chương trình con và được gọi tuần tự theo yêu cầu công nghệ hay chức năng của các thiết bị điều khiển.

Trong chương trình người dùng, bạn liên kết các trạng thái hay giá trị của các biến địa chỉ (*addresses*) với nhau và lưu trữ các kết quả hay đưa chúng ra đối tượng điều khiển (máy móc, thiết bị). Các vùng nhớ, còn gọi là các vùng địa chỉ, ví dụ như các ngõ vào và ra để giao tiếp với đối tượng điều khiển hay bit nhớ cho việc lưu trữ các dữ liệu. Người ta phân chia ra các địa chỉ ra làm hai loại: địa chỉ toàn cục được sử dụng xuyên suốt trong toàn bộ chương trình người dùng, và các địa chỉ cục bộ chỉ có giá trị trong một khối xác định (v.d các dữ liệu cục bộ tạm thời cho các bộ nhớ của các kết quả trung gian động).

Để có thể truy cập một địa chỉ cần có một *sự định vị ô nhớ* (memory location). Địa chỉ tuyệt đối sử dụng sự định vị ô nhớ để nhận dạng. Bạn cũng có thể gán một địa chỉ tuyệt đối một cái tên và sau đó sử dụng địa chỉ ký hiệu. Một vài ngôn ngữ lập trình cũng cho phép địa chỉ gián tiếp mà sự định vị ô nhớ không được tính toán cho đến thời gian chạy.

Kiểu dữ liệu (data types) xác định dãy giá trị và cấu trúc nội (lưu trữ dữ liệu) của các biến. Có nhiều kiểu dữ liệu sơ cấp mà có thể được xử lý với

các câu lệnh “bình thường” của ngôn ngữ lập trình, và kiểu dữ liệu phức. Kiểu dữ liệu người dùng định nghĩa (UDT) được bạn định nghĩa, được gán một tên và sau đó được sử dụng trong toàn bộ chương trình.

5.1 Các khối tổ chức và cấp ưu tiên

Chương trình toàn bộ của bộ xử lý trung tâm (CPU: central processing unit) bao gồm hệ điều hành và chương trình người dùng.

Hệ điều hành là tổng tất cả các câu lệnh và các thông báo của các chức năng điều hành trong thiết bị (v.d. cứu lấy dữ liệu khi mất nguồn cung cấp, kích hoạt các cấp độ ưu tiên, v.v.). Mặc dù hệ điều hành là thành phần cố định của CPU không thể thay đổi, bạn có thể nạp nó trở lại từ card nhớ trong khi cập nhật chương trình.

Chương trình người dùng là tổng tất cả các câu lệnh của người dùng đã được lập trình và các thông báo cho xử lý tín hiệu mà thông qua nó hệ thống hay dự án được điều khiển bị ảnh hưởng bởi sự phối hợp của nhiệm vụ điều khiển.

Các *khối tổ chức* (Organization block) là sự giao tiếp giữa hệ điều hành và chương trình người dùng. Các khối tổ chức là thành phần của chương trình người dùng được gọi và xử lý bởi hệ điều hành khi xảy ra các sự cố xác định. Các khối tổ chức được chia ra *cấp độ ưu tiên* xác định tuần tự xử lý chương trình (khả năng ngắt lẫn nhau) khi xảy ra nhiều sự cố.

Số lượng các khối tổ chức là cố định. Bạn có thể tự xác định quyền xử lý ưu tiên trong khuôn khổ cho phép của các CPU. Không có một CPU nào có thể xử lý tất cả các sự cố được xác định trong STEP 7. Bảng sau chỉ ra tất cả các khối tổ chức; Một CPU cụ thể sử dụng một phần trong bảng này.

Bảng 5.1 Các khối tổ chức của SIMATIC S7

Khối tổ chức	Được gọi	Cấp độ ưu tiên
Quét chương trình chính (Main program scan) OB1	Theo chu kỳ (lặp đi lặp lại) bởi hệ điều hành	1
Ngắt thời gian của ngày (Time-of-day interrupts) OB10 đến OB 17	Tại thời điểm xác định hay trong một khoảng thời gian (v.d: hàng tháng)	2
Ngắt thời gian trễ (Time-delay interrupts) OB 20 đến OB 23	Sau một khoảng thời gian có thể điều chỉnh, được điều khiển bằng chương trình người dùng	3 đến 6



Chương trình chính được đặt trong khối tổ chức OB1. OB1 có cấp độ ưu tiên xử lý thấp nhất. Nó có thể bị ngắt bởi tất cả các ngắt và lỗi. Bạn không thể thay đổi quyền ưu tiên xử lý của OB1.

Sau khi xử lý chương trình trong OB1 kết thúc, CPU ngay lập tức gọi OB 1 trở lại (có nghĩa OB1 được CPU xử lý theo chu kỳ). Sự xử lý chu kỳ này là kiểu xử lý “bình thường” được sử dụng cho PLC. Đối với nhiều ứng dụng, tổng toàn bộ chương trình người dùng chỉ đặt trong OB1.

Bạn có thể đặt thời gian xử lý cho chương trình chính để có thời gian cho xử lý nền. CPU gọi OB 90 – tùy thuộc vào thời gian sử dụng- được xử lý theo phương pháp “piece by piece” quen thuộc với một chu kỳ đầy đủ của chương trình chính. Cũng như OB1, OB 90 cũng có thể bị ngắt bởi tất cả các sự cố ngắt và lỗi.

Chương trình ngắt, cấp độ ưu tiên

Hệ điều hành CPU có thể ngay lập tức đến một ưu tiên xác định (một ngắt) và gọi khối tổ chức đã được gán. Bạn viết chương trình ngắt trong khối này, biểu diễn đáp ứng với các ngắt. Nếu chương trình đang xử lý hiện hành có quyền ưu tiên thấp hơn ngắt, thì nó bị ngắt tại câu lệnh kế tiếp. CPU “ghi nhớ” tất cả các dữ liệu liên quan, để sau khi xử lý chương trình ngắt, nó có thể tiếp tục xử lý chương trình có quyền ưu tiên thấp tại điểm mà nó bị ngắt.

Ví dụ, các ngắt do các thiết bị được điều khiển (ngắt cứng), từ CPU (ngắt chu kỳ, ngắt thời gian của ngày) hay bị kích bởi chương trình người dùng (ngắt thời gian trễ và ngắt đa xử lý). Quyền ưu tiên xử lý của các ngắt này có thể được đặt trong hầu hết các CPU. Nếu có nhiều khối tổ chức cho một kiểu ngắt, thì bạn có thể gán cho mỗi OB một cấp độ ưu tiên riêng. Khi hai sự cố xảy ra tại một thời điểm, sự cố có quyền ưu tiên cao hơn được xử lý trước. Với việc xác định các tham số của CPU, bạn xác định được cấp độ ưu tiên cho nhiều khối tổ chức. Các khối không sử dụng bạn cho quyền ưu tiên 0.

Chương trình lỗi

Khi các lỗi xác định xảy ra, CPU đáp ứng ngay bằng cách gọi các khối tổ chức chứa chương trình lỗi. Người ta phân biệt ra giữa lỗi đồng bộ và lỗi không đồng bộ. Lỗi không đồng bộ là lỗi không phụ thuộc đến sự xử lý chương trình (v.d. mất nguồn cung cấp ở thiết bị mở rộng hay ngắt ở việc thay đổi một module). Lỗi đồng bộ là lỗi được tạo ra bởi sự xử lý chương trình (v.d định địa chỉ đến một địa chỉ không tồn tại hay một lỗi trong chuyển đổi kiểu dữ liệu).



Khởi động có thể được kích bằng tay bởi công tắc chọn chế độ hay một chức năng truyền thông, hay tự động bằng cách bật nguồn cung cấp

Không có sự giới hạn ở độ dài của chương trình khởi động và không có hạn chế thời gian ở việc xử lý chương trình khởi động. Không có một ngắt nào được xử lý trong khi chương trình khởi động đang được xử lý.

Trong khi khởi động, CPU cập nhật các chức năng thời gian, bộ đếm thời gian hoạt động và đồng hồ thời gian thực. Các module ngõ ra không cho phép tín hiệu ra trong khi khởi động.

Cold restart (Khởi động nguội)

Trong cold restart, CPU tự đặt nó và các module vào trạng thái cơ bản đã thiết kế, xóa tất cả các dữ liệu từ bộ nhớ hệ thống (cũng như dữ liệu cố định retentive) và gọi OB102. Chương trình hiện hành và các dữ liệu trong bộ nhớ làm việc bị xóa. Chương trình được nạp lại từ bộ nhớ chương trình (nói cách khác là reset bộ nhớ, bộ nhớ chương trình RAM không bị xóa). Sau khi khởi động nguội, CPU xử lý chương trình chính trong OB1 từ vị trí bắt đầu.

Hot restart (Khởi động nóng hay khởi động lại toàn bộ)

Trong khởi động nóng, CPU tự đặt nó và các module vào trạng thái cơ bản đã thiết kế, xóa các dữ liệu không cố định từ bộ nhớ hệ thống và gọi OB100. Chương trình hiện hành và dữ liệu hiện hành trong bộ nhớ làm việc được giữ lại. Sau khi khởi động nóng, CPU xử lý chương trình chính trong OB1 từ vị trí bắt đầu.

Warm restart (Khởi động lại)

Khi STOP hay mất nguồn, CPU lưu tất cả các sự cố ngắt và các thanh ghi trong CPU cung cấp để xử lý chương trình người dùng. Do đó, khi khởi động lại, nó có thể tiếp tục tại thời điểm chương trình bị ngắt. Có thể là chương trình chính, nhưng cũng có thể là chương trình ngắt hay lỗi. Tất cả các sự cố ngắt “cũ” được lưu và được xử lý. “chu kỳ còn lại” (remaining cycle) là từ điểm trong chương trình mà CPU tiếp tục sau khi khởi động lại đến điểm kết thúc chương trình chính thì vẫn được đếm như khởi động, không có ngắt mới nào được xử lý. Các module ngõ ra bị chặn lại và ở trong trạng thái cơ bản. Khi đặt tham số cho CPU, bạn có thể xác định sau bao lâu của ngắt, CPU được phép thực hiện khởi động lại (100 ms đến 1 giờ). Nếu thời gian ngắt dài hơn, thì chỉ có khởi động nguội hay khởi động nóng được cho phép.



5.4 Reset bộ nhớ, thuộc tính nhớ lâu dài (Retention)

Reset bộ nhớ

Reset bộ nhớ đặt CPU vào “trạng thái cơ bản”. Reset bộ nhớ chỉ có thể được thực hiện với thiết bị lập trình khi CPU ở trạng thái STOP, hay với công tắc chọn chế độ: công tắc được giữ ở MRES ít nhất 3s, sau đó thả ra, sau thời gian lớn nhất là 3s để công tắc lại vị trí MRES lần nữa ít nhất 3s.

CPU xóa toàn bộ chương trình người dùng trong bộ nhớ làm việc và bộ nhớ RAM. Bộ nhớ hệ thống với các vùng địa chỉ cũng bị xóa, không phụ thuộc vào việc đặt thuộc tính retentive (ghi nhớ). CPU reset tất cả các tham số của tất cả các module –cũng như bản thân nó- về trạng thái chuẩn. Ngoại trừ các tham số MPI. Chúng không bị thay đổi để CPU bị reset bộ nhớ vẫn được địa chỉ trên MPI bus. Khi reset, vùng đệm chẩn đoán, đồng hồ thời gian thực, và bộ đếm thời gian hoạt động cũng không bị reset. Nếu một card nhớ flash EPROM được cài đặt, thì CPU sao chép chương trình người dùng từ card nhớ đến bộ nhớ làm việc. Nếu card nhớ chứa các dữ liệu thiết kế, điều này cũng được CPU sử dụng.

Thuộc tính nhớ lâu dài (Retentive)

Một vùng nhớ là retentive khi nội dung của nó được giữ lại ở khởi động nóng. Vùng địa chỉ retentive có thể là bit nhớ, timer, counter và ở S7-300 là vùng nhớ dữ liệu. Số lượng dữ liệu retentive tùy thuộc vào CPU. Bạn có thể xác định ở các tham số của CPU trong thanh “Retentive Memory” vùng nào là nhớ lâu dài (retentive).

Sự đặt chỉnh cho thuộc tính retentive được đặt trong các khối dữ liệu hệ thống (SDB) trong bộ nhớ làm việc (có nghĩa là card nhớ). Nếu card nhớ là card RAM, thì pin backup phải được sử dụng cho PLC, để nhớ sự đặt chỉnh retentive.

Nếu bạn sử dụng pin backup, trạng thái tín hiệu của bit nhớ, timer và counter được định nghĩa như là retentive được lưu giữ. Chương trình và dữ liệu người dùng không bị thay đổi. Nếu card nhớ là flash EPROM và không có pin backup, thì S7-300 và S7-400 có các phản ứng khác nhau. Với S7-300, thì trạng thái tín hiệu của bit nhớ, timer và counter định nghĩa là retentive được lưu giữ. Với S7-400 thì không. Nội dung của các khối dữ liệu đặt là retentive cũng được giữ ở S7-300. Hãy nhớ lại rằng, với S7-300, nội dung của các mảng dữ liệu retentive được chứa trong CPU chứ không phải trong card nhớ.

Các khối dữ liệu còn lại (đối với S7-300) và tất cả các khối dữ liệu (đối với S7-400) được chép từ card nhớ đến bộ nhớ làm việc. Các khối mã cũng được chép. Chỉ có các khối dữ liệu từ card nhớ là được giữ retentive. Các khối dữ liệu được tạo với SFC22 CREAT_DB không phải là retentive. Sau khi khởi động, các khối dữ liệu có nội dung chứa trong card nhớ, có nghĩa là nội dung đã được lập trình.

5.5 Chương trình chính

Chương trình chính là chương trình người dùng được xử lý theo chu kỳ. Sự xử lý chương trình theo chu kỳ là sự xử lý “bình thường” đối với PLC. Đa số các bộ điều khiển chỉ sử dụng xử lý chương trình kiểu này. Khi xử lý chương trình điều khiển sự cố được sử dụng, thì nó chỉ thường là bổ sung cho chương trình chính.

CPU chỉ xử lý chương trình chính khi ở chế độ RUN. CPU hoạt động khi công tắc chọn chế độ ở mặt trước của CPU được đặt ở RUN hay RUN-P. Ở RUN-P, bạn có thể lập trình CPU bằng thiết bị lập trình kết nối với nó. Ở vị trí RUN, bạn có thể lấy chìa khóa ra để không một ai có thể thay đổi ở chế độ hoạt động. Ở vị trí này, chương trình chỉ có thể được đọc.

Tổ chức chương trình

Chương trình người dùng thường được chia ra thành nhiều phần riêng (các khối). Khi một khối được xử lý, nó phải được gọi. Chỉ có các khối tổ chức không được gọi bởi chương trình người dùng mà là được khởi động bằng hệ điều hành của CPU.

Với cấu trúc của chương trình, bạn xác định CPU xử lý khối ở sự cố nào và sự tuần tự nào. Cho ví dụ, cấu trúc chương trình thô được tạo khi bạn lập trình gọi các khối cho chương trình chính trong khối tổ chức OB1. Trong các khối “cấp độ cao hơn” này, bạn có thể gọi các khối khác và cấu trúc chương trình người dùng chi tiết hơn, và v.v..

Theo nguyên tắc, có 2 cách để chia các nhiệm vụ tự động tổng thành các nhiệm vụ thành phần nhỏ hơn. Một cấu trúc chương trình công nghệ (technological program structure) nguyên thủy dựa trên cơ sở thiết lập thiết bị được điều khiển và được chia thành thiết bị, thiết bị thành phần và các bộ phận. Các phần riêng của thiết bị hoặc dự án được điều khiển tương thích với các khâu chương trình riêng. Ví dụ: thiết bị thành phần “băng tải” có thể bao gồm: các phần tử băng tải khác nhau, xe, các trạm nâng. Các thiết bị thành phần này cũng bao gồm các phụ tùng riêng như các động cơ, van, và các phần tử chỉ thị. Một cấu trúc chương trình chức năng (functional

program structure) dựa trên cơ sở chức năng điều khiển được thực hiện. Các khối cấp độ thấp hơn chứa chương trình của các chức năng thành phần. Ví dụ: chức năng “thông báo nhận được” có thể bao gồm thông báo điều kiện, thông báo lưu trữ và thông báo ngõ ra.

Lồng các khối gọi

Trong thời gian hoạt động, CPU tạo một lối đi vào trong ngăn xếp khối (B-Stack) cho mỗi khối được gọi. Với thông tin này, CPU có thể tiếp tục xử lý ở khối được gọi, sau khi xử lý các khối được gọi kết thúc. Các khối được gọi tiếp theo viết đè dữ liệu khối gọi trước trong B-stack. Nếu sự gọi khối “lồng nhau” được tạo trong khối được gọi, thì CPU thiết lập một phần tử mới trong B-stack. Số lượng các phần tử B-stack bị giới hạn ở giá trị cực đại của một CPU xác định. Nếu có quá nhiều sự gọi khối “lồng nhau” được tạo lớn hơn giá trị cực đại cho phép, CPU đi vào chế độ STOP với thông báo lỗi “Tràn ngăn xếp khối (block Stack overflow)”.

5.6 Thông tin khởi động

Hệ điều hành của CPU chuyển đổi thông tin khởi động trong 20 byte đầu tiên của dữ liệu cục bộ tạm thời đến mỗi khối tổ chức. Bạn có thể tạo thông báo cho thông tin khởi động theo ý bạn, hoặc có thể sử dụng khuôn mẫu từ Standard Library trong chương trình Organization blocks.

Bảng dưới đây chỉ thông tin khởi động này cho OB1, tên ký hiệu chuẩn và các kiểu dữ liệu. Tên có thể được đổi sự định danh phù hợp với yêu cầu của bạn. Mặc dù vậy, nếu bạn không sử dụng thông tin khởi động, 20 byte đầu tiên của dữ liệu cục bộ tạm thời phải được phục vụ cho mục đích này (có nghĩa là một mảng 20 byte).

Thông tin khởi động của OB1 cho bạn biết sự xử lý chương trình trong chu kỳ đầu tiên sau khi khởi động có hay không và khởi động nào được thực hiện. Thông tin này có thể được sử dụng để đặt trước trong chương trình người dùng. Ba giá trị INT trong thông tin khởi động cung cấp thông tin về thời gian chu kỳ của chu kỳ sau cùng và thời gian chu kỳ cực đại và cực tiểu từ lần đóng mạch sau cùng.

Nên nhớ rằng bạn chỉ có thể đọc thông tin khởi động của khối tổ chức, có thể tự đọc trực tiếp trong khối tổ chức vì dữ liệu cục bộ tạm thời được đặt ở đây. Nếu bạn cũng cần các giá trị từ thông tin khởi động trong các khối được định vị ở các cấp độ thấp hơn, thì hàm hệ thống SFC RD_SINFO gọi tại điểm tương ứng trong chương trình. SFC RD_SINFO không chỉ được gọi tại điểm bất kỳ trong chương trình chính mà là trong mỗi lớp ưu tiên. Trong

chương trình, khối tổ chức báo lỗi hay trong khối động cũng thế. Hàm hệ thống SFC RD_SINFO cung cấp cho bạn thông tin khởi động của các khối tổ chức hiện hành và thông tin khởi động của OB khởi động được xử lý sau cùng tại cấp độ gọi thấp hơn.

Bảng 5.2 Thông tin khởi động cho OB1

<i>Tên</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>	<i>Sự chỉ định</i>
OB1_EV_CLASS	BYTE	Nhóm sự cố	B#16#11= gọi OB chuẩn
OB1_STRT_INFO	BYTE	Thông tin khởi động	B#16#01= Chu kỳ đầu tiên sau khi khởi động nóng B#16#02= Chu kỳ đầu tiên sau khi khởi động lại B#16#03= Mỗi chu kỳ tiếp theo B#16#04= Chu kỳ đầu tiên sau khi khởi động nguội
OB1_PRIORITY	BYTE	Cấp độ ưu tiên	B#16#01
OB1_OB_NUMBR	BYTE	Số OB	B#16#01
OB1_RESERVED_1	BYTE	Dự trữ	-
OB1_RESERVED_2	BYTE	Dự trữ	-
OB1_PREV_CYCLE	INT	Thời gian chu kỳ trước	Thời gian xử lý (ms) của chu kỳ chương trình trước
OB1_MIN_CYCLE	INT	Thời gian chu kỳ cực tiểu	Thời gian chu kỳ cực tiểu (ms) tính từ khởi động sau cùng
OB1_MAX_CYCLE	BYTE	Thời gian chu kỳ cực đại	Thời gian chu kỳ cực đại (ms) từ khởi động sau cùng
OB1_DATE_TIME	BYTE	Xảy ra sự cố	Ngày và giờ của OB gọi trong chu kỳ hiện hành

5.7 Các chức năng của CPU

Real-time clock (đồng hồ thời gian thực)

Đồng hồ thời gian thực cung cấp ngày và giờ. Các ngắt và các xung thời gian của ngày cho các bộ đếm thời gian chạy được bắt nguồn từ đồng hồ thời gian thực. Nếu pin backup đang hoạt động, thì đồng hồ thời gian thực tiếp tục chạy dù cho nguồn điện cung cấp bị mất. Reset bộ nhớ CPU không ảnh hưởng đến đồng hồ thời gian thực.



5.8 Bộ đệm (Process Images)

Các module trong bộ đệm (Process Images)

Process Images chứa các trạng thái tín hiệu của một vài module ngõ vào và ra. Có hai loại bộ đệm: bảng bộ đệm ngõ vào (Process Image input table: PII) và bảng bộ đệm ngõ ra (Process Image output table: PIQ). Bảng bộ đệm ngõ vào được địa chỉ bởi các vùng địa chỉ ngõ vào (I), và bảng bộ đệm ngõ ra được địa chỉ bởi các vùng ngõ ra (Q). Các ngõ vào ra thường được sử dụng để điều khiển máy hay dự án.

Process Image là một phần của bộ nhớ hệ thống nội CPU. Nó bắt đầu tại địa chỉ I/O 0 và kết thúc tại giới hạn địa chỉ trên đã xác định bởi CPU. Giới hạn này có thể được đặt với CPU được thiết kế tương ứng.

Khi bạn xác định các tham số của các địa chỉ module, bạn cũng có thể xác định trạng thái tín hiệu của module có được đặt trong Process Image hay không. Bạn có thể sử dụng các địa chỉ của Process Image không được gán cho các module tương tự các vùng bit nhớ M. Các module mà các địa chỉ của nó không nằm trong Process Image được địa chỉ với các vùng địa chỉ ngõ vào ngoại vi (PI) và ngõ ra ngoại vi (PQ).

Cập nhật Process Image

Tất cả các module số thường được định vị trong vùng địa chỉ của process Image. Tất cả các module analog nhận một địa chỉ bên ngoài Process Image. Nếu CPU có sự phân định địa chỉ tự do, bạn có thể sử dụng bảng cấu hình để xác định cho mỗi module hoặc không được địa chỉ qua Process Image.

Sau khi CPU khởi động, trước khi OB1 được xử lý lần đầu tiên, hệ điều hành chuyển đổi trạng thái tín hiệu của bảng bộ đệm ngõ ra (PIQ) đến các module ngõ ra và đọc các trạng thái tín hiệu của các module ngõ vào vào bảng bộ đệm ngõ vào (PII). Sau đó OB1 được xử lý, trong khi trạng thái tín hiệu của bảng bộ đệm ngõ vào vẫn được cập nhật giá trị với các ngõ vào được liên kết và các ngõ ra được điều khiển. Sau khi OB1 kết thúc xong, một chu kỳ mới bắt đầu với việc cập nhật Process Image.

Nếu một lỗi xảy ra trong khi cập nhật tự động Process Image (ví dụ một module không còn tương thích nữa), khối tổ chức OB85 (chương trình xử lý lỗi) được gọi. Nếu khối OB85 không tồn tại, CPU đi vào chế độ STOP.

Bộ đệm thành phần (Part Process Image)



Sự phân bố các khối tổ chức ngắt

Một vài CPU cho phép bạn gán các bộ đệm thành phần đến các khối tổ chức cho ngắt và các lỗi không đồng bộ. Bạn làm điều này khi định nghĩa các tham số cho CPU. Chúng tôi đề nghị đặt trong các bộ đệm thành phần này các địa chỉ của các module mà bạn địa chỉ trong chương trình ngắt. Khi một sự cố ngắt xảy ra, bộ đệm thành phần đã gán được cập nhật.

Hàm hệ thống cho bộ đệm thành phần

Các hàm hệ thống SFC 26 UPDAT_PI và SFC 27 UPDAT_PO cập nhật các bộ đệm thành phần đã định nghĩa của các ngõ vào hoặc ra. Hàm hệ thống có thể được gọi tại bất kỳ điểm nào trong chương trình để cập nhật bộ đệm thành phần. Với sự cho biết của bộ đệm thành phần 0, bạn cũng có thể cập nhật bộ đệm của chương trình chính, ví dụ khi bạn đã tắt chế độ cập nhật tự động. Nếu một lỗi xảy ra trong cập nhật, thì lỗi được tường thuật qua các giá trị thu được của hàm hệ thống.

5.9 Thời gian chu kỳ, thời gian phản ứng

Thời gian giám sát chu kỳ

Sự xử lý chương trình trong khối tổ chức OB1 được giám sát theo thời gian. Điều này được xảy ra bởi chức năng “giám sát thời gian chu kỳ”. Thời gian giám sát chuẩn là 150 ms. Bạn có thể đặt giá trị này trong phạm vi từ 1 ms đến 6s khi định nghĩa các tham số của CPU. Nếu xử lý chương trình dài hơn thời gian giám sát chu kỳ đặt thì CPU gọi khối tổ chức OB 80 (“lỗi thời gian”). Nếu OB này không tồn tại, thì CPU đi vào chế độ STOP.

Thời gian giám sát chu kỳ bao gồm: thời gian xử lý tổng của OB1. Ở đây cũng bao gồm thời gian xử lý cho lớp có quyền ưu tiên cao hơn (trong chu kỳ hiện hành). Các xử lý truyền thông bởi hệ điều hành (ví dụ truy xuất PG trên CPU) cũng cộng thêm vào thời gian chạy của chương trình chính.

Thời gian chu kỳ tối thiểu, xử lý nền OB90

Một vài CPU cho phép bạn xác định trước chu kỳ tối thiểu. Nếu sự xử lý chương trình chính, bao gồm các ngắt, dài bằng thời gian đã xác định trước, CPU sẽ đợi cho đến khi thời gian chu kỳ tối thiểu đạt tới. Sau đó CPU bắt đầu chu kỳ kế tiếp bằng cách gọi lại OB 1. Thời gian chu kỳ tối thiểu bị mất khi mất nguồn. Bạn có thể xác định thời gian chu kỳ tối thiểu trong phạm vi từ 1ms đến 6s trong thanh ghi “Cycle/Clock Memory” trong hộp thoại “Object Properties” của CPU.



điều khiển đưa ra sự thay đổi và đặt ngõ vào. Một chu kỳ nữa đi qua trước khi ngõ ra module phản ứng.

Thời gian phản ứng ở sự thay đổi tín hiệu ngõ vào cũng có thể nằm giữa một và hai thời gian chu kỳ. Thời gian trễ cho các module ngõ vào và thời gian đóng mạch của khởi động từ và v.v. cũng được cộng thêm cho thời gian phản ứng.

Thời gian xử lý chương trình người dùng chứa tất cả các thủ tục trong một chu kỳ chương trình, như là xử lý các ngắt, xử lý trong hệ điều hành như cập nhật các timer, điều khiển giao tiếp MPI và cập nhật bộ đệm.

Trong trường hợp riêng, bạn có thể giảm thời gian phản ứng bằng cách địa chỉ trực tiếp I/O hay sử dụng hay gọi điều khiển sự cố các phần chương trình.



5.10 Xử lý ngắt (tổng quát)

Xử lý ngắt là một thủ tục xử lý chương trình điều khiển sự cố. Khi một sự cố xảy ra, hệ điều hành ngắt xử lý chương trình chính và gọi khối tổ chức được gán cho sự cố. Trong khối tổ chức ngắt đặc biệt này (OB), bạn viết một chương trình điều khiển phản ứng đến sự cố. Bạn có thể lập cấu trúc chương trình trong OB ngắt như chương trình chính. Sau khi chương trình này được xử lý, hệ điều hành tiếp tục xử lý chương trình chính tại điểm ngắt. Như vậy một ngắt có thể xảy ra sau mỗi câu lệnh. Các sự cố mà dẫn đến ngắt chương trình chính có thể là các ngắt và các lỗi. Khi có nhiều sự cố ngắt xảy ra thì xử lý ngắt được điều khiển tuần tự theo quyền ưu tiên. Nhiều sự cố ngắt có thể được tóm tắt theo lớp ưu tiên.

Thông tin khởi động

Hệ điều hành cung cấp cho các khối tổ chức ngắt thông tin khởi động. Thông tin khởi động nằm trong 20 byte đầu của dữ liệu cục bộ tạm thời của khối tổ chức riêng. Bạn phải chắc chắn khai báo 20 byte này cho mọi khối tổ chức nếu bạn không sử dụng chúng. Chương trình các khối tổ chức trong *standart library* đề nghị một mẫu cho khai báo. Bạn cũng có thể sử dụng cấu trúc biến khác và gán cho chúng các tên riêng.

Bạn hãy xác định số lượng dữ liệu cục bộ tạm thời cho mỗi khối tổ chức khi bạn định nghĩa các tham số cho CPU. Ở đây bạn cũng phải để ý đến dữ liệu cục bộ tạm thời của khối được gọi bởi OB ngắt. Đối với một khối tổ chức không sử dụng, bạn đặt số dữ liệu cục bộ là “0”.

Bảng 5.3 Thông tin khởi động cho các khối tổ chức ngắt

Byte	Ngắt đa xử lý OB 60	Hardware Interrupts OB40 .. OB 47	Cyclic Interrupts OB 30..OB38	Delay Time Interrupts OB 20..OB23	Tim-of-Day Interrupts OB 10..OB17
0	Cấp sự cố	Cấp sự cố	Cấp sự cố	Cấp sự cố	Cấp sự cố
1	Sự cố khởi động	Sự cố khởi động	Sự cố khởi động	Sự cố khởi động	Sự cố khởi động
2	Cấp ưu tiên	Cấp ưu tiên	Cấp ưu tiên	Cấp ưu tiên	Cấp ưu tiên
3	OB số	OB số	OB số	OB số	OB số
4	-	-	-	-	-
5	-	Nhận biết địa chỉ	-	-	-
6-7	Nhận biết nhiệm vụ(INT)	Địa chỉ bắt đầu module (WORD)	Độ lệch pha ms (INT)	Nhận biết nhiệm vụ (WORD)	Khoảng thời gian (WORD)
8-9	-	Thông tin	-	Khoảng thời gian	-
10-11	-	Hardware interrupt (DWORD)	Xung đồng hồ ms (INT)	trễ đã hết (TIME)	-
12-19	Điểm thời gian sự cố (DT)	Điểm thời gian sự cố (DT)	Điểm thời gian sự cố (DT)	Điểm thời gian sự cố (DT)	Điểm thời gian sự cố (DT)

5.11 Ngắt cứng (hardware Interrupts)

Ngắt cứng (còn gọi là ngắt xử lý) được sử dụng để thu nhận các sự cố nghiêm trọng từ hệ thống được điều khiển hay thiết bị ngay lập tức trong chương trình người dùng và phản ứng với một chương trình tương thích. Các khối tổ chức OB 40 đến OB 47 trong STEP 7 được sử dụng cho *Hardware*

Interrupts. Một trong 8 khối tổ chức này có giá trị tùy thuộc vào CPU sử dụng.

Kích một ngắt cứng

Ngắt cứng được kích bởi một module có khả năng ngắt. Ví dụ, có thể là một module ngõ vào số thu nhận tín hiệu từ quá trình hay một module chức năng mà thông qua quá trình của một module kích ngắt cứng. Kích ngắt cứng được tạo bởi chuẩn đặt “*disable*”. Khi bạn định nghĩa các tham số cho module, bạn có thể cho phép xử lý các ngắt cứng (tham số tĩnh) trong thanh công cụ *HW Config* trong cấu hình phần cứng. Ở đây bạn cũng có thể xác định ngắt cứng có được kích cho một sự cố xảy ra hay không (tham số động). Tham số động có thể thay đổi trong thời gian hoạt động với SFC. Khi bạn định nghĩa các tham số cho module, bạn cũng có thể gán một khối tổ chức cho ngắt cứng.

Khi một ngắt cứng xảy ra, các khối tổ chức được gán cũng phải tồn tại. Nếu không, CPU đưa ra một bản tin trong vùng đệm chẩn đoán và gọi khối tổ chức báo lỗi không đồng bộ OB 85 (“Lỗi xử lý chương trình”) hay đi vào chế độ STOP. Các ngắt cứng chỉ được kích khi CPU ở chế độ RUN. Ngắt cứng bị loại bỏ trong khi khởi động.

Đọc thông tin khởi động

Trong khối tổ chức ngắt, bạn có thể biết được module nào kích các ngắt. Byte 5, 6 và 7 của thông tin khởi động chứa các địa chỉ khởi động của module này. Byte 8 đến 11 chứa trạng thái của các ngõ vào digital, và các trạng thái ngắt của module đối với các module khác.

Thiết kế các ngắt cứng trong CPU

Để thiết kế các ngắt cứng trong CPU, gọi cấu hình phần cứng với công cụ *HW Config*. Thanh “*Interrupts*” trong hộp thoại “*Objet Properties*” chỉ ra các khối tổ chức ngắt cứng trong CPU bạn đang sử dụng. Bạn đặt cấp ưu tiên xử lý ở đây. Đối với mỗi OB ngắt cứng được sử dụng cũng như đối với cấp độ ưu tiên phải được đặt sẵn ít nhất 20 bytes ở dữ liệu cục bộ tạm thời.

Ngăn chặn và làm trễ các ngắt cứng

Bạn có thể điều khiển xử lý các ngắt cứng với các hàm hệ thống sau: SFC 39 DIS_IRT (ngăn chặn ngắt), SFC 40 EN_IRT (Cho phép các ngắt bị ngăn chặn). SFC 41 DIS_AIRT (Trì hoãn ngắt) và SFC 42 EN_AIRT (cho phép các ngắt đã trì hoãn).

5.12 Ngắt chu kỳ (Cyclic Interrupt)

Ngắt chu kỳ được kích theo chu kỳ ở một khoảng thời gian đặt trước. Ngắt chu kỳ (còn gọi là Watchdog Interrupt) gọi một khối tổ chức đặc biệt được đặt cho các kiểu ngắt này. Với ngắt chu kỳ cho phép bạn có thể tạo ra một chương trình xác định trong một khoảng thời gian mà nó không phụ thuộc vào thời gian xử lý của chương trình chu kỳ. Các khối tổ chức OB 30 đến OB 38 ở STEP 7 để xử lý các ngắt chu kỳ. Một trong chín khối tổ chức này có giá trị tùy thuộc vào loại CPU bạn đang sử dụng.

Thiết kế ngắt chu kỳ trong CPU

Ngắt chu kỳ được định nghĩa trong *HW Config* khi bạn định nghĩa các tham số cho CPU trong cấu hình phần cứng trong mục “Cyclic Interrupt” của hộp thoại “Object Properties”. Một ngắt chu kỳ có 3 tham số: Khoảng thời gian để thực hiện, độ dịch pha và cấp độ ưu tiên. Các giá trị có thể điều chỉnh cho ở khoảng thời gian và ở độ dịch pha là từ 1ms đến 1 phút; cấp độ ưu tiên tùy theo CPU có thể chọn lựa từ 2 đến 24 cũng như là 0 (= ngắt không tác động). Đối với mỗi OB ngắt chu kỳ sử dụng cũng như đối với cấp độ ưu tiên phải được đặt tối thiểu 20 byte trên thanh “Memory” của hộp thoại “Object Properties”.

Độ dịch pha (Phase offset)



các ngắt bị ngăn chặn), SFC 41 DIS_AIRT (trì hoãn ngắt) và SFC 42 EN_AIRT (cho phép ngắt bị trì hoãn).

5.13 Ngắt thời gian trong ngày (Time-of-Day Interrupts)

Bạn sử dụng ngắt thời gian trong ngày khi bạn muốn xử lý một chương trình tại một thời gian xác định, ví dụ một lần, hay từng chu kỳ hay hàng ngày. OB 10 đến OB 17 của STEP 7 được sử dụng cho mục đích này. Một trong tám khối tổ chức này có tác dụng tùy thuộc vào CPU sử dụng. Một ngắt thời gian của ngày chỉ có thể được tạo ra chính xác, nếu đồng hồ thời gian thực được điều chỉnh đúng.

Cài đặt ngắt thời gian của ngày trong CPU

Ngắt thời gian trong ngày được cài đặt trong cấu hình phần cứng HW Config. Mục *“Time-Of-Day-Interrupt”* trong hộp thoại *“Object Properties”* liệt kê các OB có giá trị cho mục đích này của CPU. Bạn có thể đặt cấp độ ưu tiên xử lý, thực hiện và khởi động điểm thời gian. Ngắt thời gian trong ngày được khởi động tự động với option *“Active”* khi CPU ở chế độ RUN.

Ngắt thời gian trong ngày có thể được xử lý ở một trong hai cách:

- Một lần; khối tổ chức tương ứng được gọi một lần tại điểm thời gian đã đặt hay
- chu kỳ; khối tổ chức tương ứng được khởi động ở hàng tháng, hàng giờ, hàng ngày, hàng tuần, hàng tháng, vào cuối tháng hay hàng năm.

Đối với các OB ngắt được sử dụng cũng như đối với cấp độ ưu tiên của nó phải được đặt sẵn ít nhất 20 byte ở dữ liệu cục bộ tạm thời ở mục *“Memory”* trong hộp thoại *“Object Properties”* của CPU. Bạn phải chú ý rằng ngắt thời gian trong ngày đã đặt cấu hình phải tồn tại trong chương trình người dùng, nếu không thì CPU sẽ gọi OB báo lỗi OB 85 (“lỗi xử lý chương trình”) hay đi vào chế độ STOP.

Xử lý ngắt thời gian trong ngày với các chức năng hệ thống

Để khởi động ngắt thời gian trong ngày, trước tiên phải đặt chỉnh thời gian khởi động và sau đó kích hoạt ngắt thời gian trong ngày. Cả hai khả năng này có thể được tách ra và thực hiện ở HW Config, và với các chức năng hệ thống. Với hàm hệ thống SFC 28 SET_TINT, bạn đặt điểm thời gian khởi động và chu kỳ của ngắt thời gian trong ngày; Với hàm hệ thống SFC 30 ACT_TINT, bạn khởi động ngắt thời gian trong ngày.



điểm SFC được gọi. Bạn có thể hủy bỏ ngắt trễ thời gian đã khởi động bằng cách gọi hàm hệ thống SFC 33 CAN_DINT một thời gian thứ hai; thì khối tổ chức liên hệ không được gọi nữa. Bạn hỏi trạng thái ngắt trễ thời gian với chức năng hệ thống SFC 34 QRY_DINT. Phải chắc chắn rằng OB ngắt trễ thời gian tồn tại, nếu không CPU sẽ gọi OB báo lỗi OB 85 (“lỗi xử lý chương trình”) hay đi vào chế độ STOP.

Ngắt trễ thời gian chỉ được thực hiện trong chế độ RUN của CPU. Bạn có thể khởi động ngắt trễ thời gian trong chương trình khởi động với việc gọi SFC 32 SRT_DINT. Trong trường hợp này, CPU phải ở chế RUN khi thời gian trễ đã chạy xong. Nếu không, CPU chờ gọi khối tổ chức cho đến khi kết thúc chương trình khởi động và gọi OB ngắt trễ thời gian trong chế độ RUN trước khi khởi động chương trình chính.

Ngăn chặn và trì hoãn ngắt trễ thời gian

Bạn có thể sử dụng các chức năng hệ thống sau đây để ảnh hưởng đến ngắt trễ thời gian: SFC 39 DIS_IRT (chặn ngắt), SFC 40 EN_IRT (cho phép ngắt đã bị trì hoãn), SFC 41 DIS_AIRT (Trì hoãn ngắt) và SFC 42 EN_AIRT (cho phép ngắt đã bị trì hoãn).

5.15 Ngắt đa xử lý

Ngắt đa xử lý có thể được sử dụng trong hoạt động đa xử lý để phản ứng với tất cả các CPU tham dự đồng bộ với một sự cố. SFC 35 MP_ALM được gọi để kích ngắt đa xử lý. Khối tổ chức OB 60 được sử dụng cho việc xử lý ngắt đa xử lý với cấp độ ưu tiên cố định 25. Chế độ đa xử lý chỉ có ở các trạm S7-400.

Đặt cấu hình ngắt đa xử lý trong CPU

Ngắt đa xử lý không được đặt cấu hình trong cấu hình phần cứng; nó có trong các CPU có khả năng đa xử lý. Mặc dù vậy, cũng phải đặt sẵn ít nhất 20 byte cho dữ liệu tạm thời ở mục “Memory” trong hộp thoại “Object Properties” của CPU, nó ở cấp ưu tiên 25.

Kích ngắt đa xử lý

Gọi hàm hệ thống SFC 35 MP_ALM để tạo ngắt đa xử lý. Bạn có thể gán một nhận biết để gọi hàm hệ thống, để chương trình có thể xác định CPU nào tạo ra ngắt. OB ngắt đa xử lý được gọi ở tất cả các CPU tại cùng một thời điểm. Điều này có nghĩa là ở CPU được gọi hàm hệ thống SFC 35 chờ gọi khối tổ chức OB 60 cho đến khi tất cả các CPU khác đã được thông báo sẵn sàng. Khối tổ chức cho ngắt đa xử lý chỉ được khởi động ở RUN.

Việc gọi hàm hệ thống SFC 35 MP_ALM vào chương trình khởi động bị loại trừ với một thông báo lỗi trở về.

Trì hoãn và ngăn chặn ngắt đa xử lý

Bạn có thể sử dụng các hàm hệ thống sau đây để ảnh hưởng đến sự xử lý của ngắt đa xử lý: : SFC 39 DIS_IRT (chặn ngắt), SFC 40 EN_IRT (cho phép ngắt đã bị trì hoãn), SFC 41 DIS_AIRT (Trì hoãn ngắt) và SFC 42 EN_AIRT (cho phép ngắt đã bị trì hoãn).



và sau đó lấy ra các nguyên nhân gây lỗi. Chức năng chẩn đoán hệ thống có thể phát hiện ra các module lỗi và đưa các thông tin lỗi này vào vùng đệm chẩn đoán. Vùng đệm chẩn đoán cũng chứa các thông tin về chế độ chuyển tiếp của CPU, ví dụ như sự cố là nguyên nhân đưa CPU vào chế độ STOP. Nội dung của vùng đệm chẩn đoán được giữ lại khi CPU ở chế độ STOP, trong khi reset bộ nhớ và khi mất nguồn cung cấp. Nó có thể được đọc trở lại sau khi nguồn cung cấp có trở lại và sau khi khởi động CPU.

Các lỗi đồng bộ

Hệ điều hành của CPU tạo ra một sự cố lỗi đồng bộ, nếu một lỗi xuất hiện trực tiếp trong xử lý chương trình. Nếu OB lỗi không đồng bộ không được lập trình, CPU sẽ đi vào chế độ STOP khi một sự cố lỗi đồng bộ xảy ra. Có hai kiểu lỗi khác nhau:

- Lỗi lập trình : khối OB 121 được gọi, và
- Lỗi truy xuất : khối OB 122 được gọi

OB lỗi đồng bộ có cùng cấp ưu tiên với khối tạo ra lỗi. Bởi vậy, các thanh ghi của khối truy xuất có thể bị truy xuất bởi các OB lỗi không đồng bộ. , và bởi vậy chương trình trong OB lỗi đồng bộ cũng có thể trả về các thanh ghi với nội dung thay đổi cho khối bị ngắt.

Nên chú ý rằng, khi một OB lỗi đồng bộ được gọi, thì 20 byte của thông tin khởi động được chứa trong L-Stack của cấp ưu tiên nguyên nhân gây lỗi, thêm vào đó dữ liệu cục bộ tạm thời mở rộng của OB lỗi không đồng bộ và dữ liệu cục bộ tạm của tất cả các khối được gọi vào OB này.

Nhập vào một giá trị thay thế

Bạn có thể gọi SFC 44 REPL_VAL để ghi một giá trị thay thế vào accumulator 1 từ một OB lỗi đồng bộ. Ví dụ: Giá trị có thể không được đọc nữa từ module ngõ vào. OB 122 (“lỗi truy xuất: access error”) được gọi và một giá trị thay thế được điền vào với SFC 44 REPL_VAL.

Lỗi không đồng bộ

Các lỗi không đồng bộ là các lỗi xảy ra không phụ thuộc vào sự xử lý chương trình. Khi một lỗi không đồng bộ xảy ra, hệ điều hành gọi một trong các khối tổ chức sau:

- Lỗi thời gian – OB 80

Lỗi thời gian có nghĩa là một OB được yêu cầu trong khi nó vẫn được xử lý, hay một ngắt thời gian của ngày trôi qua qua việc đặt chỉnh trước đồng hồ thời gian thực, ví dụ thời gian giám sát chu kỳ bị vượt quá.

➤ *Lỗi nguồn cung cấp – OB 81*

Khối này được gọi khi pin trên giá của CPU hay một giá mở rộng đã hết, pin backup hoàn toàn bị mất hay nguồn cung cấp 24 V bị mất.

➤ *Ngắt chẩn đoán – OB 82*

Với ngắt chẩn đoán sẽ thông báo một khối tương ứng của CPU sự cố chẩn đoán

➤ *Ngắt chèn vào/tháo ra – OB 83*

Hệ điều hành giám sát cấu hình module theo từng giây. Mỗi lần một module bị tháo ra hay chèn vào trong khi ở RUN, STOP và STARTUP, một lỗi vào được tạo trong vùng đệm chẩn đoán và danh sách trạng thái hệ thống, và OB 83 được gọi. Nếu một module thích hợp được cắm vào chỗ cắm được thiết kế, thì module được nạp tự động với các tham số đặt và bản ghi dữ liệu bởi CPU. Chỉ sau đó OB 83 được gọi để tường thuật module đã được cài trở lại.

➤ *Lỗi phần cứng CPU – OB 84*

Hệ điều hành gọi khối tổ chức OB 84 khi một lỗi giao tiếp xảy ra hay không xuất hiện (Mạng MPI, PROFIBUS-DP).

➤ *Lỗi xử lý chương trình – OB 85*

Hệ điều hành gọi khối tổ chức OB 85 khi một khối tổ chức không tồn tại được gọi, khi một khối xảy ra lỗi truy xuất của hệ điều hành (v.d. khối dữ liệu lỗi khi một khối hàm hệ thống SFB được gọi), và khi một lỗi truy xuất I/O xảy ra trong khi bộ đệm đang được cập nhật trên mạch hệ thống (tự động).

➤ *Sai giá module – OB 86*

Hệ điều hành gọi khối tổ chức OB 86 khi nó phát hiện sai giá module (nguồn mất, đứt dây, kết nối sai), hay hư hỏng một mạng hoặc một trạm I/O phân bố.

➤ *Lỗi truyền thông – OB 87*

Hệ điều hành gọi khối tổ chức OB 87 khi một lỗi truyền thông xảy ra (ví dụ: nhận biết điện tín sai hay lỗi chiều dài điện tín ở truyền thông dữ liệu toàn cục, lỗi trong khi đồng bộ thời gian).

5.17 Xử lý các sự cố ngắt

Hệ điều hành của CPU cho phép các sự cố ngắt xếp theo thứ tự để được xử lý tại thời điểm sau cùng. Sự xử lý các sự cố ngắt được điều khiển bởi các hàm hệ thống. Được phân biệt giữa các lỗi đồng bộ và lỗi không đồng bộ và các ngắt.

Che và không che các lỗi đồng bộ

Nếu một lỗi đồng bộ xuất hiện, thì hệ điều hành gọi khối tổ chức tương ứng hoặc – nếu OB không tồn tại- thì chuyển sang chế độ STOP. Hãy chắc chắn rằng các sự cố lỗi đồng bộ có thể được “che chắn” để chúng không làm cho CPU chuyển sang chế độ STOP hay gọi khối tổ chức báo lỗi. Khi một sự cố lỗi đồng bộ được che, CPU ghi nhớ điều này vào thanh ghi trạng thái sự cố.

Bạn có thể gọi hàm hệ thống SFC 36 MSK_FLT để che các sự cố lỗi đồng bộ. Mỗi lỗi đồng bộ được gán một bit vào mặt nạ lỗi. Sự che này chỉ có ảnh hưởng trong cấp ưu tiên hiện hành (cấp xử lý chương trình), mà SFC được gọi vào. Thanh ghi trạng thái sự cố được đọc bởi hàm hệ thống SFC 38 READ_ERR để phát hiện một lỗi đồng bộ đã xảy ra trong sự che. Hãy gọi hàm hệ thống SFC 37 DMSK_FLT nếu bạn muốn không che lỗi đồng bộ và cho phép xử lý lại.

Chặn, trì hoãn và cho phép các lỗi đồng bộ và các ngắt

Các lỗi đồng bộ và các ngắt trong xử lý chương trình chu kỳ thường không thể tiên đoán trước được. Chúng có thể ngắt chương trình chính tại bất kỳ thời điểm nào gọi chương trình ngắt hoặc chương trình lỗi. Các chương trình ngắt có ưu tiên thấp hơn cũng có thể bị ngắt theo con đường này. Sự ngắt này có thể có ảnh hưởng xấu khi phần chương trình ngắt phải được xử lý trong một thời gian xác định, ví dụ để đạt tới một thời gian phản ứng

ngắt, hay khi tuân tự câu lệnh không được phép ngắt, ví dụ ở việc đọc các giá trị liên quan của một module ngoại vi.

Bạn có thể sử dụng hàm hệ thống SFC 39 DIS_IRT để không cho phép các ngắt và các lỗi không đồng bộ. Sau khi chặn các sự cố xảy ra được loại bỏ; chúng gọi khối tổ chức liên quan để đưa CPU vào trạng thái STOP nếu OB ngắt không tồn tại. Bạn có thể chọn giữa chặn tất cả các ngắt và lỗi không đồng bộ, giữa chặn lỗi không đồng bộ và cấp độ ngắt (v.d. chỉ các ngắt cứng) và chặn khối tổ chức ngắt hay không đồng bộ riêng lẻ. Hoạt động chặn có tác dụng trong tất cả cấp độ ưu tiên (cho tất cả cấp độ xử lý chương trình). Nếu bạn muốn hủy bỏ hàm chặn, hãy gọi hàm hệ thống SFC 40 EN_IRT. Sau khi khởi động nóng (warm restart) và khởi động nguội (cold restart) thì sự xử lý tất cả các ngắt và lỗi không đồng bộ được cho phép trở lại.

Sự trì hoãn các ngắt và lỗi không đồng bộ với hàm hệ thống SFC 41 DIS_AIRT triệt tiêu sự xử lý của chúng. Sự xử lý được lấy lại, khi sự trì hoãn được hủy bỏ với hàm SFC 42 EN_AIRT. Sự trì hoãn xử lý ngắt và lỗi không đồng bộ cũng có thể được lồng vào. Trong trường hợp này, phải chắc chắn rằng sự trì hoãn được cho phép nhiều lần như nó được kích hoạt.

5.18 Các khối người dùng(tổng quan)

STEP 7 cho phép bạn có thể chia chương trình của bạn ra thành các phần riêng. Các phần riêng có một hàm xác định, cấu trúc hay mối liên hệ đối với nhiệm vụ công nghệ. Các phần chương trình này được gọi là các khối. Với các chương trình lớn và phức tạp thì cấu trúc chương trình được chia ra thành các khối riêng được đề nghị và đôi khi cần thiết. Các khối được gọi cho sự xử lý chương trình hoặc là nối tiếp nhau hoặc là tuần tự được lồng vào nhau.

Các khối tổ chức –OB (Organization blocks)

Các khối tổ chức (OBs) biểu diễn sự giao tiếp giữa hệ điều hành và chương trình người dùng. Hệ điều hành của CPU gọi các khối tổ chức khi xảy ra sự cố xác định, như các ngắt cứng hay ngắt thời gian của ngày. Chương trình chính được đặt trong khối tổ chức OB1. Các khối tổ chức khác được nhận dạng bởi các số xác định của các sự cố được gọi. Khả năng ngắt sự xử lý chương trình thực hiện theo qui tắc quyền ưu tiên.

Các khối hàm –FBs (Function blocks)

Các khối hàm (FBs) là thành phần có thể gắn tham số của chương trình người dùng, mà một phần có thể nhớ các dữ liệu cục bộ của chúng trong một khối dữ liệu cố định. Một khối dữ liệu khác có thể được gắn theo mỗi lần gọi khối FB. Khối dữ liệu được gắn theo này được gọi là “khối dữ liệu thứ cấp”(instance data block). Tổ hợp của việc gọi khối hàm và khối thứ cấp được gọi là “gọi thứ cấp”(call instance) hay gọi tắt là “thứ cấp”(instance). Các khối hàm cũng có thể lưu trữ các biến của chúng trong các khối dữ liệu thứ cấp của sự gọi các khối hàm. Điều này được gọi là “thứ cấp cục bộ”(local instance”.

Hàm – FCs (Functions)

Bạn sử dụng các hàm (FCs) để lập trình thường xuyên các hàm tự động được lặp lại. Bạn định nghĩa các đặc tính của chúng bằng cách gắn các tham số cho chúng. FCs đưa trở về giá trị cho các khối gọi. Giá trị của hàm là không bắt buộc. Ngoài giá trị hàm, các FC cũng có thể có các tham số ngõ ra. Các hàm không nhớ thông tin, chúng không có khối dữ liệu đi theo.

Khối dữ liệu –DBs (Data blocks)

Các khối dữ liệu (DBs) chứa dữ liệu của chương trình người dùng. Bằng cách lập trình các khối dữ liệu, bạn có thể xác định dữ liệu được lưu ở dạng nào. Bạn có thể bảo vệ các khối dữ liệu chống lại sự ghi đè (chỉ cho phép đọc), hay chỉ lưu trữ chúng trong bộ nhớ chương trình. Có hai cách sử dụng các khối dữ liệu: là khối dữ liệu toàn cục và khối dữ liệu thứ cấp. Khối dữ liệu toàn cục còn được gọi là khối dữ liệu “tự do” trong chương trình người dùng. Nó không được gắn một khối mã. Một khối dữ liệu thứ cấp được gắn cố định cho một khối hàm (FB). Một phần dữ liệu của khối hàm được chứa trong nó.

5.19 Cấu trúc của một khối

Các khối dữ liệu và các khối mã cơ bản được xây dựng từ 3 thành phần:

- Đầu khối
 - Nó chứa các tính chất của khối, v.d tên khối
- Phần khai báo
 - Các biến cục bộ của khối được khai báo trong phần này
- Phần chương trình (khối mã) hay phần khởi tạo (khối dữ liệu)



The data block is write-protected in the programmable controller là một tính chất chỉ cho các khối dữ liệu. Có nghĩa là bạn chỉ có thể đọc khối dữ liệu này bằng chương trình của bạn. Việc viết đè dữ liệu thì không thể và một thông báo lỗi xuất hiện.

Một khối dữ liệu với tính chất *Unlinked* thì chỉ được đặt trong bộ nhớ chương trình. Nó không liên quan đến sự xử lý. Các khối dữ liệu trong bộ nhớ chương trình không thể được ghi và chỉ có thể đọc với hàm hệ thống SFC 20 BLKMOV.

Tính chất *Standard Block* được đặt ở đầu khối của các khối chuẩn của Siemens.

Bảng khai báo

Bảng khai báo của một khối mã chứa định nghĩa của các biến cục bộ của khối. Đây là các biến mà chỉ được sử dụng trong khối này. Mỗi khối chứa temporary local data thì các kết quả trung gian được chứa trong nó. Hàm (FCs) và các khối hàm (FBs) có thể trao đổi dữ liệu với việc gọi khối bằng block parameters. Các tham số ngõ vào, ra và vào/ra chỉ có thể được đọc, ghi hay đọc và ghi. Các khối hàm chứa static local data trong khối dữ liệu thứ cấp.

Bạn xác định tên và kiểu dữ liệu của các biến cho mỗi loại biến trong bản khai báo. Bạn có thể đặt trước các tham số khối và dữ liệu cục bộ tĩnh của các khối hàm với các giá trị ban đầu.

Trong bảng khai báo của một khối dữ liệu, bạn xác định các địa chỉ dữ liệu được lưu trong khối dữ liệu này. Mỗi địa chỉ dữ liệu có một tên, một kiểu dữ liệu, một giá trị ban đầu và một chú thích. Bạn có thể tiết kiệm không gian nhớ bằng cách tóm tắt các biến BOOL thành một số của 8 và các biến BYTE thành các cặp. Với lập trình theo incremental, trong việc xem dữ liệu, bạn cũng có thể gán một actual value cho một địa chỉ dữ liệu được chuyển đổi trong khi nạp khởi tạo vào bộ nhớ làm việc. Các giá trị ban đầu được giữ trong bộ nhớ chương trình, trong khi chương trình làm việc với giá trị hiện hành trong bộ nhớ làm việc.

Phần chương trình

Phần chương trình của một khối mã chứa các câu lệnh được xử lý bởi CPU. Màn hình soạn thảo chương trình chỉ các câu lệnh ở dạng văn bản hay đồ họa, tùy thuộc vào ngôn ngữ lập trình bạn đang sử dụng. Khối đã biên dịch chỉ chứa các câu lệnh MC7. Hộp thoại “Object Properties” của khối hiển thị yêu cầu không gian của khối trong bộ nhớ chương trình và bộ nhớ

6 Truyền Thông

Truyền thông là sự trao đổi dữ liệu giữa các module lập trình. Hệ điều hành của SIMATIC đáp ứng tất cả các chức năng truyền thông. Hai module CPU có thể trao đổi dữ liệu với chỉ một đường cáp nối giữa chúng và không cần thêm phần cứng. Các module CP cung cấp các kết nối mạng hiệu quả cũng như kết nối với các hệ thống bên ngoài.

SIMATIC NET là thuật ngữ chung sử dụng cho truyền thông SIMATIC. Nó đảm nhiệm việc trao đổi thông tin giữa các bộ điều khiển lập trình và giữa các bộ điều khiển với các giao tiếp vận hành. Các phương pháp truyền thông khác nhau hiện có đáp ứng đầy đủ các yêu cầu điều khiển.

Việc trao đổi dữ liệu đòi hỏi các module phải có khả năng truyền thông (các CPU hay CP) và được nối mạng với nhau. Network là kết nối phần cứng giữa các trạm truyền thông. Việc trao đổi dữ liệu sử dụng một *connection* theo chương trình cụ thể (*communication utility*). Chương trình người dùng điều khiển dữ liệu truyền thông bằng các chức năng truyền thông (*communication functions*).

Mạng (Network), mạng con (Subnet)

Một mạng là một nhóm các thiết bị nối kết với nhau nhằm mục đích truyền thông. Nó gồm một hay nhiều mạng con liên kết với nhau. Những mạng con này có thể cùng hoặc khác loại.

Một mạng con kết hợp với tất cả các trạm truyền thông kết nối cứng, có cùng đặc tính vật lý và các thông số truyền (ví dụ tốc độ truyền) và sử dụng cùng một phương pháp truyền để trao đổi dữ liệu. Các mạng con SIMATIC có các chức năng khác nhau. MPI là mạng có giá thành hạ dùng cho lượng dữ liệu nhỏ. PROFIBUS truyền lượng dữ liệu từ nhỏ đến trung bình ở tốc độ cao. Industrial Ethernet điều khiển lượng dữ liệu lớn ở tốc độ cao. Cuối cùng là các kết nối point-to-point dùng kết nối nối tiếp với các thủ tục riêng.

Ứng Dụng Truyền Thông

Một ứng dụng truyền thông cho biết cách thức mà các trạm truyền thông trao đổi dữ liệu và xử lý dữ liệu. Việc xử lý dựa trên một thủ tục hoặc các thủ tục kết hợp giữa các đối tác truyền thông. SIMATIC nhận biết các

ứng dụng S7 functions, Global Data Communication, PROFIBUS_DP, PROFIBUS_FMS, PROFIBUS_FDL (SDA), ISO-Transport và ISO-on-TCP

Giao tiếp MPI kết nối trạm SIMATIC S7, M7 và C7. Nó cũng kết nối các trạm này với thiết bị lập trình và thiết bị giám sát

Kết nối tới Industrial Ethernet bằng các module CP cho phép truyền lượng lớn dữ liệu

Giao tiếp DP của S7 CPU và thiết bị lập trình qua bus làm cho PROFIBUS trở nên lý tưởng cho mạng con

Hình 6.1 Các mạng với SIMATIC

Kết nối

Kết nối xác định mối quan hệ giữa hai trạm truyền thông. Nó biểu diễn thị sự phân chia hợp lý của hai trạm để thực hiện một ứng dụng truyền thông. Nó cũng chứa các đặc tính đặc biệt như dạng kết nối (động và tĩnh) và cách thiết lập kết nối. SIMATIC nhận biết các dạng kết nối sau đây : kết nối S7, kết nối point-to-point, kết nối FMS và FDL, kết nối ISO-Transport và các kết nối ISO-on-TCI và UDP.

Các Chức Năng Truyền Thông

Các chức năng truyền thông là giao tiếp của chương trình người sử dụng với ứng dụng truyền thông. Các chức năng truyền thông được tích hợp trong hệ điều hành của CPU và được gọi bằng các khối hệ thống dùng cho truyền thông nội bộ của SIMATIC S7. Các khối có thể nạp hiện có để truyền thông với các thiết bị bên ngoài nhờ các vi xử lý truyền thông.

6.1 SIMATIC Subnet

MPI, mạng có giá thành thấp sử dụng cho lượng dữ liệu nhỏ

Mỗi CPU của SIMATIC S7 được trang bị một “giao tiếp đa điểm” (MPI, multi point interface). Giao tiếp này thiết lập một mạng con mà CPU, HMI và các thiết bị lập trình có thể trao đổi dữ liệu với nhau. Sự trao đổi dữ liệu được thực hiện qua giao thức riêng của Siemens.

MPI sử dụng cáp 2 dây hoặc một cáp sợi quang bằng thủy tinh hay nhựa làm đường truyền. Chiều dài tối đa của cáp cho một đoạn bus là 50 mét. Sử dụng bộ lặp RS 485 làm tăng chiều dài tối đa lên đến 1100 mét. Module nối cáp quang Optical Link làm tăng chiều dài cho phép thậm chí hơn 100 km. Tốc độ truyền thường là 187,5kbit/giây.

Số trạm tối đa là 32. Mỗi trạm nhận số lượng thời gian nhất định để truy cập bus và gửi dữ liệu. Khi thời gian này kết thúc, trạm sẽ chuyển tín hiệu (các quyền truy cập) sang trạm kế tiếp. Phương pháp này được gọi là “chuyển giao tín hiệu” (Token Passing).

Bạn có thể dùng truyền thông dữ liệu toàn cục (global), truyền thông trạm ngoài SFC hay truyền thông SFB để chuyển dữ liệu giữa các CPU với một mạng con MPI mà không cần thêm module phụ nào.

PROFIBUS, trao đổi lượng dữ liệu nhỏ và vừa ở tốc độ cao

PROFIBUS là các chữ viết tắt của “Process Field Bus”. Nó là tiêu chuẩn riêng của hãng được trình bày trong tập 2 của EN 50170 và cho nối mạng các thiết bị.

Đường truyền truyền thông là cáp 2 dây bọc hoặc cáp quang bằng thủy tinh hay nhựa. Tốc độ truyền quyết định chiều dài của cáp trong đoạn truyền. Chiều dài tối đa ở tốc độ truyền cao nhất (12Mbit/giây) là 100m và 1000 m ở tốc độ truyền thấp nhất (9,6 kbit/giây). Bạn có thể kéo dài mạng bằng các bộ khuếch đại (Repeater) hoặc các module kết nối quang (Optical Link Module).

Số trạm tối đa là 127. Các trạm có thể là trạm chủ động hay thụ động. Một trạm chủ động nhận lượng thời gian nhất định để truy cập bus và gửi dữ liệu. Sau khi thời gian ấn định kết thúc, trạm sẽ chuyển tín hiệu (các quyền truy cập) sang trạm kế tiếp. Phương pháp này được gọi là chuyển giao tín hiệu. Khi các trạm thụ động (slaves) được gán với một trạm chủ động (master), trạm chủ sẽ liên lạc với các trạm tớ của nó khi có tín hiệu. Một trạm thụ động không thể tự nhận tín hiệu.

Bạn có thể dùng một mạng con PROFIBUS để thiết lập liên kết với ngoại vi phân tán, cần có ứng dụng truyền thông PROFIBUS_DP. Nhiều CPU S7 đã có sẵn một DP master. Ví dụ DP master IM 467 hoặc CP 342-5. Bạn có thể sử dụng mạng con này để truyền thông các SFC hoặc SFB.

Dữ liệu có thể được truyền bằng PROFIBUS-FMS và PROFIBUS-FDL khi sử dụng phù hợp các Module CP. Các khối có thể nạp (giao tiếp FMS hoặc giao tiếp SEND/RECEIVE) có thể sử dụng như giao tiếp với chương trình người dùng.

Industrial Ethernet, trao đổi lượng dữ liệu lớn ở tốc độ cao

Industrial Ethernet là mạng con dùng để nối các máy vi tính và các bộ điều khiển lập trình. Mạng con này chủ yếu dùng cho các ứng dụng trong công nghiệp. Nó đáp ứng tiêu chuẩn IEEE 802.3.

Về mặt điện, kết nối vật lý là một cáp đồng trục 2 dây bọc hoặc một cáp xoắn đôi ("industrial twisted pair"). Về mặt quang học, kết nối vật lý là một cáp quang thủy tinh. Độ 1,5 km ở kết nối điện, còn ở cáp quang là 4,5 km. Tốc độ truyền được ấn định là 10 Mbit/giây.

Mạng Công Nghiệp (Industrial Ethernet) có thể nối mạng trên 1000 trạm. Trước khi truy cập mạng, mỗi trạm kiểm tra để xem có trạm nào đang gửi dữ liệu hay không. Khi một trạm đang gửi dữ liệu, trạm kia phải đợi một thời gian trước khi thử truy cập mạng lần nữa. Đây được gọi là phương pháp truy cập CSMA/CD. Tất cả các trạm có quyền như nhau.

Bạn có thể trao đổi dữ liệu với mạng truyền thông SFB qua Industrial Ethernet và sử dụng các chức năng S7. Industrial Ethernet đòi hỏi các module CP phù hợp. Bạn có thể dùng các CP này để thiết lập các kết nối ISO-Transport hoặc ISO-on-TCP và điều khiển các kết nối này bằng giao tiếp SEND/RECEIVE.

Kết nối point-to-point, kết nối nối tiếp bằng các giao thức đặc biệt

Kết nối point-to-point (PTP) dùng kết nối nối tiếp để điều khiển việc trao đổi dữ liệu. SIMATIC Manager xử lý kết nối point-to-point như một mạng con. Việc cấu hình cũng tương tự như một mạng con.

Đường truyền truyền thông là một cáp điện. Vị trí của nó tùy thuộc vào giao tiếp. Hiện có giao tiếp RS 232C (V.24), 20mA (TTY) và RS 422/485. Tốc độ truyền từ 300 bit/giây đến tối đa là 19,2 kbit/giây đối với giao tiếp 20 mA, 76,8 kbit/giây cho RS 232C và RS 422/485. Chiều dài cáp tùy thuộc vào thiết kế giao tiếp và tốc độ truyền. Chiều dài 10 m cho RS 232C, 1000m cho giao tiếp 20mA với 9.6 kbit/giây, và 1200m cho RS422/585 với 19,2 kbit/giây.

3964 (R), RK 512 là giao thức để thực hiện trình điều khiển máy in và ASCII (ASCII driver). Bạn có thể dùng ASCII driver định nghĩa phương pháp của riêng bạn. Các trình điều khiển đặc biệt có thể nạp sử dụng trong những trường hợp bất thường.

Giao tiếp AS, mạng có giá thành thấp nhất

Các mạng giao tiếp AS (giao tiếp cảm biến/cơ cấu chấp hành) thiết kế các bộ cảm biến và cơ cấu chấp hành phù hợp với giao tiếp AS (AS Interface) theo chỉ tiêu kỹ thuật IEC TG 178. SIMATIC Manager không hiển thị giao tiếp AS như dạng một mạng con. Chỉ giao tiếp AS master mới được cấu hình bằng công cụ cấu hình phần cứng HW Config hoặc công cụ cấu hình mạng NetPro.

Đường truyền truyền thông là một cáp đôi không bọc. Cáp này cung cấp cho các bộ cảm biến và cơ cấu chấp hành cả dữ liệu và nguồn (cần nguồn thứ cấp). Với một khuếch đại, chiều dài tối đa của mạng là 300 m. Tốc độ truyền được ấn định là 167 kbit/giây.

Giao tiếp AS chủ (AS-Interface-Master) điều khiển tới 31 giao tiếp AS tớ (AS-Interface-slave) bằng cách xoay vòng theo chu kỳ. Điều này cần một thời gian phản hồi.

6.2 Các Ứng Dụng Truyền Thông

Tùy thuộc vào kết nối được chọn mà các ứng dụng truyền thông khác nhau sẽ điều khiển việc trao đổi dữ liệu qua các mạng con. Dưới đây mô tả cách sử dụng các ứng dụng này :

- Các chức năng PG : các chức năng phục vụ, truyền và kiểm tra. Chẳng hạn như, PC sử dụng chúng để thực hiện các chức năng “giám sát biến”

(monitor variable) hoặc “đọc bộ đếm chẩn đoán” (read diagnostic bufer) hoặc nạp chương trình người dùng.

- Các chức năng giao tiếp vận hành : Các OP (Operator Panels) sử dụng các chức năng này để đọc và ghi các biến.
- Truyền thông SFB : ứng dụng cho việc điều khiển có lượng dữ liệu lớn. Khởi động bằng cách gọi SFB trong chương trình người dùng với các chức năng điều khiển và giám sát; tĩnh, các kết nối được đặt cấu hình.
- Truyền thông SFC : ứng dụng cho việc điều khiển có lượng dữ liệu tới 76 bytes cho một lần truyền. Khởi động bằng cách gọi SFC trong chương trình người dùng; động, các kết nối không được cấu hình.

Các chức năng S7 có thể được thực hiện bằng các mạng con MPI, PROFIBUS và Industrial Ethernet

Global data communication (Truyền dữ liệu toàn cục) truyền đi lượng dữ liệu nhỏ giữa một vài CPU. Nó không đòi hỏi lập trình phụ trong chương trình người dùng. Việc truyền có thể theo chu kỳ hoặc theo sự kiện điều khiển. Truyền thông dữ liệu toàn cục là một phương pháp Broadcast. Tình trạng truyền thông được ghi nhận. Truyền thông dữ liệu toàn cục cần MPI bus hoặc bus truyền thông.

PROFIBUS-DP (Decentralized Peripherals) chuyển dữ liệu giữa chủ và tớ cho ngoại vi phân tán. Nó được tiêu chuẩn hóa theo tiêu chuẩn 50170 volume 2. Bạn có thể sử dụng ứng dụng này để đặt địa chỉ cho các slaves S7 và các slaves không theo tiêu chuẩn SIMATIC với một mạng con PROFIBUS.

PROFIBUS-FMS (Fieldbus Message Specification) đưa ra các ứng dụng cho việc truyền các biến cấu trúc (các biến FMS) được tiêu chuẩn hóa theo tiêu chuẩn 50170 volume 2. Truyền thông dùng các kết nối tĩnh và chỉ một mạng con PROFIBUS.

PROFIBUS-FDL (Fieldbus Data Link) truyền dữ liệu bằng chức năng SDA (Send Data với Acknowledge). Nó được tiêu chuẩn hóa theo volume 2 của EN 50170. Truyền thông dùng các kết nối tĩnh. Chẳng hạn, một PROFIBUS-FDL dùng một mạng con PROFIBUS để trao đổi dữ liệu với một bộ điều khiển SIMATIC S5.

ISO-Transport truyền dữ liệu theo chuẩn ISO 8973, class 4. Truyền thông dùng các kết nối tĩnh. Chẳng hạn như, ISO-Transport dùng Industrial Ethernet để trao đổi dữ liệu với một bộ điều khiển SIMATIC S5.

	Kết nối SFB (chỉ dùng cho S7-400)	Bảng kết nối Gọi SFB
Mạng PTP (point - to - point)		
Các module	Ứng dụng truyền thông	Cấu hình, giao tiếp.
CP340 CP 441 –1	ASC II protocol, 3964 (R) Printer driver	Công cụ cấu hình sẵn, khối có thể nạp bằng CP 441 : SFB
CP 341	ASC II protocol, 3964 (R) RK 512, Printer driver Special driver	Công cụ cấu hình sẵn, khối có thể nạp bằng CP 441 : SFB

6.3 Cấu Hình Mạng

Cơ sở của truyền thông SIMATIC là việc nối mạng các trạm. Các mạng con và các module cáp truyền thông trên các đối tượng yêu cầu. Bạn có thể tạo ra các mạng con và các trạm mới theo trình tự thư mục hình cây (project hierarchy) với SIMATIC Manager và sau đó thêm các module có khả năng truyền thông (CPU và CP) bằng cấu hình phần cứng. Đồng thời, bạn ghép các giao tiếp truyền thông của các module này với một mạng con. Sau đó, bạn định rõ những liên hệ truyền thông giữa các module này (các kết nối) trong bảng kết nối.

NetPro cung cấp biểu diễn đồ họa, tài liệu của các mạng và các trạm của chúng. Bạn cũng có thể cài đặt các mạng con và các trạm của bạn bằng NetPro. Sau đó, ghép các trạm với các mạng con và xác định những đặc tính từng trạm của các module truyền thông. Các bước dưới đây sẽ chỉ cho bạn biết làm thế nào để sử dụng công cụ cấu hình mạng NetPro xác định các mối liên hệ truyền thông trong một dự án:

- Mở mạng con MPI chuẩn trong thư mục Project. Nếu không có, hãy set up mới bằng cách chèn một mạng con
- Bây giờ hãy set up trạm. Nếu cần, hãy thêm các mạng con phụ bằng NetPro.
- Hãy mở các trạm và cung cấp cho chúng các module truyền thông.
- Kết nối module tới mạng con thích hợp
- Điều chỉnh các thông số mạng nếu cần.
- Nếu cần, định rõ các kết nối truyền thông trong bảng kết nối.

Cấu hình mạng phải được dịch trước khi nạp các thông số truyền thông tới hệ thống đích. Khi được dịch dữ liệu được lưu trong mục *system data* ở Blocks folder. Bạn có thể gán các thông số truyền thông tới các trạm riêng lẻ bằng NetPro hoặc chuyển tất cả các dữ liệu cấu hình đến các trạm bằng SIMATIC Manager.

Cửa sổ mạng

Bạn phải cài đặt một Project trước khi khởi động công cụ cấu hình mạng NetPro. SIMATIC Manager tự động cài đặt một mạng con MPI khi bạn tạo một dự án. Nhấp đúp vào mạng con này hoặc mạng con bất kỳ để khởi động NetPro. Bạn cũng có thể khởi động NetPro bằng cách mở mục *Connections* trong thư mục CPU. Phần trên của cửa sổ NetPro sẽ hiển thị tất cả các mạng con, các trạm (dạng các nodes) và các kết nối của chúng đã cài đặt cho dự án. Nếu một module “connectable“ (CPU S7 –400 chẳng hạn) được chọn trong phần trên của cửa sổ, thì phần dưới sẽ chỉ ra bảng kết nối.

Hình 6.2 Ví dụ về cấu hình mạng bằng cửa sổ mạng, bảng kết nối, network object catalog

Cửa sổ thứ hai hiển thị danh mục mạng với các trạm SIMATIC, các mạng con và các trạm DP hiện có. Bạn có thể mở và đóng hiển thị danh

mục với VIEW –CATALOG, và có thể định vị cửa sổ danh mục ở bên phải của cửa sổ cấu hình bằng cách nhấp đúp vào thanh tiêu đề.

Chọn và sắp xếp các phần tử

Bắt đầu cấu hình bằng cách chọn một mạng con (subnet). Nhấp vào mạng con trong danh mục và kéo nó đến cửa sổ mạng. Mạng con xuất hiện thành một đường ngang trong cửa sổ. Sử dụng phương pháp tương tự cho các trạm, nhưng chưa cần nối chúng với một mạng con. Nếu bạn cứ cố kéo các đề mục bất hợp lý đến cửa sổ mạng, thì một “dấu hiệu cấm” (prohibited) sẽ xuất hiện trên con trỏ chuột.

Các trạm vẫn còn “trống”. Nhấp đúp một trạm để mở cấu hình phần cứng. Bây giờ bạn có thể cấu hình trạm hoặc ít nhất là (các) module bằng một kết nối mạng. Hãy lưu trạm rồi quay về NetPro. Bạn sẽ thấy được giao tiếp của module truyền thông hiển thị như một hộp nhỏ phía dưới phần hiển thị của các module. Nhấp vào hộp này, giữ chuột và kéo nó đến mạng con thích hợp. Kết nối với mạng con sẽ làm xuất hiện một đường dọc.

NetPro cũng biểu diễn đồ họa của hệ thống DP master với tất cả các DP slaves của nó. Sau đó, chúng có thể được soạn thảo. Bạn có thể cho phép hoặc không cho phép hiển thị các DP slaves. Với các mạng mở rộng hay với hệ thống master xác định, chúng tôi đề nghị tập trung vào một hệ thống master với tất cả các slaves.

6.4 Kết nối cấu hình

Các kết nối mô tả mối liên hệ truyền thông giữa hai thiết bị. Kết nối cấu hình theo một trong hai điều kiện sau đây :

- Truyền thông SFB được thiết lập giữa hai thiết bị SIMATIC S7.
(truyền thông nhờ kết nối đã được đặt cấu hình).
- Đối tác truyền thông không phải là một trạm SIMATIC.

Bảng kết nối

Các kết nối truyền thông được cấu hình trong bảng kết nối. Để cấu hình kết nối, hãy chọn một CPU S7-400 trong công cụ cấu hình mạng NetPro. Bảng kết nối xuất hiện bên dưới cửa sổ mạng. Nhập một kết nối truyền thông mới bằng cách dùng INSERT → CONNECTION hoặc bằng cách nhấp đúp vào đường trống.

Tạo một bảng kết nối cho mỗi “active” CPU. Hãy nhớ rằng bạn không thể tạo một kết nối cho một CPU S7-300 được bởi vì các CPU này chỉ có thể là các đối tác “passive” trong kết nối S7. Trong hộp thoại “New connection”, hãy chọn đối tác kết nối trong hộp “Station” và “module”. Trạm và các module phải tồn tại. Bạn cũng xác định rõ dạng kết nối trong cửa sổ này. Nếu bạn muốn bố trí các đặc điểm kết nối bổ sung, sử dụng “Show Properties dialog box”.

Các kết nối một phía và hai phía

Số lượng kết nối có thể có tùy thuộc vào CPU riêng biệt. STEP 7 tạo một kết nối ID cho mỗi kết nối và mỗi thông số. Bạn sẽ cần thông tin này khi sử dụng các khối truyền thông trong chương trình của mình.

Với *one-sided connection*, truyền thông chỉ có thể kích được bởi một đối tác. Ví dụ : Truyền thông SFB giữa một CPU S7-400 và S7-300. Mặc dù một S7-300 không có chức năng truyền thông SFB, nhưng các dữ liệu có thể được trao đổi với một S7- 400 bằng SFB 14 GET và SFB 15 PUT. Hệ điều hành điều khiển việc trao đổi dữ liệu này và chương trình người dùng không sử dụng truyền thông này trên CPU S7-300. Bạn hãy cấu hình một kết nối một phía trong bảng kết nối của “active” CPU STEP 7 chỉ gán một *ID local* bởi vì bạn chỉ có thể nạp kết nối này vào trạm cục bộ.

Với *two-sided connection*, cả hai đối tác đều tham gia truyền thông. Ví dụ : Hai CPU S7-400 với các chức năng truyền thông SFB 8 BSEND và SFB 9 BRCV. Bạn chỉ cấu hình kết nối hai phía một lần cho một trong hai đối tác. STEP 7 gán một *ID local* (cho trạm hiện thời đang mở) với một *partner ID*, và tạo dữ liệu kết nối cho cả hai trạm. Bạn phải nạp vào cả hai bảng kết nối để mỗi đối tác đều có bảng kết nối.

Dựa vào mạng con và thủ tục truyền, hãy chọn một dạng kết nối. Đây có thể là một kết nối S7 (theo tiêu chuẩn giữa hai thiết bị S7), PTP (point-to-point) hoặc kết nối ISO-on-TCP (theo tiêu chuẩn cho kết nối thiết bị ngoài).

Kết nối mở rộng nhiều dự án

Để truyền thông dữ liệu giữa hai module S7 thuộc về các dự án (project) khác nhau của SIMATIC, hãy cấu hình đối tác kết nối trong bảng kết nối như “unspecified” (cho mỗi trạm cục bộ trong cả hai dự án).

Hãy chắc chắn các dữ liệu kết nối của cả hai dự án đều giống nhau bởi vì STEP 7 không kiểm tra điều này. Lưu lại và dịch, sau đó nạp dữ liệu kết nối vào từng trạm cục bộ.

Kết nối với những trạm không phải S7

Bạn cũng có thể định rõ các đối tác kết nối khác ngoài trạm S7 trong một dự án. Các trạm không phải là trạm S7 gồm thiết bị của các nhà sản xuất khác và các trạm S7 trong dự án khác, các thiết bị lập trình PG/PC và SIMATIC S5.

Bạn không thể cấu hình kết nối trừ phi trạm non-S7 ở dạng một object trong dự án (project), và sau khi bạn đã nối trạm non-S7 theo các đặc tính trạm với mạng con thích hợp. Ví dụ : Hãy chọn trạm trong NetPro, chọn các thuộc tính EDIT → OBJECT và nối trạm với mạng con mong muốn trong mục “interfaces”.

6.5 Cấu hình cho thiết bị ngoại vi phân tán

Về cơ bản, cấu hình các ngoại vi phân tán giống như của các module tập trung (centralize). Thay vì sắp xếp các module trong một giá đỡ (rack), bạn ghép các trạm DP (PROFIBUS nodes) với một hệ thống DP master. Chúng tôi đề nghị thực hiện theo thứ tự sau đây :

1. Trong SIMATIC Manager, hãy cài đặt một dự án (project) hoặc mở cái có sẵn.
2. Sử dụng SIMATIC Manager, cài đặt một mạng con PROFIBUS trong dự án, và nếu cần, set bus.
3. Dùng SIMATIC Manager một lần nữa, cài đặt trạm chủ trong dự án là DP chủ (master), chẳng hạn như trạm S7-400. Nếu hệ thống của bạn chứa các DP slaves thông minh, hãy cài đặt các trạm slaves phù hợp, ví dụ trạm S7-300. Khởi động cấu hình phần cứng bằng cách mở trạm chủ.
4. Dùng cấu hình phần cứng đặt một DP master trong trạm chủ. Đây có thể là một CPU có giao tiếp DP. Gán mạng con PROFIBUS mà bạn cài đặt trước đó với giao tiếp DP. Giờ đây, bạn đã có một hệ thống DP master. Nếu thích, bạn có thể cấu hình module khác sau đó. Lưu lại và biên dịch (compile) station.
5. Nếu bạn đã cài đặt một trạm S7 cho một DP slave thông minh rồi, hãy mở nó trong cấu hình cứng và “chèn” module với giao tiếp DP mong muốn. Đây có thể là một CPU S7-300 với giao tiếp DP cấu thành hoặc một ET 200X, module nền BM 147/CPU chẳng hạn. Nối giao tiếp DP với “DP slave”. Ghép PROFIBUS mạng con được cài đặt trước với giao tiếp DP. Sau đó, cấu hình giao tiếp dữ liệu người dùng theo quan điểm của DP slave. Nếu thích, bạn có thể cấu hình các module khác sau đó. Lưu lại và biên dịch (compile) station.

Thực hiện tương tự cho các trạm phụ mà bạn muốn sử dụng như intelligent DP slaves

6. Hãy mở trạm chủ bằng hệ thống DP master và dùng chuột kéo trạm PROFIBUS (DP slaves nhỏ gọn và kiểu module) từ catalog phần cứng đến hệ thống DP master. Gán địa chỉ trạm, set địa chỉ đầu và địa chỉ chẩn đoán của module.
7. Nếu đã cài đặt các intelligent DP slaves, dùng chuột kéo biểu tượng phù hợp (trong catalog phần cứng dưới mục “PROFIBUS” và “Configured Stations”) từ catalog đến hệ thống DP master.

Mở biểu tượng và gán một intelligent DP slave đã được cấu hình (“coupling”). Định rõ địa chỉ trạm. Cấu hình giao tiếp dữ liệu người dùng theo DP master hoặc của CPU master trung tâm. Sử dụng cách tương tự cho intelligent DP slave.

8. Lưu trữ và biên dịch tất cả các trạm. Hệ thống DP chủ được cấu hình. Bây giờ, có thể thêm các CPU hoặc các DP tớ (slave) khác vào cấu hình hoặc nhập bất kỳ các thông số module còn thiếu.

Hình 6.4 ví dụ về DP master trong cấu hình phần cứng

Hình 6.5 Ví dụ về DP master trong cấu hình với netpro

6.6 Địa chỉ trong hệ thống DP master

Hệ thống DP chủ (master) có DP chủ và tất cả các DP tớ (slave) được tích hợp trong cấu trúc địa chỉ của CPU trung tâm. Bởi vì CPU trung tâm đặt địa chỉ cho các DP tớ như các module tập trung (centralized). Hệ thống DP chủ chứa các địa chỉ sau đây:

➤ Địa chỉ trạm

Mỗi trạm trong mạng PROFIBUS mạng con có một địa chỉ duy nhất. Địa chỉ trạm này là số trạm để phân biệt nó với các trạm khác trong mạng con.

➤ Địa chỉ vị trí (Geographic Address)

Địa chỉ vị trí của một DP slave là địa chỉ slot của một module tập trung. Nó chứa hệ thống DP chủ có ID được xác định trong khi cấu hình, và địa chỉ trạm trên PROFIBUS tương ứng với số giá đỡ (rack number). Với các DP slave, địa chỉ cũng chứa số slot (slot number). Khi các module có những module phụ (submodule) thì địa chỉ cũng chứa slot của module phụ. Đối với S7-300 thì việc đánh số slot của các module I/O bắt đầu từ 4.

➤ Địa chỉ đầu của module

Địa chỉ đầu của module giúp bạn truy cập vào dữ liệu người dùng của một DP slave hoặc một module trong modular DP slave. Địa chỉ này là địa chỉ đầu module của một CPU. Các bytes dữ liệu người dùng của một DP slave được đặt trong vùng truyền của DP master trên P bus của CPU. Giống như bất kỳ module tập trung nào, các bytes dữ liệu người dùng

này có thể được nạp hoặc chuyển đến vùng nhớ của CPU. Nếu dữ liệu là 3 bytes hoặc lớn hơn 4 bytes, thì phải sử dụng các chức năng hệ thống SFC .

➤ Địa chỉ chẩn đoán

Các module- như DP master hoặc một nguồn cung cấp phụ không có dữ liệu người dùng nhưng có dữ liệu chẩn đoán- được gán một địa chỉ chẩn đoán. Địa chỉ chẩn đoán chiếm một bytes trong vùng địa chỉ của các ngõ vào ngoại vi.

Vùng địa chỉ
logic của CPU

Modules kết nối tập trung

Modules
kết nối tập
trung

Địa chỉ module

Địa chỉ trạm

Địa chỉ vị trí
(ví dụ DP master)

Hình 6.6 các địa chỉ trạm SIMATIC S7 có hệ thống DP master.

Vùng Nhớ Truyền dữ liệu Bằng Các intelligent DP slaves

Địa chỉ ngõ vào và ra của các DP tớ được đặt trong vùng nhớ ngoại vi của CPU trung tâm. Bây giờ, CPU trung tâm được xem như là CPU chủ. Nó không truy cập trực tiếp vào các module input/output của các DP tớ. Vì lý do này mà mỗi DP tớ có một vùng nhớ truyền được chia ra các vùng phụ có chiều dài khác nhau và đồng nhất dữ liệu. Tùy thuộc vào sự phân chia ấy mà mỗi DP tớ sẽ giao tiếp với CPU chủ như một DP slave compact - modular.

Bạn nhập địa chỉ của vùng nhớ truyền (transfer memmory) trong lúc cấu hình. Địa chỉ được định rõ theo quan điểm của CPU tớ suốt thời gian cấu hình của các DP tớ thông minh. Địa chỉ được định theo CPU chủ khi DP tớ được thêm vào hệ thống DP chủ. Theo DP chủ (đúng hơn là CPU chủ trung tâm), địa chỉ vùng nhớ truyền không thể xung đột với các địa chỉ của các

module khác trong trạm S7 (trung tâm). Theo CPU tổ, địa chỉ vùng nhớ truyền không thể trùng lặp với địa chỉ module của DP slave.

Đối với việc truy cập dữ liệu người dùng và đồng nhất dữ liệu, các vùng địa chỉ của vùng nhớ truyền được xử lý như các module riêng lẻ (ví dụ địa chỉ thấp nhất của vùng địa chỉ là địa chỉ đầu của module).

6.7 Các Chức Năng DP Đặc Biệt

Đồng bộ hoá DP slaves bằng SYNC và FREEZE

DP master đọc dữ liệu ngõ vào của DP slaves liên tục và ghi nhận các trạng thái tín hiệu theo trình tự thời gian hiện hành. Đồng thời, lệnh FREEZE đọc dữ liệu input liên quan được phân phối cho DP slave. Lệnh này kết hợp các DP slaves thành một nhóm để đóng băng (freeze) các trạng thái tín hiệu input hiện hành (một cách đồng bộ), để DP master có thể gọi các dữ liệu này theo chu kỳ. Những tín hiệu input này vẫn duy trì giá trị của chúng cho đến khi một lệnh FREEZE khác yêu cầu các DP slaves đọc và duy trì các tín hiệu input hiện thời lần nữa hoặc cho đến khi có lệnh UNFREEZE huỷ bỏ FREEZE.

Cách tương tự cũng được dùng cho các tín hiệu output. DP master ghi các dữ liệu output cho các DP slave liên tục, và các tín hiệu được cung cấp theo thứ tự này. Dữ liệu ngõ ra liên quan đến lệnh SYNC được phân phối cho một số DP slave được xử lý đồng thời. Đồng thời lệnh SYNC kết hợp các DP slaves thành một nhóm đưa ra các trạng thái tín hiệu ngõ ra của chúng cùng lúc (một cách đồng bộ) và duy trì các trạng thái này không đổi. Bây giờ DP master có thể ghi các trạng thái tín hiệu mới liên tiếp vào các DP slaves. Sau khi truyền xong, dùng lệnh SYNC một lần nữa để bảo đảm đưa ra đồng bộ các tín hiệu ngõ ra mới. Các DP slave vẫn duy trì trạng thái của tín hiệu ngõ ra cho đến khi bạn đưa ra những giá trị hiện hành bằng lệnh SYNC hoặc hủy lệnh SYNC bằng lệnh UNSYNC.

Việc gọi chức năng hệ thống SFC II DPSYC_FR kích một lệnh SYNC hoặc FREEZE chương trình người dùng. Sau đó, DP master sẽ gửi lệnh này tới tất cả DP slaves cùng lúc.

Chỉ có thể dùng SYNC và FREEZE nếu DP master và các DP slave có

thể xử lý chức năng này và xem như bạn đã cấu hình nhóm lệnh SYNC/FREEZE bằng cấu hình phần cứng.

Bạn có thể tạo được 8 nhóm lệnh SYNC/FREEZE trên mỗi hệ thống DP master. Những nhóm này thực hiện hoặc một lệnh SYNC hoặc một lệnh FREEZE hoặc cả hai. Cấu hình nhóm SYNC/FREEZE sau khi cấu hình hệ

thống DP master và tất cả các DP slave hiện có trong hệ thống DP master. Để thực hiện điều này, hãy chọn hệ thống DP master (thanh có dấu chấm đen trắng), sau đó, soạn thảo Object properties. Trong hộp thoại xuất hiện, nhấp chuột vào nút “Properties” và xác định lệnh nào mà các nhóm phải thực hiện. Sau đó, gán các DP slave cho các nhóm riêng lẻ.

Các ngõ ra có thời gian điều khiển bằng nhau

DP master thường điều khiển các DP slave của nó theo chu kỳ liên tục. Tuy nhiên, khoảng thời gian có thể thay đổi do truyền thông S7 lúc điều khiển thực hiện chương trình đã thực hiện các chức năng điều khiển qua mạng con PROFIBUS. Nếu DP master được thiết kế để điều khiển việc này, bạn có thể cài đặt các chu kỳ bus có khoảng cách bằng nhau (được xem là chu kỳ bus không đổi) khi bạn muốn các ngõ ra được điều khiển bởi các ngoại vi phân tán với khoảng thời gian bằng nhau bằng nhau thì DP chủ chỉ có thể là chủ cấp 1 (Class 1 Master) trên PROFIBUS.

Cấu hình chu kỳ bus có khoảng cách bằng nhau sau khi có tất cả các DP slave trong hệ thống DP master. Trong SIMATIC Manager, hãy chọn mạng con PROFIBUS và sau đó là EDIT→ OBJECT PROPERTIES. Trong hộp thoại Object Properties, trên mục “Network settings” nhấp nút “options” và nhập giá trị mục “Constant Bus Cycle”. STEP 7 tính toán khoảng cách thời gian bằng nhau theo ý muốn dựa trên thông số đã nhập. Các thông số này gồm các yếu tố sau: chuẩn (standard), failure-tolerant hoặc tối ưu hoá tốc độ (speed-optimized).

“Listening In” Bằng Truyền Thông Chéo Trên PROFIBUS

Trong hệ thống DP master, một DP master điều khiển các DP slave gán với nó. Nếu các trạm được thiết kế nhằm mục đích này thì nút khác (chủ hay tớ được gọi là *receiver*) có thể “listen in” PROFIBUS để xác định dữ liệu ngõ vào nào mà một DP slave (gọi là *sender*) gửi cho DP master. Đây được gọi là truyền thông chéo (hay truyền thông trực tiếp).

Cấu hình truyền thông chéo sau khi tất cả các trạm được nối với mạng con PROFIBUS. Chọn giao tiếp DP trong trạm nhận (*receiver*) và chọn EDIT→ OBJECT PROPERTIES. Trong hộp thoại Object Properties, nhấp nút “New” trên mục “Direct Communication”. Sau đó, chỉ định trạm gửi và các vùng địa chỉ để *sender* và *receiver* trao đổi thông tin với nhau.

6.8 Truyền Thông Dữ Liệu Toàn cục (global data)

Truyền thông dữ liệu toàn cục (truyền thông GD khoảng cách gần) là ứng dụng truyền thông được tích hợp trong hệ điều hành của CPU. Nó truyền thông qua mạng MPI với lượng dữ liệu nhỏ mà thời gian không phải là yếu tố quyết định. Dữ liệu toàn cục có thể được truyền gồm các vùng đệm input và output, bit nhớ, dữ liệu trong các khối dữ liệu, giá trị thời gian và đếm là dữ liệu được gửi đi. Truyền thông GD theo chu kỳ không cần chương trình người dùng. Các chức năng hệ thống dùng cho truyền thông GD điều khiển theo sự kiện (event-driven) bằng S7-400. Các CPU phải được nối mạng với nhau qua giao tiếp MPI và chúng phải được nối với bus truyền thông như trong giá đỡ module S7-400. Tất cả các CPU phải cùng thuộc dự án S7 cùng loại.

Chu kỳ GD

Việc gửi và nhận xảy ra bất đồng bộ giữa sender và receiver tại điểm kiểm tra chu kỳ quét. Điều này có nghĩa là sau khi xử lý chương trình theo chu kỳ nhưng trước chu kỳ mới dữ liệu vùng đệm được cập nhật. Dữ liệu được trao đổi theo dạng gói dữ liệu (data packet). Các CPU trao đổi gói dữ liệu toàn cục theo dạng chu kỳ GD. Có tới 15 CPU có thể trao đổi dữ liệu với nhau trong một chu kỳ GD, một CPU cũng có thể chiếm hơn một chu kỳ GD.

Tỉ lệ quét

Tỉ lệ quét điều khiển việc gửi và nhận thông tin toàn cục bởi vì truyền thông GD đặt tải trong thời gian xử lý của hệ điều hành CPU và cũng cần thời gian truyền trên MPI. Bạn xác định số chu kỳ sau khi CPU phải gửi hoặc nhận dữ liệu. Với các CPU S7-400, tỉ lệ quét 0 không thể thực hiện truyền thông dữ liệu theo chu kỳ. Hãy chọn tỉ lệ quét này nếu bạn chỉ muốn gửi và nhận dữ liệu theo sự kiện điều khiển bằng các SFC.

Hình 6.9 ví dụ chu kỳ GD.

Hình 6.10 Nguyên tắc truyền thông dữ liệu toàn cục theo sự kiện điều khiển và chu kỳ

Bảng dữ liệu toàn cục

Sau khi đã cài đặt một dự án: tồn tại một mạng con MPI; sau khi cấu hình các trạm S7. Các trạm phải chứa ít nhất một CPU kết nối với MPI.

Bạn hãy cấu hình truyền thông GD bằng cách nhập các giá trị cho bảng. Khi biểu tượng dùng cho mạng con MPI được chọn trong Manager SIMATIC hoặc trong công cụ cấu hình mạng NetPro, bạn có một bảng dữ liệu toàn cục rỗng bằng cách chọn OPTIONS→DEFINE GLOBAL DATA. Một bảng dữ liệu toàn cục có thể chứa đến 15 cột CPU.

Trong mỗi hàng dưới mỗi CPU, hãy nhập các địa chỉ có giá trị mà bạn muốn truyền. Một hàng có thể chứa vài receiver nhưng chỉ có một sender. Một gói dữ liệu không chỉ chứa các địa chỉ riêng lẻ mà còn chứa vùng địa chỉ. Chẳng hạn như, MB 0:15 là vùng các bytes nhớ từ MB từ 0 đến MB15, DB 20.DBW 14:8 cho biết vùng dữ liệu trong khối dữ liệu DB 20, chứa 8 từ dữ liệu bắt đầu từ dữ liệu DBW 14.

Sau khi nhập số liệu bảng, bạn hãy chọn menu GD TABLE→COMPILE..... Sau khi biên dịch lần đầu bạn xác định tỉ lệ quét và địa chỉ dùng cho tình trạng truyền nếu cần. Sau khi thực hiện xong hãy biên dịch bảng GD một lần nữa.

6.9 Truyền Thông SFC

Truyền thông SFC có nhiệm vụ truyền dữ liệu giữa các module lập trình SIMATIC. Truyền thông SFC cần có các hàm truyền thông, gọi là các hàm SFC. Chúng được lưu trữ trong hệ điều hành của CPU. Khi cần người sử dụng chỉ việc gọi các hàm này.

Khi được gọi các hàm truyền thông SFC sẽ tự động thiết lập kết nối truyền thông. Chúng không được cấu hình trong bảng kết nối vì đây là dạng truyền thông sử dụng kết nối không cấu hình. Sau khi đã thiết lập truyền dữ liệu thì bạn có thể xác định trong cài đặt thông số việc kết nối có bị ngắt hay không để sử dụng các tài nguyên khác, hoặc việc kết nối vẫn được duy trì cho các truyền thông phụ. Cũng có thể chỉ kết nối tới một đối tác truyền thông tại một thời điểm.

Chẳng hạn nếu tạm thời không đủ các tài nguyên kết nối để thiết lập kết nối thì có thể bỏ xung sau. Trong truyền thông SFC thì đối tác truyền thông không cố định. Sau khi việc kết nối đã hoàn tất thì bạn có thể chọn một đối tác truyền thông khác bằng cách thay đổi thông số. Có nghĩa là số

lượng đối tác truyền thông không bị giới hạn bởi các tài nguyên kết nối hiện có.

Truyền thông SFC truyền một biến hay một vùng dữ liệu cho mỗi một nhiệm vụ. Vùng dữ liệu có dung lượng tối đa là 76 byte. Có 2 dạng truyền thông SFC là: truyền thông trạm nội bộ – truyền thông trạm ngoài.

Truyền thông SFC trạm ngoài

Truyền thông SFC trạm ngoài truyền dữ liệu giữa các trạm SIMATIC. Dữ liệu trao đổi qua kết nối MPI, có thể sử dụng MPI hiện có. Chẳng hạn truyền thông SFC trạm ngoài có thể truyền dữ liệu điều khiển song song với truyền thông dữ liệu chung (global).

Truyền thông SFC trạm ngoài gồm có các hàm hệ thống sau:

➤ **SFC 65X_SEND** Gửi dữ liệu

Kết nối 2 chiều: SFC 66 X_SEND phải được gọi từ phía đối tác truyền thông.

➤ **SFC 66X_RCV** Nhận dữ liệu

Kết nối 2 chiều: SFC 65 X_SEND phải được gọi từ phía đối tác truyền thông.

➤ **SFC 67X_GET** Đọc dữ liệu

Kết nối 1 chiều: đọc một biến từ đối tác truyền thông mà không có chương trình người dùng trong đối tác truyền thông.

➤ **SFC 67X_GET** Ghi dữ liệu

Kết nối 1 chiều: ghi một biến tới đối tác truyền thông mà không có chương trình người dùng trong đối tác truyền thông.

➤ **SFC 69 X_ABORT** Kết nối đầu cuối

Tự động thiết lập kết nối đầu cuối.

Hình 6.11 Nguyên tắc truyền thông SFC

Truyền thông SFC nội bộ

Truyền thông SFC nội bộ truyền dữ liệu giữa các modules lập trình trong một trạm SIMATIC. Dữ liệu trao đổi qua PROFIBUS. Có thể sử dụng PROFIBUS Subnet đã có. Chẳng hạn truyền thông SFC nội bộ có thể truyền dữ liệu điều khiển song song với truyền thông dữ liệu theo chu kỳ giữa CPU chủ và CPU tổ thông qua PROFIBUS-DP.

Truyền thông SFC nội bộ gồm có các hàm hệ thống sau:

- SFC 72 L_GET Đọc dữ liệu

Kết nối 1chiều: đọc một biến từ đối tác truyền thông mà không có chương trình người dùng trong đối tác truyền thông.

- SFC 73 L_PUT Ghi dữ liệu

Kết nối 1chiều: ghi một biến tới đối tác truyền thông mà không có chương trình người dùng trong đối tác truyền thông.

- SFC 74 I_ABORT Kết nối đầu cuối

Tự động thiết lập kết nối đầu cuối.

6.10 Truyền Thông SFB

Truyền thông SFC truyền phần lớn dữ liệu giữa các trạm SIMATIC S7. các trạm được kết nối với nhau bằng một mạng phụ (subnet), mạng phụ này có thể là MPI, PROFIBUS hoặc Ethernet-subnet. Các kết nối truyền thông là cố định, chúng được cấu hình trong bảng kết nối bởi vì đây là truyền thông mở rộng có kết nối cấu hình.

Các hàm truyền thông SFB là các khối hàm hệ thống được tích hợp sẵn trong hệ điều hành của các S7-400 CPU. Cũng giống như khối hàm FB, mỗi một SFB cần có một khối dữ liệu (instance data block) để lưu trữ dữ liệu nội bộ của nó. Khối dữ liệu dùng cho một SFB nằm trong vùng nhớ chương trình người dùng (user memory).

Để sử dụng truyền thông SFB, hãy copy phần mô tả giao tiếp của SFB từ *Standard Library* trong *System Function Block* vào thư mục *Blocks*, tạo một khối dữ liệu cho mỗi lần gọi và sau đó gọi SFB. Cũng có thể chọn SFB từ “program elements” catalog và khối dữ liệu cho SFB sẽ được tạo ra tự động. Bạn cũng có thể gọi SFB từ một khối hàm, sau đó SFB lưu trữ dữ liệu của nó vào trong khối dữ liệu của khối hàm này.

Hình 6.12 Nguyên tắc truyền thông SFB.

Đặt cấu hình truyền thông SFB

Truyền thông qua các khối hàm hệ thống cần có bảng kết nối để xác định các kết nối truyền thông.

Mỗi đối tác truyền thông khi kết nối truyền thông được xác định bằng một ID kết nối. Khi soạn thảo kết nối thì STEP 7 sẽ gán ID kết nối. Sử dụng “local ID” trong đặt (set) thông số cho các SFB trên module có kết nối thấy được. Sử dụng “partner ID” trong đặt thông số cho các SFB của đối tác truyền thông.

Có thể sử dụng cùng một kiểu kết nối cho các nhiệm vụ phát/thu (sending/receiving) khác nhau. Tuy nhiên sau đó cần nhập thêm một ID nhiệm vụ (job ID) cho ID kết nối để phân biệt giữa các nhiệm vụ và thiết lập quan hệ giữa các khối phát/thu (sending/receiving blocks).

Trao đổi dữ liệu một chiều và hai chiều

Trong truyền thông dữ liệu một chiều (*one-way data communication*) thì truyền thông SFB chỉ được gọi bằng CPU cục bộ (local). Hệ điều hành sẽ

đáp ứng các hàm truyền thông cần thiết cho CPU đối tác. Truyền thông dữ liệu một chiều cũng bao gồm các chức năng điều khiển và giám sát.

Trong *truyền thông dữ liệu hai chiều (two-way data communication)* thì tại các đầu cuối của mỗi kết nối cần có một khối SEND và một khối RECEIVE. Cả hai khối cần có các ID kết nối nằm chung một hàng trong bảng kết nối. Có thể sử dụng nhiều “cặp khối” cho một kết nối có các ID nhiệm vụ.

Bảng 6.2 Khối hàm hệ thống trong truyền thông SFB

Truyền thông dữ liệu 2 chiều		Truyền thông dữ liệu 1 chiều	
SFB 8 USEND	Gửi không kết hợp toàn bộ dữ liệu	SFB 14 GET	Đọc dữ liệu vào một CPU cụ thể
SFB 9 URCV	Nhận không kết hợp toàn bộ dữ liệu	SFB 15 PUT	Ghi dữ liệu vào một CPU cụ thể
SFB 12 BSEND	Gửi một khối dữ liệu tới 76 Kbyte	Truyền dữ liệu máy in	
SFB 13 BRCV	Nhận một khối dữ liệu tới 76 Kbyte	SFB 16 PRINT	Truyền dữ liệu tới một máy in
Chức năng điều khiển		Chức năng giám sát	
SFB 19 START	Thực hiện khởi động nóng trên đối tác truyền thông	SFB 22 STATUS	Hỏi tình trạng đối tác
SFB 12 STOP	Chuyển đối tác truyền thông sang trạng thái STOP	SFB 23 USTATUS	STATUS ghi nhận tình trạng đối tác
SFB 21 RESUME	Thực hiện khởi động nóng trên đối tác truyền thông	SFC 62 CONTROL	Hỏi tình trạng của một SFB

7 Bảng điều khiển và giám sát xử lý

Tổng quát về thiết bị và công cụ cấu hình

Để điều khiển máy móc và nhà máy có nghĩa giám sát quá trình sản xuất và liên quan đến xử lý. Họ SIMATIC HMI (Human Machine Interface) cung cấp các thiết bị và công cụ cần thiết cho các công việc này. SIMATIC HMI bao gồm các thiết bị – bảng nút nhấn (PP, Push Button Panels) – hiển thị chữ (TD, Textdisplays) – bảng vận hành (OP, Operator Panels) – và bảng điều khiển sờ (TP, Touch Panels) – dùng cho tất cả các công việc điều khiển và giám sát máy móc. Phần bảng điều khiển của họ SIMATIC HMI cũng có thể hiểu là giao tiếp người – máy.

Các thiết bị họ SIMATIC HMI hoàn toàn phù hợp với các bộ điều khiển lập trình SIMATIC. Bạn sử dụng phần mềm ProTool để thiết lập cấu hình cho bảng điều khiển và thiết bị giám sát xử lý, cho phép chúng giao tiếp với CPU qua giao tiếp MPI hoặc mạng PROFIBUS. Hệ điều hành của CPU giải quyết phần lớn các công việc truyền thông. Với các thiết bị họ SIMATIC HMI tiên tiến nhất thì WinCC là một công cụ hoàn hảo cho việc thiết lập các hệ thống có hiển thị cho các PC chuẩn. Có thể coi đây là hệ thống SCADA rất mạnh (Supervisory Control And Data Acquisition) thỏa mãn các yêu cầu cao nhất trong điều khiển vận hành máy móc và các quá trình xử lý trong công nghiệp.

Chương này mô tả một số thiết bị thuộc họ SIMATIC HMI

7.1 Đặt Cấu hình SIMATIC HMI

ProTool và WinCC là các công cụ đặt cấu hình cho các thiết bị họ SIMATIC HMI. ProAgent là một phần mềm tùy chọn cung cấp các hiển thị chẩn đoán chuẩn.

ProTool là công cụ cấu hình cho tất cả các thiết bị HMI – từ các bảng hiển thị text, hiển thị đồ họa cho tới các hệ thống chạy trên Windows là MP270 và OP37/Pro. ProTool hiện có 3 version, mỗi version cung cấp các phạm vi chức năng khác nhau: ProTool/Lite, ProTool và ProTool/Pro. Bạn cài đặt ProTool như một “thành phần tích hợp” của STEP 7 và sử dụng nó để tạo ra một dự án ProTool (thư mục). Thư mục ProTool chứa tất cả dữ liệu

HMI và cho biết cách giao tiếp giữa thiết bị HMI với các trạm SIMATIC. HMI station sẵn sàng hoạt động khi nó được nạp với dữ liệu cấu hình.

Có thể sử dụng ProAgent trong kết nối với Engineering Tools để tạo ra các hiển thị chẩn đoán dùng cho chẩn đoán lỗi trong máy móc và nhà máy.

WinCC là phần mềm hoàn hảo cho phép bạn thiết lập các giao tiếp vận hành phức tạp có hiển thị, điều khiển hoạt động và thu nhận dữ liệu của các hệ thống HMI chạy trên Windows hoặc các PC chuẩn.

Hình 7.1 Tổng quan về thiết bị họ SIMATIC HMI và phần mềm thiết lập cấu hình



Hình 7.2 Tích hợp SIMATIC Manager và ProTool trong một dự án SIMATIC.

7.2 Bảng điều khiển nút nhấn PP7 và PP17

Bảng điều khiển nút nhấn PP7 và PP17 là loại bảng điều khiển được ráp sẵn dùng cho vận hành các máy đơn giản. Các phím gõ có thể đánh nhãn hoặc có sẵn – số lượng phím phụ thuộc vào dạng bảng điều khiển, các LED có bề mặt hai màu. Kết nối bảng điều khiển tới bộ điều khiển nhờ một giao tiếp nối tiếp, hoặc dùng MPI nối trực tiếp tới giao tiếp của bộ điều khiển lập trình, hoặc DP trong kết nối PROFIBUS-DP tới DP master bất kỳ.

Việc lắp đặt bảng điều khiển rất đơn giản chỉ cần cắt một hình chữ nhật. Phía trước bảng điều khiển có cấp bảo vệ IP 65 và phía sau là IP 20. Các ngõ vào ra digital hiện có cho phép thêm các phần tử chuẩn 22,5 mm, chẳng hạn như các nút nhấn, đèn báo có thể được chèn vào các lỗ đục sẵn của các version chuẩn. Hãng đã đặt trước địa chỉ MPI là 10 và địa chỉ nút nhấn đèn báo bắt đầu từ byte nhớ MB 100. Có thể sử dụng bảng điều khiển ngay lập tức mà không cần xác định thêm thông số nào. bạn có thể thay đổi các thông số giao tiếp bằng hiển thị phía sau thiết bị mà không cần công cụ

đặc biệt. Sử dụng công cụ ProTool để cấu hình cho bảng điều khiển nút nhấn.

Mặc dù đã có sẵn nhiều chức năng khác nhau như thử đèn và nút nhấn, tần số chớp 0,5 Hz và 2 Hz, bạn có thể thay đổi bảng điều khiển nút nhấn PP7 và PP17 theo yêu cầu của mình với chú ý tới số lượng, trình bày hiển thị, phần tử hoạt động và đặt nhấn.

Nếu muốn sử dụng bảng điều khiển nút nhấn như một DP slave trong mạng PROFIBUS, bạn cần xác định các thông số cần thiết khi cấu hình phần cứng STEP 7, thực hiện tương tự như đối với một trạm ET200. Tất cả các thông số được lưu trữ trong một module nhớ phụ (submodule) và có thể thay thế dễ dàng. Tất cả các trạng thái hoạt động được chỉ báo phía sau thiết bị và được dùng cho mục đích chẩn đoán.

Bảng điều khiển nút nhấn PP7: có 8 phím, 8 LED, 4 ngõ vào digital phụ, 3 lỗ cắt sẵn cho các phần tử chuẩn 22,5 mm như đèn, nút nhấn, ...

Bảng điều khiển nút nhấn PP17-I: có 16 phím, 16 LED, 16 ngõ vào digital, 16 ngõ ra digital phụ, 12 lỗ cắt sẵn cho các phần tử chuẩn 22,5 mm như đèn, nút nhấn, ...

Bảng điều khiển nút nhấn PP7-II: có 32 phím, 32 LED, 16 ngõ vào digital phụ, 16 ngõ ra digital.

PP7

PP17-II

Hình 7.3 Bảng điều khiển nút nhấn PP7 và PP17

7.3 Text Display TD17

Text Display TD17 là bộ hiển thị chữ theo hàng. Phía trước đáp ứng cấp bảo vệ IP 65 (lắp đặt xong) cho phép lắp đặt TD17 trong vùng lân cận máy.

Kết nối bộ hiển thị tới bộ điều khiển lập trình bằng giao tiếp MPI hoặc mạng PROFIBUS. TD17 có một đồng hồ tích hợp bên trong lưu trữ hoạt động trong nhiều ngày, cho phép vận hành TD17 mà không cần pin backup.

Bạn có thể cấu hình LED backlit hiển thị LC dạng 4 hàng với 20 ký tự mỗi hàng hoặc 8 hàng 40 ký tự mỗi hàng (cỡ chữ: 6 mm). TD17 có một bàn phím dạng màng có khả năng chống dầu, mỡ, và các chất tẩy thông thường. Sử dụng bàn phím để gọi các chức năng chuẩn hoặc hiển thị lưu trữ chuẩn của hãng sản xuất.

Sử dụng thiết bị lập trình và công cụ ProTool để thiết lập cấu hình TD17 ở offline. Sau khi đặt cấu hình, nạp dữ liệu cấu hình vào TD17 và kết nối tới bộ điều khiển lập trình. Bây giờ TD17 đã sẵn sàng làm việc.

Thông báo xử lý được hiển thị dưới dạng chữ cố định. Bạn có thể tạo những lời giải thích thêm cho Info text, lưu trữ vào TD17 và gán cho một thông báo dạng chữ (text message). Các thông báo dạng chữ có thể hiển thị dưới dạng các ngôn ngữ khác nhau như: Đức, Anh, Pháp, Ý, Tây Ban Nha và Nga (Cyrillic). Có thể nạp tới 3 ngôn ngữ trên vào TD17 để sử dụng chúng ở online.

Bạn có thể định nghĩa tới 999 thông báo xử lý (process message) với 80 ký tự mỗi thông báo. Vùng đệm thông báo xử lý có thể lưu trữ tới 256 thông báo. Các thông báo được hiển thị kèm theo ngày tháng (date and time).

Vận hành TD17 bằng cách gọi ra các hiển thị chuẩn đã chọn từ bảng nội dung: Display process message buffer, Delete process message buffer, Display system message buffer, Set language/contrast, Set date/time, và thay đổi chế độ hoạt động (online, offline và nạp dữ liệu cấu hình). Chức năng password cho phép bạn điều khiển quyền truy cập và chống vận hành trái phép.

Vùng dữ liệu trong vùng nhớ người dùng thực hiện việc trao đổi dữ liệu giữa TD17 và PLC. TD17 nhận lệnh điều khiển từ chương trình người dùng, như các ngõ ra, hoặc xoá vùng đệm thông báo (process message buffer), thay đổi ngôn ngữ, cài đặt ngày tháng. Mỗi một nút nhấn ở bàn phím được gán cho một bit riêng biệt trong vùng dữ liệu của

Hình 7.4 *Hiển thị text TD17*

CPU. Khi nhấn một nút nhấn ở bàn điều khiển sẽ set hoặc reset bit tương ứng trong CPU và được xử lý trong chương trình người dùng.

7.4 Hiện Thị Text OP3, OP7 và OP17

Phía trước hiển thị OP3, OP7 và OP17 có cấp bảo vệ IP 65 cho phép các thiết bị này có thể đặt trong vùng lân cận máy móc. Chúng được trang bị một hiển thị LED-backlit LC và một bàn phím dạng màng có khả năng chống dầu, mỡ, các chất tẩy thông thường. Bảng điều khiển vận hành cho phép giám sát và điều khiển sự cố, trạng thái hoạt động và các giá trị xử lý hiện hành.

Các giá trị muốn hiển thị thường theo một chức năng công nghệ riêng biệt được ráp sẵn thành một “hiển thị”. Một số hiển thị chuẩn đã được lưu trữ trong bảng điều khiển vận hành cho phép đưa OP vào sử dụng ngay mà không cần phải chuẩn bị nhiều. Bạn có thể tạo các hiển thị để chúng phù hợp với việc điều khiển máy móc hoặc một nhà máy cụ thể. Các hiển thị gồm một text cố định và các biến nhập vào được cập nhật liên tục. Người vận hành có thể nhập các giá trị mới chẳng hạn khi nhập một setpoint để điều khiển máy. Phụ thuộc vào dạng hiển thị, bạn có thể sử dụng các phím để gọi các màn hình hiển thị khác hoặc điều khiển một biến nhị phân trong chương trình người dùng.

Sử dụng phần mềm cấu hình ProTool để cài đặt (set up) và cấu hình bảng điều khiển. Các OP này có thể nối tới các bộ điều khiển S7-200 (bằng giao tiếp PPI) hoặc S7-300/400 (bằng giao tiếp MPI). Mật khẩu và các cấp bảo vệ cho phép ngăn cản việc vận hành trái phép. Họ OP7 và OP17 có thể kết nối tới mạng PROFBUS. Nếu bạn muốn in ra các bảng liệt kê, thông báo lỗi, vùng xử lý thông báo, nội dung màn hình hiển thị bạn có thể kết nối OP7 và OP17 với một máy in.

Bảng điều khiển hiển thị OP3 có 18 phím hệ thống, 5 trong số đó là phím mềm (softkey), hiển thị 2 hàng x 20 ký tự và một vùng nhớ flash EPROM 128 KB để lưu trữ dữ liệu cấu hình.

Bảng điều khiển hiển thị OP7 có 22 phím hệ thống, 8 phím chức năng cho phép người dùng dán nhãn, 8 trong số các phím trên là phím mềm (softkey), hiển thị 4 hàng x 20 ký tự và một vùng nhớ flash EPROM 128 KB để lưu trữ dữ liệu cấu hình.

Bảng điều khiển hiển thị OP17 có 22 phím hệ thống, 24 phím chức năng cho phép người dùng dán nhãn, 22 trong số các phím trên là phím mềm

(softkey), hiển thị có thể cài đặt cho máy in 4 hàng x 20 ký tự hoặc 8 hàng 40 ký tự và một vùng nhớ flash EPROM 256 KB để lưu trữ dữ liệu cấu hình.

Hình 7.5 Bảng điều khiển hiển thị OP3, OP7 và OP17.

7.5 Bảng điều khiển Đồ Họa OP27 và OP37

Phía trước bảng điều khiển đồ họa OP27 và OP37 có cấp bảo vệ IP 65 cho phép các thiết bị này có thể đặt trong vùng lân cận máy móc. Chúng được trang bị một hiển thị CCFL-backlit LC và một bàn phím dạng màng có khả năng chống dầu, mỡ, các chất tẩy thông thường. Bảng điều khiển vận hành cho phép bạn giám sát và điều khiển sự cố, trạng thái hoạt động và các giá trị xử lý hiện hành.

Hiển thị đồ họa chỉ ra các thông báo lỗi, xử lý và các biến xử lý (địa chỉ). Chúng tương ứng cho các biến như các giá trị ngõ ra, đường biểu đồ, đường cong hoặc chỉ ra trạng thái dưới dạng ký hiệu có thay đổi màu sắc. Người vận hành có thể nhập các giá trị trong vùng ngõ vào đặc biệt (special input) hoặc dạng các giá trị ký hiệu. Các phần tử hiển thị cố định của hiển thị là đồ họa ký tự hoặc đồ họa điểm. Bạn tạo đồ họa điểm từ Windows graphic editor bất kỳ và chèn đồ họa vào sơ đồ. Người vận hành sử dụng các phím mềm và các phím chức năng của bảng điều khiển để điều khiển quá trình xử lý. Bạn có thể tạo mật khẩu (password) và các cấp bảo vệ để ngăn cản việc vận hành trái phép.

Sử dụng phần mềm cấu hình ProTool để cài đặt (set up) và cấu hình bảng điều khiển vận hành OP27 và OP37. Các OP này có thể nối tới các bộ điều khiển S7-200 (bằng giao tiếp PPI) hoặc S7-300/400 (bằng giao tiếp MPI), nối tới mạng PROFIBUS và có thể kết nối với một máy in.

Bảng điều khiển vận hành OP27 có bộ nhớ flash EPROM 1 MB, có thể có một hiển thị STN monochrome LC (8 cấp độ xám), hoặc hiển thị màu STN color LC (8 màu). Cả hai dạng hiển thị đều có 320 x 240 chấm điểm,

24 phím hệ thống (4 phím có LED) có chức năng cố định, 24 phím chức năng (18 phím có đèn) trong đó có 14 phím là phím mềm có thể thay đổi chức năng.

Bảng điều khiển vận hành OP37 có bộ nhớ flash EPROM 2 MB, có thể có một hiển thị màu STN color LC hoặc một hiển thị TFT color LC (8 màu). Cả hai dạng hiển thị đều có 640 x 480 chấm điểm, 32 phím hệ thống (4 phím có LED) có chức năng cố định, 36 phím chức năng (28 phím có đèn) trong đó có 20 phím là phím mềm có thể thay đổi chức năng.

Hình 7.6 Bảng điều khiển vận hành hiển thị đồ họa OP27 và OP37.

7.6 Bảng điều khiển kiểu sờ TP27 và TP37

Các bảng điều khiển sờ (touch panels) là một phương pháp mới trong điều khiển máy móc hoặc nhà máy: người vận hành điều khiển bằng cách nhấn trực tiếp vào màn hình hiển thị. Bàn phím được thay thế bằng một màn hình, để thực hiện một thao tác điều khiển người vận hành chỉ cần sờ vào nút nhấn tương ứng trên màn hình.

Bảng điều khiển kiểu sờ TP27 và TP37 cho phép thể hiện trung thực đồ họa của máy hay nhà máy được điều khiển. Phía trước của bảng điều khiển có cấp bảo vệ IP 65 (lắp đặt xong) cho phép chúng có thể được sử dụng trong vùng lân cận máy. Màn hình là một hiển thị CCFL-backlit LC và có khả năng chống dầu, mỡ, và các chất tẩy thông thường.

Sử dụng công cụ cấu hình ProTool để cài đặt và cấu hình cho bảng điều khiển. Kết nối tới trạm SIMATIC nhờ giao tiếp MPI hoặc mạng PROFIBUS-DP. Khi kết nối với máy in các bảng điều khiển có thể theo kết nối nối tiếp hoặc song song (chỉ có ở TP 37).

Các bảng điều khiển kiểu sờ có cùng phạm vi chức năng như các bảng điều khiển vận hành cùng họ. TP 27 và TP 37 giám sát giá trị giới hạn, hiển thị các giá trị xử lý, các vùng động (đồ họa ký tự và đồ họa điểm), các sơ đồ có yếu tố tĩnh, đường cong, đồ họa có các đường ngang và các dạng đồ họa khác. Chúng cũng tham gia quản lý các thông báo xử lý và sự cố, cài đặt mật khẩu, ở chế độ online cho phép thay đổi tới 3 ngôn ngữ. Đặc điểm riêng của bảng điều khiển kiểu sờ là các nút nhấn đã cài đặt trước cho text, text lists, graphics, graphics lists và các nút nhấn dùng cho các chức năng riêng biệt của người dùng.

Tính năng của **TP 27 touch panel** tương tự như chức năng của OP27, chỉ khác ở màn hình sờ. Bảng điều khiển TP 26 hiện có một hiển thị 5.7“ STN color/monochrome (8 màu hoặc 8 cấp độ xám) với độ phân giải 320 x 240 điểm ảnh (pixel). Bảng điều khiển TP 27-10 hiện có một hiển thị 10.3“ STN color (8 màu) với độ phân giải 640 x 480 điểm ảnh.

Tính năng của **TP 37 touch panel** tương tự như chức năng của OP37, chỉ khác ở màn hình sờ. Bảng điều khiển TP 26 hiện có một hiển thị 10.4“ TFT color (8 màu) với độ phân giải 640 x 480 điểm ảnh.

Hình 7.7 *Bảng điều khiển kiểu sờ TP 27 và TP 37.*

7.7 Hệ thống MP270 Và OP37/Pro chạy trên Windows

Các hệ thống MP270 và OP37/Pro chạy trên windows, chúng không những đáp ứng được các yêu cầu của hệ thống giao tiếp vận hành là phải làm việc được trong môi trường khắc nghiệt mà còn có tính năng như một PC. Sử dụng công cụ cấu hình ProTool/Pro CS, phần mềm ProTool/Pro RT để cài đặt và cấu hình cho MP270 và OP37/Pro. Cả hai hệ thống này đem lại cho bạn nhiều khả năng mở rộng với các chức năng người dùng xác định.

Chẳng hạn đối với ProTool/Pro cho phép bạn chạy chương trình được viết bằng Microsoft Visual Basic Script.

Multi-Panel MP270 sử dụng Win CE

Multi-panel MP270 sử dụng hệ điều hành Microsoft Windows CE được trang bị một flash EPROM 8MB dùng để lưu trữ dữ liệu chương trình và vùng nhớ chính có dung lượng 16MB. Một slot card PC dùng cho mục đích mở rộng.

Phía trước bảng điều khiển có cấp bảo vệ IP 67 với độ sâu lắp đặt chỉ có 50 mm, trang bị một hiển thị màu color STN hoặc color TFT với độ phân giải 640 x 480 điểm ảnh, 256 màu và CCFL backlighting. Bàn phím dạng màng có 33 phím hệ thống (4 phím có đèn), 16 phím chức năng có LED do người dùng dán nhãn, 20 phím mềm do người dùng dán nhãn (12 phím có đèn).

Ngoài các tính năng cơ bản của một bảng điều khiển dạng đồ họa, MP270 còn có các đặc điểm như định vị động các đối tượng đồ họa, lưu trữ thông báo và giá trị xử lý vào một PC card, mô phỏng cấu hình. Một cửa sổ có chức năng cuộn (scroll) và zoom cho phép thể hiện tới 8 đường cong. Các hiển thị thông báo, cửa sổ thông báo, hàng thông báo, vùng thông báo đã được cấu hình sẵn.

OP37/Pro sử dụng Windows 95

OP37/Pro là bảng điều khiển vận hành OP37 có một vi xử lý Pentium 166 MHz và chạy trên hệ điều hành Windows 95. Có vùng nhớ chính 32 MB, một ổ đĩa mềm và đĩa cứng 2.1GB. Sử dụng công cụ PtoTool/Pro CS, phần mềm ProTool/Pro RT để cấu hình OP37/Pro panel.

Hình 7.8 Hệ thống MP270 và OP37/Pro chạy trên Windows.

7.8 Thực hiện tối ưu các thao tác với Operator Interfaces

Module nút nhấn điều khiển trực tiếp 24V

Module nút nhấn điều khiển trực tiếp được gắn vào phía sau của bảng điều khiển vận hành (operator panel) hoặc bảng điều khiển kiểu sờ, có 8 hoặc 16 ngõ ra digital 24 VDC được điều khiển bởi bàn phím (operator panel) hoặc màn hình sờ (touch panel) hoặc bằng phần mềm.

Nút nhấn điều khiển trực tiếp DP

Các nút nhấn điều khiển điều khiển trực tiếp DP cho phép bạn điều khiển các địa chỉ nhị phân trong trạm SIMATIC qua mạng PROFIBUS-DP. Mặt khác, chúng cho phép bạn điều khiển các LED của bảng điều khiển từ trạm SIMATIC qua PROFIBUS-DP. Modul ngoại vi I/O có cùng địa chỉ trạm PROFIBUS với bảng điều khiển vận hành. Trên bảng điều khiển vận hành bạn có thể sử dụng tất cả các phím mềm và phím chức năng như các nút nhấn trực tiếp DP. Trên bảng điều khiển dạng sờ bạn có thể sử dụng 24 hoặc 48 nút nhấn sờ như các nút nhấn trực tiếp DP.

Giao tiếp bảng điều khiển (CPI)

Module CPI điều khiển hiển thị và các phần tử hoạt động qua mạng PROFIBUS DP. Module được gắn vào phía sau bảng điều khiển vận hành hoặc bảng điều khiển kiểu sờ. Phụ thuộc vào kiểu module mà có từ 16 hoặc 32 ngõ vào digital 24VDC và 16 hoặc 32 ngõ ra digital 24VDC được nối dây trực tiếp tới bảng điều khiển của bàn điều khiển hoặc máy. Bạn kết nối các hiển thị và các phần tử hoạt động như ngoại vi I/O tới bảng điều khiển vận hành hay bảng điều khiển sờ qua PROFIBUS-DP. Bạn cũng có thể sử dụng các nút nhấn điều khiển trực tiếp DP (không phải module nút nhấn điều khiển trực tiếp 24VDC) song song với module CPI.

Bảng 7.1 giao tiếp vận hành trong liên kết với thiết bị HMI

	Module nút nhấn điều khiển trực tiếp 24 V	Nút nhấn điều khiển trực tiếp DP	giao tiếp bảng điều khiển
Được sử dụng với	OP 27/37, TP 27/37	OP 7/DP, OP 17/DP, Op 27/37, TP 27/37	OP 27/37, TP 27/37
Số lượng các ngõ digital input/output			
OP 27/OP 37	8/16 Q		16 hoặc 32 I, 16 hoặc 32 Q

TP 26-6	8 Q		16 I, 16 Q
TP 27-10, TP 37	16 Q		16 hoặc 32 I, 16 hoặc 32 Q
Điều khiển bằng các phím mềm/phím chức năng hoặc nút nhấn trong bảng điều khiển kiểu sờ (sờ)			
OP 7 OP 17		F1...F4, K1...K4 F1...F8, K1...K16	
OP 27 OP 37	F1...F8 F1...F12	F1...F14, K1...K10 F1...F20, K1...K16	
TP 27-6/TP 27-10 TP 37	8/16 16	24/24 47	
Điều khiển bằng nút nhấn có LED			
OP 7 OP 17 OP 27 OP 37		K1...K4 K1...K16 F1...F8, K1...K10 F1...F12, K1...K16	

7.9 Kết Nối Với SIMATIC S7 Stations

Các giao tiếp dưới đây dùng để kết nối trạm HMI tới trạm SIMATIC. Chú ý rằng các giao tiếp cũng phụ thuộc vào version của thiết bị HMI đang sử dụng:

- S7-200 qua PPI
- S7-300/400 qua MPI
- S7-300/400 qua PROFIBUS như “MPI station”

Khi cấu hình cho hệ thống bằng ProTool bạn chọn trạm SIMATIC kết nối với thiết bị HMI, xác định dạng kết nối và địa chỉ trạm.

Kết nối giao tiếp PPI

Mặc dù kết nối giao tiếp PPI với S7-200 là kết nối point to point cơ bản nhưng nó vẫn có thể kết nối nhiều trạm S7-200 tới một trạm HMI. Trạm S7-200 luôn luôn coi kết nối này như một kết nối point to point khi chỉ có duy nhất một kết nối được kích hoạt tại một thời điểm thậm chí có nhiều trạm HMI hay thiết bị lập trình kết nối tới trạm S7-200.

Kết nối giao tiếp MPI

Có tới 32 nút (node) có thể liên kết trong một mạng MPI. Các “tài nguyên kết nối” hiện có sẽ quyết định số lượng trạm có thể kết nối tới CPU của một trạm S7. Chẳng hạn một CPU 314 có thể thiết lập 4 kết nối (tĩnh), một CPU 414 có thể thiết lập tới 32 kết nối. CPU S7-300 có hai kết nối cố định – một cho kết nối thiết bị lập trình và một cho trạm HMI- mà bạn không thể dùng cho mục đích khác. Cũng có thể dùng các kết nối tự do để kết nối các trạm HMI phụ.

Địa chỉ mặc định MPI của trạm HMI là “1“, thiết bị lập trình là “0“ và S7-CPU là “2“. Do đó bạn có thể sử dụng một trạm HMI, một thiết bị lập trình với một S7-CPU mà không cần phải thay đổi địa chỉ.

Kết nối PROFIBUS

Bạn dùng công cụ cấu hình ProTool để xác định địa chỉ trạm cho HMI. Xác định trạm kết nối với HMI, hoặc đối tác truyền thông của nó để trao đổi dữ liệu trực tiếp. Định nghĩa thiết bị MHMI như một trạm năng động. Nếu bạn sử dụng module nút nhấn điều khiển trực tiếp hoặc module giao tiếp bảng điều khiển bạn cũng phải cấu hình thiết bị HMI như một DP slave (SIMATIC Manager dùng HW-Config trong hardware catalog chọn PROFIBUS-DP → CONFIGURED STATION → SIMATIC OP).

S7 functions

Truyền thông giữa các trạm HMI và các trạm S7 do *S7 Function* đảm nhiệm, đây là một dịch vụ truyền thông riêng biệt của SIMATIC. Trạm HMI tượng trưng cho khách hàng (yêu cầu dữ liệu) và trạm S7 là dịch vụ (cung cấp dữ liệu). Không cần phải có các chức năng truyền thông riêng biệt trong chương trình người dùng bởi vì hệ điều hành đáp ứng toàn bộ việc truyền thông qua S7 functions.

7.10 Đặt Cấu Hình Thiết Bị HMI với SIMATIC Protocol

SIMATIC ProTool là một phần mềm định cấu hình cho các thiết bị SIMATIC HMI. ProTool hiện có là ứng dụng 32 bit chạy trên Microsoft Windows 95/98 và Windows NT. Hiện có 3 phiên bản với các phạm vi chức năng khác nhau:

- *ProTool/Lite* dùng cấu hình cho các thiết bị HMI hiển thị kiểu hàng (line-oriented) như TD 17, OP3 , OP17, C7-621, C7-633, C7-634.
- *ProTool* dùng cấu hình cho các thiết bị HMI hiển thị kiểu hàng và kiểu đồ họa như OP 27, Op 37, TP 27, TP 37, C7-626.

- *ProTool/Pro CS* dùng cấu hình cho các thiết bị HMI hiển thị kiểu hàng và kiểu đồ hoạ và chạy trên Windows như MP270, OP37/Pro và các PC chuẩn có phần mềm *ProTool/Pro RT*

Bạn sử dụng ProTool để tạo ra một dự án (project) ProTool cho mỗi một thiết bị HMI, chứa tất cả các dữ liệu cần thiết để chạy thiết bị như các sơ đồ, biến, thông báo, cách thực hiện. ProTool có thể được cài đặt như một phiên bản đơn lẻ, tuy nhiên chúng tôi khuyên bạn nên sử dụng nó như một phần của STEP 7 để ProTool có thể truy xuất trực tiếp tới các bảng ký hiệu và các thông số truyền thông của các chương trình người dùng liên quan. Bằng cách này bạn không phải nhập các thông tin hai lần, và tất nhiên có các thuận lợi khác: dữ liệu HMI được lưu trữ cùng với S7 project; bạn có thể dùng công cụ cấu hình phần cứng HW Config để xác định các ngõ vào và ra PROFIBUS-DP; bạn có thể truy cập vào chương trình cấu hình mạng NetPro để kết nối thiết bị HMI tới một mạng con và cho hiển thị trong cài đặt mạng (network setup).

Tạo một ProTool project

Khi đã cài đặt ProTool như một phần tích hợp của STEP 7. Để xác định một đối tượng (object) cho một thiết bị HMI trong SIMATIC Manager tạo một cái mới hay mở một cái đã có, chọn thiết bị HMI và sau đó chọn INSERT STATION → SIMATIC OP để chèn HMI object. Tạo một object cho mỗi hệ thống giao tiếp vận hành, cho điều khiển vận hành và giám sát các phần của trạm SIMATIC C7. Trước khi cấu hình HMI object bạn nên xác định mạng subnet (MPI, PROFIBUS-DP) và thiết lập kết nối cho trạm SIMATIC dùng HW Config hoặc NetPro.

Nhấp đúp vào object HMI “rỗng” để khởi động trợ giúp ProTool project, hộp thoại *Project Wizard-Device Selection* xuất hiện trên màn hình cho phép bạn gán các thiết bị thực vào object chẳng hạn như bảng điều khiển vận hành dạng đồ hoạ OP 37, nhấp vào nút “Parameter” sẽ mở ra hộp thoại kế tiếp cho phép bạn xác định kết nối mạng con và cài đặt bộ điều khiển, sau đó ProTool sẽ xét dữ liệu HMI liên quan tới trạm này như các ký hiệu hoặc các thông số truyền thông.

ProTool cung cấp các hiển thị xử lý chuẩn chứa các chức năng chung cho việc chọn thiết bị HMI và kết nối chương trình người dùng tới bộ điều khiển. Sau đó bạn có thể thay đổi và chỉnh sửa các hiển thị chuẩn này và viết chương trình theo yêu cầu của mình.

Dữ liệu cần trao đổi giữa thiết bị HMI và trạm SIMATIC được chứa trong các biến xử lý hoặc các vùng truyền thông. Thiết bị HMI đọc hoặc ghi các biến xử lý một cách trực tiếp mà không cần chương trình người dùng. Bạn phải xác định các vùng truyền thông trong vùng nhớ người dùng ở CPU.

Hình 7.9 Chọn thiết bị đích trong ProTool/Pro CS

Cấu hình một thiết bị HMI dùng ProTool

ProTool cho phép bạn tạo các hiển thị xử lý cho thiết bị HMI một cách đơn giản mà không cần bất cứ kỹ năng lập trình đặc biệt nào. Bạn tạo thư mục ProTool cho mình ở chế độ offline không cần kết nối tới thiết bị HMI. Thiết bị lập trình hiển thị các sơ đồ xử lý mà bạn đã tạo ra và chúng cũng xuất hiện trên thiết bị HMI. Sau khi cấu hình, hãy biên dịch (compile) thư mục ProTool và dữ liệu vào thiết bị HMI hay còn gọi là “thiết bị đích”(target device).

Cửa sổ Project của ProTool được chia làm 2 phần. Ở phần bên trái bạn sẽ thấy các dạng đối tượng (object) mà bạn đã cấu hình phụ thuộc vào thiết bị HMI muốn sử dụng. Ở phần bên phải của cửa sổ là các đối tượng riêng lẻ, nhấp đúp vào một đối tượng để khởi động công cụ phần mềm liên quan tới đối tượng đó.

Xác định vùng truyền thông

Vùng giao tiếp (interface area) là một vùng dữ liệu trong chương trình người dùng xác định giao tiếp giữa các trạm SIMATIC và thiết bị HMI. Nó chứa dữ liệu liên quan và nhiệm vụ dùng cho việc trao đổi dữ liệu. Cần bảo đảm xác định mỗi thiết bị HMI một vùng giao tiếp riêng biệt.

Dữ liệu được trao đổi trực tiếp qua các vùng dữ liệu chương trình. Các vùng dữ liệu chương trình dùng cho thông báo xử lý, thông báo lỗi, bàn phím, LED image, số sơ đồ...phụ thuộc vào thiết bị HMI mà bạn muốn sử dụng, ví dụ: khi một bit được set từ chương trình người dùng sẽ kích việc hiển thị một thông báo lỗi trong thiết bị HMI; nhấn một phím chức năng trên thiết bị HMI sẽ set một bit trong vùng đệm phím chức năng được đánh giá bằng chương trình người dùng.

Để xác định các vùng truyền thông, hãy mở ProTool project và chọn SYSTEM → AREA POINTERS, chọn vùng (ví dụ vùng dữ liệu), chọn “insert” để mở hộp thoại định nghĩa cho vùng dữ liệu.

Các sơ đồ xử lý cho người vận hành

Các sơ đồ xử lý là phần cốt lõi của ProTool project. Chúng mô tả trạng thái của máy móc hay nhà máy. Đồng thời cũng chứa các phần tử mà qua đó người vận hành điều khiển các bộ phận trong nhà máy. Ở dạng đơn giản nhất thì một sơ đồ xử lý chỉ gồm có 2 hoặc 4 hàng với text và các biến được chen, giống như ở các thiết bị hiển thị text TD và thiết bị hiển thị theo hàng (line oriented) HMI. Một sơ đồ xử lý có thể là một sơ đồ phức tạp chứa các phần tử đồ họa thay đổi theo tình trạng hoạt động của nhà máy.

Sau khi cho hiển thị Opening diagram các sơ đồ xử lý khác được gọi trên màn hình bằng các phương pháp khác nhau: bằng chương trình người dùng trong CPU, bằng các tín hiệu từ HMI: do người vận hành chọn bằng cách nhấn các phím chức năng. Nội dung và thứ tự gọi các sơ đồ xử lý tạo ra giao tiếp với người dùng (user interface).

Các sơ đồ xử lý chứa các phần tử hiển thị (display element) tĩnh và động cho biết quá trình xử lý, trạng thái hoạt động và sự cố. Các phần tử hiển thị thay đổi theo loại HMI sử dụng. Các phần tử hiển thị tĩnh là texts, điểm ảnh (pixel graphic) và đồ họa ký tự. Các phần tử hiển thị động là các vùng ngõ ra, các đường cong, biểu đồ và đèn báo. Chúng thay đổi kiểu hiển thị khi biến gán cho chúng thay đổi giá trị.

Các *phần tử hoạt động* (Operating elements) cho phép bạn can thiệp vào quá trình xử lý, chẳng hạn như : nhập các điểm đặt (setpoints), kích một hoạt động trong nhà máy. Các phần tử hiển thị thay đổi theo loại HMI sử dụng. Bạn có thể “xây dựng” một sơ đồ xử lý bằng các vùng ngõ vào, vùng ngõ vào/ra, các phím chức năng và các nút nhấn. Bạn có thể gán các thuộc tính động để tín hiệu hoá, chẳng hạn cho một hoạt động vận hành riêng biệt. Bạn cũng có thể chen, xoá các phần tử hoạt động khi một sự kiện điều khiển đang diễn ra. Một phần tử hoạt động như vậy được gọi là “event-driven”.

Dữ liệu nhập và xuất sử dụng biến

Một biến chứa đựng các thông tin có được từ quá trình xử lý hay nhà máy. Đó có thể là một trạng thái tín hiệu, một setpoint hay thời gian trong ngày. Các biến có một tên và dạng dữ liệu. Biến cục bộ (local variable) chỉ có ở thiết bị HMI. Một biến chung (global variable) có trong CPU và thiết bị HMI, nó chiếm vùng nhớ trong CPU và có thể được đặt địa chỉ từ chương trình người dùng và bằng các sơ đồ xử lý trong thiết bị HMI. Các biến global trao đổi dữ liệu giữa thiết bị HMI và trạm SIMATIC.

Các thông báo xử lý và trạng thái hoạt động

ProTool phân chia ra các loại thông báo sau:

- *Process messages* thông báo các trạng thái thông thường như chế độ hoạt động, tình trạng xử lý và trình tự xử lý.
- *Fault messages* cho biết lỗi xử lý như tới hạn hay trạng thái xử lý nguy hiểm. Thông báo lỗi phải làm cho người vận hành hiểu được và có thể kết hợp theo nhóm.
- *System messages* cho biết các trạng thái và lỗi của thiết bị HMI và bộ điều khiển SIMATIC, đây là một phần của thiết bị HMI do đó không cần phải cấu hình.

Các thông báo được hiển thị phụ thuộc vào tính ưu tiên của chúng (quan trọng) và có thể in ra bằng máy in. Các hệ thống HMI sử dụng Windows cho phép bạn lưu trữ các thông báo này.

7.11 Chẩn Đoán Xử Lý Dùng SIMATIC ProAgent

Phần mềm tùy chọn SIMATIC ProAgent cung cấp nhiều chương trình chuẩn dùng cho chẩn đoán lỗi trong máy móc hay nhà máy. Các chương trình chuẩn này có thể sử dụng được trên các bảng điều khiển vận hành kiểu đồ họa hay bảng điều khiển kiểu tiếp xúc (sờ). ProAgent tương thích với

STEP7 Engineering Tool được sử dụng cho các bộ PLC SIMATIC S7-300/400 và thiết bị tự động.

ProAgent cung cấp cho bạn các khái niệm chẩn đoán đã chuẩn hoá cho phép bạn áp dụng cho các phần mềm của STEP7. Các chẩn đoán xử lý trên thiết bị HMI làm giảm gánh nặng cho trạm SIMATIC. Thiết bị HMI thực hiện phân tích điều kiện, lưu trữ thông tin cấu trúc, chú thích, ... Tất cả các chức năng chẩn đoán HMI chiếm vùng nhớ và thời gian xử lý chương trình trong CPU của trạm SIMATIC.

Bạn cài đặt ProAgent cùng với ProTool. Trong quá trình dịch chương trình người dùng STEP7 lưu trữ tất cả dữ liệu cần thiết cho các chẩn đoán xử lý trong một cơ sở dữ liệu chung (common database). Khi cấu hình thiết bị HMI, ProTool sẽ truy xuất cơ sở dữ liệu này. Bạn tích hợp các sơ đồ chẩn đoán vào giao tiếp người dùng HMI và chọn đối tượng chẩn đoán (chẳng hạn “units”, các khối người dùng, trình tự các bước) và chọn cách báo hiệu.

Trong khi hoạt động, các công cụ này cung cấp cho ProAgent bằng dữ liệu xử lý riêng biệt và tĩnh. Nội dung của các sơ đồ chẩn đoán luôn luôn liên quan tới một thông báo hoặc thiết bị được chọn.

Bắt đầu chẩn đoán: Message Diagram hiển thị các lỗi xảy ra trong quá trình xử lý

Sơ đồ thông báo hiển thị tất cả các thông báo xử lý. Bạn có thể chọn một thông báo và từ đó rẽ nhánh sang các sơ đồ chẩn đoán khác.

Hình ảnh tổng quát: Unit Overview

Unit Overview hiển thị tất cả nhà máy hoặc thiết bị máy móc, các thiết bị phụ và các chế độ làm việc hiện hành của chúng. Do đó người vận hành có thể cách ly thiết bị có sự cố ngay lập tức và thực hiện các thao tác cần thiết được xác định bằng cấu hình. Người vận hành có thể xem chi tiết để phân tích lỗi hoặc chuyển sang sơ đồ chuyển động (motion diagram) để sửa lỗi.

Chỉ báo trình tự lỗi trong Step Diagram

Khi kết nối với S7-GRAPH bạn có thể chọn toàn bộ trình tự báo lỗi, khởi tạo hoặc vô hiệu hoá (deactive) trình tự này. Bạn có thể chọn một bước cụ thể trong một trình tự và chỉ kích hoạt (active) hoặc vô hiệu hoá bước này.

Điều khiển nhà máy và chỉnh sửa lỗi dùng Motion Diagram

Sơ đồ chuyển động (motion diagram) cho phép bạn khởi tạo các chuyển động cụ thể trong các thiết bị riêng lẻ, chẳng hạn bạn dùng các phím mềm hoặc các phím điều khiển trực tiếp của thiết bị HMI để điều khiển một thiết bị có sự cố tới vị trí an toàn nhất.

Phân tích điều kiện trong Diagnostic Detail View

Phân tích điều kiện (condition analysis) có nghĩa là đánh dấu vị trí một lỗi trong chương trình. Mã chương trình được hiển thị dạng Statement List (STL) hoặc Ladder Logic (LAD), phụ thuộc vào ngôn ngữ lập trình đã sử dụng. Bạn có thể thấy trong chương trình tín hiệu nào gây ra thông báo lỗi.

7.12 Quan sát, điều khiển vận hành dùng SIMATIC WinCC

SIMATIC WinCC (Windows Control Center) là một hệ thống SCADA có tính năng mạnh mẽ được sử dụng trong tất cả các lĩnh vực sản xuất tự động và xử lý trong công nghiệp. WinCC có thể lắp đặt được trên máy tính PC chuẩn (tương thích AT) hoặc trong bảng điều khiển vận hành OP47-12-lắp đặt trong vùng lân cận máy. WinCC là một phần mềm 32 bit chạy trên hệ điều hành Microsoft Windows 95/98 hoặc Microsoft Windows NT.

WinCC hiện có ở dạng phần mềm runtime hay phần mềm hoàn chỉnh (runtime và hệ thống cấu hình). WinCC cho phép bạn tạo ra các sơ đồ xử lý để điều khiển và giám sát máy móc hay nhà máy, lưu trữ dữ liệu và các sự kiện bằng cơ sở dữ liệu SQL, quản lý tất cả các thông tin về nhà máy và in ra dưới dạng các bản báo cáo (report). Trong phiên bản WinCC runtime, người vận hành giám sát và điều khiển máy hay nhà máy sử dụng giao diện đồ họa thiết kế bằng WinCC.

Ưu điểm của WinCC

Control Center là cấp cao nhất trong hệ thống WinCC. Bạn khởi động tất cả các module từ cấp này.

User Administrator gán các quyền sử dụng đối với các module runtime của từng người soạn thảo. Mỗi lần có người truy cập, hệ thống sẽ kiểm tra xem có truy cập đúng quyền không và sau đó cho phép các vùng project (dự án) được truy cập tương ứng với quyền được phép của người dùng.

Graphics Designer cho phép bạn tạo các sơ đồ xử lý, nó có giao diện để sử dụng bằng công cụ, bảng đồ họa, hỗ trợ cấu hình hệ thống nhờ các đối tượng tích hợp (integrated object) và thư viện ký hiệu. Mở các giao diện cho phép bạn nhập các đồ họa và các đối tượng đồ họa được tạo ra trước đó bằng

công cụ đồ họa khác; có giao diện OLE 2.0. Một Assistant giúp bạn tạo các đáp ứng động của các đối tượng sơ đồ xử lý.

Global Scripts là thuật ngữ chung dùng cho các chức năng và thao tác C. Các chức năng Project và các thao tác được giới hạn trong project tạo ra chúng. Các chức năng nội bộ và chức năng chuẩn mở rộng tính hợp lệ vượt quá phạm vi một project: được gọi là “project crossing functions”. Ví dụ: lưu trữ xử lý động, lưu trữ dữ liệu người dùng và lưu trữ chọn lọc.

Tag Logging chứa các chức năng đọc thông tin xử lý vào hệ thống HMI và chuẩn bị dữ liệu này cho hiển thị và lưu trữ.

Alarm Logging: các sự kiện xử lý, thông báo, cảnh báo được chọn lựa, chuẩn bị, hiển thị, nhận biết và lưu trữ. Ưu điểm này cho phép bạn hiểu các thông tin về sự cố và trạng thái hoạt động cho phép bạn nhận biết trước các tình huống. Giúp giảm bớt hoặc ngăn ngừa thời gian ngừng trệ và tăng chất lượng sản phẩm.

Use Text Library để soạn thảo text sử dụng cho các module WinCC của phiên bản runtime.

Report Designer giúp đỡ bạn trong trình bày (layout) các bản báo cáo và truyền bản báo cáo ra máy in. Khi bản báo cáo được gọi và hiển thị hoặc được in thì vị trí chiếm khi cấu hình bản báo cáo sẽ được thay thế bằng các giá trị xử lý hiện hành.

WinCC và các phụ kiện WinCC

Hiện có một số phụ kiện cho hệ thống WinCC SCADA. Ở đây chúng tôi chỉ đề cập 3 trong số các phụ kiện này: *WinCC/Server* cho phép bạn sử dụng WinCC trong môi trường client/server. Module này điều khiển tới 16 client. *WinCC/User Archives* là một module trợ giúp lưu trữ dữ liệu trong random data structure. Module này cho phép bạn lưu trữ các thông số hoạt động của máy trong cơ sở dữ liệu WinCC để hiển thị dữ liệu dưới dạng một bảng hoặc để nhập và xuất dữ liệu. *WinCC/Storage* được sử dụng cho lưu trữ dữ liệu và xuất các giá trị đo được, báo cáo và thông báo ra ngoài.

Các phụ kiện modules WinCC khác hiện có sử dụng cho nhiều công việc khác nhau chẳng hạn như bảo trì, quản lý năng lượng và truyền thông với hệ thống thứ 3. Các module phụ kiện WinCC được cung cấp dưới dạng các kênh truyền thông tới WinCC, ActiveX-Object, Graphic Object hoặc ở dạng các phần mềm độc lập.

Bảng điều khiển OP 47-12

Bảng điều khiển OP 47-12 được sử dụng cho các công việc điều khiển và giám sát nhà máy và máy phức tạp. Phía trước bảng điều khiển có cấp bảo vệ IP 65- phù hợp cho sử dụng trong các môi trường khắc nghiệt trong công nghiệp. Bảng điều khiển OP 47-12 chạy trên Windows NT (Workstation) và kèm phiên bản WinCC runtime, hiện có tiếng Anh và Đức.

OP47-12 có một vi xử lý Intel Pentium 200MHz, bộ nhớ ẩn phụ (Second Level Cach) 512 KB, bộ nhớ chính 64 MB, một đĩa cứng 2.1 GB và một ổ đĩa mềm 3.5”.

Hiển thị TFT 12” có độ phân giải 800 x 600 điểm ảnh và 256 màu. Bàn phím dạng màng có 33 phím hệ thống (3 phím có đèn), 32 phím chức năng (28 phím có đèn) và một chuột. Bạn có thể kết nối thêm một module nút nhấn điều khiển trực tiếp 24 V với 16 ngõ ra.

Trạm vận hành OS 57 và OS 77

Các trạm vận hành SIMATIC (operator station) đã được lắp ráp và cài đặt sẵn dùng cho điều khiển máy và nhà máy. Trong kết nối với phiên bản trọn bộ WinCC chúng có thể được sử dụng như một trạm cấu hình. Trạm OS 57 là một desktop sử dụng hệ điều hành Windows 95; trạm OS 77 là một trạm hình tháp (tower) sử dụng hệ điều hành Windows NT 4.0 (Workststion).

Hệ thống hiển thị Flat (bảng điều khiển PC)

Hệ thống hiển thị phẳng là các PC nhỏ gọn và rất mạnh sử dụng trong công nghiệp. Nhờ có thiết kế nhỏ gọn, bền về cơ, phía trước có cấp bảo vệ IP 65, chúng có thể sử dụng trong khoảng không gian chật hẹp ở gần máy và trong môi trường dơ bẩn, bụi và dao động.

Nhỏ gọn nhất là *SIMATIC PC F110* có các đặc tính sau:vi xử lý Intel Pentium MMX 200 MHz, vùng nhớ chính có thể mở rộng tới 128 MB, hiển thị màu 10.4” DSTN với độ phân giải 640 x 480 điểm ảnh và 256 màu, ổ đĩa mềm 3.5” và ổ đĩa cứng 3.2 GB. Phiên bản mới nhất là *SIMATIC PC F145*- được trang bị vi xử lý Intel Pentium II MMX 333 MHz, vùng nhớ chính có thể mở rộng tới 384 MB, hiển thị màu 13.3” TFT với độ phân giải 1024 x 768 điểm ảnh và 64k màu, ổ đĩa mềm 3.5” và ổ đĩa cứng 3.2 GB và một ổ đĩa CD-ROM.

7.13 Chẩn đoán xử lý trong chương trình người dùng bằng S7-PDIAG

S7-PDIAG là một phần mềm tùy chọn. Nó cho phép bạn tích hợp các chẩn đoán xử lý trong chương trình người dùng của một trạm SIMATIC (từ

CPU 314 trở lên) bằng các ngôn ngữ lập trình LAD, FBD và STL. Chức năng chẩn đoán xử lý điều hành hoạt động của một máy hay nhà máy, nhận biết, cách ly, thông báo lỗi và đề nghị biện pháp sửa chữa. Do đó S7-PDIAG làm giảm thời gian ngưng trệ.

Khi được sử dụng trong liên kết với phần mềm cấu hình HMI là SIMATIC ProTool và phần mềm SIMATIC ProAgent đặc biệt phù hợp trong các công việc chẩn đoán xử lý. S7-PDIAG là một công cụ rất mạnh để thiết kế một hệ thống chính xác và đáng tin cậy dùng cho quan sát xử lý, chẩn đoán và sửa chữa sự cố.

Phương pháp giám sát của PDIAG

PDIAG sử dụng các phương pháp khác nhau để phát hiện lỗi.

General monitoring dùng giám sát các phép toán logic trên các địa chỉ

Address monitoring giám sát mức tín hiệu hoặc cạnh tín hiệu của từng địa chỉ, tham gia vào việc tính thời gian trì hoãn khi cần.

Motion monitoring kiểm tra xem các chuyển động vật lý trong xử lý có được thực hiện đúng và đúng tốc độ hay không. PDIAG có các chức năng giám sát đã được định nghĩa trước là:

- Action Monitoring (sau khi khởi động, thiết bị đã đạt tới vị trí cuối được chọn trong thời gian trì hoãn xác định chưa ?)
- Startup Monitoring (sau khi khởi động, thiết bị đã đi qua vị trí cuối được chọn trong thời gian trì hoãn xác định chưa ?)
- Reaction Monitoring (sau khi qua một cờ vị trí (position flag) thiết bị đã đạt tới vị trí cuối được chọn trong thời gian trì hoãn chưa ? hoặc sau khi thiết bị đã đạt tới vị trí cuối nó có đi qua vị trí cuối này trong thời gian trì hoãn không?)
- Interlock Monitoring (sau khi khởi động, điều kiện khoá chéo có được thỏa mãn trong thời gian trì hoãn không ?)

Cấu trúc theo Unit

Mỗi khối trong chương trình người dùng đều có một *unit*, đó là khả năng của các chẩn đoán xử lý. Một unit thường kết hợp các phần có chung đặc tính kỹ thuật hoặc xử lý và do đó cho phép định vị nhanh chóng một lỗi. Bạn có thể kết hợp các định nghĩa lỗi riêng lẻ, chuyển động hay các sub-unit, thành một unit.

Motions là các hoạt động xử lý được giám sát theo 2 hướng, 2 vị trí cuối hoặc hơn. Khi chuyển động bắt đầu, nó di chuyển theo một hướng đã được gán trước. Bạn sử dụng chức năng *Error Definition* để xác định lỗi trong thông báo lỗi. Chẳng hạn điều này có thể là mức tín hiệu của một địa chỉ sau thời gian trì hoãn hoặc một phép toán logic tự định nghĩa dùng các phần tử PDIAG.

PDIAG Unit Overview hiển thị các unit đã được cấu hình. Thư mục *Blocks* hiển thị tất cả các khối có thể truy cập chức năng chẩn đoán xử lý cùng với các sub-unit và các định nghĩa lỗi của chúng.

Hình 7.10 Ví dụ tổng quát phần mềm S7-PDIAG.

Thiết kế với PDIAG

Để cấu hình một chương trình giám sát (monitoring routine) bạn dùng trình soạn thảo hoặc thực hiện trực tiếp từ chương trình chẩn đoán PDIAG.

Trong trình soạn thảo, di chuyển chuột tới địa chỉ mà bạn muốn cài đặt một chương trình giám sát liên kết với một lệnh gán, một cuộn dây hoặc một hộp (box) và sau đó chọn EDIT → SPECIAL OBJECT PROPERTIES → MONITORING. Trong mục “Process Monitoring” chọn phương pháp giám sát.

Địa chỉ mà bạn đã chọn bằng cách trên là “địa chỉ chẩn đoán đầu”. Một phân tích điều kiện được khởi động từ địa chỉ này và sẽ chỉ báo các trạng thái tín hiệu gây ra thông báo lỗi. Thiết bị hiển thị thực hiện phân tích điều kiện không cần đến chương trình người dùng phụ.

Bạn có thể đặt cấu hình chương trình giám sát trực tiếp từ PDIAG. Để khởi động PDIAG, mở SIMATIC Manager chọn OPTIONS →

CONFIGURE PROCESS MONITORING. Tuy nhiên phương pháp này không thể phân tích điều kiện. Bạn nên chọn phương pháp này, chẳng hạn cho các ngõ vào không có tham chiếu trong chương trình.

Giám sát chuyển động cần một chương trình đặc biệt. Ví dụ một chương trình LAD phù hợp với hiển thị chẩn đoán chuẩn ProAgent (khối chức năng FB 100) được phân phối cùng với công cụ PDIAG. Bạn có thể sử dụng chương trình này như một mẫu (model) cho các chương trình giám sát của mình.

Để hoàn tất cấu hình, kích hoạt chương trình giám sát. Nếu phù hợp, nhập văn bản thông báo cho thiết bị hiển thị. Lắp lại cách thức này cho tất cả các chương trình giám sát.

Trong trình soạn thảo chọn OPTIONS → CUSTOMIZE ... và mở mục “Create Block”, ở đây chọn mục “Store process diagnostics data”. Để dịch các chương trình giám sát trong PDIAG, chọn PROCESS DIAGNOSTICS → COMPLIE. Để các chương trình giám sát có hiệu lực bạn phải gọi các khối chức năng chương trình giám sát trong một khối được xử lý theo chu kỳ, chẳng hạn khối tổ chức OB1.