

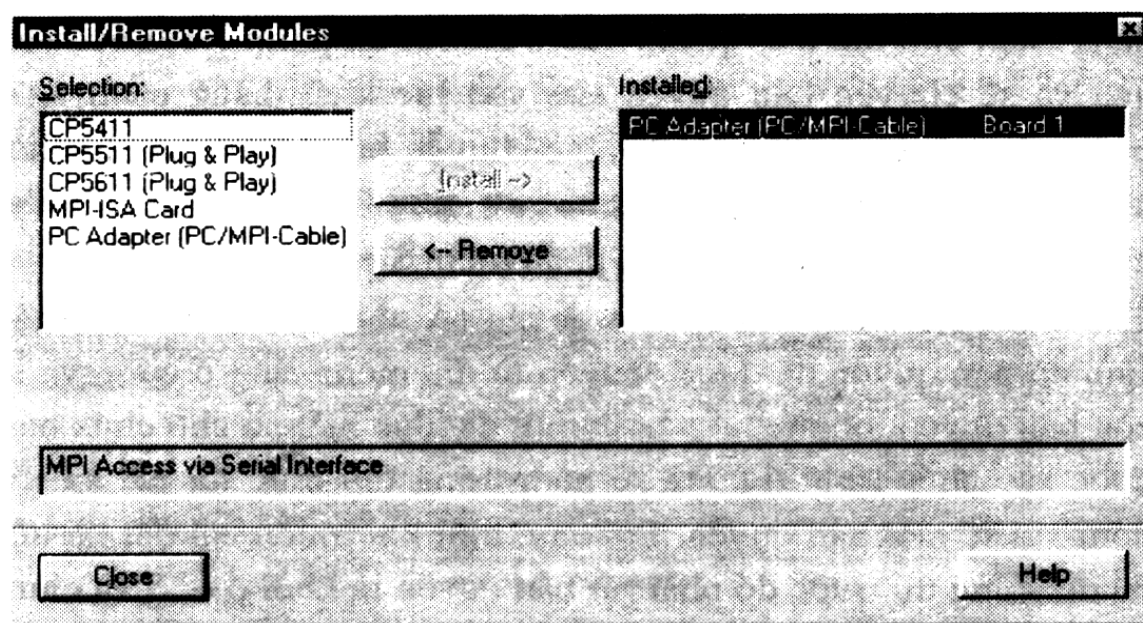
CHƯƠNG II: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

I- KẾT NỐI PC/PLC

1- Kết nối PC/PLC

a) Khi cài đặt STEP7

Khi cài đặt chương trình **Step7** vào PC, ta cần chọn sẵn các thiết bị phần cứng có khả năng sử dụng. Lúc này các chương trình của các thiết bị kết nối được cài đặt vào phần mềm của ta. Nói chung, nếu bộ nhớ của PC không bị hạn chế ta nên cài đặt tất cả các phần mềm này để tiện sử dụng về sau.



Hình 2-1 : Lựa chọn thiết bị giao tiếp khi cài đặt

b) Khi sử dụng

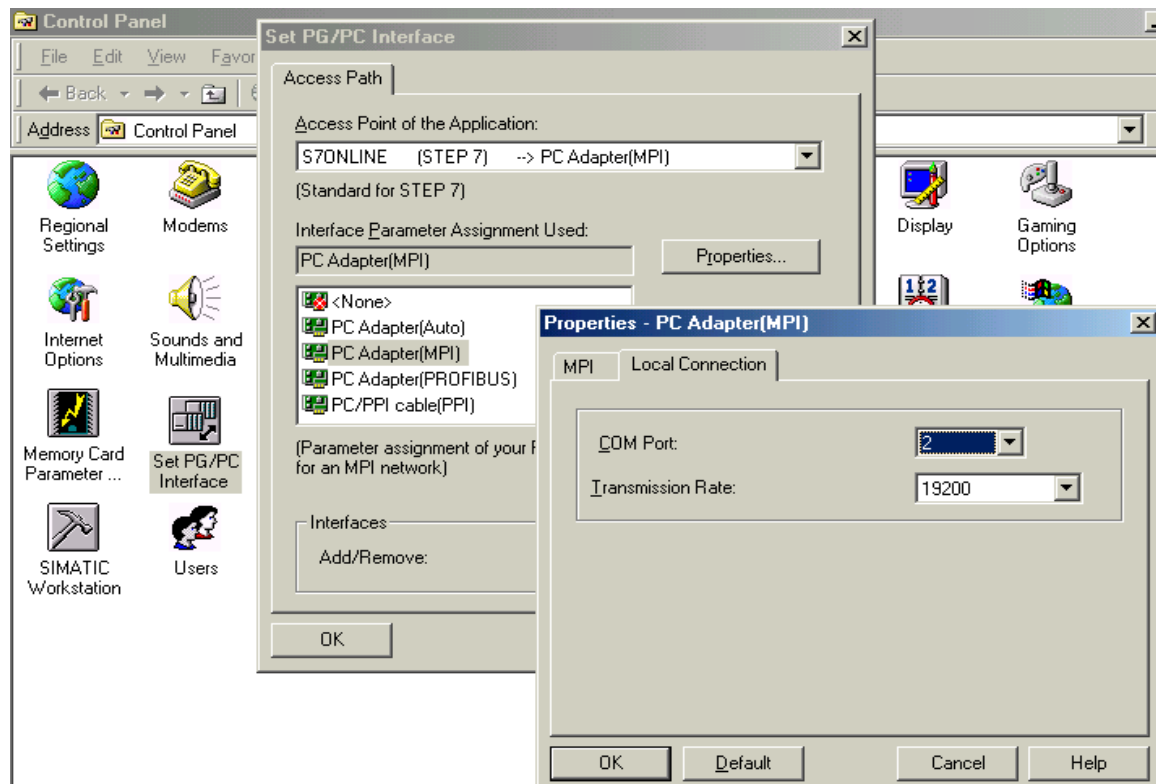
* Chọn thiết bị

- Từ menu **Start** chọn **Settings** → **Control panel** → **Set PC/PG Interface**.
- Trong hộp thoại được mở, chọn đúng thiết bị giao tiếp mà ta đang sử dụng.

* Gán tham số

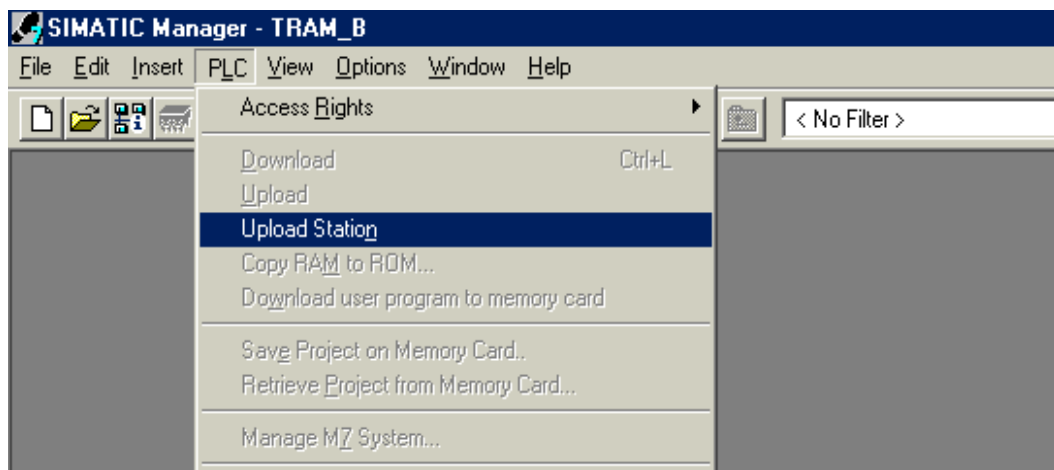
Chọn **Properties** để xác định tham số. Ta chọn đúng cổng nối tiếp của PC mà ta nối đến **PCAdapter**. Tốc độ truyền được chọn tương ứng với tốc độ truyền của **PC Adapter**. (Thường là 187,5KB)

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-2 : Gán tham số cho thiết bị giao tiếp

2- Upload



Hình 2-3 : Upload một trạm

- Upload cho phép ta đọc lại cấu hình và chương trình đã nạp vào PLC.
- Từ menu lệnh, ta chọn lệnh **Upload Station**

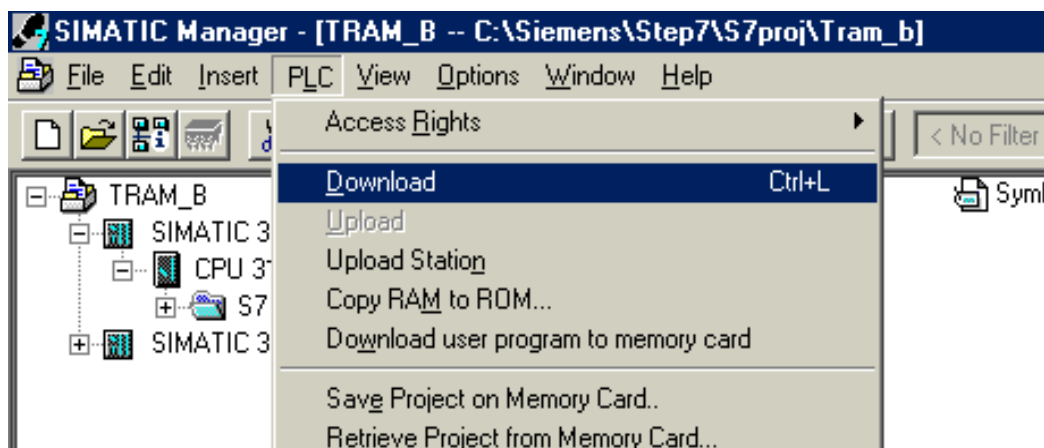
CHẨN ĐOÁN VÀ XỬ LÝ LỖI PLC S7 – 300

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

- Trạm PLC thực hoặc giả lập (PLCSIM) sẽ được chọn, kể cả chương trình đã nạp cho nó, và địa chỉ MPI của trạm.
- Với PLC thực, việc Upload chỉ đọc được kiểu các module I/O, không biết được chính xác loại I/O thực tế của nó. Tuy nhiên, nếu trước đó cấu hình chính xác kiểu loại các module SM đã được Dowload xuống bộ nhớ PLC và bộ nhớ không bị mất dữ liệu, khi Upload ta sẽ đọc được đầy đủ kiểu loại của các Module I/O.
- Khi làm việc với PLCSIM, chỉ với loại CPU315-2DP ta mới có thể Upload cấu hình I/O mở rộng (các module SM), các loại CPU khác ta chỉ có thể Upload kiểu loại CPU, địa chỉ MPI.

3- Download

- **Download** nạp các thông số cấu hình (Ví dụ các tham số hoạt động của các module) và các khối chương trình cho PLC.
- Khi Download cần phải chọn đúng địa chỉ MPI.
- Muốn Download địa chỉ MPI cho CPU phải tiến hành riêng cho từng trạm PLC, khi cấu hình phần cứng.
- CPU cần phải ở trạng thái STOP hay RUN-P
- Nên xóa bộ nhớ trước khi Download để tránh các khối logic cũ còn lại.
- Với PLCSIM, cấu hình I/O phần cứng không thể download và upload, ngoại trừ với CPU loại CPU 315_2DP, CPU316_2DP, CPU318-2.



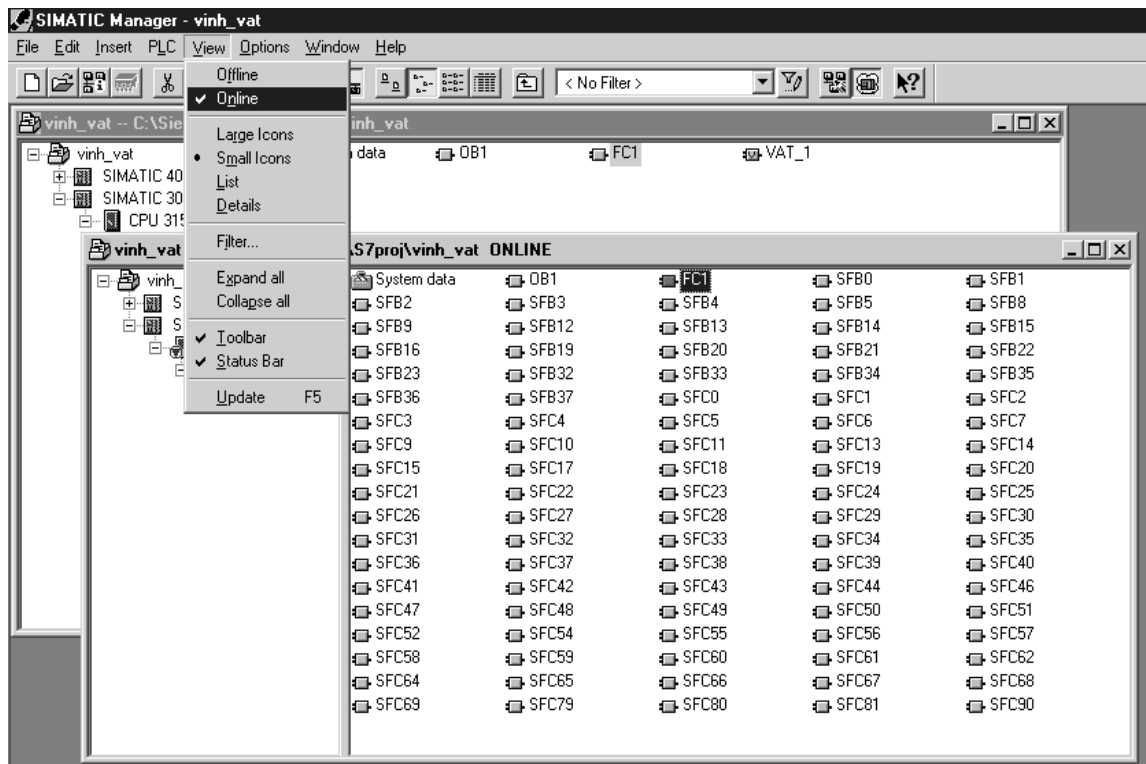
Hình 2-4: Download một trạm

4- Kết nối ONLINE

- *Kết nối **ONLINE** cho phép PC liên kết với PLC để theo dõi trạng thái hoạt động của PLC.
- * Hai cửa sổ trên PC cho ta quan sát cấu hình, các khối logic trên PC và trên PLC.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

* Kết nối ONLINE có thể được sử dụng để download, upload các khối logic chương trình và theo dõi hoạt động của PLC.



Hình 2-5 : Kết nối ONLINE

III- SỬ DỤNG CHỨC NĂNG DEBUG

1- Công dụng

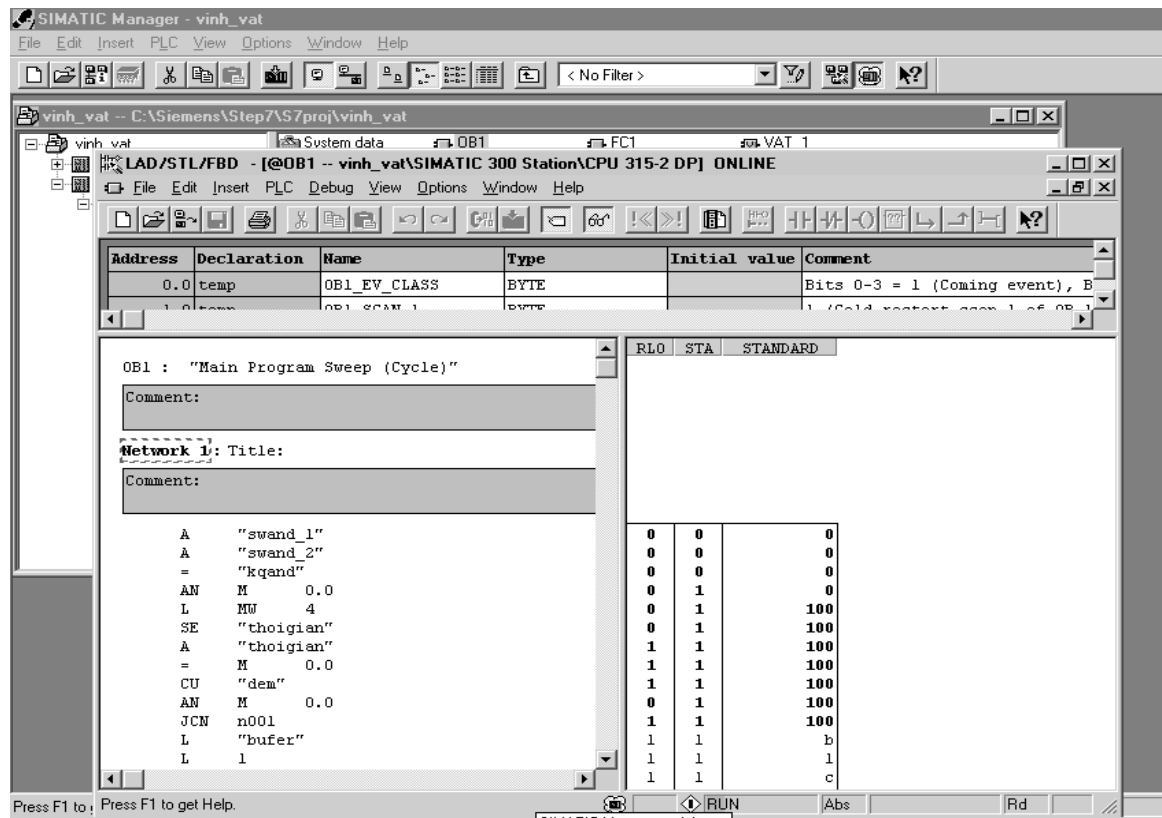
* Chức năng DEBUG được sử dụng để theo dõi (**Monitor**) hoạt động của chương trình trong PLC thực hay PLCSIM. Ta cần mở từng khối logic (OB, FC, FB) và theo dõi từng kết quả thực hiện theo từng dòng lệnh , trong STL, LAD hay FBD.

* Khi sử dụng ngôn ngữ STL, trong cửa sổ trạng thái ta có thể theo dõi giá trị RLO, STA, ACCU1 (STANDARD) , ACCU2, AR1, AR2, STATUS WORD, DB, DI với các kiểu dữ liệu BOOL, DECIMAL, HEX, REAL. Các giá trị được thể hiện tương ứng với từng dòng lệnh.

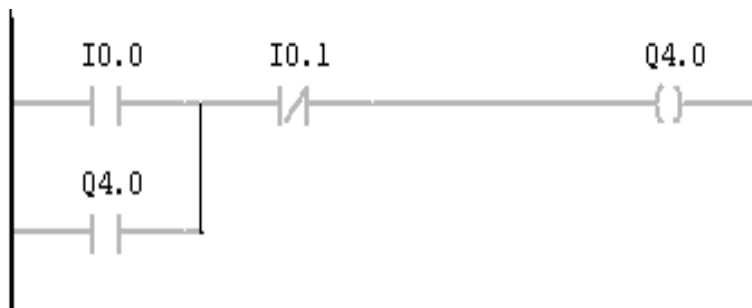
* Khi sử dụng ngôn ngữ LAD, hình ảnh, màu sắc của hình vẽ cho ta biết trạng thái hoạt động của chương trình.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

* Khi sử dụng chức năng DEBUG, ta có thể theo dõi hoạt động của chương trình, nhờ đó ta có thể sửa đổi các lỗi logic để chương trình hoạt động đúng như mong muốn.



Hình 2-6 : Chức năng DEBUG(STL)



Hình 2-7: Chức năng DEBUG (LAD)

CHẨN ĐOÁN VÀ XỬ LÝ LỖI PLC S7 – 300

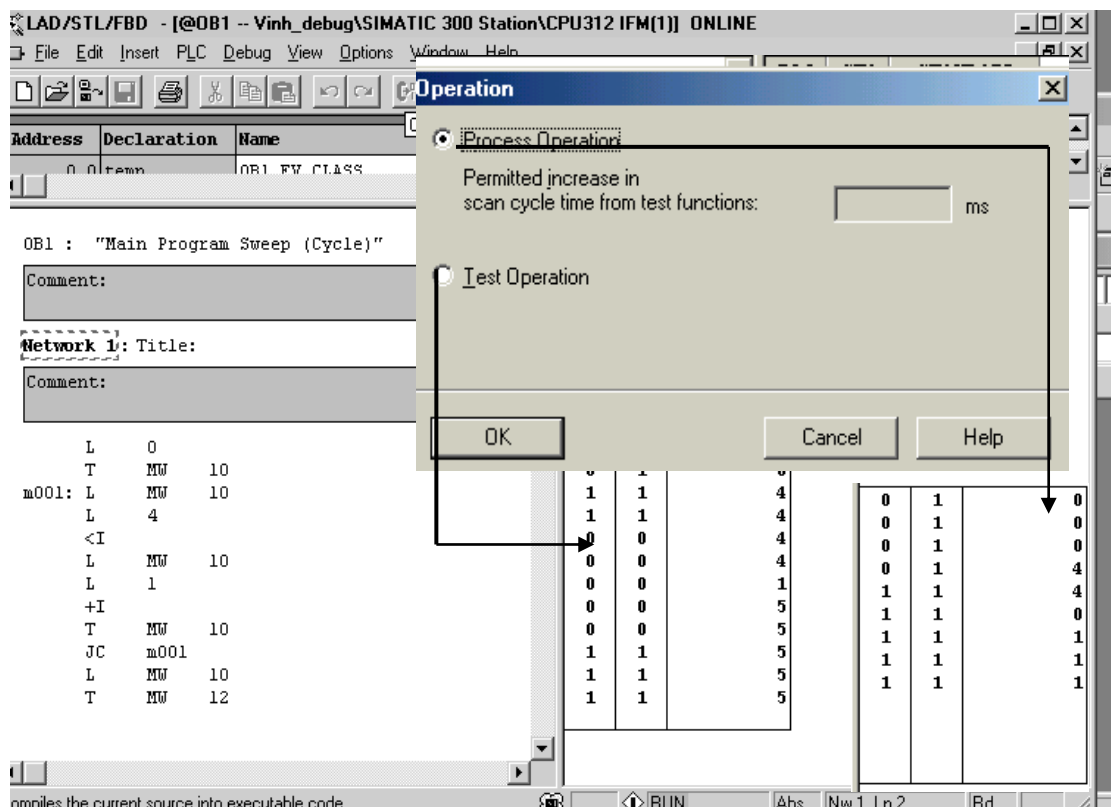
CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

2- Chọn chế độ :

Trong chức năng Debug ta có thể sử dụng các chế độ khác nhau .

* Khi sử dụng PLCSIM , chọn bằng menu **DEBUG > OPERATION**.

* Với PLC thực, việc chọn chế độ thực hiện khi xác lập cấu hình phần cứng và download cho CPU.



Hình 2-8 : Chọn chế độ DEBUG

* Trong chức năng **Process Operation** các vòng lặp chỉ thể hiện kết quả vòng đầu tiên trên cửa sổ DEBUG. Trong chế độ này có thể tăng chu kỳ quét của CPU (scan cycle) , thực hiện trong cấu hình phần cứng.

* Trong chức năng **Test Operation** trạng thái của vòng lặp được theo dõi sau khi thực hiện tất cả các lần quét.

Trong hình 2-8 ta thấy chương trình thực hiện 5 vòng (MW10 lấy các giá trị từ 0 đến 4) . Khi chọn chức năng **Process Operation**, ta chỉ thấy hoạt động qua vòng lặp đầu được thể hiện (MW10 = 1).

3- Đặt điểm dừng Breakpoint

* Điểm dừng xác định dòng lệnh mà khi thực hiện đến đó, CPU sẽ chuyển sang chế độ treo HOLD. Ta có thể đặt nhiều điểm dừng và cho CPU chạy từng đoạn (**Resume**) từ điểm dừng này đến điểm dừng tiếp theo, hoặc cho CPU thực hiện

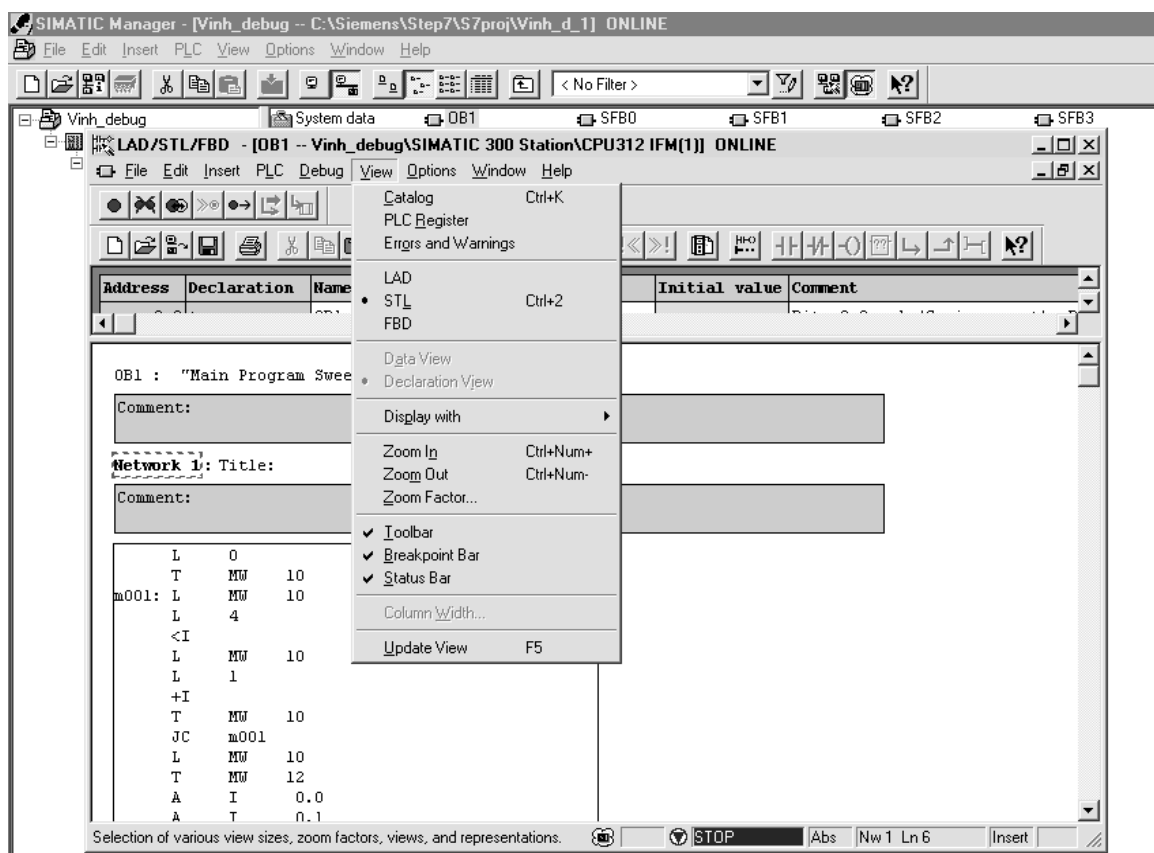
CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

từng lệnh (**Next Statement**) .

Ta có thể đặt điểm dừng, xóa điểm dừng theo yêu cầu kiểm tra.

* Để đặt được điểm dừng, ta cần chọn chế độ hoạt động của DEBUG là **OPERATION >TEST OPERATION**

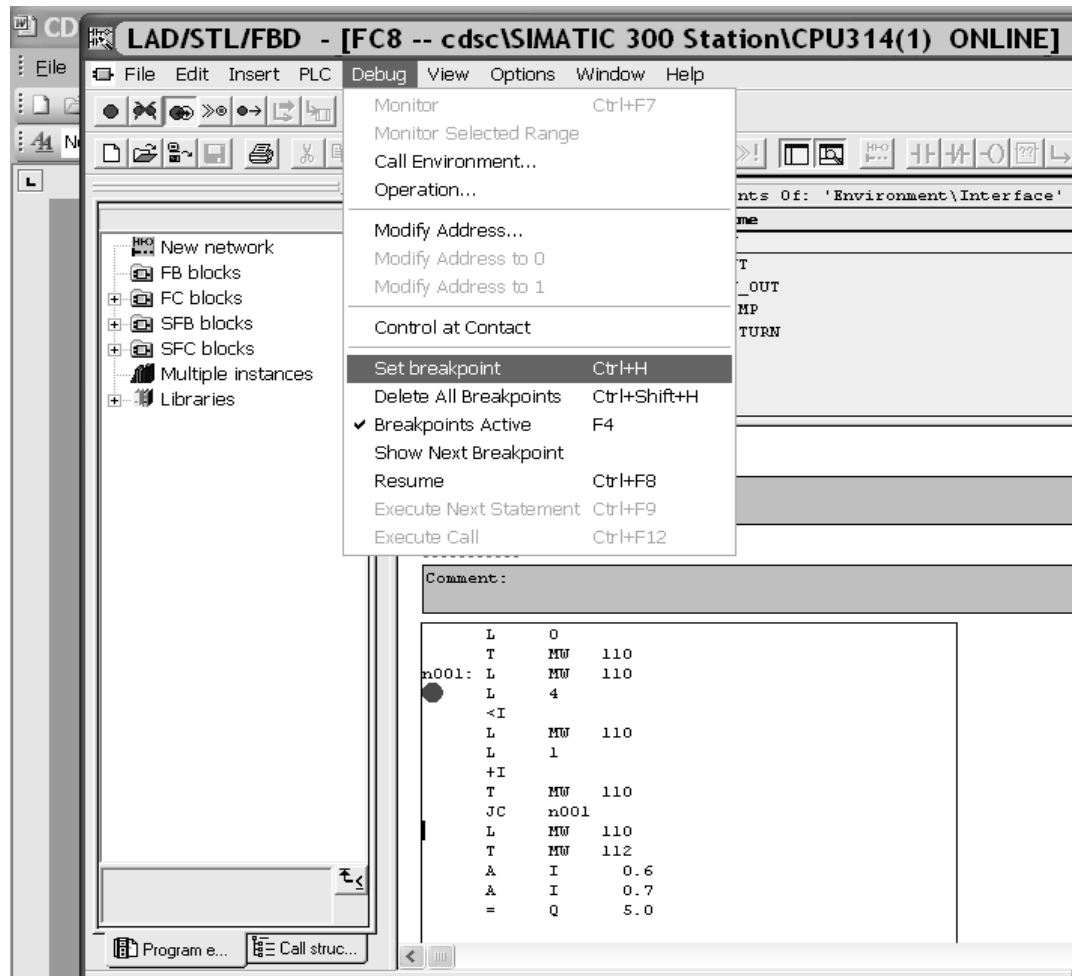
* Để chọn điểm dừng, từ Menu Manager ta chọn **VIEW>ONLINE**. Trong cửa sổ ONLINE ta mở khối cần theo dõi, trong menu VIEW chọn **Breakpoint Bar**, ta sẽ có thanh lệnh cho phép chọn các lệnh đặt điểm dừng, xóa điểm dừng, cho tác động. Ta cũng có thể chọn lệnh đặt điểm dừng hoặc xóa điểm dừng từ menu **Debug**.



Hình 2-9 : Mở thanh lệnh chọn điểm dừng

- * Khi cần đặt điểm dừng, ta cần phải :
 - Chọn dòng lệnh mà CPU cần dừng
 - Cho chế độ điểm dừng hoạt động (Active)
 - Dùng lệnh **Resume** để CPU thực hiện đoạn lệnh tiếp theo, đến điểm dừng mới.
- * CPU cần ở chế độ RUN_P.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-10: Đặt điểm dừng cho DEBUG

4-Sử dụng điểm trigger:

*Nếu một hàm được gọi nhiều lần, việc Debug nó có thể được chọn theo lần gọi mà ta muốn quan sát. Lúc đó ta chọn mode của **Debug** là **Test Operation**. Trong menu Debug gọi lệnh **Call Enviroment** và chọn đường dẫn gọi hàm (3 lần cuối). Ta có thể xem lược đồ cấu trúc gọi hàm để chọn điểm trigger.

Ví dụ :

*Trong OB1:

```
CALL FC 1
CALL FC 3
AN "goi1"
BEC
CALL FC 2
```

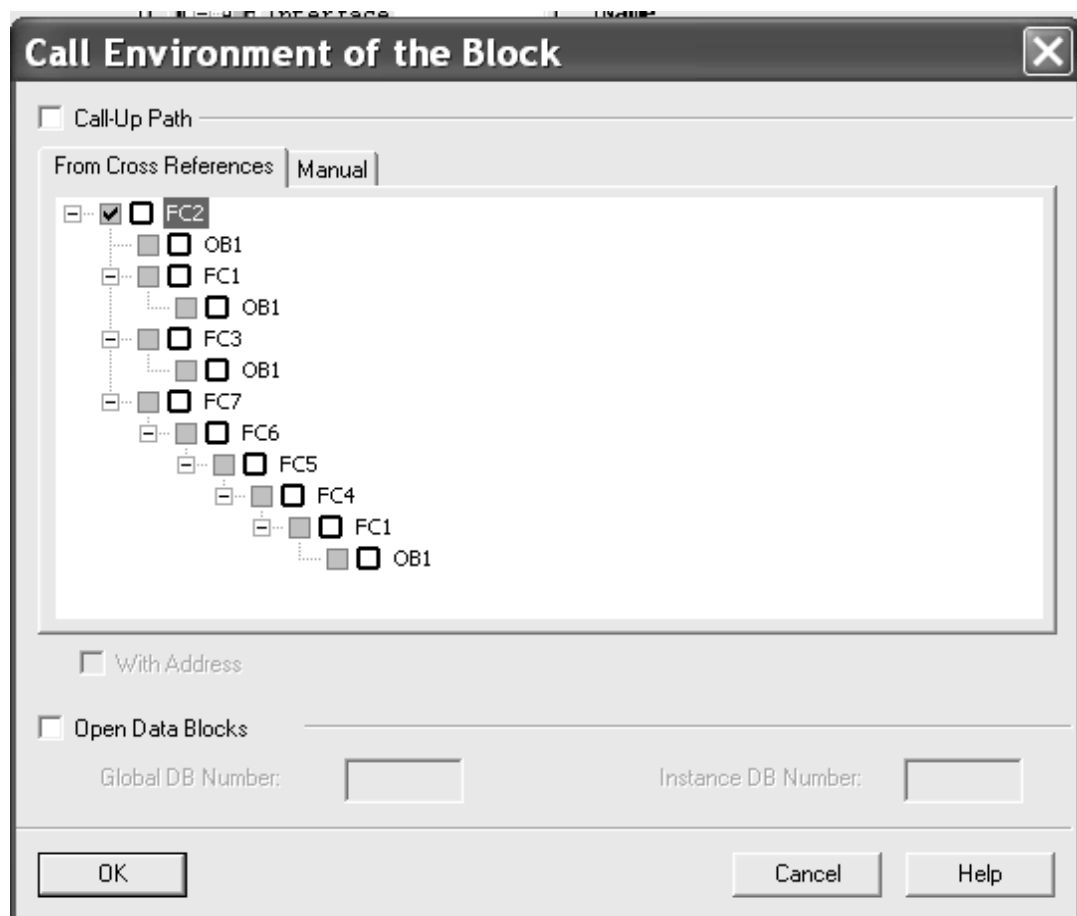

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

```
so1 := "so1_1"
so2 := "so2_1"
ketqua:="ketqua1"
*Trong FC 1:
CALL FC 4
AN "goi2"
BEC
CALL FC 2
so1 := "so1_2"
so2 := "so2_2"
ketqua:="ketqua2"
*Trong FC3 :
CALL FC 2
so1 := "so1_3"
so2 := "so2_3"
ketqua:= "ketqua3"
*Trong FC 4:
CALL FC 5
*Trong FC 5:
CALL FC 6
*Trong FC 6 :
CALL FC 7
*Trong FC 7:
CALL FC 2
so1 := "so1_4"
so2 := "so2_4"
ketqua:= "ketqua4"
```

*Mở chức năng DEBUG cho hàm FC2, chọn **Debug> Operation > Test Operation**. Sau đó quay về chọn **Debug > Call Enviroment** , ta có các đường dẫn gọi hàm, chọn đường dẫn muốn Debug. Sau đó chọn **Debug> Monitor**

*Lúc này kết quả của DEBUG thể hiện theo lần gọi mà ta đã chọn.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-11 : Sơ đồ cấu trúc chương trình

IV- SỬ DỤNG BẢNG BIẾN VAT

1- Công dụng

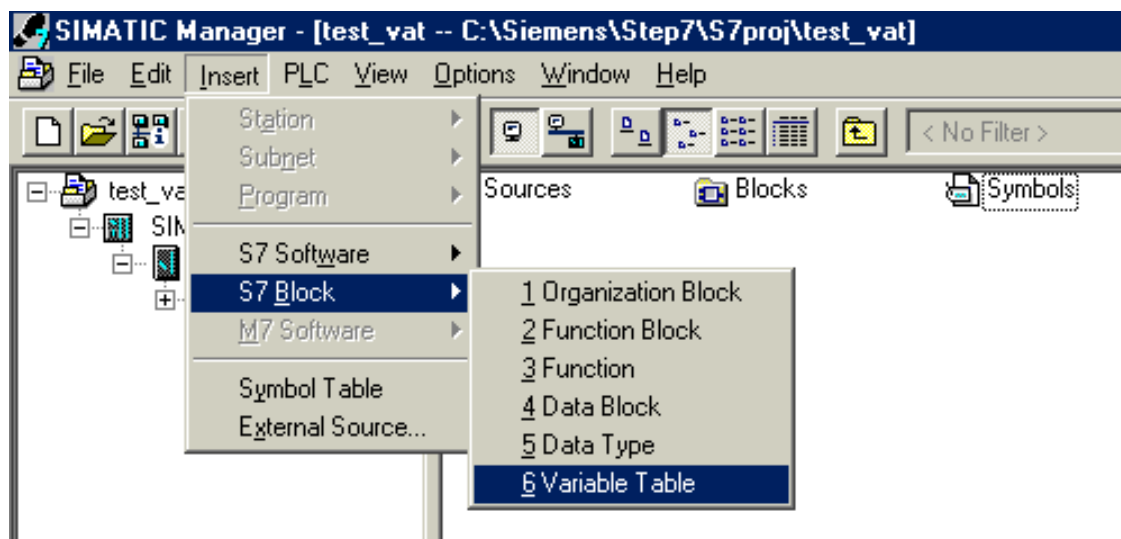
* Sử dụng bảng biến cho phép theo dõi giá trị các biến cần thiết trong quá trình hoạt động của PLC, qua đó ta có thể xác định chương trình hoạt động có đúng ý muốn hay không. Với các chương trình lớn, việc theo dõi theo DEBUG gặp khó khăn, sử dụng VAT cho ta khả năng theo dõi các biến chủ yếu.

* Việc sử dụng bảng biến có thể thực hiện với PLC thực hay PLCSIM.

2- Tạo bảng biến VAT

* Từ **SIMATIC Manager** ta có thể mở bảng biến.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-12: Mở bảng biết VAT

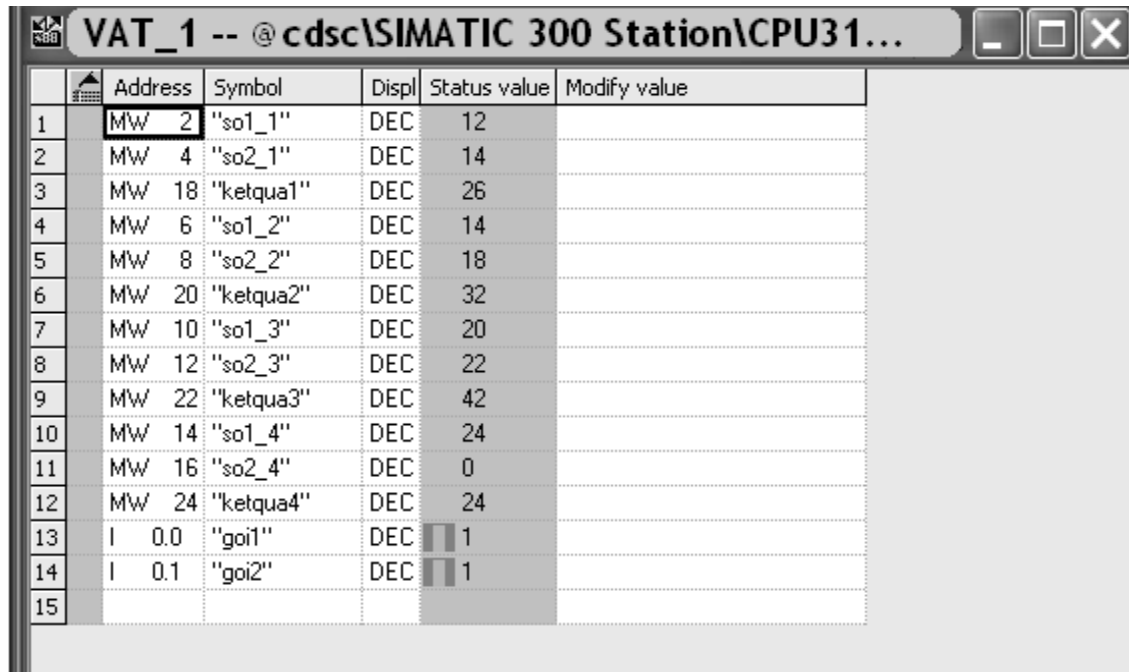
* Trong bảng biến ta điền các biến (địa chỉ hoặc ký hiệu, nội dung còn lại S7 tự động điền theo bảng ký hiệu **Symbol** đã có trong chương trình)

| | Address | Symbol | Displ | Status value | Modify value |
|----|---------|-----------|-------|--------------|--------------|
| 1 | MW 2 | "so1_1" | DEC | | |
| 2 | MW 4 | "so2_1" | DEC | | |
| 3 | MW 18 | "ketqua1" | DEC | | |
| 4 | MW 6 | "so1_2" | DEC | | |
| 5 | MW 8 | "so2_2" | DEC | | |
| 6 | MW 20 | "ketqua2" | DEC | | |
| 7 | MW 10 | "so1_3" | DEC | | |
| 8 | MW 12 | "so2_3" | DEC | | |
| 9 | MW 22 | "ketqua3" | DEC | | |
| 10 | MW 14 | "so1_4" | DEC | | |
| 11 | MW 16 | "so2_4" | DEC | | |
| 12 | MW 24 | "ketqua4" | DEC | | |
| 13 | I 0.0 | "goi1" | DEC | | |
| 14 | I 0.1 | "goi2" | DEC | | |
| 15 | | | | | |

Hình 2-13: Điền các biến cần theo dõi vào bảng

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

2- Theo dõi hoạt động



The screenshot shows a window titled "VAT_1 -- @cdsc\SIMATIC 300 Station\CPU31...". It contains a table with the following columns: Address, Symbol, Displ, Status value, and Modify value. The table lists 15 rows of data, including memory words (MW) and inputs (I).

| | Address | Symbol | Displ | Status value | Modify value |
|----|---------|-----------|-------|--------------|--------------|
| 1 | MW 2 | "so1_1" | DEC | 12 | |
| 2 | MW 4 | "so2_1" | DEC | 14 | |
| 3 | MW 18 | "ketqua1" | DEC | 26 | |
| 4 | MW 6 | "so1_2" | DEC | 14 | |
| 5 | MW 8 | "so2_2" | DEC | 18 | |
| 6 | MW 20 | "ketqua2" | DEC | 32 | |
| 7 | MW 10 | "so1_3" | DEC | 20 | |
| 8 | MW 12 | "so2_3" | DEC | 22 | |
| 9 | MW 22 | "ketqua3" | DEC | 42 | |
| 10 | MW 14 | "so1_4" | DEC | 24 | |
| 11 | MW 16 | "so2_4" | DEC | 0 | |
| 12 | MW 24 | "ketqua4" | DEC | 24 | |
| 13 | I 0.0 | "goi1" | DEC | 1 | |
| 14 | I 0.1 | "goi2" | DEC | 1 | |
| 15 | | | | | |

Hình 2-14: Theo dõi giá trị các biến khi PLC hoạt động

*Mở bảng biến và chọn menu **Variable > Monitor** (PC đã được liên kết với PLC thật hay PLCSIM). Lúc đó các giá trị trạng thái Status Value phản ánh trạng thái và kết quả hoạt động của CPU.

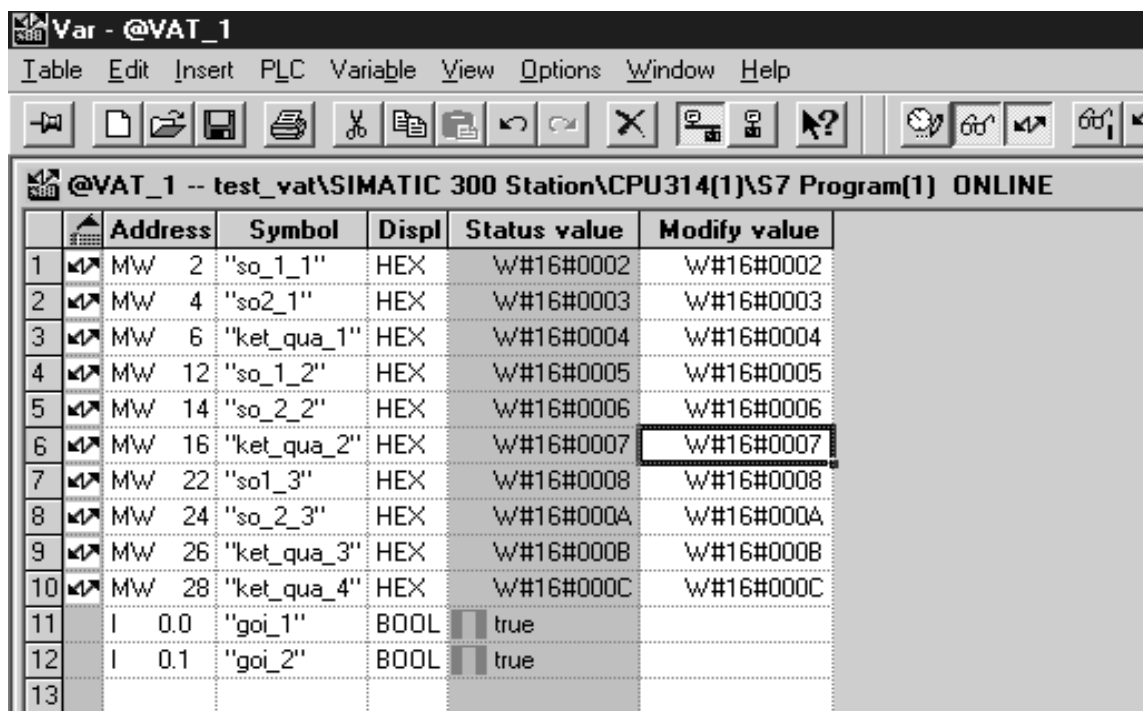
* Nếu không sử dụng chức năng **Monitor**, biến không được kiểm tra. Nếu sử dụng chức năng **Update Monitor Value**, ta có thể hiển thị bảng biến một lần và tức thời.

*Căn cứ vào giá trị các biến, ta biết chương trình có hoạt động đúng dự kiến hay không.

3- Nạp giá trị cho biến (Modify)

* **Modify** các biến là đặt giá trị mà ta muốn gán cho biến. Điều này chỉ thực hiện cho các biến điều khiển (đọc vào), với các biến ra, khi CPU hoạt động nó sẽ bị ghi đè bằng kết quả xử lý của chương trình.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



| | Address | Symbol | Displ | Status value | Modify value |
|----|---------|-------------|-------|--------------|--------------|
| 1 | MW 2 | "so_1_1" | HEX | W#16#0002 | W#16#0002 |
| 2 | MW 4 | "so_2_1" | HEX | W#16#0003 | W#16#0003 |
| 3 | MW 6 | "ket_qua_1" | HEX | W#16#0004 | W#16#0004 |
| 4 | MW 12 | "so_1_2" | HEX | W#16#0005 | W#16#0005 |
| 5 | MW 14 | "so_2_2" | HEX | W#16#0006 | W#16#0006 |
| 6 | MW 16 | "ket_qua_2" | HEX | W#16#0007 | W#16#0007 |
| 7 | MW 22 | "so_1_3" | HEX | W#16#0008 | W#16#0008 |
| 8 | MW 24 | "so_2_3" | HEX | W#16#000A | W#16#000A |
| 9 | MW 26 | "ket_qua_3" | HEX | W#16#000B | W#16#000B |
| 10 | MW 28 | "ket_qua_4" | HEX | W#16#000C | W#16#000C |
| 11 | I 0.0 | "goi_1" | BOOL | true | |
| 12 | I 0.1 | "goi_2" | BOOL | true | |
| 13 | | | | | |

Hình 2-15: Nạp giá trị cho từng biến

* Chế độ hoạt động của CPU phải là **RUN-P** để các giá trị từ PC có thể ghi vào CPU.

* Nếu không sử dụng chức năng **Modify**, ta có thể sử dụng chức năng **Activate Modify Value** cho mỗi lần gán giá trị. Khi đang sử dụng chức năng **Modify** ta không thể gán giá trị mới cho biến, còn khi sử dụng chức năng **Activate Modify Value** ta có thể gán giá trị cho biến.

* Chỉ những biến trong vùng nhìn thấy được **Modify**.

4- Chọn điểm trigger :

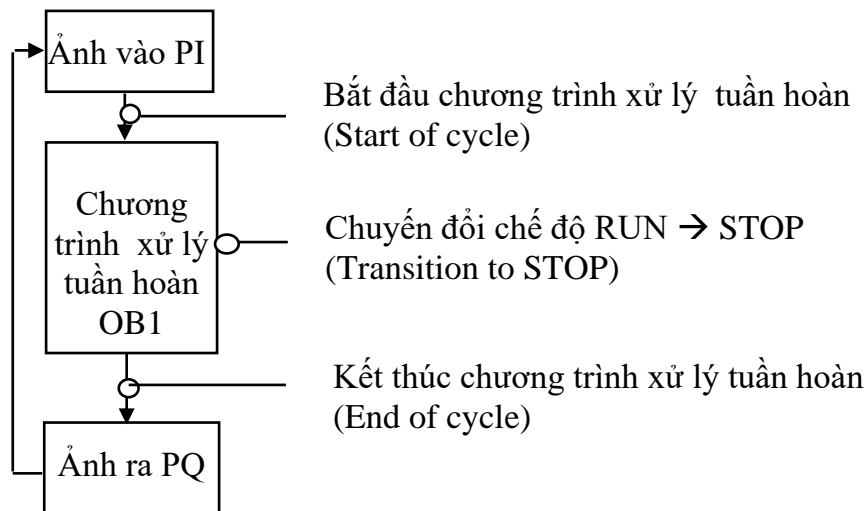
Lệnh Trigger chọn điểm cập nhật dữ liệu modify của biến vào chương trình .

- **Start of Cycle** : cập nhật sau khi đọc đầu vào từ PII, dữ liệu sẽ được viết đè lên ảnh đầu vào và dùng để xử lý.

- **End of cycle** : cập nhật trước khi truyền ra PIQ, viết đè lên kết quả đã xử lý của CPU

- **Transition to stop** : cập nhật lúc CPU chuyển từ trạng thái RUN sang trạng thái STOP.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-16 : Chọn điểm trigger

* Chọn kiểu **trigger** : Ta có thể cập nhật giá trị **Modify** cho các biến một lần hay liên tục qua các lần quét.

- Cập nhật giá trị một lần, với các chu kỳ tiếp sau giá trị của biến được ghi theo kết quả xử lý của chương trình.

Ví dụ :

Chương trình trong OB1 :

```
AN    M      0.0
L      SST#1S
SE     T      1
A      T      1
=      M      0.0
A      M      0.0
BEC
L      MW     100
INC    1
T      MW     100
BE
```

Đoạn chương trình trên, biến **MW100** được tăng thêm **1** sau mỗi giây. Nếu ta nạp cho biến **MW100** theo kiểu 1 lần (**Trigger condition for modifying > once**) **MW100** nhận giá trị mới, sau đó biến **MW100** tiếp tục tăng thêm **1** sau mỗi giây tiếp theo. Nếu ta nạp biến theo kiểu qua mỗi lần quét (**Trigger condition for modifying > Every cycle**) biến **MW100** lấy giá trị nạp sau mỗi chu kỳ quét, không thay đổi qua mỗi chu kỳ của T1.

5- Đặt giá trị cho biến ra trong chế độ STOP

Trong chế độ STOP của CPU, ta vẫn có thể đặt giá trị đầu ra cho biến. Điều này giúp ta kiểm tra riêng rẽ hoạt động của một thiết bị nối với đầu ra CPU (thiết bị chấp hành). Chức năng này thực hiện với PLC thực.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

* Mở bảng biến, nối với PLC : **PLC > Connect to > [Chọn CPU đã được cấu hình trong phần cứng của chương trình (Configured), CPU có thể kết nối (Accessible) hay CPU kết nối trực tiếp đến PC(Direct)]**

* Chọn chế độ làm việc của PLC là STOP.

* Nạp giá trị đầu ra mong muốn cho biến ngoại vi (PQB, PQW, PQD) trong cột **Modify Value**.

* Chọn menu lệnh **Variable > Enable Peripheral Outputs**

* Nạp giá trị cho biến bằng lệnh **Variable > Activate Modify Values**.

* Chức năng này bị xóa khi bỏ dấu chọn ở menu lệnh **Variable > Enable Peripheral Outputs**

V-CHỨC NĂNG FORCE

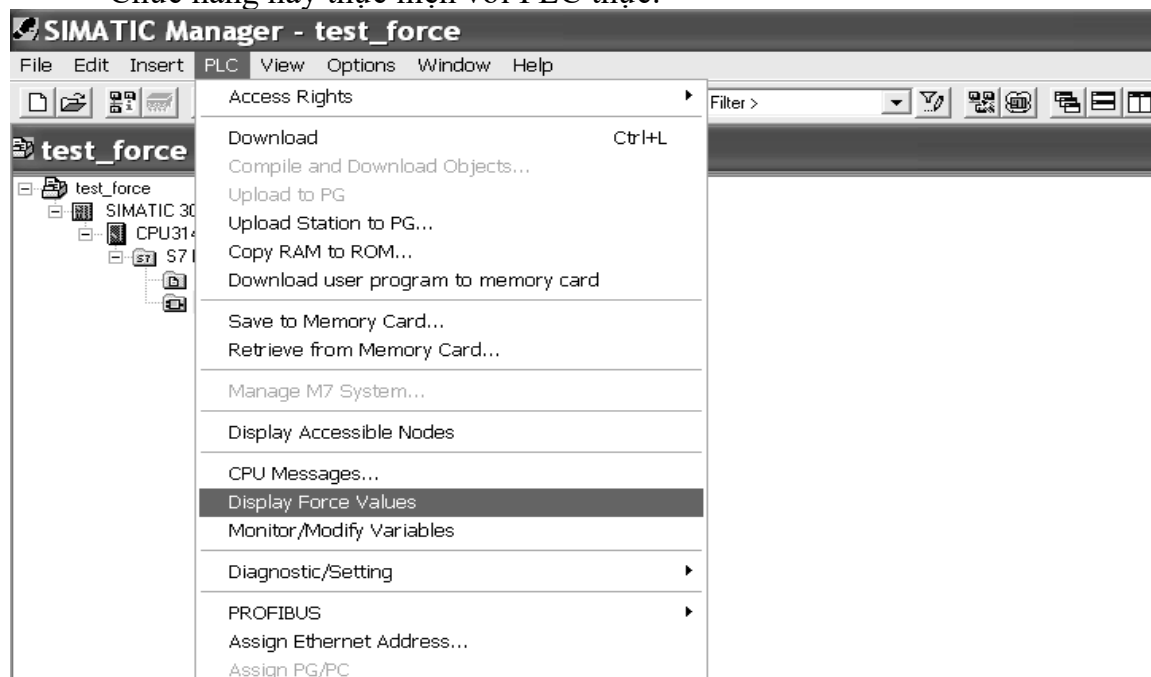
1- Công dụng

*Người ta dùng chức năng FORCE để gán cho các biến vào ra các giá trị cố định nhằm thử hoạt động hệ thống.

*Cần phải rất thận trọng khi sử dụng chức năng này vì dễ gây nguy hiểm cho người sử dụng hoặc thiết bị.

*Không phải mọi PLC đều có chức năng này.

* Chức năng này thực hiện với PLC thực.



Hình 2-17: Chọn hiển thị giá trị Force

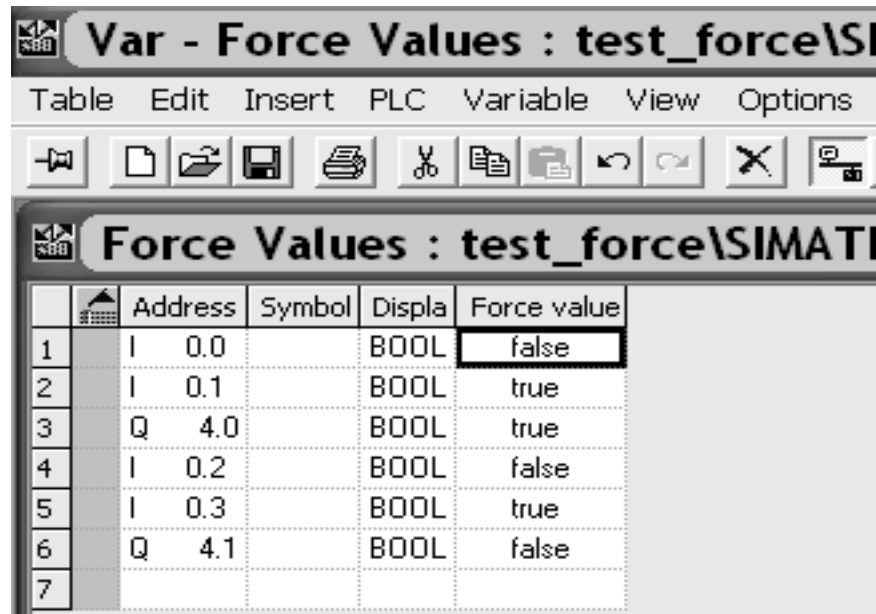
2- Tạo bảng biến FORCE

* Từ menu PLC, ta mở **Display Force Values** để hiển thị bảng giá trị cưỡng

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

bức (**force**).

*Trong bảng biến **Force**, ta điền các biến cần cưỡng bức. Với CPU S7-300, chỉ có các biến **I, Q** được FORCE. Với CPU S7-400, ta còn có thể cưỡng bức các biến M, PIW, PQW.

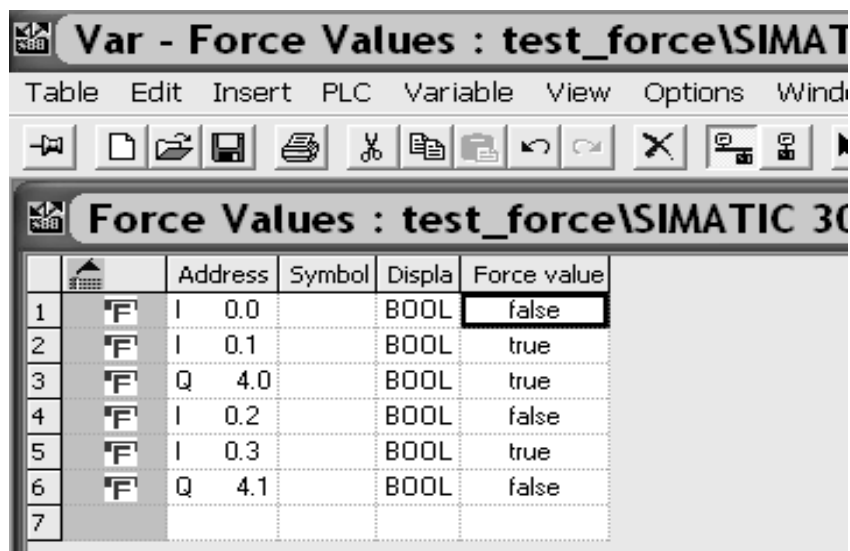


| | Address | Symbol | Displa | Force value |
|---|---------|--------|--------|-------------|
| 1 | I 0.0 | | BOOL | false |
| 2 | I 0.1 | | BOOL | true |
| 3 | Q 4.0 | | BOOL | true |
| 4 | I 0.2 | | BOOL | false |
| 5 | I 0.3 | | BOOL | true |
| 6 | Q 4.1 | | BOOL | false |
| 7 | | | | |

Hình 2-18 : Điền và đặt giá trị cưỡng bức cho biến

3- Thực thi lệnh FORCE

* Từ menu lệnh **Variable** chọn **Force**, lệnh được thực thi



| | Address | Symbol | Displa | Force value |
|---|---------|--------|--------|-------------|
| 1 | I 0.0 | | BOOL | false |
| 2 | I 0.1 | | BOOL | true |
| 3 | Q 4.0 | | BOOL | true |
| 4 | I 0.2 | | BOOL | false |
| 5 | I 0.3 | | BOOL | true |
| 6 | Q 4.1 | | BOOL | false |
| 7 | | | | |

Hình 2-21: Các biến đã bị cưỡng bức theo giá trị đặt

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

*Giá trị có thể FORCE là **I và Q**. Thực tế, khi FORCE một đầu vào I, giá trị PIW vẫn phụ thuộc tín hiệu từ bên ngoài nhưng ảnh I không phụ thuộc giá trị biến ngoại vi. Còn với đầu ra Q, biến ngoại vi nhận giá trị cưỡng bức còn giá trị ảnh Q phụ thuộc vào xử lý của CPU.

VI- ĐỌC THÔNG TIN TỪ PLC

1- Công dụng

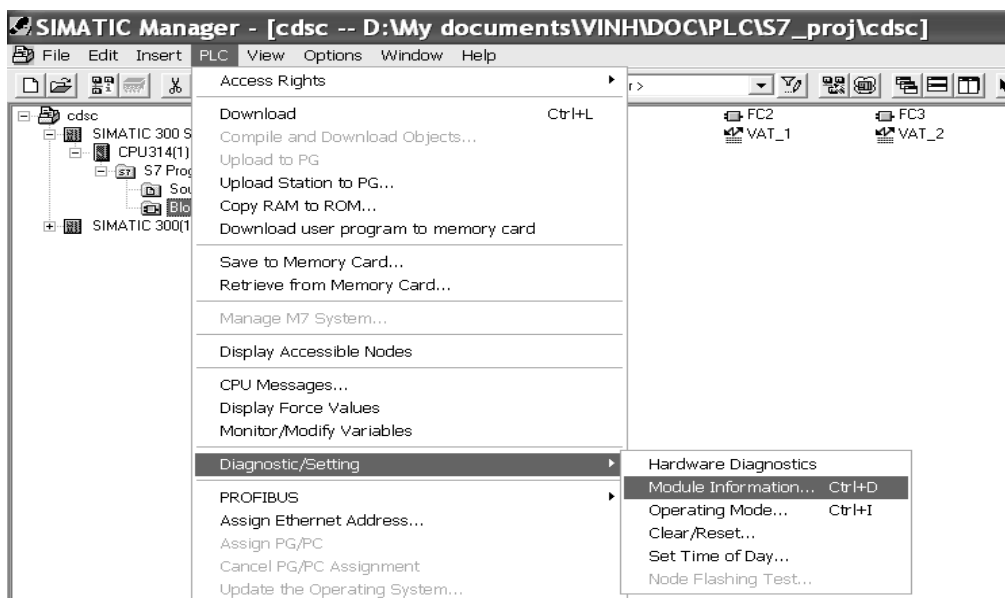
* Các thông tin từ PLC cho phép ta tìm ra hầu hết các lỗi của chương trình. Các thông tin này được lưu giữ trong PLC ngay cả khi ta xóa bộ nhớ của nó.

* Khi PLC có lỗi, nó có thể chuyển sang trạng thái STOP hay không tùy thuộc loại lỗi và chương trình đã nạp cho PLC (Các khối OB). Khi viết chương trình xử lý lỗi, ta cũng có thể đưa ra các thông báo sơ bộ về lỗi qua các cổng ra. Tuy nhiên số cổng ra không thể quá nhiều nên lượng thông tin này cũng hạn chế. Nếu ta kết nối PLC với PC ngay cả khi nó đang ở trạng thái STOP hoặc sau khi bị mất điện, ta có thể đọc được các thông tin cần thiết từ PLC.

* Việc xử lý các thông tin cho phép ta biết rõ lỗi đã xảy ra.

2- Các thông tin từ PLC

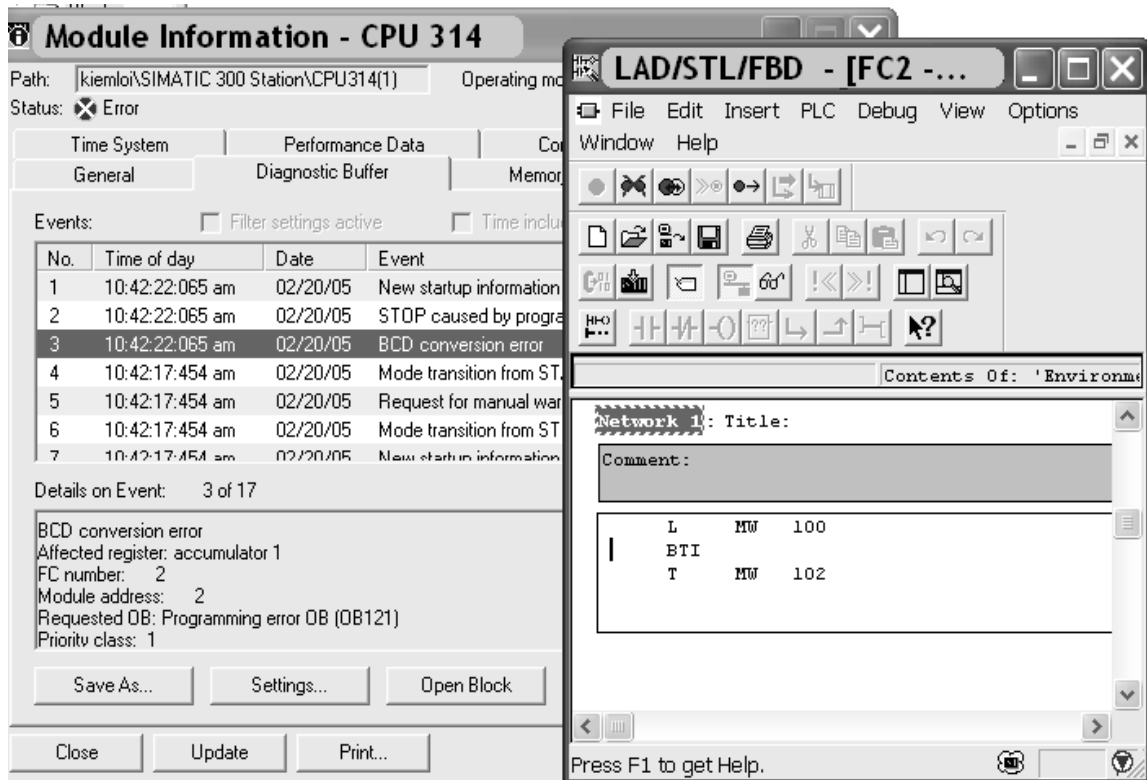
*Khi hệ thống có lỗi, ta có thể tìm thông tin về lỗi qua menu lệnh **PLC > Module Information**. Lúc này ta mở cửa sổ vùng đệm chứa các thông tin chẩn đoán để xem lỗi của hệ thống. Các thông tin này không bị xóa ngay cả khi resets bộ nhớ (MRES).



Hình 2-22: Mở cửa sổ bộ nhớ đệm chẩn đoán

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC

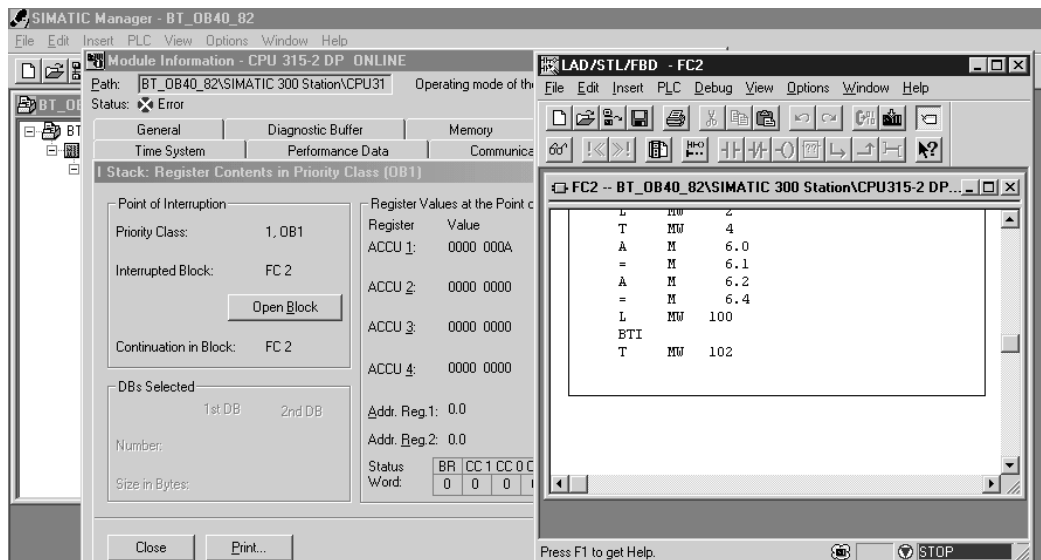
* Đọc thông tin trong cửa sổ chẩn đoán, ta biết thời gian và sự kiện xảy ra đối với CPU. Ta có thể chọn sự kiện mà ta muốn xem xét. Cửa sổ sự kiện sẽ cho ta chi tiết hơn về nội dung của sự kiện.



Hình 2-23: Xác định lệnh bị lỗi trong khối chứa lỗi

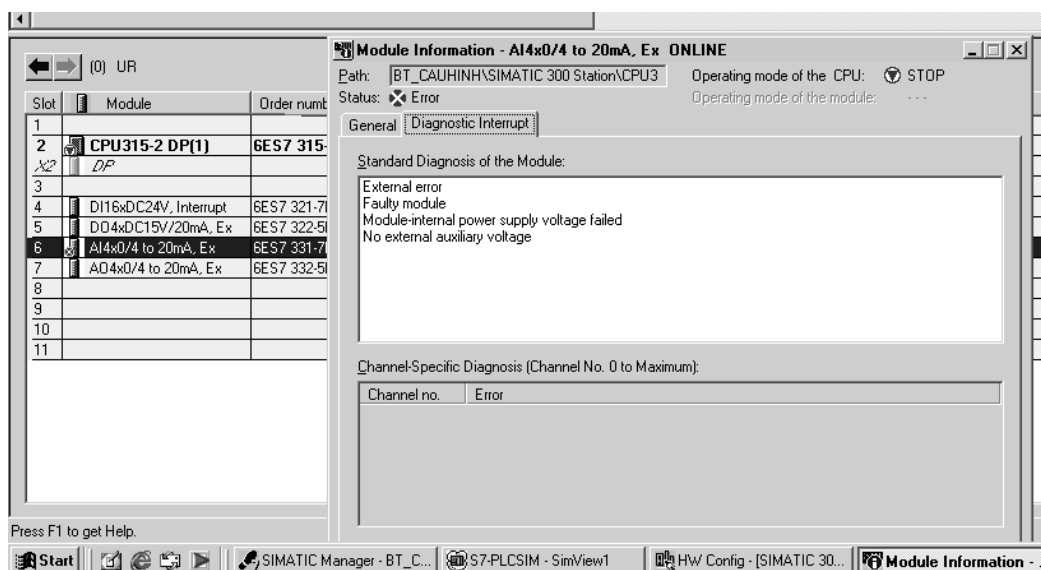
- * Đọc thông tin **Diagnostic Buffer**, ta biết được thời gian và sự kiện gây lỗi.
- * Đọc thông tin trong **Details on Event** ta biết được chi tiết hơn về lỗi
- * Nếu ta muốn xem khối chứa lỗi, ta mở khối bằng lệnh Open Block. Khi khối được mở, con trỏ sẽ xuất hiện ở dòng lệnh có lỗi.
- * Để có thông tin chi tiết hơn, ta có thể sử dụng **I Stack** để đọc giá trị của ngăn ghi **I Stack** và xác định lỗi của chương trình. Trong ví dụ, khối bị lỗi là **FC2**, dòng lệnh lỗi là **BTI**, giá trị thanh ghi ACCU1 lúc có lỗi là **W#16#A**, không hợp lệ với giá trị BCD.

CHƯƠNG 2: THEO DÕI ĐIỀU KHIỂN HOẠT ĐỘNG CỦA PLC



Hình 2-24: Sử dụng I Stack để xác định nguyên nhân lỗi

*Ta cũng có thể xem thông tin lỗi của một module SM bị lỗi.



Hình 2-25: Xem xét thông tin lỗi trên module SM

TỪ SIMATIC Manager chọn **Hardware**, trong menu chọn **View>Online**, mở **Hardware**.

Các module bị hỏng được đánh dấu. Chọn module cần xem thông tin, mở **Module Information > Diagnostic Interrupt** để xem chi tiết về lỗi xảy ra ở module.

* Cần lưu ý là module có thể được xem xét là loại module có ngắt chẩn đoán (**Diagnostic Interrupt**) và được lập trình cho phép.