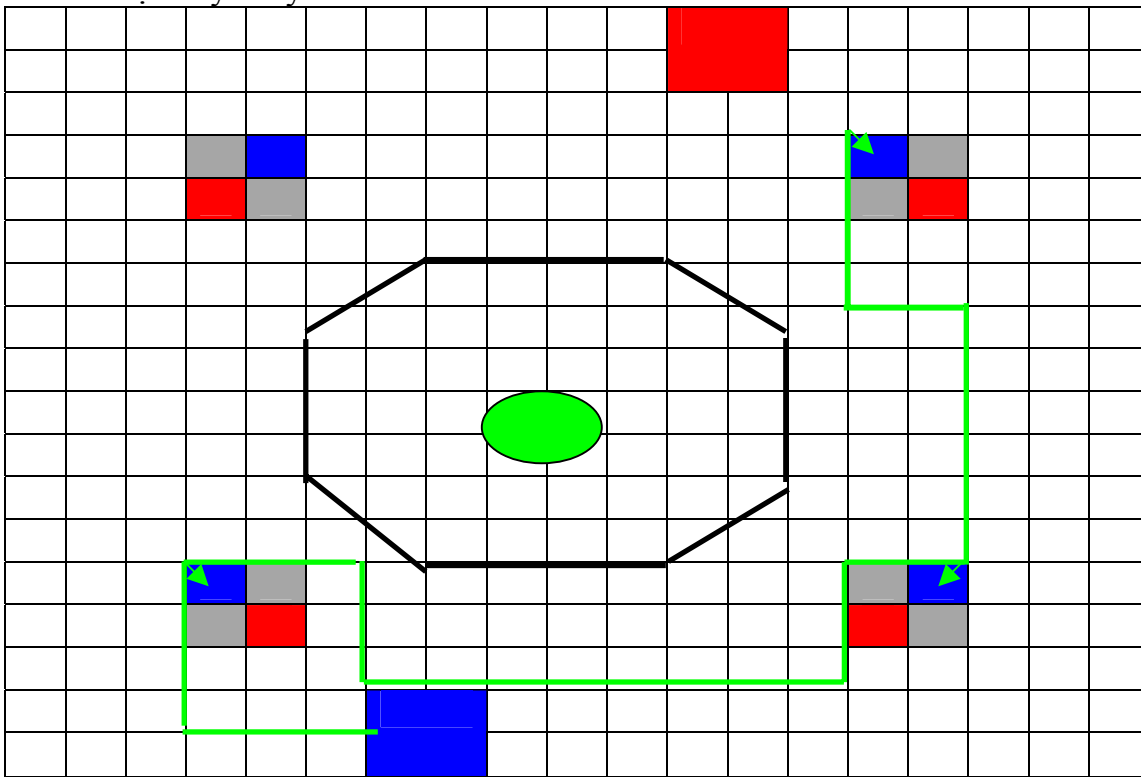


CHƯƠNG 7 NHỮNG ỨNG DỤNG CỦA PLC (5 LT)

7.1. Ứng dụng PLC trong lĩnh vực điều khiển robot:

Về vấn đề robot công nghiệp chủ yếu là các cánh tay máy làm việc trong các nhà máy lắp ráp và sản xuất ô tô, mô tô, tại các bến cảng, kho bãi chứa hàng...thì PLC có những vai trò rất lớn.

Ở đây chỉ giới thiệu đến bạn đọc chủ yếu là các bạn sinh viên tham dự các cuộc thi robocon. Đây là chương trình thường xuyên tổ chức hàng năm, việc cho robot tự động dò theo các vạch trắng là đề tài chính mà rất nhiều bạn trong cuộc tốn rất nhiều thời gian. Sau đây tôi sẽ đưa ra một giải pháp để các bạn tham khảo trong quá trình ứng dụng PLC vào lĩnh vực này. Đây là mô hình sơ đồ sân đấu:



Hình 1: Sơ đồ sân đấu và hành trình mà robot cần phải thực hiện

Bảng 1: Mô tả hành trình làm việc của Robot

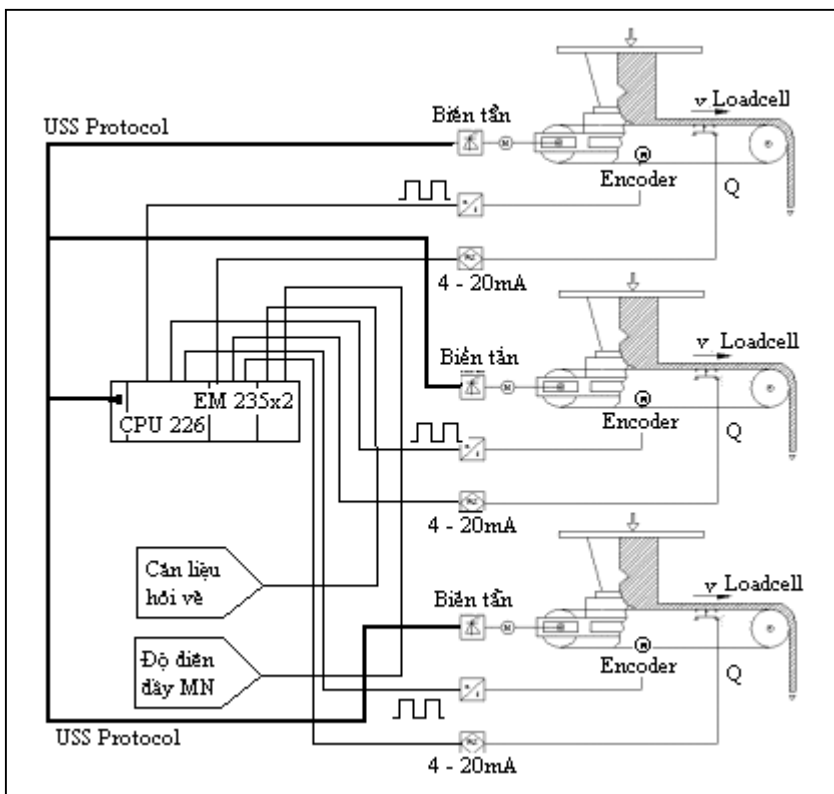
Số vạch	4	5	8	9	10	12	13	16	17	23	24	27	28	29	30
Bánh trái	T1	T1	T1	cộng thêm 1 vạch	T1	T1	T1	T1	L1	T1	L1	T1	T1	T1	cộng thêm 1 vạch
Bánh phải	T2	L2	T2		L2	T2	L2	T2	T2	T2	T2	T2	T2	L2	T2

chế độ làm việc	chạy thẳng	rẽ phải	chạy thẳng	bỏ bóng	rẽ phải	chạy thẳng	rẽ phải	chạy thẳng	rẽ trái	chạy thẳng	rẽ trái	chạy thẳng	rẽ phải	chạy thẳng	bỏ bóng
Số vạch	31	33	36	41	42	43	>43								
Bánh trái	L1	T1	L1	T1	T1	T1	ngừng & bỏ bóng								
Bánh phải	T2	T2	T2	T2	L2	T2									
chế độ làm việc	rẽ trái	chạy thẳng	rẽ trái	chạy thẳng	rẽ phải	chạy thẳng									

Lấy tập File từ ĐANTN Dương để bổ sung. (Sơ đồ bố trí cảm biến trên robot; Sơ đồ phân bố độ rộng xung; Sơ đồ băm điện áp; sơ đồ thuật toán của chương trình; Chương trình viết dưới dạng STL).

7.2. Ứng dụng PLC trong hệ thống sản xuất linh hoạt:

Hiện nay, hệ thống cân bằng định lượng được ứng dụng rất rộng rãi trong các nhà máy xí nghiệp công nghiệp. Ở nơi đâu có sự phối trộn các chất theo tỉ lệ định trước (bài toán phối liệu) thì ở đó có sự tham gia của cân bằng định lượng, đặc biệt là các nhà máy chế biến vật liệu xây dựng, nhà máy phân bón, cao su... Hệ thống này có khả năng điều chỉnh tự động được từng chất ứng với tỉ lệ đặt trước dựa trên cơ sở các vòng lặp điều chỉnh ví dụ PI, PID.

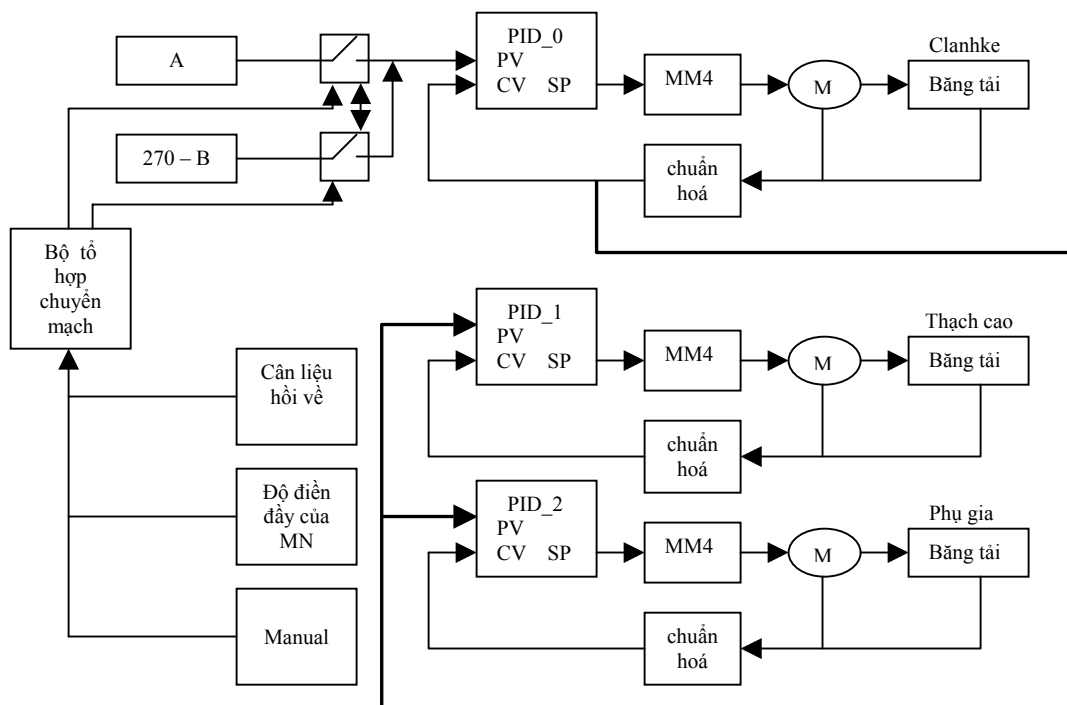


Hình 2: Sơ đồ công nghệ của hệ thống phối liệu

Đặt vấn đề:

Làm thế nào để phối liệu theo giá trị đặt trước của 3 chất clanhke, thạch cao, phụ gia tương ứng là 70%, 20% 10% để nghiền xi thành xi măng. Trong đó tổng khối lượng cần phải đổ vào máy nghiền là **A tấn/h**. Ngoài ra, hệ thống làm việc còn phụ thuộc vào cân liệu hồi về ở đầu vào của máy nghiền (sau khi ra khỏi máy nghiền những hạt có khối lượng lớn được hồi về nhờ phân ly động) và độ điện đầy của máy nghiền. Có nghĩa là lúc nào hệ thống

làm việc cũng phải đảm bảo được yếu tố đầu tiên là độ điền đầy của máy nghiền là 80%. Nếu lượng liệu từ hệ thống phối liệu đưa đến cộng với liệu hồi về làm cho độ điền đầy của máy nghiền vượt mức **80%B tấn/h** thì hệ sẽ tự động giảm lượng liệu cung cấp đến nhưng vẫn đảm bảo được tỉ lệ phần trăm của bài toán phối liệu mặc dù khối lượng xuất không đạt **A tấn/h**, ngoài ra vòng lặp điều chỉnh còn phải nhận biết được sự thay đổi tham số của Clanhke để kịp thời điều chỉnh phụ gia và thạch cao. Sự thay đổi thông số trên thạch cao và phụ gia là hàm bậc nhất với biến là clanhke. Như vậy mỗi băng tải có một vòng lặp điều chỉnh PI với thông số phản hồi là tín hiệu tổng hợp từ hai tín hiệu của loadcell và encoder, đầu ra là giá trị setpoint xuống biến tần theo đường USS. Ngoài ra hệ thống còn làm việc được ở chế độ Manual, trong chế độ này hệ thống không quan tâm đến lượng liệu hồi về và độ điền đầy của máy nghiền. Như vậy hệ thống làm việc ở 3 chế độ: $\text{ĐỀM} \leq 80\%B$; $\text{ĐỀM} > 80\%B$; Manual. giả sử ở 80% lượng liệu trong máy nghiền là **270 tấn**. Có thể mô tả sơ đồ khối điều khiển hệ thống như sau:



Hình 3: Sơ đồ mô tả hệ thống điều khiển CSDL cho hệ thống nghiền xi măng

Đương nhiên là hệ thống phải quản lý trong giới hạn nhất định, nếu một trong 3 băng tải gặp sự cố giả đầy mà lượng liệu vượt mức ngưỡng được đặt tại đầu cân bằng, lúc đó dòng đưa về vượt mức 20mA thì hệ thống sẽ dừng làm việc thông qua chương trình con xử lý sự cố. Hoặc bất cứ 1 băng chuyên nào cũng có giám sát trượt đai, nếu xảy ra thì chương trình xử lý sự cố cũng sẽ được gọi.

Yêu cầu phân cứng của hệ thống:

+ 1 PLC_CPU 226

- + 2 EM235 Module
- + 3 Biến tần MM3 hoặc MM4 (điều chỉnh tốc độ 3 băng tải)
- + 4 Load cell (lấy tín hiệu về đầu cân)
- + 4 Đầu cân (chuẩn hoá tín hiệu về từ 4÷20 mA; cài đặt giá trị giới hạn trên)
- + 1 Sound Sensor
- + Bộ cáp đồng trục nối từ RS-485 Port đến PLC
- + 3 Encoder

Sơ đồ đấu nối hệ thống như hình 2.

Quá trình tính toán, chuẩn hoá, setpoint cho từng vòng lặp được thực hiện như sau:

Bài toán đặt ra là làm thế nào để tín hiệu phản hồi về được chuyển sang đơn vị tấn/h để so sánh với giá trị setpoint.

Tính tải trọng Q:

$$Q = \frac{q \cdot L}{2}$$

Với: Q: Tải trọng điểm [Kg]

q: Tải trọng trên băng tải [Kg/m]

L: Chiều dài tính toán [m]

Công thức tính trọng lượng P trên băng tải theo thời gian :

$$P = q \cdot V = \frac{2 \cdot Q}{L} \cdot V (*)$$

Với: P: Trọng lượng băng tải theo thời gian [Kg/s]

V: Vận tốc dài của băng tải [m/s]

Theo (*), tính q bằng cách tính Q vì L đã biết trước. Để chuẩn hoá q từ [0÷1] ta chia q cho q_{max} ; tính q_{max} ta dựa vào P_{max} đó là giá trị đặt tương ứng với % của mỗi chất và V_{max} của băng tải (dựa vào tốc độ định mức của động cơ).

Như vậy, tín hiệu phản hồi về chính là P [tấn/h]. Cần phải tìm vận tốc V [m/h] và tải trọng trên băng tải q [tấn/m].

Xây dựng công thức tính toán tốc độ của băng tải nhờ vào Encoder:

Giả sử chọn thời gian tính toán là 250ms tương ứng với số xung tính được là x xung

Nếu ta chọn loại encoder có thông số 500xung/vòng thì sau x xung:

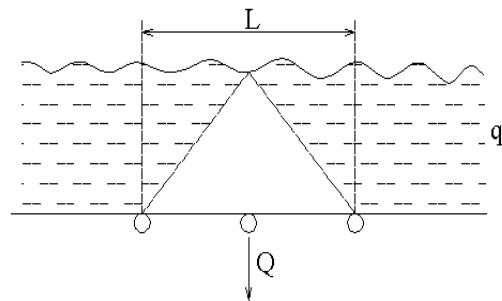
Bánh xe encoder quay được: $\frac{x}{500}$ (vòng) tương ứng với góc quay $\frac{2\pi x}{500}$ (rad).

Sau 1ms bánh xe encoder quay được: $\frac{2\pi x}{250 \cdot 500}$ (rad).

Như vậy vận tốc góc của bánh xe là:

$$\omega = 1000 \cdot \frac{1}{250} \cdot \frac{2\pi x}{500} \left(\frac{\text{rad}}{\text{s}} \right)$$

Vận tốc dài của bánh xe bằng vận tốc dài của băng tải:



Hình 3: Mặt cắt dọc của băng tải

$$V_{ed} = \omega \cdot \frac{d}{2} = 1000 \cdot \frac{1}{250} \cdot \frac{2\pi x d}{500} \cdot \frac{d}{2} \left(\frac{m}{s} \right) = V_{bt}$$

d: Đường kính của bánh xe: chọn $d = 0,1$ [m]; $L=0.5$ [m]

$$V_{bt} = V_{ed} = 3600 \cdot \frac{1000 \cdot 2\pi x d}{250 \cdot 500} = 18,086x \left(\frac{m}{h} \right)$$

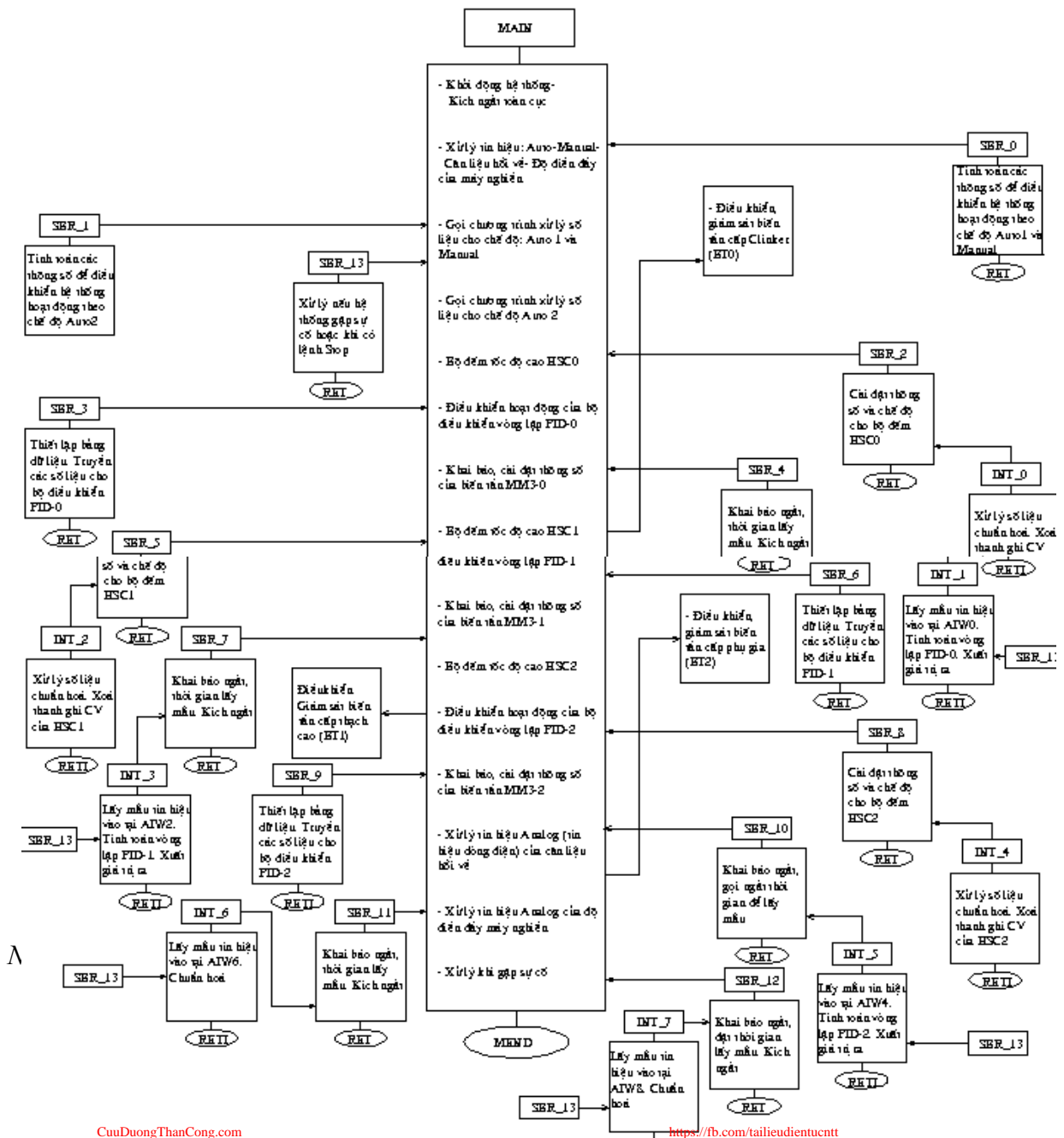
Chuẩn hoá về giá trị từ $[0 \div 1]$ tiến hành chia cho V_{max} ; tính V_{max} dựa vào tốc độ định mức của động cơ.

Chuẩn hoá và đưa về đầu vào CV (Current Value) của bộ PID của PLC:

$$\frac{q}{q_{max}} = \frac{[0 \div q_{max}]}{q_{max}} = [0,0 \div 1,0]$$

$$\frac{P}{P_{max}} = \frac{q \cdot V_{bt}}{q_{max} \cdot V_{max}} = [0,0 \div 1,0][0,0 \div 1,0] = [0,0 \div 1,0]$$

Quá trình thực hiện được thực hiện theo sơ đồ khối sau:



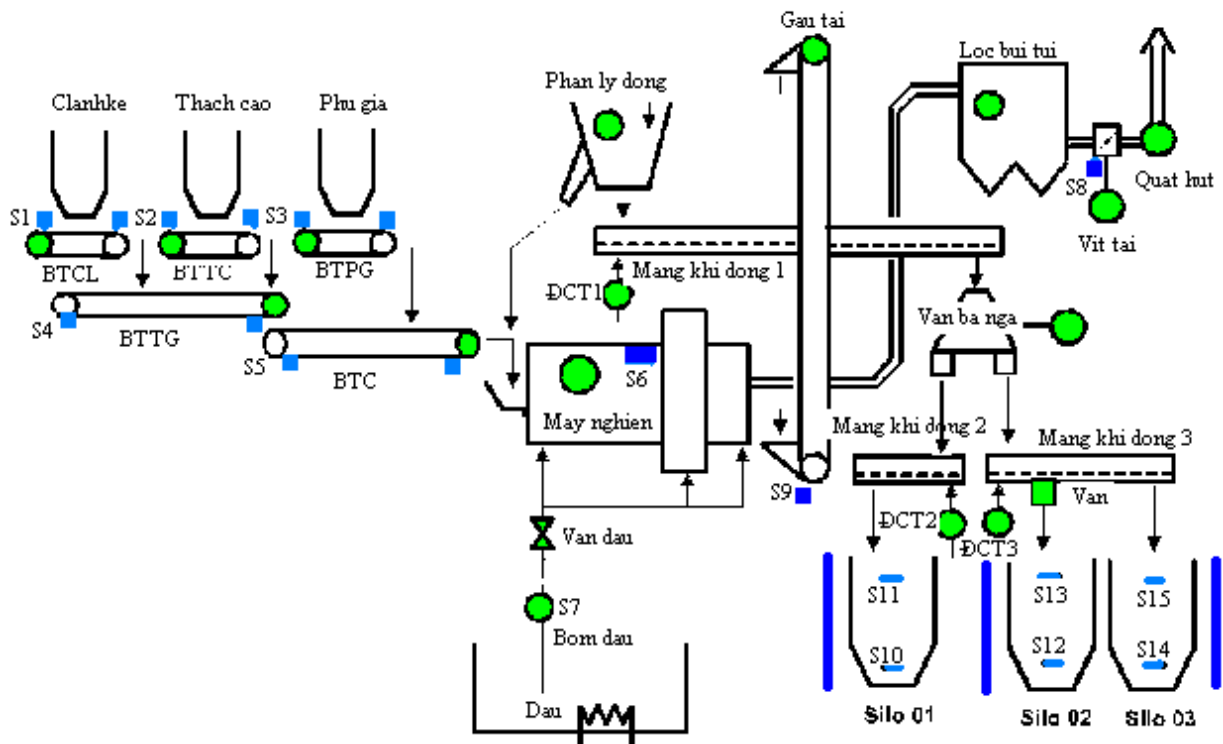
Các quá trình tính toán này thực hiện trong PLC. Để tính Q dựa vào đầu vào analog tương ứng, tại đây ta phải thực hiện các bước lấy mẫu. Đối với bộ PID ta phải có chương trình khai báo các tham số cần thiết của bộ PID **liên hệ chương 3**. Để tính vận tốc V phải sử dụng bộ đếm tốc độ cao... Điều khiển các biến tần trong mạng dùng giao thức USS protocol như trình bày ở phần 7.5.

7.3. Ứng dụng PLC trong điều khiển quá trình:

Dưới đây là sơ đồ công nghệ của hệ thống phối liệu, nghiền, phân loại và phân phối xi măng vào trong các silô. Ở đây không xét đến việc điều khiển hệ thống phối liệu như nêu ở trên phần 7.2. Chỉ thực hiện công việc tương đối đơn giản:

Chọn silô muốn nhập vào thông qua các van sau: chuyển vị trí của van 3 ngã và chọn vị trí của van trên máng khí động 3. Nếu chuyển van 3 ngã sang bên phải thì silô 1 được nhập. Sang vị trí giữa thì silô 1 và silô 2 hoặc silô 3, nếu van trên máng khí động 3 đóng thì silô 2 được nhập, van ở trạng thái mở thì silô 3 được nhập. Nếu van 3 ngã chuyển sang trái thì silô 2 hoặc silô 3, nếu van trên máng khí động 3 đóng thì silô 2 được nhập, van ở trạng thái mở thì silô 3 được nhập. Giả sử khi đang nhập cho silô 1 (van 3 ngã ở vị trí bên trái) mà phát hiện đầy thì hệ thống sẽ tự động chuyển van 3 ngã sang bên phải để nhập cho silô 2 hoặc 3 (nếu 2 đã đầy thì nhập vào 3 và ngược lại).

Điều kiện để khởi động hệ thống:



Hình 5: Sơ đồ công nghệ của hệ thống cấp liệu, nhiên, phân loại, phân phối xi măng

Các silô chứa đầy.

Dầu thủy lực cấp cho máy nghiền đã đủ áp suất.

Các băng chuyền không bị trượt đai.

Sau khi đã chọn silô và kiểm tra đủ các điều kiện an toàn cho việc khởi động, hệ thống phải được khởi động theo trình tự như sau: MKĐ2, MKĐ3 → MKĐ1 → Mở van → Quạt hút (lọc bụi) → Gàu tải → Phân ly động → Mở van dầu → Máy nghiền → BTC → BTTG → BTCL, BTPG, BTTC.

Từ yêu cầu công nghệ như trên ta tiến hành thiết kế chương trình như sau:

1. Vẽ giản đồ thời gian hoặc viết lưu đồ thuật toán.
2. Tính chọn PLC và module mở rộng.
3. Phân công I/O.
4. Quy định các ô nhớ để giám sát lỗi, khởi động hoặc dừng từ xa.
5. Tiến hành dịch sang ngôn ngữ của PLC từ giản đồ thời gian hoặc viết lưu đồ thuật toán.

7.4. Ứng dụng PLC trong mạng thu nhận dữ liệu từ biến tần:

Để điều khiển biến tần thông qua PLC người ta thường dùng các cách sau:

1. Dùng các đầu vào/ra số của PLC, nhưng chỉ thực hiện được những chức năng đơn giản như dừng, khởi động, đảo chiều còn việc thay đổi thời gian khởi động hoặc dừng, đặt lại tốc độ...không thể thực hiện được ở chế độ này.

2. Để thay đổi giá trị setpoint trong điều khiển phản hồi, mỗi biến tần mất đi 1 đầu vào analog và 1 đầu ra analog. Ngoài ra còn phải dùng các đầu vào/ra số để điều khiển biến tần.
3. Điều khiển biến tần qua mạng Profibus, đối với loại MM3, MM4 của Siemens đã có sẵn giao diện Profibus trên RS458 Port. Nhưng đối với những ứng dụng nhỏ thì việc thiết kế một mạng Profibus sẽ đưa giá thành lên cao, do đó không kinh tế.
4. Dùng Port 0 của PLC để kết nối tới các Port của biến tần, 1 PLC có thể điều khiển tối đa 1 mạng gồm 31 biến tần. Mạng này gọi là mạng USS. Dạng kết nối là điểm-điểm. Ta có thể điều khiển toàn bộ các chức năng của biến tần thông qua mạng này, ngoài ra còn có thể giám sát được dòng điện, điện áp, tốc độ, hướng quay...dùng vào các vùng nhớ mà PLC dành riêng cho mỗi biến tần. Chi phí cho mạng này là thấp và tối ưu nhất cho các ứng dụng nhỏ và vừa.
5. Chuẩn điều khiển mạng biến tần (giao thức USS)

Sau đây là phương pháp điều khiển mạng biến tần dùng PLC qua giao thức USS:

7.4.1. Điều kiện sử dụng giao thức USS:

Thư viện lệnh của STEP 7 - Micro/Win cung cấp 14 chương trình con, 3 thủ tục ngắt và một tập lệnh (gồm 8 lệnh) hỗ trợ cho giao thức USS.

- + Giao thức USS sử dụng Cổng 0 (Port 0) cho truyền thông USS.

Sử dụng lệnh USS_INIT để lựa chọn Port 0 cho cả USS hoặc PPI. Sau khi đã lựa chọn Port 0 cho truyền thông với chuẩn USS, không được sử dụng Port 0 cho bất kỳ mục đích nào khác.

Để phát triển các chương trình ứng dụng sử dụng giao thức USS, nên sử dụng CPU 226, CPU 226XM hoặc module EM 277 PROFIBUS-DP kết nối đến card PROFIBUS-CP ở máy tính. Cổng truyền thông thứ hai ở các loại CPU này sẽ cho phép STEP 7 - Micro/Win giám sát được ứng dụng trong khi sử dụng giao thức USS.

- + Các lệnh USS tác động đến tất cả các bit SM với truyền thông Freeport qua Port 0.

- + Các lệnh USS sử dụng 14 chương trình con và 3 thủ tục ngắt.

+ Các giá trị của các lệnh USS yêu cầu 400 byte của miền nhớ V. Địa chỉ bắt đầu được ấn định bởi người sử dụng và phần còn lại dành cho các giá trị khác.

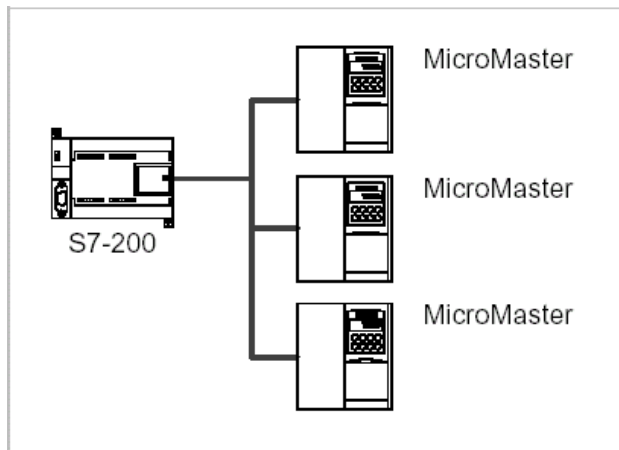
+ Vài lệnh trong lệnh USS yêu cầu một bộ đệm truyền thông 16 byte. Chẳng hạn với một tham số cho lệnh, cần phải cung cấp một địa chỉ bắt đầu trong miền nhớ V của bộ đệm này.

- + Khi thực hiện các phép tính, các lệnh USS sử dụng thanh ghi AC0 đến AC3.

Cũng có thể sử dụng các thanh ghi trong chương trình; tuy nhiên, giá trị trong các thanh ghi sẽ bị thay đổi bởi lệnh USS.

+ Các lệnh USS sẽ làm tăng bộ nhớ của chương trình lên đến 3450 byte. Tùy thuộc vào loại lệnh USS mà dung lượng của bộ nhớ có thể tăng từ 2150 byte đến 3450 byte.

+ Các lệnh USS không thể sử dụng trong chương trình con.



Hình 2.11. Kết nối PLC và biến tần theo giao thức USS

* Lưu ý:

Để thay đổi phương thức truyền thông của Port 0 trở lại PPI để truyền thông với STEP 7 - Micro/Win, cần phải sử dụng lệnh USS _ INIT khác để ấn định lại phương thức cho Port 0.

Cũng có thể định lại phương thức bằng cách chuyển S7-200 sang chế độ STOP, việc này sẽ Reset các tham số của Port 0.

7.4.2. Thời gian yêu cầu cho việc truyền thông với biến tần:

Truyền thông với các MicroMaster (MM) không đồng bộ với vòng quét của S7-200. S7-200 hoàn thành vài vòng quét trước khi một MM hoàn thành việc truyền thông. Các yếu tố giúp xác định thời gian yêu cầu: số MM có trong mạng, tốc độ baud, và thời gian vòng quét của S7-200.

Có vài loại yêu cầu thời gian trễ dài hơn khi sử dụng các lệnh truy xuất thông số. Thời gian yêu cầu cho việc truy nhập các tham số tùy thuộc loại thiết bị và tham số được truy nhập.

Sau khi lệnh USS _ INIT ấn định Port 0 cho giao thức USS, S7-200 sẽ thực hiện hỏi vòng tất cả các biến tần trong những khoảng thời gian theo dưới đây.

Bảng 2.5: Thời gian yêu cầu cho truyền thông với MM

Tốc độ	Thời gian hỏi vòng giữa các biến tần
1200	240 ms (max)
2400	130 ms (max)
4800	75 ms (max)
9600	50 ms (max)
19200	35 ms (max)
38400	30 ms (max)
57600	25 ms (max)
115200	25 ms (max)

7.4.3. Sử dụng các lệnh USS:

Để sử dụng các lệnh trong chương trình điều khiển S7-200, cần phải theo các bước sau:

1. Đưa lệnh USS _INIT vào trong chương trình và thực hiện lệnh này cho mỗi một vòng quét. Có thể sử dụng lệnh này để thiết lập các giá trị hoặc thay đổi các thông số truyền thông.

Khi sử dụng lệnh USS _INIT sẽ có vài ẩn chương trình con và thủ tục ngắt được tự động thêm vào trong chương trình.

2. Chỉ thực hiện một lệnh USS _INIT trong chương trình cho mỗi Drive.

Có thể đưa vào nhiều lệnh USS_RPM_x hay USS_WPM_x khi được yêu cầu, nhưng chỉ một lệnh được làm việc trong một thời điểm.

3. Cấp phát vùng nhớ V cho thư viện lệnh bằng cách kích chuột phải (lấy từ menu) trên Program Block trong cây thư mục.

4. Cài đặt các tham số về địa chỉ và tốc độ được sử dụng trong chương trình cho drive.

5. Dùng cáp để kết nối truyền thông từ S7-200 đến các drive.

* Chú ý:

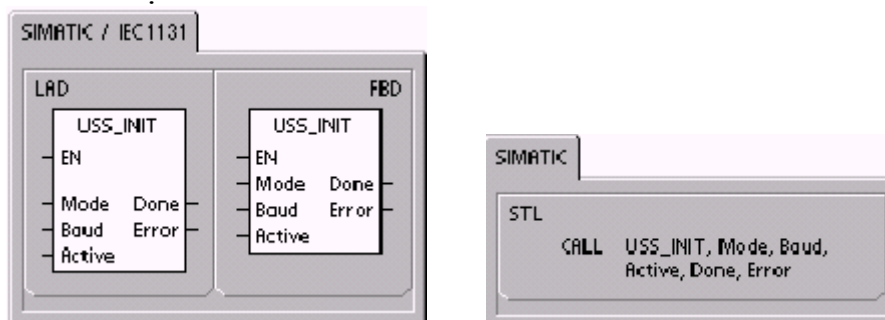
Các thiết bị kết nối với điện thế khác nhau có thể là nguyên nhân sinh ra dòng điện không mong muốn trong cáp kết nối. Dòng điện này là nguyên nhân dẫn đến các lỗi truyền thông hoặc làm hỏng thiết bị.

Cần phải chắc chắn rằng các thiết bị được kết nối với cáp đều có cùng dòng điện định mức hoặc được cách ly để ngăn ngừa dòng điện không mong muốn.

7.4.4. Các lệnh trong giao thức USS:

4.1. Lệnh USS- INIT:

Cấu trúc lệnh:



Lệnh USS_ INIT được sử dụng để cho phép thiết lập hoặc không cho phép truyền thông với các MM. Trước khi bắt kỳ một lệnh USS nào khác được sử dụng, lệnh USS_INIT phải được thực hiện trước mà không được xảy ra lỗi nào. Khi lệnh thực hiện xong và bit Done được set lên ngay lập tức trước khi thực hiện lệnh kế tiếp.

Lệnh này được thực hiện ở mỗi vòng quét khi đầu vào EN được tác động.

Thực hiện lệnh USS_INIT chỉ một lần cho mỗi sự thay đổi trạng thái truyền thông. Sử dụng lệnh chuyển đổi dương tạo một xung ở đầu vào EN. Khi thay đổi giá trị ban đầu các tham số sẽ thực hiện một lệnh USS_ INIT mới.

Người biên soạn: Lâm Tăng Đức - Nguyễn Kim Ánh

Giá trị cho đầu vào Mode lựa chọn giao thức truyền thông: đầu vào có giá trị 010 sẽ ấn định Port 0 dùng cho giao thức USS và chỉ cho phép làm việc theo giao thức này. Nếu đầu vào có giá trị 000 sẽ ấn định Port 0 dùng cho giao thức PPI và không cho phép làm việc theo giao thức USS.

Tốc độ truyền được đặt ở các giá trị: 1200, 2400, 4800, 9600, 19200, 38400, 57600 và 115200 (baud).

Đầu vào Active dùng để xác định địa chỉ của Drive. Chỉ hỗ trợ số địa chỉ Drive từ 0 đến 30.

Các tham số sử dụng trong lệnh USS_INIT.

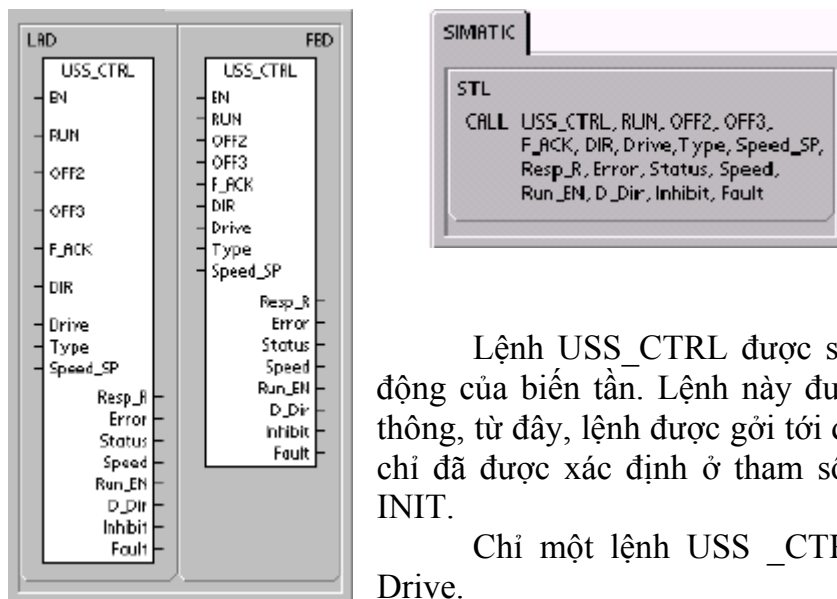
Bảng 2.6: Kiểu dữ liệu và toán hạng của các đầu vào/ra trong lệnh USS_INIT

Đầu vào/ra	Kiểu dữ liệu	Toán hạng
Mode	Byte	VB,IB,QB,MB,SB,SMB,LB,AC,Constant,*VD,*AC,*LD
Baud,Active	Dword	VD,ID,QD,MD,SD,SMD,LD,Constant,AC,*VD,*AC,*LD
Done	Bool	I, Q, M, S, SM, T, C, V, L
Error	Byte	VB, IB, QB, MB, SB, SMB, LB, AC,*VD,*AC,*LD

Khi lệnh USS_INIT kết thúc, đầu ra Done được set lên. Đầu ra Error (kiểu byte) chứa kết quả thực hiện lệnh.

4.2. Lệnh USS - CTRL:

Cấu trúc lệnh:



Lệnh USS_CTRL được sử dụng để điều khiển hoạt động của biến tần. Lệnh này được đưa vào bộ đệm truyền thông, từ đây, lệnh được gửi tới địa chỉ của biến tần, nếu địa chỉ đã được xác định ở tham số Active trong lệnh USS_INIT.

Chỉ một lệnh USS_CTRL được ấn định cho mỗi Drive.

- Bit EN phải được set lên mới cho phép lệnh USS_CTRL thực hiện. Lệnh này luôn ở mức cao (mức cho phép).

Người biên soạn: Lâm Tăng Đức - Nguyễn Kim Ánh

- RUN (RUN/STOP) cho thấy drive là on hoặc off. Khi bit RUN ở mức cao, MM nhận lệnh khởi động ở tốc độ danh định và theo chiều đã chọn trước. Để Drive làm việc, các điều kiện phải theo đúng như sau:
 - + Địa chỉ Drive phải được lựa chọn từ đầu vào Active trong lệnh USS_INIT.
 - + Đầu vào OFF2 và OFF3 phải được set ở 0.
 - + Các đầu ra Fault và Inhibit phải là 0.
- Khi đầu vào RUN là \square OFF \square , một lệnh được chuyển đến MM để điều khiển giảm tốc độ động cơ xuống cho đến khi động cơ dừng.
- Đầu vào OFF2 được sử dụng để cho phép điều khiển MM dừng với tốc độ chậm.
- Đầu vào OFF3 được sử dụng để cho phép điều khiển MM dừng với tốc độ nhanh.
- Bit Resp_R báo nhận phản hồi từ Drive. Tất cả các hoạt động của MM được thăm dò thông tin trạng thái. Tại mỗi thời điểm, S7-200 nhận một phản hồi từ Drive, bit Resp_R được set lên và tất cả các giá trị tiếp theo được cập nhật.
- Bit F_ACK (Fault Acknowledge) được sử dụng để nhận biết lỗi từ Drive. Các lỗi của Drive được xoá khi F_ACK chuyển từ 0 lên 1.
- Bit Dir (Direction) xác định hướng quay mà MM sẽ điều khiển.
- Đầu vào Drive (Drive address) là địa chỉ của MM mà lệnh USS_CTRL điều khiển tới. Địa chỉ hợp lệ: 0 đến 31.
- Đầu vào Type (Drive type) dùng để lựa chọn kiểu MM. Đối với thể hệ MM3 (hoặc sớm hơn) đầu vào Type được đặt 0; còn đối với MM4 giá trị đặt là 1.
- Speed-SP (speed setpoint): là tốc độ cần đặt theo tỉ lệ phần trăm. Các giá trị âm sẽ làm động cơ quay theo chiều ngược lại.
 - Phạm vi đặt: $-200\% \div 200\%$.
- Error: là một byte lỗi chứa kết quả mới nhất của yêu cầu truyền thông đến Drive.
- Status: là một word thể hiện giá trị phản hồi từ biến tần.
- Speed là tốc độ động cơ theo tỉ lệ phần trăm. Phạm vi: -200% đến 200% .
- D-Dir: cho biết hướng quay.
- Inhibit: cho biết tình trạng của the inhibit bit on the drive (0 - not inhibit, 1- inhibit). Để xoá bit inhibit này, bit Fault phải trở về off, và các đầu vào RUN, OFF2, OFF3 cũng phải trở về off.
- Fault: cho biết tình trạng của bit lỗi (0 - không có lỗi, 1- lỗi). Drive sẽ hiển thị mã lỗi. Để xoá bit Fault, cần phải chữa lỗi xảy ra lỗi và set bit F_ACK.

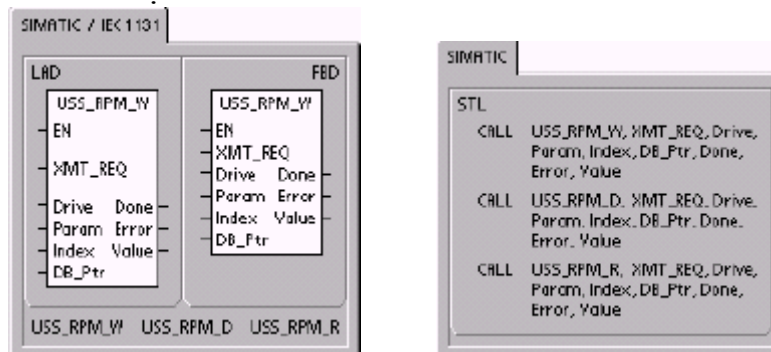
Bảng 2.7: Kiểu dữ liệu và toán hạng của các đầu vào/ra trong lệnh USS_CTRL

Đầu vào/ra	Kiểu dữ liệu	Toán hạng
RUN, OFF2, OFF3, F_ACK, DIR	BOOL	I, Q, M, S, SM, T, C, V, C, L, Power Flow
Resp_R, Run_EN, D_Dir, Inhibit, Fault	BOOL	I, Q, M, S, SM, T, C, V, C, L

Drive, Type	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Constant
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD
Status	WORD	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
Speed_SP	REAL	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, *LD, Constant
Speed	REAL	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, *LD

4.3. Lệnh USS_RPM_x:

Cấu trúc lệnh:



Có 3 lệnh đọc cho giao thức USS.

USS_RPM_W: là lệnh đọc một tham số Word.

USS_RPM_D: là lệnh đọc một tham số Double Word.

USS_RPM_R: là lệnh đọc một tham số thực.

Chỉ một lệnh đọc (USS_RPM_x) hoặc ghi (USS_WPM_x) có thể làm việc tại một thời điểm.

Lệnh USS_RPM_x hoàn thành việc thực hiện lệnh khi MM nhận biết cách thức của lệnh, hoặc khi một lỗi trạng thái được thông báo. Vòng quét vẫn tiếp tục thực hiện trong khi quá trình chờ sự phản hồi.

- Bit EN phải được set để cho phép truyền đi các yêu cầu, và nên giữ lại ở trạng thái đó cho đến khi bit Done được set lên - tín hiệu hoàn thành quá trình (Ví dụ: một lệnh USS_RPM_x truyền đến MM ở mỗi vòng quét khi đầu vào XMT_REQ là on). Do đó, đầu vào XMT-REQ nên được kích xung khi nhận được sườn xung lên để truyền một yêu cầu cho mỗi chuyển tiếp dương của đầu vào EN.

Bảng 2.8: Kiểu dữ liệu và toán hạng của các đầu vào/ra trong lệnh USS_RPM_x

Đầu vào/ra	Kiểu dữ liệu	Toán h?ng
XMT-REQ	BOOL	I, Q, M, SM, T, C, V, L
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Constant

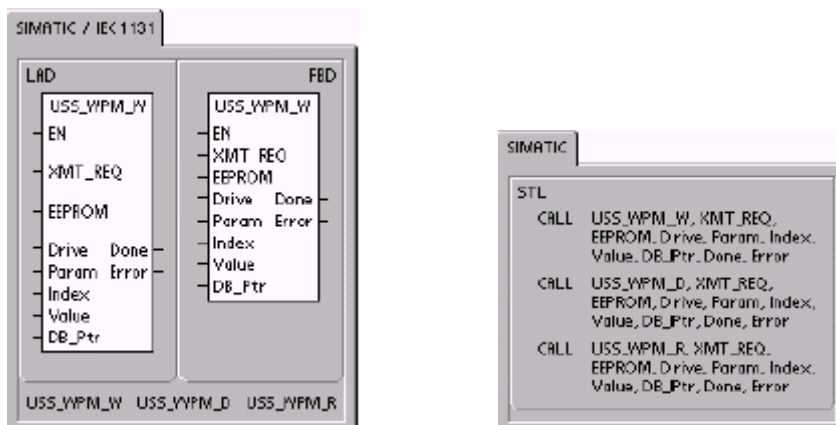
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, Constant
DB-Ptr	DWORD	&VB
Value	WORD DWORD,REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

- Đầu vào Drive là địa chỉ của MM mà lệnh USS_RPM_x được chuyển tới. Địa chỉ hợp lệ là 0 đến 31.
- Param là số tham số (là giá trị cần đọc từ MM).
- Index là con trỏ chỉ vào giá trị để đọc.
- Value là giá trị của thông số phản hồi.
- Đầu vào DB_Ptr được cung cấp bởi địa chỉ của bộ đệm 16 byte. Trong lệnh USS_RPM_x, bộ đệm này dùng chứa kết quả của lệnh đưa đến từ MM.

Khi lệnh USS_RPM_x đã hoàn tất, đầu ra Done được set lên và đầu ra Error (kiểu byte) và đầu ra Value chứa các kết quả của việc thực hiện lệnh. Đầu ra Error và Value sẽ không hợp lệ cho đến khi đầu ra Done được set lên.

4.4. Lệnh USS_WPM_x:

Cấu trúc lệnh:



Có 3 lệnh ghi cho giao thức USS:

USS_WPM_W: là lệnh ghi một tham số Word.

USS_WPM_D: là lệnh ghi một tham số Double Word.

USS_WPM_R: là lệnh ghi một tham số thực.

Chỉ một lệnh đọc (USS_WPM_x) hoặc ghi (USS_WPM_x) có thể làm việc tại một thời điểm.

Lệnh USS_WPM_x hoàn thành việc thực hiện lệnh khi MM nhận biết cách thức của lệnh, hoặc khi một lỗi trạng thái được thông báo. Vòng quét vẫn tiếp tục thực hiện trong khi quá trình chờ sự phản hồi.

- Bit EN phải được set để cho phép truyền đi các yêu cầu, và nên giữ lại ở trạng thái đó cho đến khi bit Done được set lên - tín hiệu hoàn thành quá trình (Ví dụ: một lệnh USS-WPM-x truyền đến MM ở mỗi vòng quét khi đầu vào XMT_REQ là on). Do đó, đầu vào XMT-REQ nên được kích xung khi nhận được sườn xung lên để truyền một yêu cầu cho mỗi chuyển tiếp dương của đầu vào EN.
- Đầu vào Drive là địa chỉ của MM mà lệnh USS_WPM_x được chuyển tới. Địa chỉ hợp lệ là 0 đến 31.
- Param là số tham số.
- Index là biến chỉ vào giá trị để đọc.
- Value là giá trị của thông số cần ghi đến bộ nhớ RAM trong biến tần. Đối với MM3 cũng có thể ghi giá trị này vào EEPROM, bằng cách cài đặt ở tham số P971.
- Đầu vào DB-Ptr được cung cấp bởi địa chỉ của bộ đệm 16 byte. Trong lệnh USS_WPM_x, bộ đệm này dùng chứa kết quả của lệnh đưa đến từ MM. Khi lệnh USS_WPM_x đã hoàn tất, đầu ra Done được set lên và đầu ra Error (kiểu byte) chứa các kết quả của việc thực hiện lệnh.

Khi đầu vào EEPROM được set lên, lệnh sẽ ghi vào cả bộ nhớ RAM và EEPROM của biến tần. Khi đầu vào EEPROM không được set thì lệnh này sẽ chỉ ghi vào bộ nhớ RAM vì MM3 không hỗ trợ chức năng này, do đó, cần phải chắc chắn rằng đầu vào không được set để lệnh chỉ làm việc với MM3.

Bảng 2.9: Kiểu dữ liệu và toán hạng của các đầu vào/ra trong lệnh USS_WPM_x

Đầu vào/ra	Kiểu dữ liệu	Toán hạng
XMT-REQ	BOOL	I, Q, M, SM, T, C, V, L
EEPROM	BOOL	I, Q, M, S, SM, T, C, V, L
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Constant
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, Constant
DB-Ptr	DWORD	&VB
Value	WORD DWORD,REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

7.4.5. Kết nối và cài đặt MicroMaster Series 3 (MM3):

5.1. Kết nối MM3:

Có thể sử dụng cáp chuẩn PROFIBUS và các đầu nối để kết nối S7-200 với MicroMaster Series 3.

* *Chú ý:*

Các thiết bị kết nối với điện thế khác nhau có thể sẽ là nguyên nhân dẫn tới việc phát sinh dòng điện không mong muốn trong cáp kết nối. Dòng điện này là nguyên nhân dẫn tới các lỗi truyền thông hoặc làm hỏng thiết bị.

Cần phải chắc chắn rằng tất cả các thiết bị được kết nối vào một cáp truyền thông đều có cùng dòng điện định mức hoặc được cách ly để ngăn ngừa dòng điện phát sinh không mong muốn.

5.2. Cài đặt MM3:

Trước khi kết nối đến S7-200, cần phải chắc chắn rằng có đủ các thông số của MM. Sử dụng các keypad có sẵn trên biển tần để cài đặt như sau:

1. Reset biển tần để cài đặt lại (tùy chọn). Nhấn phím P: hiển thị P000. Nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P944. Nhấn P để nhập thông số:

P944 = 1

2. Cho phép truy xuất để đọc/ghi tất cả các thông số. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P009. Nhấn P để nhập:

P009 = 3

3. Kiểm tra lại việc cài đặt thông số động cơ cho biển tần. Việc cài đặt này phải theo loại động cơ được sử dụng. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị thông số cần cài đặt. Nhấn P để nhập:

P081 = Tần số định mức của động cơ (Hz).

P082 = Tốc độ định mức của động cơ (RPM).

P083 = Dòng điện định mức của động cơ (A).

P084 = Điện áp định mức của động cơ (V).

P085 = Công suất định mức của động cơ (kW/HP).

4. Đặt chế độ điều khiển tại chỗ hay từ xa (Local/Remove). Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P910. Nhấn P để nhập:

P910 = 1 (Remove)

5. Định giá trị tốc độ Baud cho chuẩn RS-485. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P092. Nhấn P để nhập, nhấn phím mũi tên để hiển thị đúng giá trị tốc độ Baud cho chuẩn RS-485:

P092 3 (1200 baud)

 4 (2400 baud)

 5 (4800 baud)

 6 (9600 baud - chuẩn)

 7 (19200 baud)

6. Nhập địa chỉ Slave. Mỗi drive (tối đa 31) có thể vận hành qua một bus. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P091. Nhấn P để nhập. Nhấn phím mũi tên để hiển thị địa chỉ mong muốn, nhấn P nhập:

$$P091 = 0 \mid 31$$

7. Định thời gian tăng tốc (tùy chọn). Với thời gian đặt này tốc độ động cơ sẽ tăng dần cho đến khi đạt max. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P002. Nhấn P để nhập:

$$P002 = 0 \mid 650.00$$

8. Định thời gian giảm tốc (tùy chọn). Sau khoảng thời gian này động cơ sẽ giảm đến tốc độ cho đến khi dừng. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P003. Nhấn P để nhập:

$$P003 = 0 \mid 650.00$$

9. Serial Link Time-out. Đây là khoảng thời gian lớn nhất cho phép giữa hai lần truy nhập dữ liệu.

Thời gian này được tính sau khi một dữ liệu được nhận. Nếu một dữ liệu của bức điện không được nhận, biến tần sẽ ngắt và hiển thị mã lỗi F008. Đặt giá trị 0 để ngừng việc điều khiển.

Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P093. Nhấn P để nhập. Nhấn phím mũi tên để nhập giá trị mong muốn:

$$P093 = 0 \text{ (240 (thời gian được tính bằng giây))}$$

10. Serial Link Nominal System Setpoint. Giá trị này có thể thay đổi, nhưng phải tương ứng 50Hz hoặc 60Hz, được định nghĩa tương ứng với giá 100% giá trị cho PV hoặc SP. Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P094. Nhấn P để nhập. Nhấn các phím mũi tên để chọn giá trị mong muốn:

$$P094 = 0 \mid 400.00$$

11. Tương thích USS (tùy chọn). Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P095. Nhấn P để nhập:

$$P095 = 0 \quad \begin{array}{l} \text{độ phân giải 0,1Hz} \\ \text{độ phân giải 0,01Hz} \end{array}$$

12. EEPROM điều khiển (tùy chọn). Nhấn P, nhấn phím mũi tên lên hoặc xuống cho đến khi hiển thị P971. Nhấn P để nhập:

$$P971 = 0: \text{ Thay đổi các thông số cài đặt (bao gồm cả P971) bị mất khi mất nguồn.}$$

1: Tham số cài đặt được lưu lại trong suốt thời gian mất nguồn.

13: Hiển thị vận hành. Nhấn P để thoát.

7.4.6. Kết nối và cài đặt MicroMaster Series 4 (MM4):

6.1. Kết nối MM4:

Để kết nối với MM4, ta sử dụng cáp RS-485 (nối trực tiếp S7-200 với MM4). Ngoài ra, còn có thể dùng cáp chuẩn PROFIBUS và các đầu nối để kết nối.

* Chú ý:

Các thiết bị kết nối với điện thế khác nhau có thể sẽ là nguyên nhân dẫn tới việc phát sinh dòng điện không mong muốn chạy trong cáp kết nối. Dòng điện này là nguyên nhân dẫn tới các lỗi truyền thông hoặc làm hỏng thiết bị.

Cần phải chắc chắn rằng tất cả các thiết bị được kết nối vào một cáp truyền thông đều có cùng dòng điện định mức hoặc được cách ly để ngăn ngừa dòng điện phát sinh không mong muốn.

Nếu S7-200 là điểm nút cuối trong mạng, hoặc nếu kết nối là điểm - điểm (point-to-point), cần phải sử dụng đầu A1 và B1 (không phải A2 và B2) của đầu cắm.

6.2. Cài đặt MM4:

Trước khi kết nối đến S7-200, cần phải chắc chắn rằng có đủ các thông số của MM. Sử dụng các keypad có sẵn trên biến tần để cài đặt như sau:

1. Reset để cài đặt lại cho hệ thống (tùy chọn):

P0010 = 30

P0970 = 1

Nếu bỏ qua bước này, các thông số tiếp theo sẽ được set theo các giá trị:

USS PZD length: P2012 Index0 = 2

USS PKW length: P2013 Index0 = 127

2. Cho phép truy nhập đọc/ghi các thông số:

P0003 = 3

3. Kiểm tra cài đặt thông số động cơ cho biến tần:

P0304 = điện áp động cơ (V)

P0305 = dòng điện động cơ (A)

P0307 = công suất động cơ (W)

P0310 = tần số động cơ (Hz)

P0311 = tốc độ động cơ (RPM)

Các thông số cài đặt này có thể thay đổi tùy thuộc vào loại động cơ được sử dụng.

Trước khi cài đặt các thông số P0304, P0305, P0307, P0310, P0311, cần thiết phải set thông số P0010 lên 1 trước. Sau khi kết thúc việc cài đặt, đặt thông số P0010 về 0. Các thông số P0304, P0305, P0307, P0310, P0311 chỉ có thể thay đổi trong chế độ quick commissioning.

4. Định chế độ điều khiển từ xa hay tại chỗ (Local/Remove):

P0700 Index0 = 5

5. Đặt lựa chọn tần số setpoint cho USS ở cổng COM

P1000 Index0 = 5

6. Định thời gian tăng tốc (tùy chọn), là thời gian để động cơ tăng tốc đến tốc độ max:

P1120 = 0 | 650,00 (s).

7. Định thời gian giảm tốc (tùy chọn), là thời gian để động cơ giảm dần tốc độ cho đến khi dừng:

P1121 = 0 | 650,00 (s).

- 8. Đặt tần số tham chiếu:
P2000 = 1 đến 650 Hz
- 9. Tiêu chuẩn hoá USS:
P2009 Index0 = 0
- 10. Đặt giá trị tốc độ baud cho chuẩn RS-485:
P2010 Index0 = 4 (2400 baud)
5 (4800 baud)
6 (9600 baud)
7 (19200 baud)
8 (38400 baud)
9 (57600 baud)
10 (115200 baud)

- 11. Nhập địa chỉ Slave:
P2011 Index0 = 0 đến 31

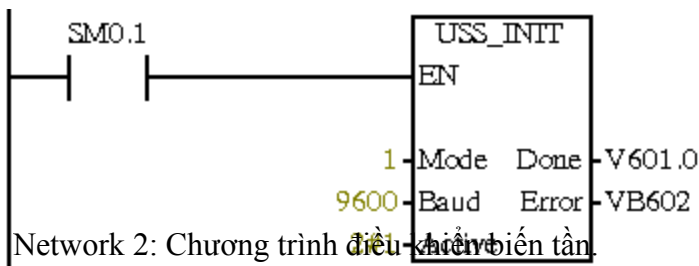
12. Đặt thời gian trễ giữa hai bức điện, đây là khoảng thời gian cho phép giữa hai lần truy nhập dữ liệu bức điện. Nó được sử dụng để cắt biến tần trong khoảng thời gian xảy ra lỗi truyền thông. Thời gian này tính từ lúc sau khi một dữ liệu hợp lệ của bức điện được nhận. Nếu có một dữ liệu không được nhận, biến tần sẽ ngắt và hiển thị mã lỗi F0070. Đặt giá trị 0 để ngừng điều khiển.

P2014 Index0 = 0 đến 65,535 ms

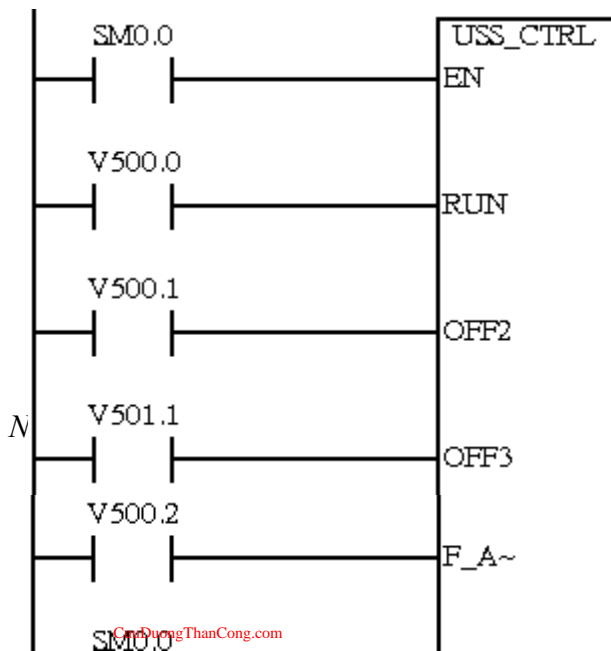
- 13. Chuyển dữ liệu từ RAM đến EEPROM:
P0971 = 1 (bắt đầu chuyển).

Lưu cài đặt sự thay đổi các thông số vào EEPROM.

7.4.7. Chương trình từ PLC để điều khiển và thu thập các thông số từ mạng biến tần
Network 1: Chương trình cài đặt tham số và truyền thông với biến tần Clanhke.



// 1_ chọn USS Protocol
 // 9600_ Tốc độ truyền giữa PLC và biến tần
 // 16#00000001_ Địa chỉ của biến tần
 // MB3_ Chứa kết quả của việc thực hiện lệnh (xem bảng mã lỗi)
 // M0.3_ bit báo trạng thái làm việc của lệnh USS_INIT



// V500.0_ Khởi động hoặc dừng động cơ.
 // V500.1=1_ Dừng động cơ với thời gian lâu hơn ở trường hợp OFF3.
 // F_ACKN Lỗi của biến tần sẽ được xoá khi tín hiệu đưa đến từ chân này tích cực.
 // SM0.0_ luôn bằng 0 có nghĩa là động cơ luôn quay thuận.
 // Drive = 0, địa chỉ của biến tần trong mạng.
 // Type = 0, kiểu biến tần là MM3, nếu bằng 1 thì tương ứng với MM4.
 // VD88_ giá trị setpoint tốc độ của động cơ (nằm trong giới hạn từ -200% đến +200%)

Chương trình cũng được tiến hành tương tự đối với n biến tần trong mạng USS, với $n = (0 \div 31)$, từ các bộ đếm, từ kép, từ đơn, byte và bit trạng thái chúng ta có thể dùng các phần mềm khác như VisualBasic, Delphi... để thiết kế giao diện, tạo lập cơ sở dữ liệu, nhật kí, cảnh báo để có thể hình thành mạng SCADA mini để điều khiển và giám sát hệ thống. Ngoài mạng biến tần dùng PLC để điều khiển, chúng ta cũng có thể tích hợp mạng bao gồm nhiều PLC và biến tần có sự điều khiển và giám sát từ PC. Để làm điều này rất khó khăn nếu như không có sự hỗ trợ của gói phần mềm Microcomputing của Siemens. Phần mềm này khi được cài đặt, nó sẽ có các tool, file .ocx, .dll rất hữu ích cho việc định nghĩa đường truyền, quét và cập nhật theo sự kiện, tạo lập giao diện, truy xuất dữ liệu từ PLC...Phần mềm này chỉ hỗ trợ cho VisualBasic.

7.5. Ứng dụng PLC trong hệ thống điều khiển giám sát:

Người biên soạn: Lâm Tăng Đức - Nguyễn Kim Ánh

Các bước tiến hành thiết lập hệ thống điều khiển và giám sát từ PC:

+ *Cấp trường*:

1. Thiết kế phần cứng dựa vào yêu cầu của quá trình công nghệ.
2. Thuật toán điều khiển cho mỗi trạm trong mạng.
3. Quy định địa chỉ của trạm, giao thức truyền thông, thuật toán truyền thông giữa các trạm trong mạng, quy định trạm master/slave nếu là trong mạng PPI; các trạm là ngang cấp nếu là mạng Freeport (giao thức này gọi là điều khiển công tự do_mỗi trạm đều là trạm chủ và cũng đều là trạm tớ). Đây là việc rất cần thiết trong quá trình tích hợp mạng.
4. Lập bảng quy định các địa chỉ, mã hiệu, lỗi... cần thiết cho việc truyền thông.
5. Viết chương trình điều khiển tương ứng cho từng trạm.

+ *Cấp điều khiển và giám sát*:

1. Cài đặt các phần mềm hỗ trợ như: Prosave, microcomputing...
2. Thiết kế giao diện HMI trên các mềm VisualBasic, Delphi...
3. Lập trình hướng đối tượng tương ứng với từng thiết bị trên sơ đồ công nghệ kết hợp với bảng quy định được tiến hành ở bước 4.
4. Thiết lập cơ sở dữ liệu, nhật kí vận hành, cảnh báo, báo cáo, vẽ đồ thị, truy xuất đến máy in nếu cần thiết.
5. Kiểm tra toàn bộ và tiến hành thủ tục kết nối.