如何使用 OB 组织块

1. 西门子技术支持网站简介

当您在使用西门子产品时可能遇到这样或是那样的问题，没关系，您可以登陆西门子技术支持与服务网站来查找您需要的信息。

网站链接为：http://www.ad.siemens.com.cn/service

登陆网站后，您可以点击相关链接，查找您想要的信息，其中在“网上课堂”可以下载西门子技术支持工程师编写的常问问题和使用入门文档，点击网页左侧的“技术资源”，进入后将显示语言切换到英文（点击网页的右上角“English”）,然后在“Search”输入框中输入您要查找的相关内容，如下载升级软件包，或是查找错误代码的解释，或是查找相关产品的信息等等，您可能会找到很多条链接，您可以从中选择您所需要的内容，您还可以通过点击“Product Support”进入西门子的产品信息库，通过点击左侧的相关文件链接可以查找到西门子相关产品的详细信息。如果您经常使用网站信息，您会觉得它已成为您解决问题的得力助手。

西门子技术支持与服务网站首页切图如下：

如果网站未能解决您的问题或者不是很清楚，您可以拨打我们的技术支持与服务热线：800 810 4288 ，手机用户可拨打 010 - 6471 9990 ,或发 E-Mail 到 adscs.china@siemens.com，将会有工程师为您解答。

2. 组织块的详细说明

请参阅文档 OB_Specification.pdf，如果想查阅英文文档，可从下面的链接下载该文档：

OB_Specification_English.pdf

您也可以通过按 F1 键查阅相应组织块的在线帮助，操作方法为：在程序中插入相应的 OB 块，然后选中该组织块并按 F1 键。

3. 常用 OB 组织块的使用说明

现以 CPU315(6ES7 315-2AG10-0AB0)，STEP7 V5.3 为例介绍常用 OB 组织块的使用方法，这些组织块包括：

程序循环组织块（OB1）；

日期时间中断组织块（以 OB10 为例）；

延时中断组织块（以 OB20 为例）；

循环中断组织块（以 OB35 为例）；

硬件中断组织块（以OB40为例）；

诊断中断组织块（以 OB82 为例）；

机架故障组织块（以 OB86 为例）；

启动的类型(CPU300 以 OB100 为例，CPU400 以 OB101，OB102 为例)；

编程故障组织块（以 OB121 为例）；

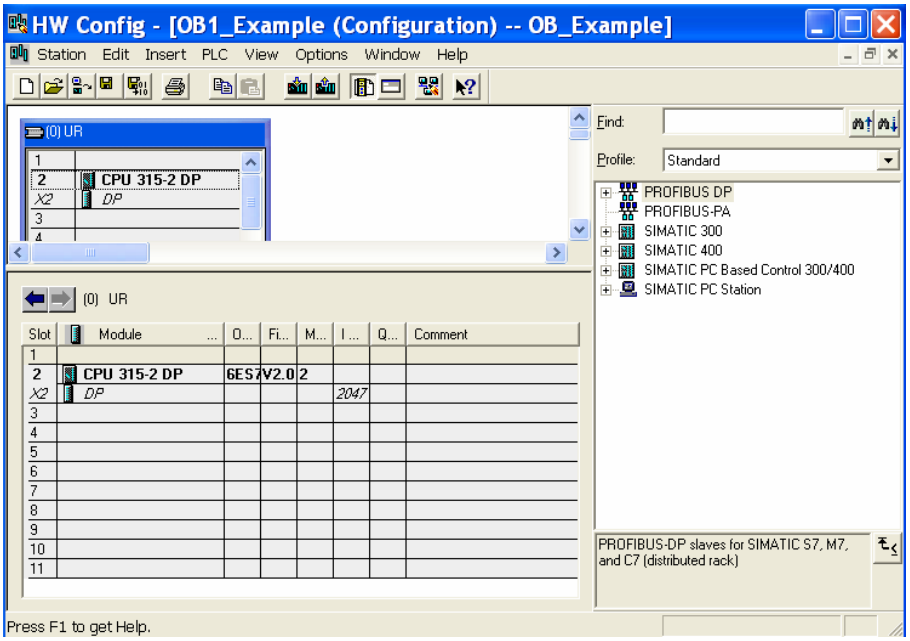
I/O 访问故障组织块（以OB122为例）；

还有其它的组织块，如：I/O 冗余故障OB（OB70），CPU 冗余故障OB（OB72），通讯冗余故障OB（OB73）请咨询CPU400H系统工程师，这里不做说明。

3．1 程序循环组织块（OB1）

3．1．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB1_Example，然后插入 CPU 315-2DP



硬件组态完成后，保存编译。

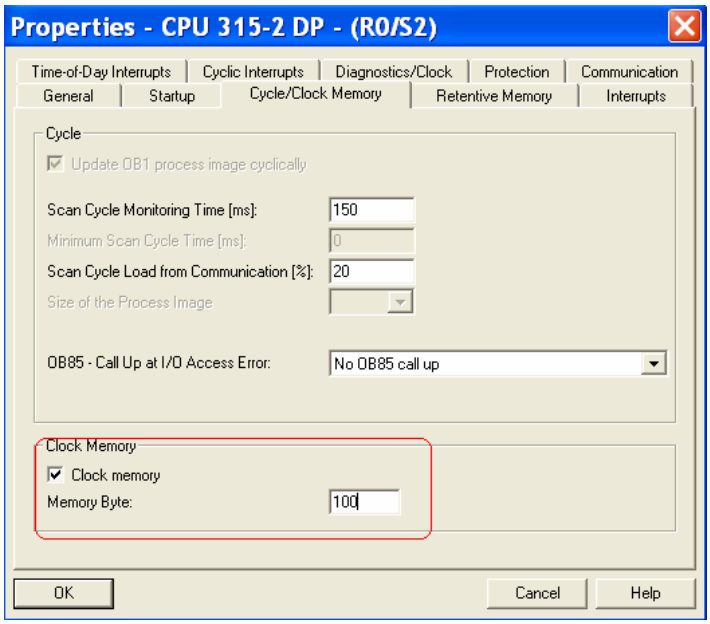
3．1．2 OB1 程序执行

OB1 的程序循环执行，用 Step7 可以时时监控程序的运行，具体程序参见 OB_Example/OB1_Example。OB1 的 STL 程序（可转成梯形图）为：

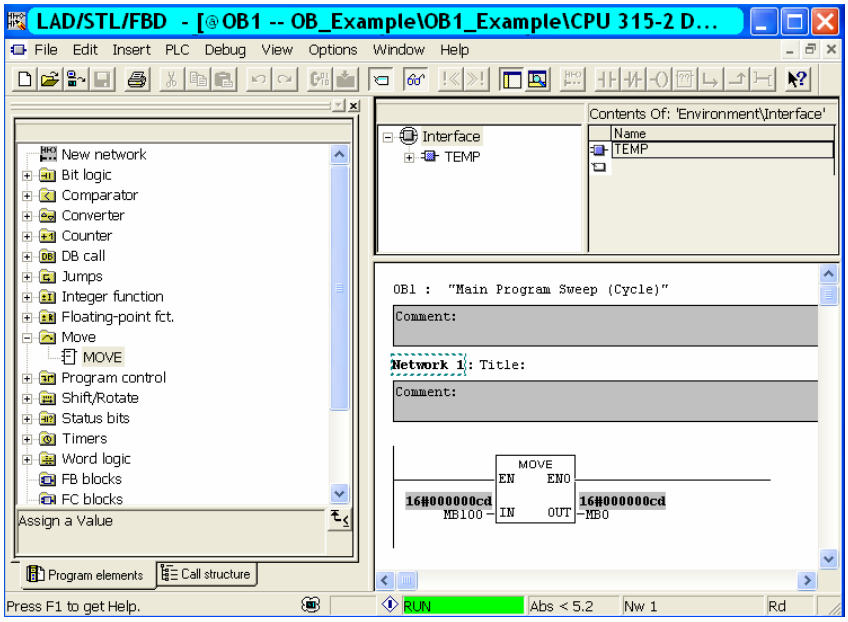
```
NetWork1:
        L       MB    100
        T       MB      0
        NOP     0
```

将 OB1 程序和硬件组态下载到 CPU 中。

其中 MB100 为时钟存储器，设置方法为进入硬件组态（HW Config），双击 CPU315-2DP，选择 Cycle/Clock Memory，具体设置画面如下：



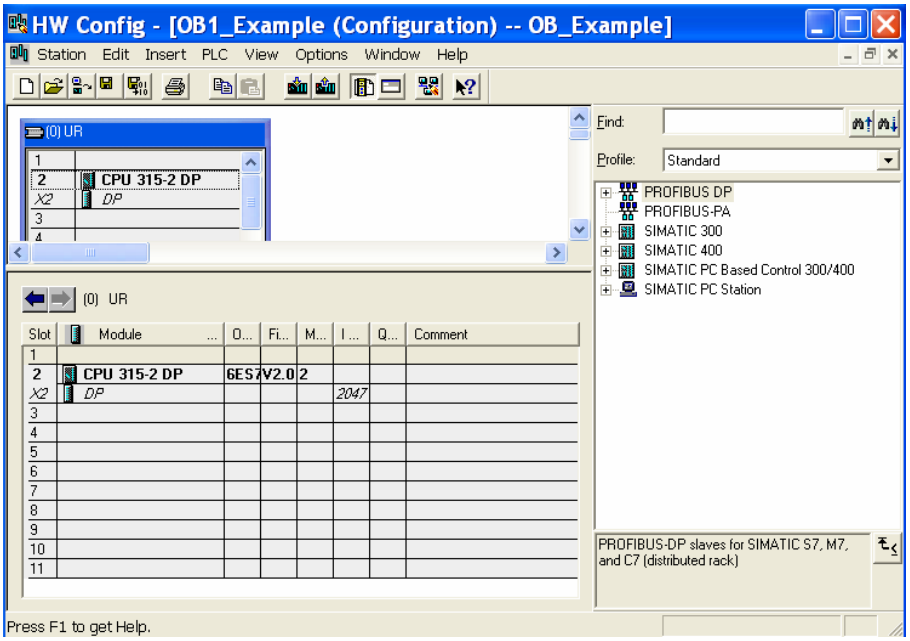
Step7 时时监控画面如下：



3．2 日期时间中断组织块（OB10）

3．2．1 硬件组态

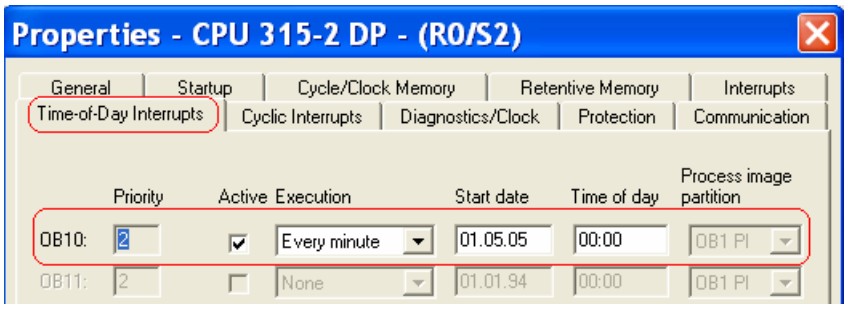
在 OB_Example 项目中插入一 S7300 站，命名为 OB10_Example，然后插入 CPU 315-2DP



双击 CPU 315-2DP，选择 Time-of-Day Interrupts 选项，选中 Active,同时设置 Execution 选项，本例选择 Every minute，Execution 选项包括：

None	不使用
Once	只执行一次
Every minute	每分钟执行一次
Every hour	每小时执行一次
Every week	每周执行一次
Every month	每月执行一次
End of month	月末执行一次
Every year	每年执行一次

设置开始执行的日期（Start date）和时间（Time of day）,设置完成后画面如下：

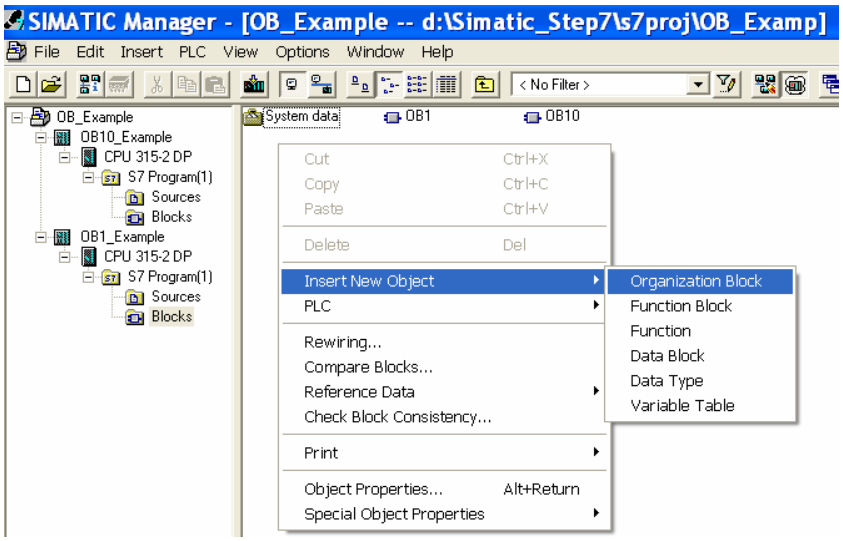


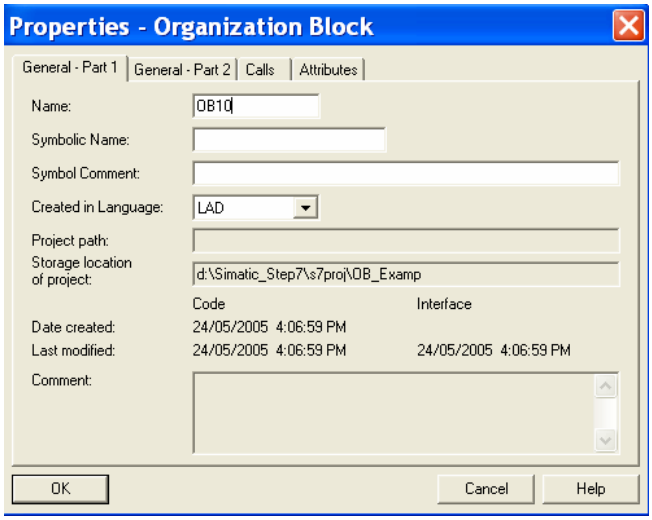
硬件组态完成后，保存编译。

3．2．2 OB10 程序执行

OB10 程序按照设定的时间执行，使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控实时数据变化。具体程序参见 OB_Example/OB10_Example。

在 OB10_Example 程序的 Blocks 中插入 OB10 组织块，画面如下：





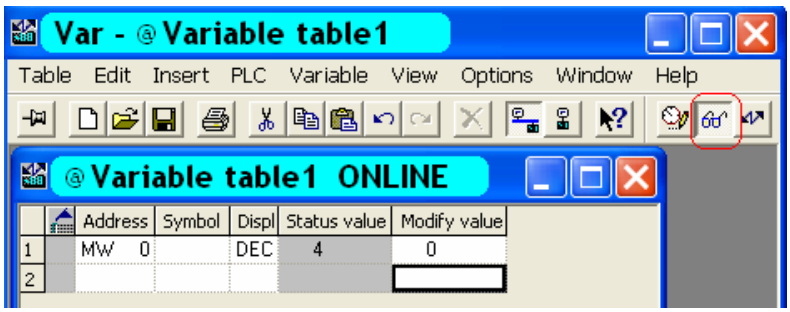
然后打开 OB10 组织块编写程序，OB10 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     MW     0
      L     1
      +I
      T     MW     0
```

NOP 0

将 OB10 程序和硬件组态下载到 CPU 中。

在 OB10_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 并点击 Monitor Variable 按钮，画面如下：

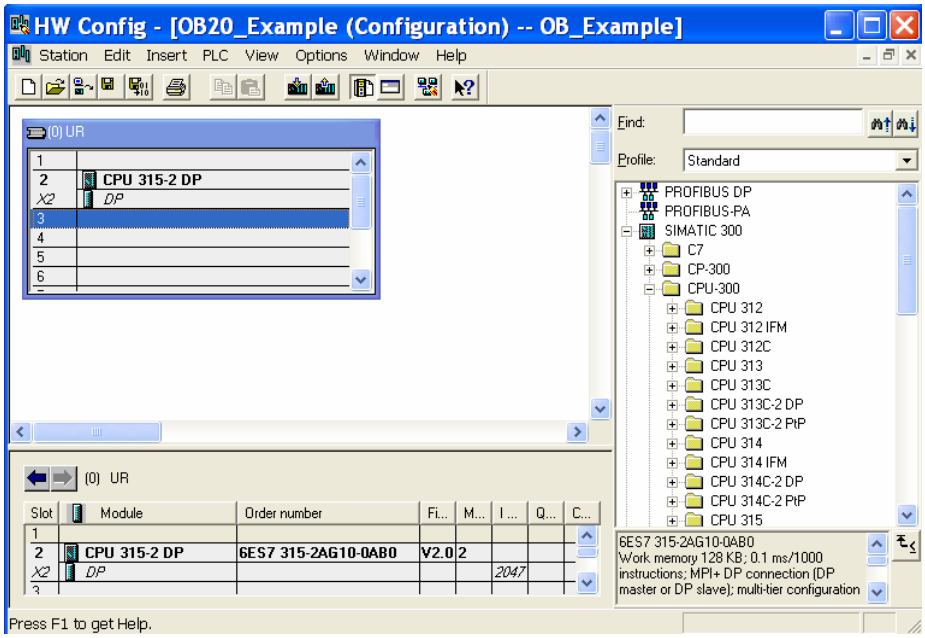


此时可以监控 MW0 每分钟加 1。

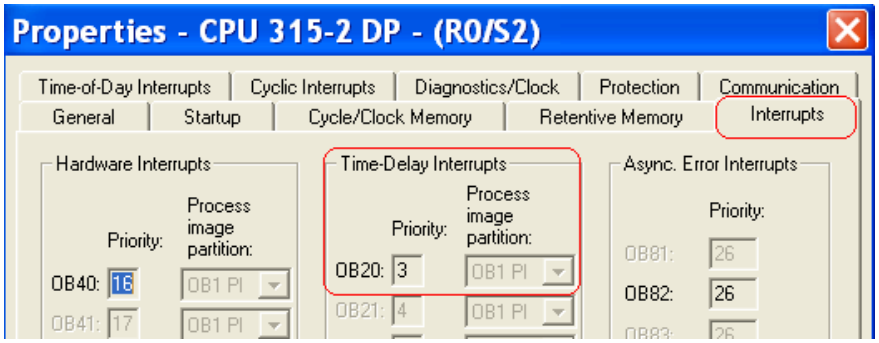
3．3 延时中断组织块（OB20）

3．3．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB20_Example，然后插入 CPU 315-2DP



双击 CPU 315-2DP，选择 Interrupts 选项，可以看到 CPU 支持 OB20，画面如下：

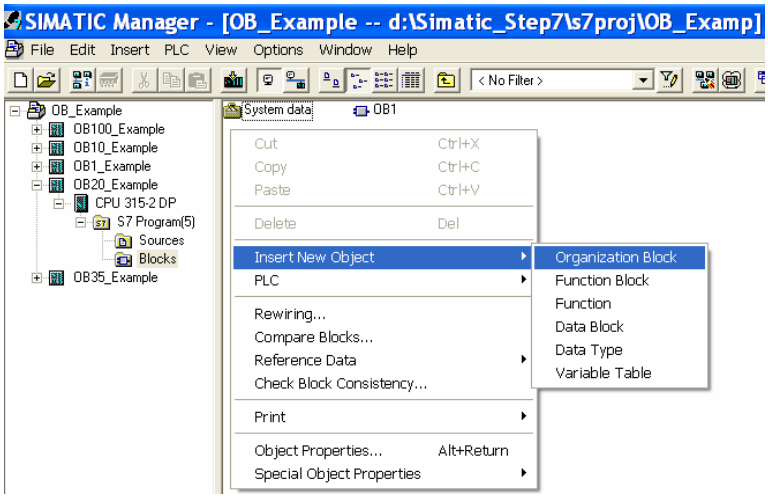


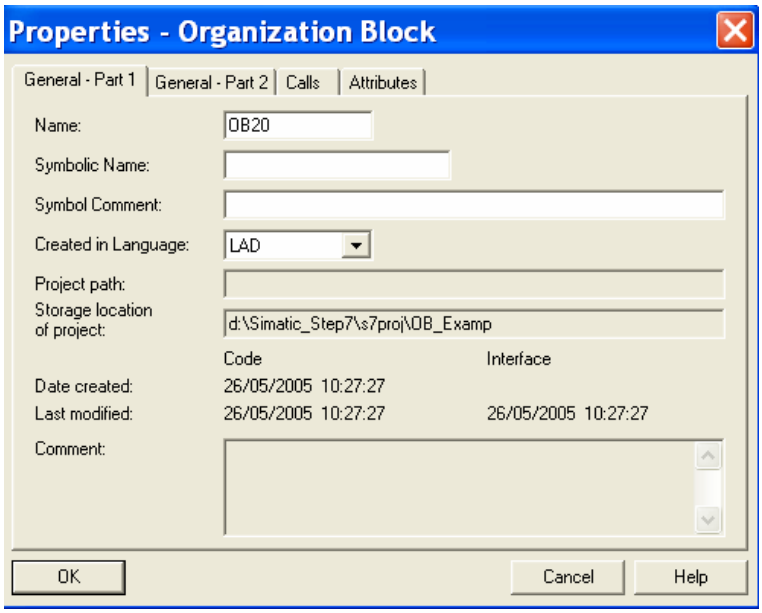
硬件组态完成后，保存编译。

3．3．2 OB20 程序执行

每一次 OB20 的程序执行，必须调用 SFC32 (SRT_DINT)，延迟时间在 SFC 的输入参数中给定，同时给定 OB 号，调用 SFC32 且设定的时间延迟到后，执行 OB 程序，如果再次执行 OB 程序，需要再次调用 SFC32。如果在延迟时间未到之前想取消程序的执行，可以调用 SFC33(CAN_DINT)，同时可以使用 SFC34 (QRY_DINT)取得延迟中断的状态，具体 SFC32/33/34 的调用方法可参考在线帮助，Step7 不能时时监控程序的运行，可用 Variable Table 监控实时数据变化。具体程序参见 OB_Example/OB20_Example。

在 OB20_Example 程序的 Blocks 中插入 OB20 组织块，画面如下：





然后打开 OB20 组织块编写程序，OB20 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     MW     0
      L     1
      +I
      T     MW     0
      NOP   0
```

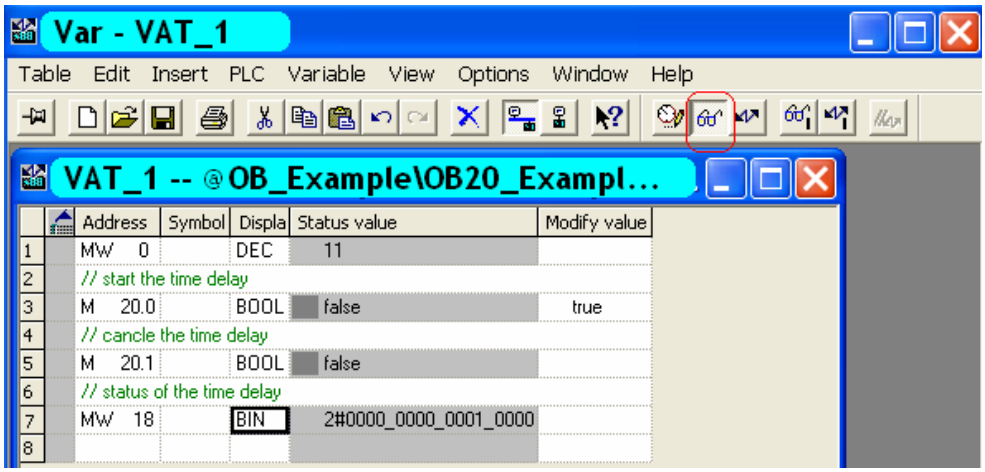
打开 OB1 组织块编写程序，OB1 的 STL 程序（可转成梯形图）为：

```
NetWork1:
     A     M     20.0
      JNB   _001
      CALL  "SRT_DINT"
       OB_NR  :=20
       DTIME  :=T#10S
       SIGN   :=MW10
       RET_VAL:=MW12
_001: A     BR
      R     M     20.0
NetWork2:
      A     M     20.1
```

```
        JNB    _002
        CALL   "CAN_DINT"
         OB_NR  :=20
         RET_VAL:=MW14
_002:  A      BR
       R      M      20.1
NetWork3:
         CALL   "QRY_DINT"
         OB_NR  :=20
         RET_VAL:=MW16
         STATUS :=MW18
        NOP    0
```

将 OB1,OB20 和硬件组态下载到 CPU 中。

在 OB20_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0,M20.0,M20.1,MW18 并点击 Monitor Variable 按钮，画面如下：

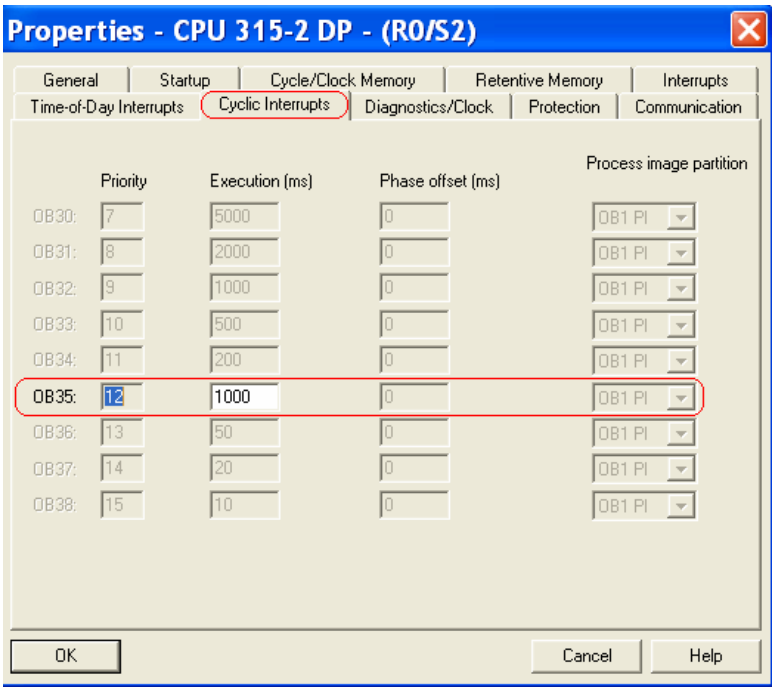


此时可以监控 MW0 的变化，将 M20.0 置为 true，10 秒钟后延迟时间到，MW0 加 1,再将 M20.0 置为 true，10 秒钟后延迟时间到，MW0 再加 1。如果当延迟时间未到，此时将 M20.1 置为 true，那么此次时间延迟中断被取消，MW0 不会加 1，每次执行的状态都可以从 MW18 种读出，具体状态的含义请参阅 SFC34(QRY_DINT)的在线帮助。

3．4 循环中断组织块（OB35）

3．4．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB35_Example，然后插入 CPU 315-2DP，参见 OB10 硬件组态，双击 CPU 315-2DP，选择 Cyclic Interrupts 选项，修改 OB35 的执行周期（Execution(ms)，范围是 1-60000ms），本例设为 1000ms，具体画面如下：

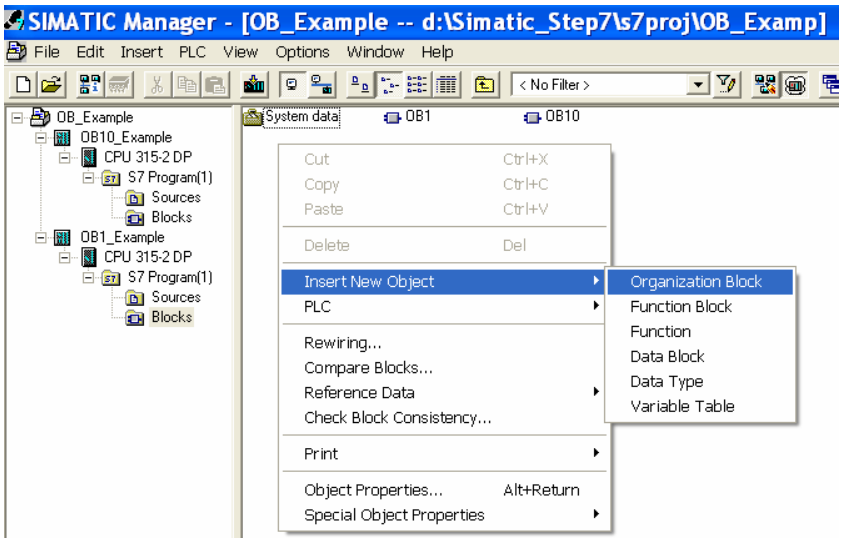


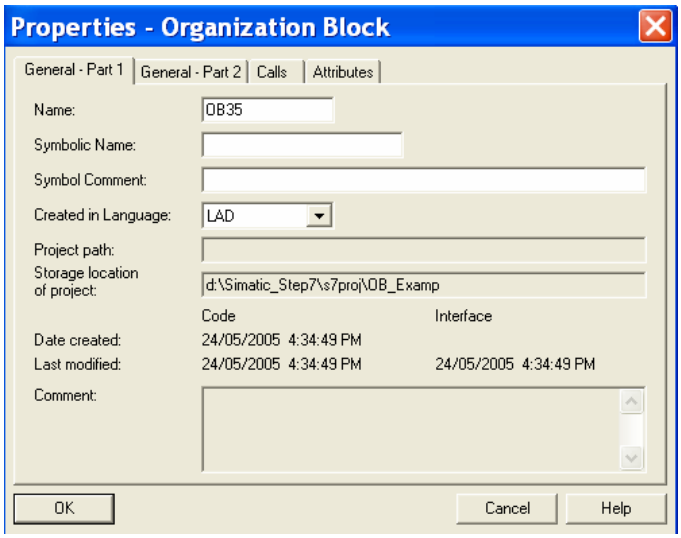
硬件组态完成后，保存编译。

3．4．2 OB35 程序执行

OB35 程序按照设定的执行周期循环执行，使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控实时数据变化。具体程序参见 OB_Example/OB35_Example。

在 OB35_Example 程序的 Blocks 中插入 OB35 组织块，画面如下：



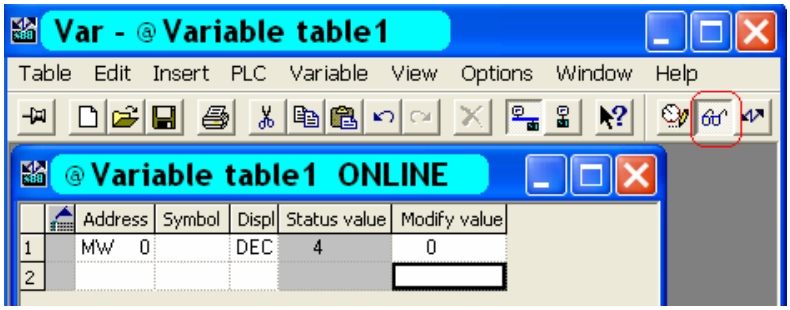


然后打开 OB35 组织块编写程序，OB35 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     MW     0
      L     1
      +I
      T     MW     0
      NOP   0
```

将 OB351 和硬件组态下载到 CPU 中。

在 OB35_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 并点击 Monitor Variable 按钮，画面如下：



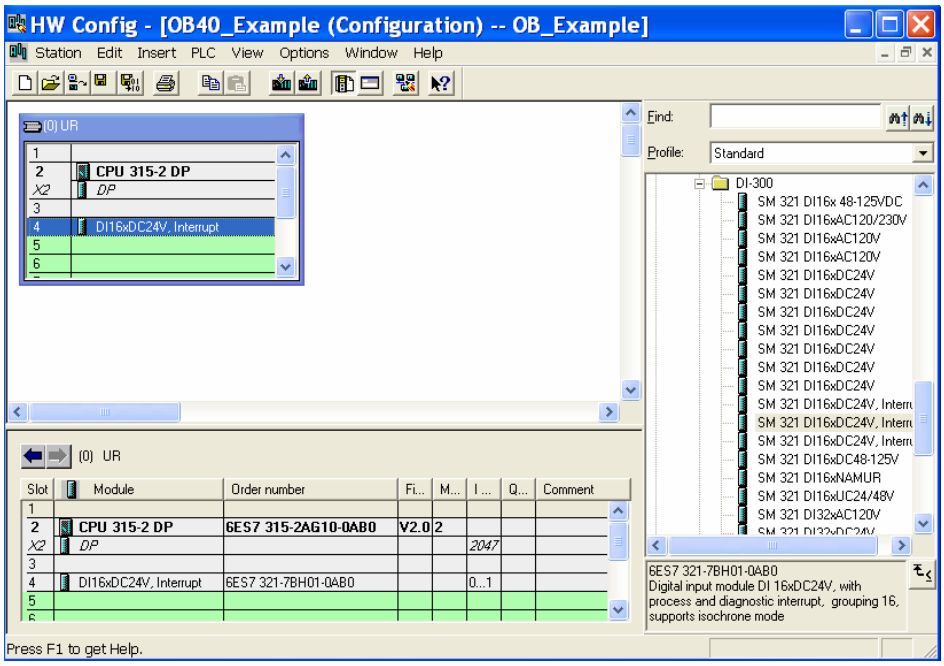
此时可以监控 MW0 每秒钟加 1。

可以在 OB35 中周期的调用 PID 模块（FB41/42/43），完成 PID 调节，也可以 OB35 中调用周期的数据发送指令，完成数据发送功能，等等。总之 OB35 是按设定的循环周期执行。

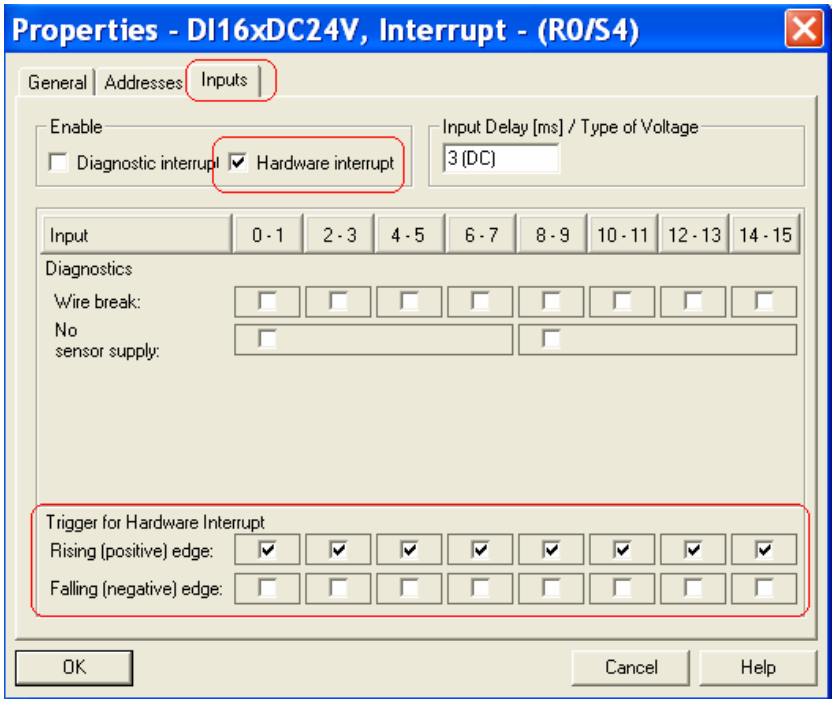
3．5 硬件中断组织块（OB40）

3．5．1 硬件组态

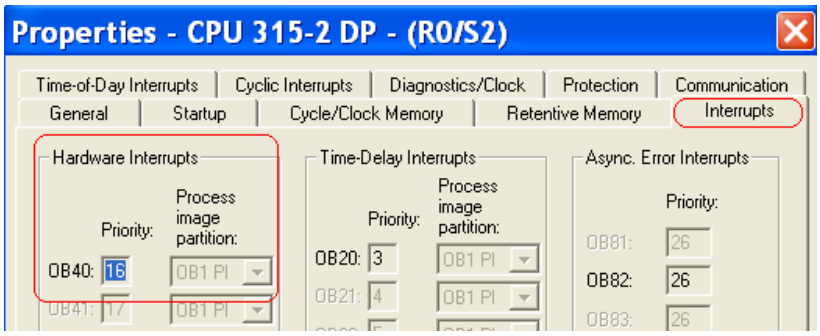
在 OB_Example 项目中插入一 S7300 站，命名为 OB40_Example，然后插入 CPU 315-2DP 和一块具有中断功能的数字量输入模板 6ES7 321-7BH01-0AB0



双击 6ES7 321-7BH01-0AB0 模板，选择 Inputs 选项，同时选中 Hardware interrupt 和 Trigger for Hardware Interrupt 选项，画面如下：



点击 OK，然后双击 CPU315-2DP，选择 Interrupts 选项，可以看到 CPU 支持 OB40，画面如下：

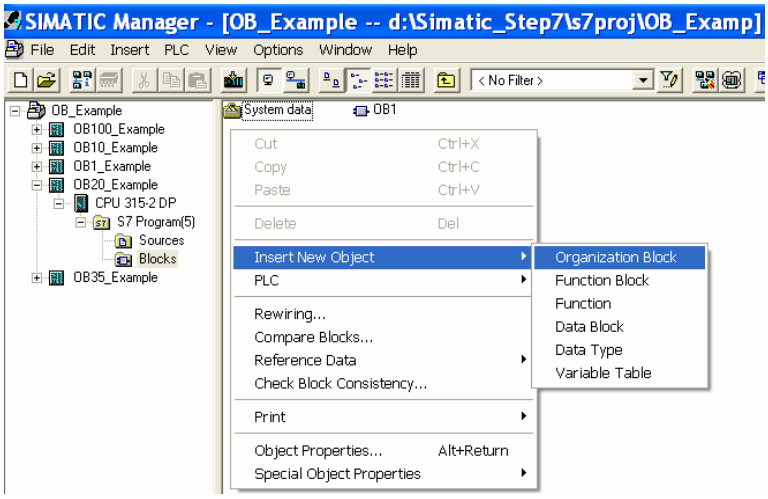


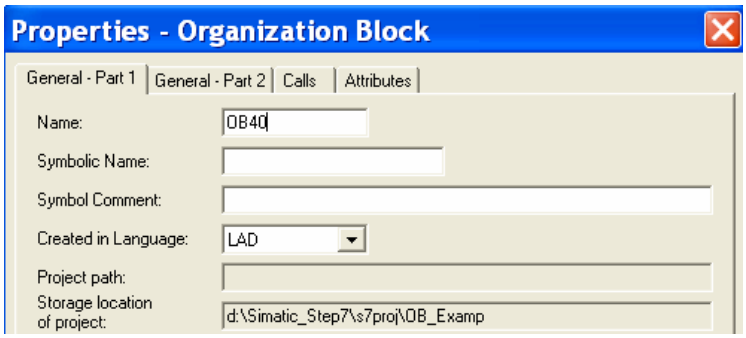
硬件组态完成后，保存编译。

3．5．2 OB40 程序执行

OB40 程序当在硬件组态中设定的硬件中断发生后执行，当 OB40 执行时可以通过它的临时变量 OB40_MDL_ADDR 读出产生硬件中断的模板的逻辑地址，通过 OB40_POINT_ADDR 可以读出产生硬件中断的通道，临时变量的具体含义请参阅在线帮助。Step7 不能时时监控程序的运行，可用 Variable Table 监控实时数据变化。具体程序参见 OB_Example/OB40_Example。

在 OB40_Example 程序的 Blocks 中插入 OB40 组织块，画面如下：



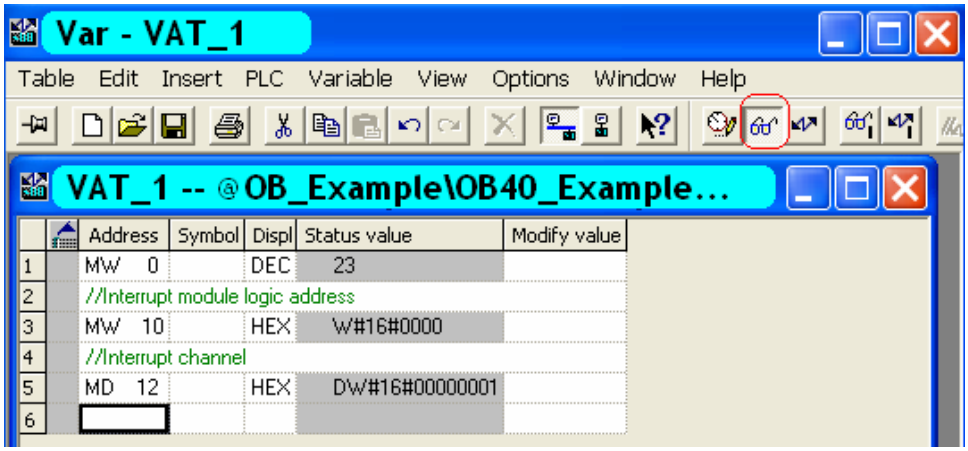


然后打开 OB40 组织块编写程序，OB40 的 STL 程序（可转成梯形图）为：

```
NetWork1:
        L       MW      0
        L       1
        +I
        T       MW      0
        NOP     0
NetWork2:
        A(
        L       #OB40_MDL_ADDR
        T       MW      10
        SET
        SAVE
        CLR
        A       BR
        )
        JNB     _001
        L       #OB40_POINT_ADDR
        T       MD      12
_001:   NOP     0
```

将 OB40 和硬件组态下载到 CPU 中。

在 OB40_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0,MW10,MD12 并点击 Monitor Variable 按钮，画面如下：



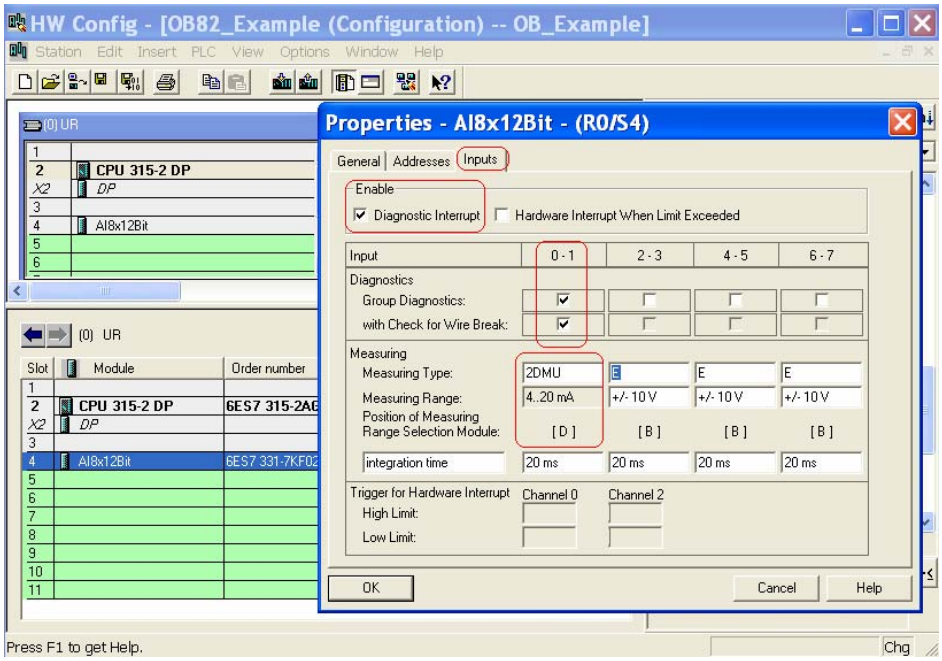
此时可以监控 MW0 的变化，每当 I0.1 有上升沿脉冲产生 MW0 加 1,MW10 为硬件中断模板的逻辑地址，本例中为 0，MD12 为中断产生的通道号，注意此值以 16 进制表示。

3．6 诊断中断组织块（OB82）

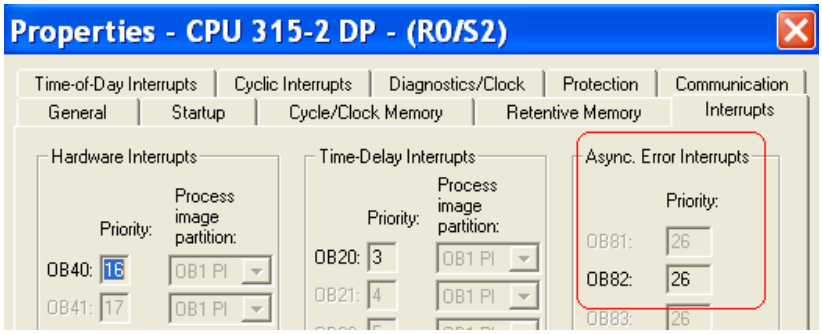
结合模板的断线检测应用和 SFC51 来说明诊断中断组织块 OB82 的使用方法。

3．6．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB82_Example，然后插入 CPU 315-2DP 和一块具有中断功能的模拟量输入模板 6ES7 331-7KF02-0AB0， 配置 SM331-7KF02-0AB0 模块的 inputs 选项，选择 0-1 通道组为 2 线制电流（2DMU），其他通道组为电压，并注意模板的量程卡与设置的相同。选中 Enable 框中的 Diagnostic Interrupt 选项，选中 Diagnostics 选项中的 0 – 1 通道组中的 Group Diagnostics 和 with Check for Wire Break 选项，配置完成的画面如下：



点击 OK，然后双击 CPU315-2DP，选择 Interrupts 选项，可以看到 CPU 支持 OB82，画面如下：



硬件组态完成后，保存编译。

3．6．2 OB82 程序执行

OB82 程序当在硬件组态中设定的诊断中断发生后执行，当 OB82 执行时可以通过它的临时变量 OB82_MDL_ADDR 读出产生诊断中断的模板的逻辑地址，OB82 其它临时变量的具体含义请参阅 OB82 的在线帮助。Step7 不能时时监控程序的运行。

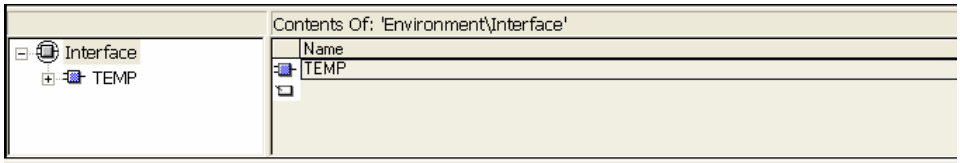
接下来完成诊断程序：

（1）在 OB_Example/OB82_Example/CPU315-2DP/S7 Program(7)/Sources 下面插入 STL Source 文件 STL Source(1)；

（2）打开空的 OB1 程序，然后选中 Libraries > Standard Libraries > System Function Blocks > SFC51 RDSYSST DIAGNSTC，按 F1 键，出现 SFC51 的在线帮助信息。可具体读一下信息的内容，然后在信息的最底部点击“Example for module diagnostics with the SFC 51”，然后选择点击“STL Source File”，选中全部 STL Source 源程序拷贝到 STL Source(1)中，存盘编译此源程序，提示没有错误；

（3） 在 Blocks 中生成 OB1，OB82，DB13 和 SFC51；

（4）打开 OB82 的程序并做简单修改，将 19 和 20 行拷贝到 go:后面并保存，具体变化如下：




```
      L     W#16#8000
      OW                                  //Set bit 15
      T     #OB82_MDL_ADDR

//Determine whether incoming or outgoing event present
go:   L     #OB82_MDL_ADDR
      T     MW    30
      L     #OB82_EV_CLASS                //Event class and IDs
      L     B#16#39
      ==I                                 //Incoming event?
      JC    come

//Read out and save diagnostic information

      L     #OB82_MDL_ADDR
      T     MW    30

//Outgoing event
      CALL  "RDSYSST"
       REQ       :=TRUE
       SZL_ID    :=W#16#B3
       INDEX     :=MW30
       RET_VAL   :=MW102
       BUSY      :=M101.7
       SZL_HEADER:=#SZL_HEADER
       DR        :=DB13.GO
```

（5）将整个 S7-300 站的程序和硬件组态下载到 CPU 中。下载完成后，将 CPU 的模式选择开关切换到 RUN 位置，此时 CPU“RUN”灯亮、“SF”灯亮，SM331 的“SF”灯亮。同时，查看 CPU 的诊断缓冲区 Hardware > Online，双击 CPU、选择 “Diagnostic Buffer”，可获得相应的故障信息；

（6）打开 DB13 数据块，在线监控，具体画面如下：

Address	Name	Type	Initial value	Actual value	Comment
0.0	COME[1]	BYTE	B#16#0	B#16#0D	
1.0	COME[2]	BYTE	B#16#0	B#16#15	
2.0	COME[3]	BYTE	B#16#0	B#16#00	
3.0	COME[4]	BYTE	B#16#0	B#16#00	
4.0	COME[5]	BYTE	B#16#0	B#16#71	
5.0	COME[6]	BYTE	B#16#0	B#16#08	
6.0	COME[7]	BYTE	B#16#0	B#16#08	
7.0	COME[8]	BYTE	B#16#0	B#16#03	
8.0	COME[9]	BYTE	B#16#0	B#16#10	
9.0	COME[10]	BYTE	B#16#0	B#16#10	
10.0	COME[11]	BYTE	B#16#0	B#16#00	
11.0	COME[12]	BYTE	B#16#0	B#16#00	
12.0	COME[13]	BYTE	B#16#0	B#16#00	
13.0	COME[14]	BYTE	B#16#0	B#16#00	
14.0	COME[15]	BYTE	B#16#0	B#16#00	
15.0	COME[16]	BYTE	B#16#0	B#16#00	

因为通道断线是一到来事件，所以诊断信息存储到 COME 数组中，具体每一字节的含义参见 S7-300 的硬件手册中 B Diagnostics Data of Signal Modules 部分的详细说明，S7-300 的硬件手册可以从西门子网站下载，下载网址为：http://support.automation.siemens.com/WW/view/en/8859629

（7）本例中 COME 数组字节的含义解释如下：

COME[1] = 16#0D 表示通道错误，外部故障 和模板问题；

COME[2] = 16#15 表示此段信息为模拟量模板的通道信息；

COME[3] = 16#00 表示 CPU 处于运行状态，无字节 2 中标示的故障信息；

COME[4] = 16#00 表示无字节 3 中标示的故障信息；

COME[5] = 16#71 表示模拟量输入；

COME[6] = 16#08 表示模板的每个通道有 8 个诊断位；

COME[7] = 16#08 表示模板的通道数；

COME[8] = 16#03 表示 0 通道错误和 1 通道错误，其它通道正常；

COME[9] = 16#10 表示 0 通道断线；

COME[10] = 16#10 表示 1 通道断线；

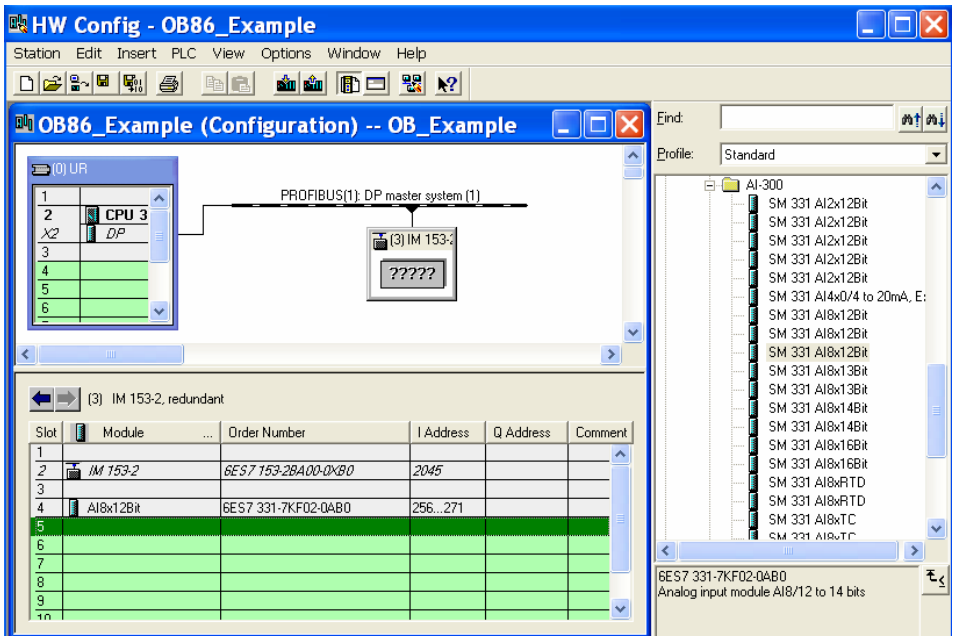
COME[11] = 16#00 表示 2 通道正常，其它通道与 2 通道相同；

（8）如何读取其他信息的诊断可详细参考 OB82、SFC51 和 S7-300 的硬件手册中 B / Diagnostics Data of Signal Modules 部分的说明。

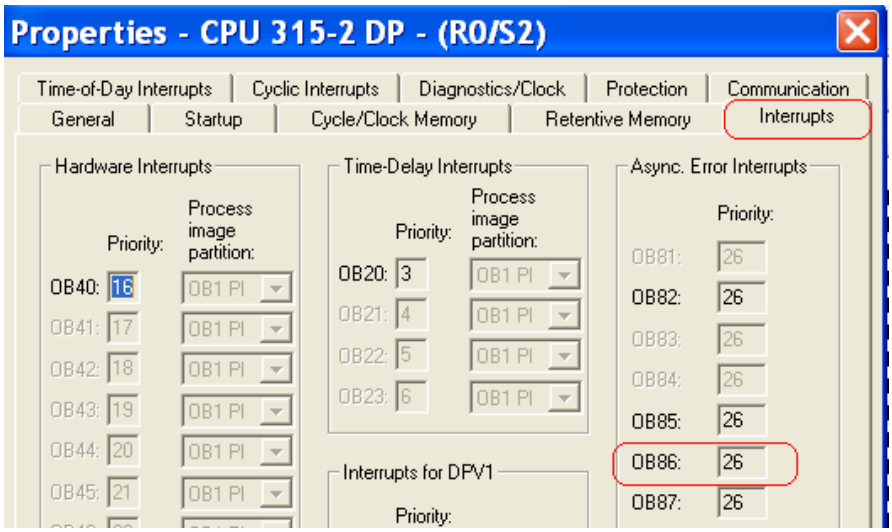
3．7 机架故障组织块（OB86）

3．7．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB86_Example，然后插入 CPU 315-2DP，选择 DP 作为主站，在 DP 主站下面添加一 ET200M 从站，并在从站中插入一模拟量模块 SM331(6ES7 331-7KF02-0AB0),同时注意 CPU 的 DP 主站地址和 ET200M 从站地址不能相同，并且 ET200M 的站地址必须和 ET200M 上的实际地址一致，组态完成后的画面如下：



然后双击 CPU315-2DP，选择 Interrupts 选项，可以看到 CPU 支持 OB86，画面如下：

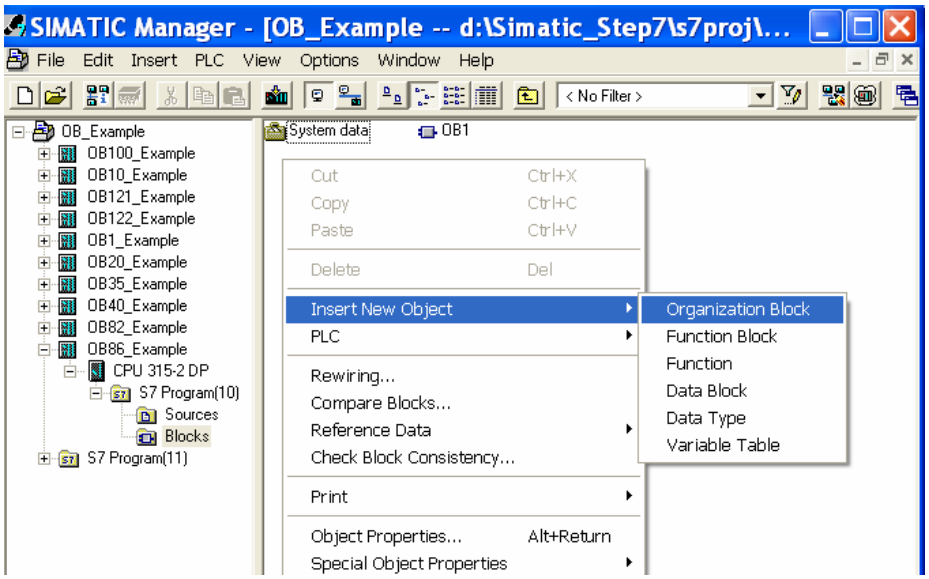


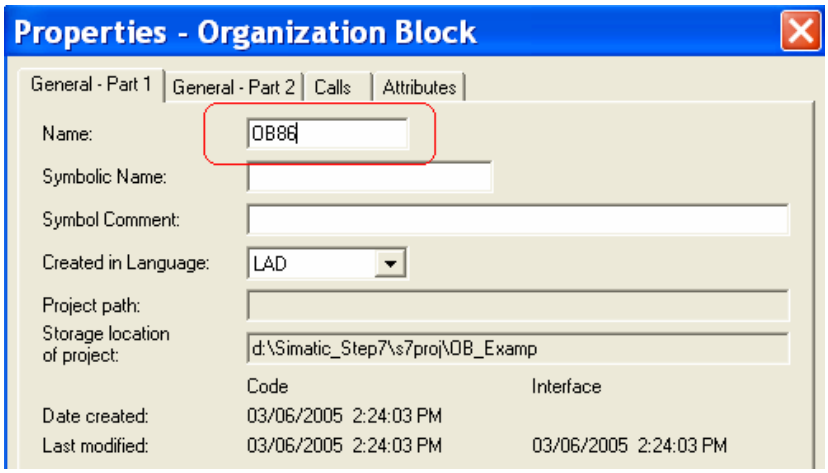
硬件组态完成后，保存编译。

3．7．2 OB86 程序执行

OB86 程序当在通讯发生问题后或者访问不到配置的机架或站时执行，此时程序还可能需要调用 OB82 和 OB122 等组织块，当 OB86 执行时可以通过它的临时变量读出产生故障的错误代码和事件类型，通过它们的组合可以得出具体的错误信息，这些信息可以通过 OB86 的在线帮助查到，同时也可以读到产生错误的模块地址和机架的信息，临时变量的具体含义请参阅在线帮助。Step7 不能时时监控程序的运行，可用 Variable Table 监控实时数据变化。具体程序参见 OB_Example/OB86_Example。

在 OB86_Example 程序的 Blocks 中插入 OB86 组织块，画面如下：





然后打开 OB86 组织块编写程序，OB86 的 STL 程序（可转成梯形图）为：

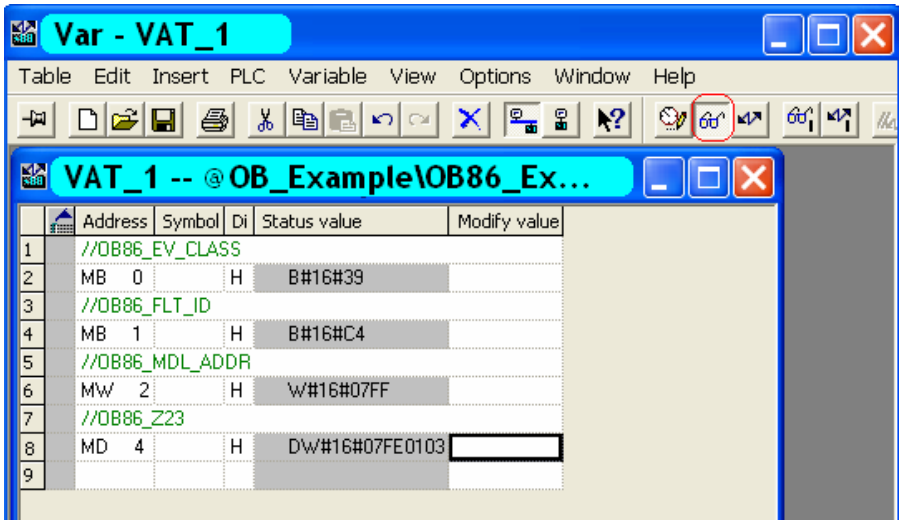
```
NetWork1:
      A(
      A(
```

```
      A(
      L     #OB86_EV_CLASS
      T     MB     0
      SET
      SAVE
      CLR
      A     BR
      )
      JNB   _001
      L     #OB86_FLT_ID
      T     MB     1
      SET
      SAVE
      CLR
_001: A     BR
      )
      JNB   _002
      L     #OB86_MDL_ADDR
      T     MW     2
      SET
      SAVE
      CLR
_002: A     BR
      )
      JNB   _003
      L     #OB86_Z23
      T     MD     4
_003: NOP   0
```

注意：将 OB86 的临时变量 OB86_RACKS_FLTD Array [0 ..31] 改为 OB86_Z23 DWORD。

将 OB86 和硬件组态下载到 CPU 中。

在 OB86_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MB0,MB1,MW2,MD4 并点击 Monitor Variable 按钮，画面如下：



此时可以读到 MB0，MB1 为 16#39 和 16#C4，可以通过它们的组合得出主站逻辑地址为 2047 的站有通讯错误，出现错误的从站地址为 3。更多的信息读取请参阅 OB86 的在线帮助。

3．8 启动的类型(OB100)

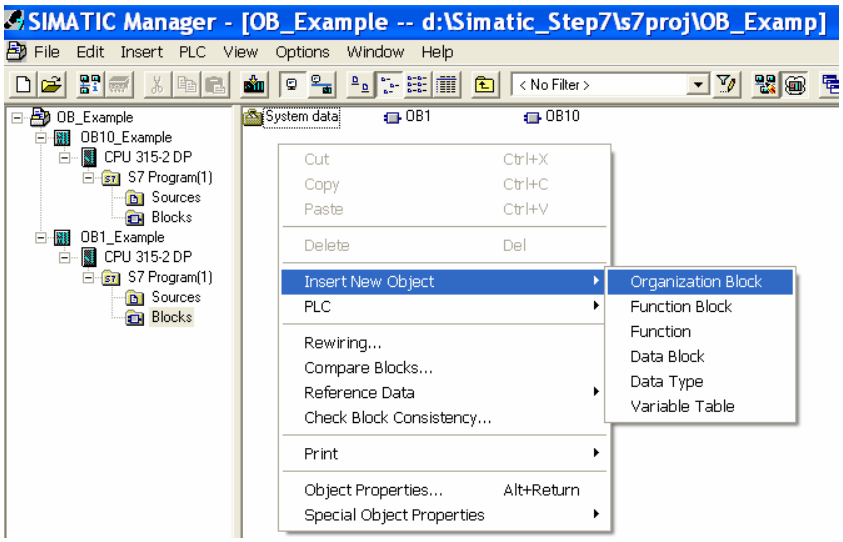
3．8．1 硬件组态

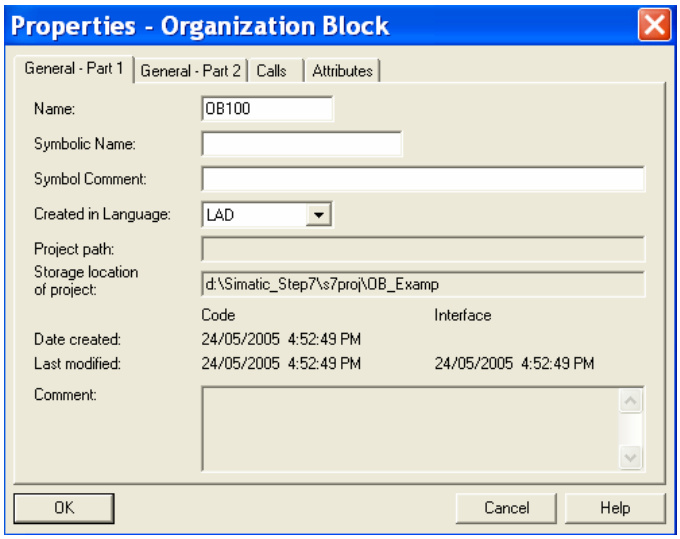
在 OB_Example 项目中插入一 S7300 站，命名为 OB100_Example，然后插入 CPU 315-2DP，参见 OB10 硬件组态。

3．8．2 OB100 程序执行

OB100 程序在 CPU 执行 Warm Restart 时执行，且只执行一次，可用于变量的初始化，使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控数据变化。具体程序参见 OB_Example/OB100_Example。

在 OB100_Example 程序的 Blocks 中插入 OB100 组织块，画面如下：

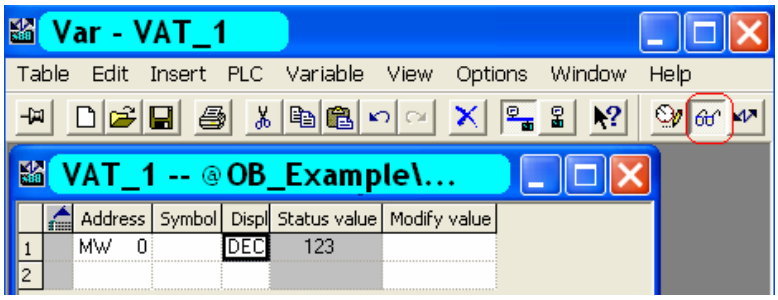




然后打开 OB100 组织块编写程序，OB100 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     123
      T     MW     0
      NOP   0
```

在 OB100_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 并点击 Monitor Variable 按钮，画面如下：



此时可以监控 MW0 为 123，如果修改 MW0 的值为 0，则 MW0 不会再被赋值为 123，只有当 CPU 再次执行 Warm Restart(重新上电或者从 Stop 切换到 Run 状态)后才会被赋值。

3．9 编程故障组织块（OB121）

3．9．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB121_Example，然后插入 CPU 315-2DP，参见 OB10 硬件组态。

3．9．2 OB121 程序执行

OB121 程序在 CPU 程序执行错误时执行，此错误不包括用户程序的逻辑错误和功能错误等，例如当 CPU 调用一未下载到 CPU 中的程序块，CPU 会调用 OB121,通过临时变量 OB121_BLK_TYPE 可以得出出现错误的程序块。使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控数据变化。具体程序参见 OB_Example/OB121_Example。

（1）在 OB121_Example 程序的 Blocks 中插入 OB121 组织块，然后打开 OB121 组织块编写程序，OB121 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     #OB121_BLK_TYPE
      T     MW     0
      NOP   0
```

（2）在 OB121_Example 程序的 Blocks 中插入 FC1，然后打开 FC1 编写程序，FC1 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      A     #in1
      =     #out1
```

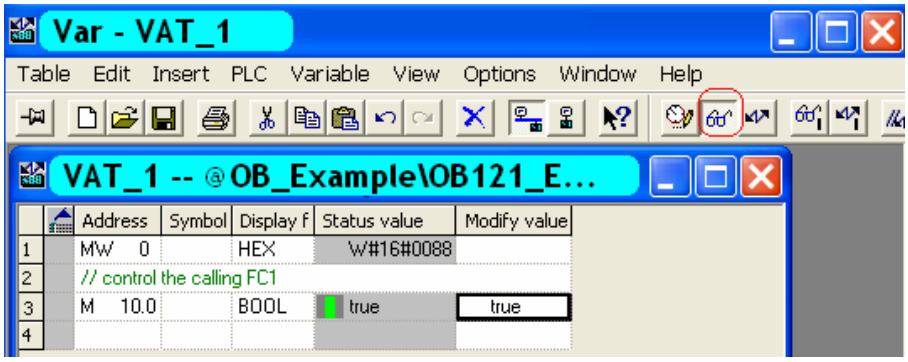
（3）打开 OB1 编写程序，OB1 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      A     M     20.1
      =     L     20.0
      BLD   103
      A     M     10.0
      JNB   _001
      CALL  FC     1
       in1 :=L20.0
       out1:=M20.2
_001: NOP   0
```

先将硬件组态和 OB1 下载到 CPU 中，此时 CPU 能正常运行。

在 OB121_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 和 M10.0 并点击 Monitor Variable 按钮，程序运行正常，将 M10.0 置为 true，CPU 报错并停机，查看 CPU 的诊断缓冲区信息，发现为编程错误，将 OB121 下载到 CPU 中，再将 M10.0 置为 true，CPU 会报错误但

不停机，MW0 立刻为 16#88，查看 OB121 的在线帮助，16#88 表示为 OB 程序错误，检查发现 FC1 未下载，具体画面如下：

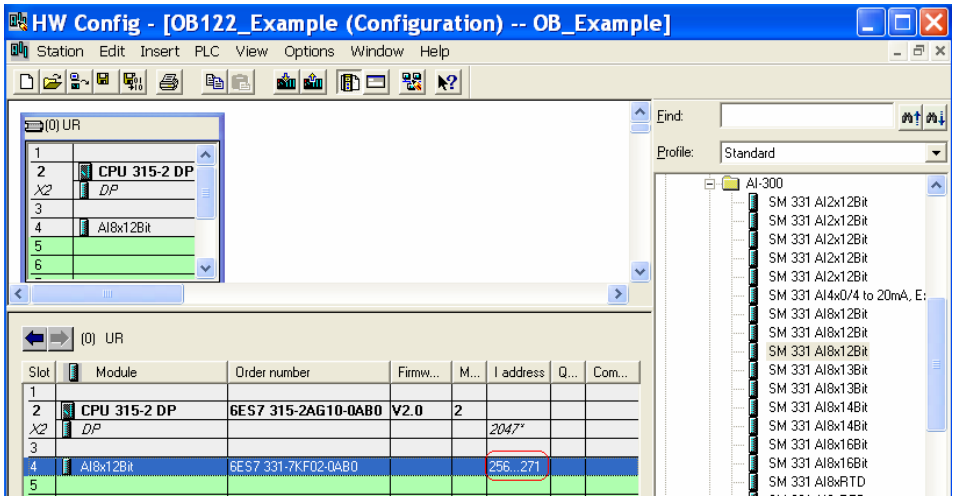


下载 FC1 后再将 M10.0 置为 true，CPU 不会再报错，程序也不会再调用 OB121。

3．10 I/O 访问故障组织块（OB122）

3．10．1 硬件组态

在 OB_Example 项目中插入一 S7300 站，命名为 OB122_Example，然后插入 CPU 315-2DP 和一块模拟量输入模板 6ES7 331-7KF02-0AB0，配置 SM331-7KF02-0AB0 模块的 inputs 选项，选择所有通道组为电压类型，注意模板的量程卡与设置的相同。模拟量的逻辑输入地址为 256...271，配置完成的画面如下：



硬件组态完成后，保存编译。

3．10．2 OB122 程序执行

OB122 程序在出现 I/O 访问错误时被调用，例如当 CPU 程序访问一未定义的 I/O 地址，CPU 会出现 I/O 访问错误，CPU 会调用 OB122,如果 OB122 未下

载，CPU 会报故障停机。通过临时变量 OB122_SW_FLT 可以读出错误代码，通过 OB122_BLK_TYPE 得出出现错误的程序块，通过 OB122_MEM_AREA 可以读出被访问的地址类型，通过 OB122_MEM_ADDR 可以读出发生错误的存储器地址。使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控数据变化。具体程序参见 OB_Example/OB122_Example。

（1）在 OB122_Example 程序的 Blocks 中插入 OB122 组织块，然后打开 OB122 组织块编写程序，OB122 的 STL 程序（可转成梯形图）为：

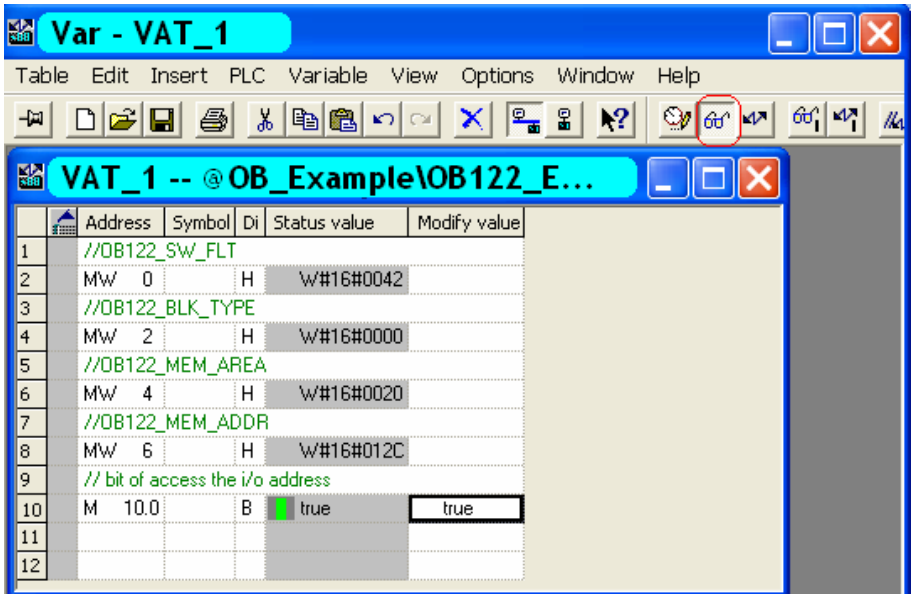
```
NetWork1:
      A(
      A(
      A(
      L     #OB122_SW_FLT
      T     MW     0
      SET
      SAVE
      CLR
      A     BR
      )
      JNB   _001
      L     #OB122_BLK_TYPE
      T     MW     2
      SET
      SAVE
      CLR
_001: A     BR
      )
      JNB   _002
      L     #OB122_MEM_AREA
      T     MW     4
      SET
      SAVE
      CLR
```

```
_002: A     BR
      )
      JNB   _003
      L     #OB122_MEM_ADDR
      T     MW     6
_003: NOP   0
```

（2）打开 OB1 编写程序，OB1 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      A     M      10.0
      JNB   _001
      L     PIW    300
      T     MW     20
_001: NOP   0
```

先将硬件组态和 OB1 下载到 CPU 中，此时 CPU 能正常运行，在 OB122_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0，MW2，MW4，MW6 和 M10.0 并点击 Monitor Variable 按钮，程序运行正常，将 M10.0 置为 true，CPU 会报错误并停机。查看 CPU 的诊断缓冲区信息，发现为 I/O 访问错误，将 OB122 下载到 CPU 中，再将 M10.0 置为 true，CPU 会报错误但不停机，MW0 为 16#0042，MW2 为 16#0000，MW4 为 16#00200，MW62 为 16#012C，查看 OB121 的在线帮助可得到相应的故障信息，具体监控画面如下：



检查并修改 OB1 程序为

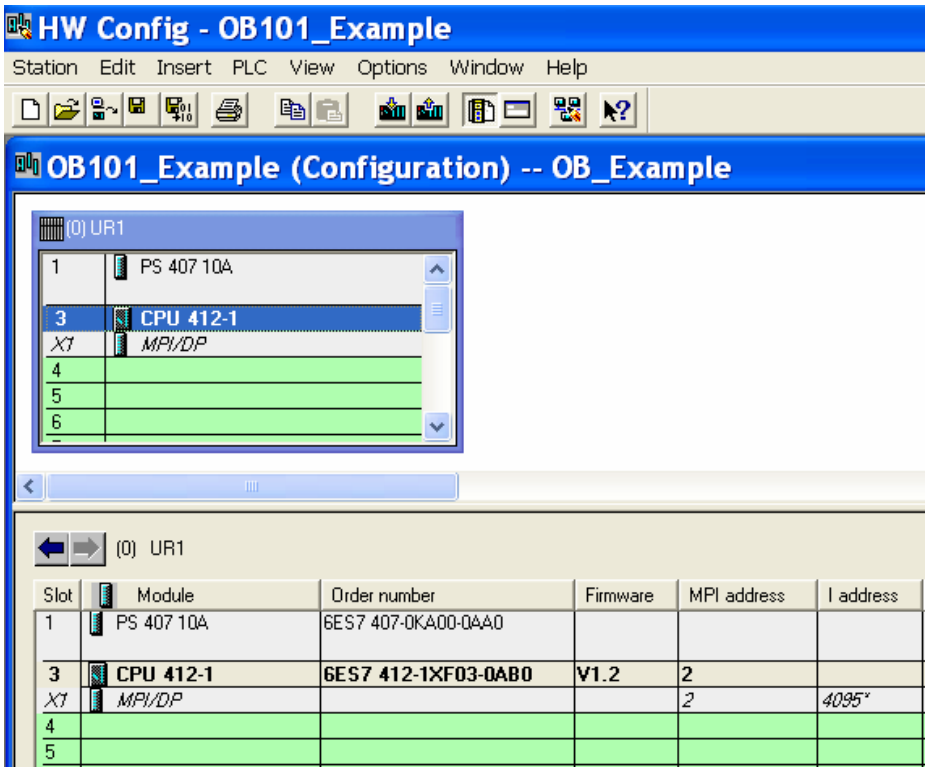
```
NetWork1:
        A       M       10.0
        JNB     _001
        L       PIW     256
        T       MW      20
_001:   NOP     0
```

重新下载 OB1，运行程序 CPU 不会再报错，程序能正常运行。

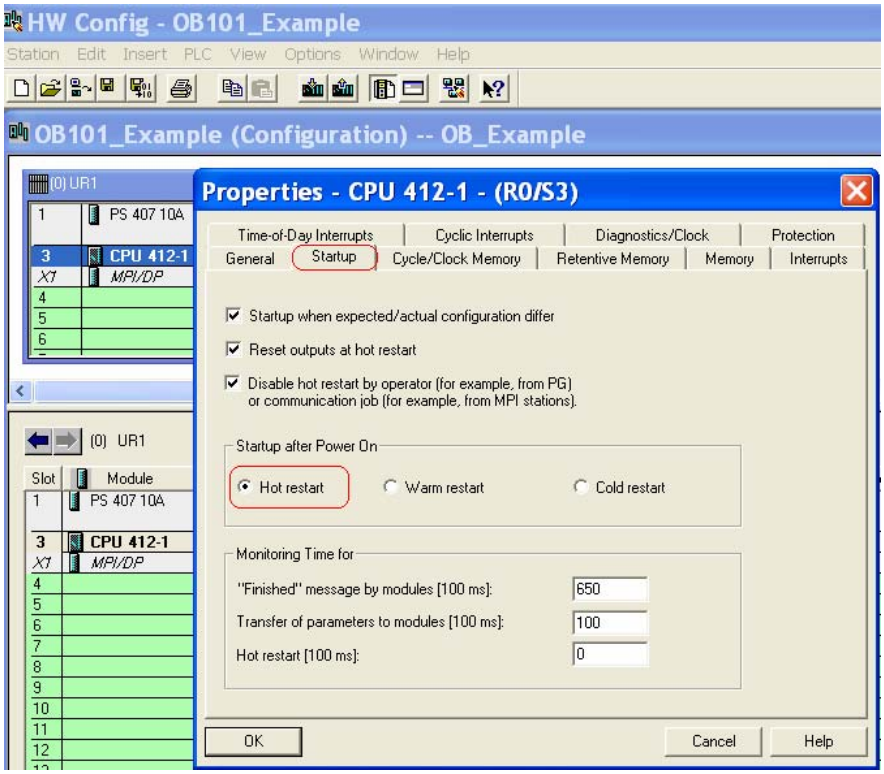
3．11 启动的类型(OB101)

3．11．1 硬件组态

在 OB_Example 项目中插入一 S7400 站，命名为 OB101_Example，然后插入 CPU 412-1(6ES7 412-1XF03-0AB0 Ver1.2)，组态完成画面如下：



双击 CPU 412-1，设置启动方式，选择 Hot Restart,具体画面如下：



组态完成后保存编译。

3．11．2 OB101 程序执行

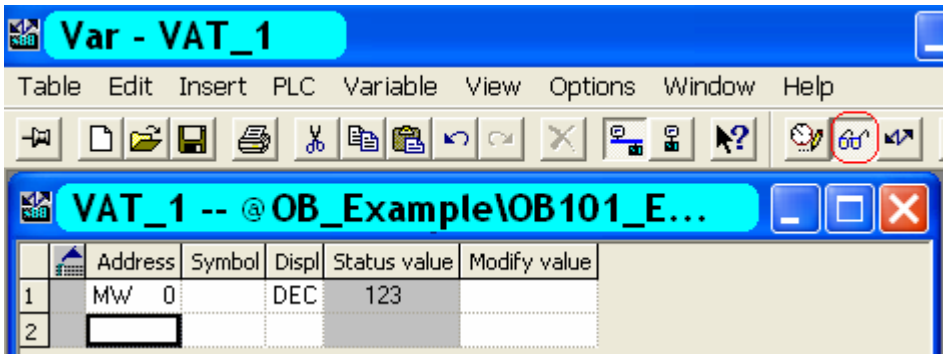
OB101 程序在 CPU 执行 Hot Restart 时执行，且只执行一次，可用于变量的初始化，使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控数据变化。具体程序参见 OB_Example/OB101_Example。

在 OB101_Example 程序的 Blocks 中插入 OB101 组织块，然后打开 OB101 组织块编写程序，OB101 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     123
      T     MW     0
      NOP   0
```

将程序和硬件组态下载到 CPU 中，然后执行 Hot Restart。

在 OB101_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 并点击 Monitor Variable 按钮，画面如下：

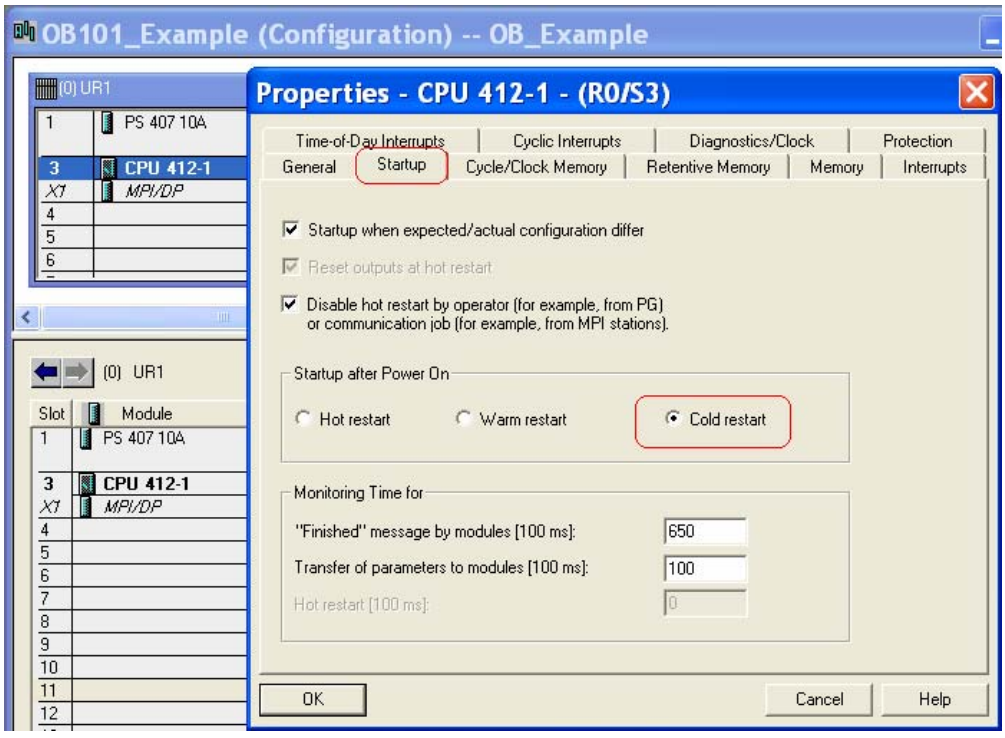


此时可以监控 MW0 为 123，如果修改 MW0 的值为 0，则 MW0 不会再被赋值为 123，只有当 CPU 再次执行 Hot Restart 后才会被赋值。

3．12 启动的类型(OB102)

3．12．1 硬件组态

在 OB_Example 项目中插入一 S7400 站，命名为 OB102_Example，然后插入 CPU 412-1(6ES7 412-1XF03-0AB0 Ver1.2)，组态参见 OB101 部分，设置启动方式，选择 Cold Restart,具体画面如下：



组态完成后保存编译。

3．12．2 OB102 程序执行

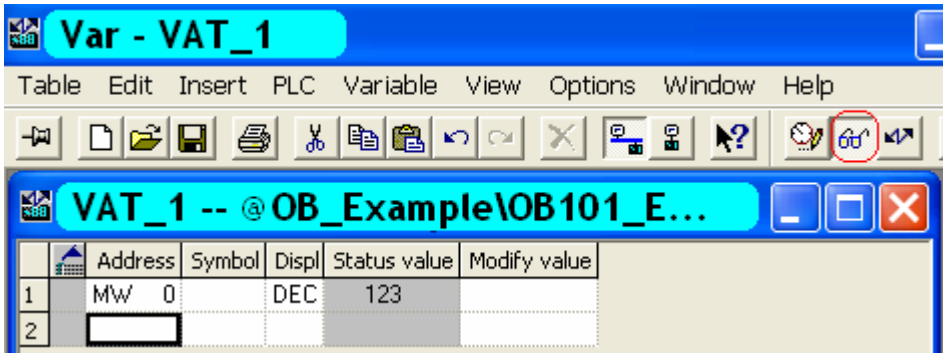
OB102 程序在 CPU 执行 Cold Restart 时执行，且只执行一次，可用于变量的初始化，使用 Step7 不能时时监控程序的运行，可用 Variable Table 监控数据变化。具体程序参见 OB_Example/OB102_Example。

在 OB102_Example 程序的 Blocks 中插入 OB102 组织块，然后打开 OB102 组织块编写程序，OB102 的 STL 程序（可转成梯形图）为：

```
NetWork1:
      L     123
      T     MW     0
      NOP   0
```

将程序和硬件组态下载到 CPU 中，然后执行 Cold Restart。

在 OB102_Example 程序的 Blocks 中插入 Variable Table，然后打开，填入地址 MW0 并点击 Monitor Variable 按钮，画面如下：



此时可以监控 MW0 为 123，如果修改 MW0 的值为 0，则 MW0 不会再被赋值为 123，只有当 CPU 再次执行 Hot Restart 后才会被赋值。

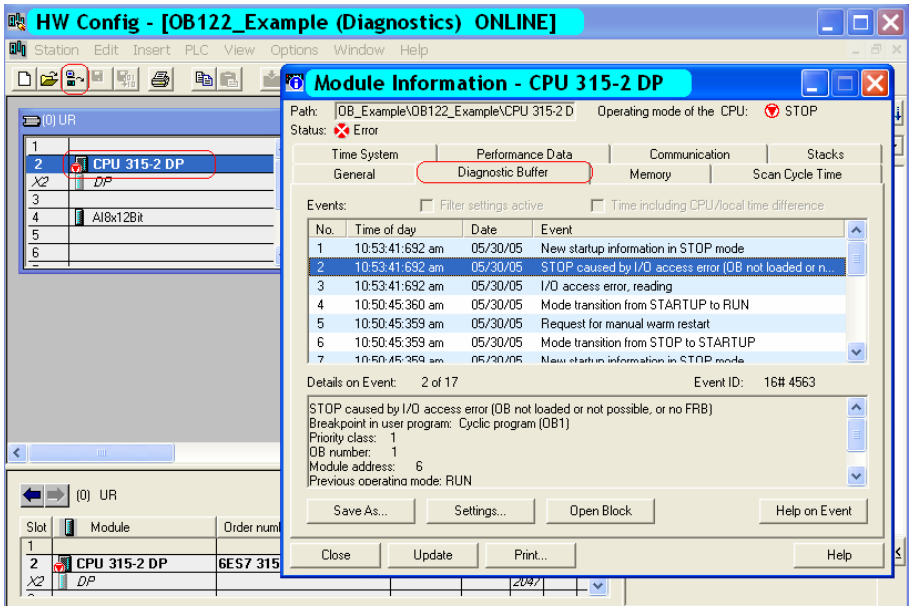
4. 有关组织块的常问问题

4．1 CPU 的 SF 红灯亮，CPU 停机是什么原因造成的？

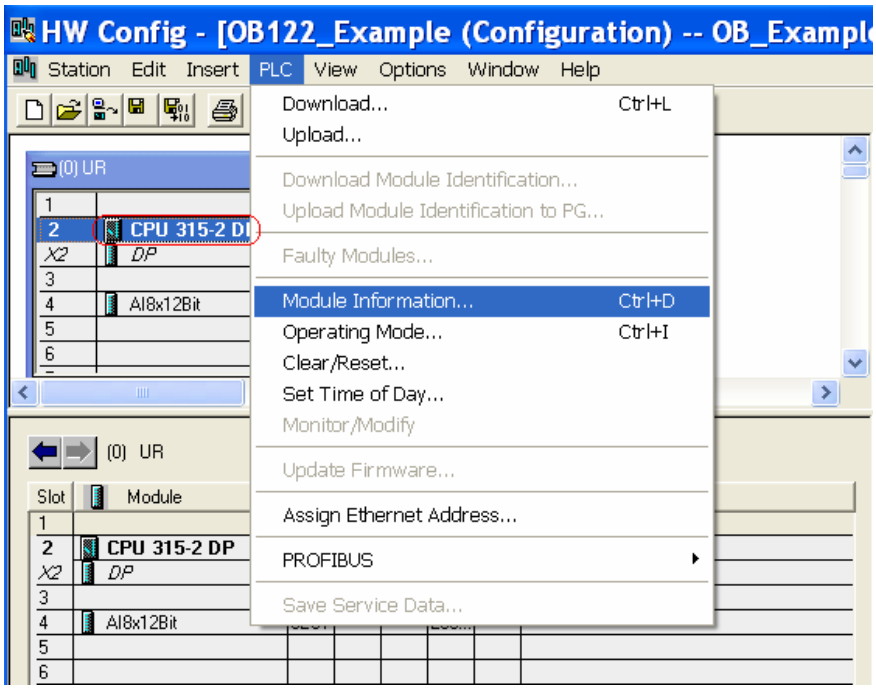
当 CPU 的 SF 红灯亮，CPU 停机后不知道是什么原因，此时怎么办呢？您需要去查看 CPU 的诊断缓冲区，根据缓冲区中提供的停机信息采取相应的措施，例如需要 OB82 ,OB86 的组织块下载等。

那么如何查看 CPU 的诊断缓冲区呢？

方法一、首先建立 Step7 与 CPU 的通讯，然后打开硬件组态，点击 Offline<->Online 按钮  ，然后双击 CPU，选择 Diagnostic Buffer 选项，可以查看 CPU 的故障信息，具体画面如下：



方法二、首先建立 Step7 与 CPU 的通讯，然后打开硬件组态，点击 CPU,然后选择 PLC->Module Information...选项，具体画面如下：



再选择 Diagnostic Buffer 选项，可以查看 CPU 的故障信息，具体画面同上。

4．2 为什么监控 OB100 程序时，感觉程序没有运行？

因为 OB100 位暖启动组织块，只有当 CPU 执行暖启动操作时才执行 OB100 的程序，且只执行一个周期，所以当监控 OB100 程序时感觉程序没有运行。

4．3 OB35 的循环时间最长为 60 秒，但想实现 5 分钟的循环周期怎么办？

将 OB35 的执行周期设为 60000ms,在 OB35 组织块中做一加法计数，当计数值等于 5 后执行相应的程序，然后将计数值清零，简单程序举例如下：

OB35 组织块 STL 程序为：

```
NetWork1:
      L     MW     0
      L     1
      +I
      T     MW     0
      NOP   0
NetWork2:
      L     MW     0
      L     5
      ==I
      =     L     20.0
      A     L     20.0
      JNB   _001
      L     1234
      T     MW    100
_001: NOP   0
      A     L     20.0
      JNB   _002
      L     0
      T     MW     0
_002: NOP   0
```

4．4 在冗余电源配置中，电源模块掉电，调用那个 OB 可以防止 CPU 停机？

通过在程序中添加 OB83 可以防止 CPU 停机而添加 OB81 不能防止 CPU 停机。通常我们很冗易以为 OB81 就是处理所有电源故障的 OB 块，但对

于冗余电源配置中，某个电源模块掉电故障，实际上 CPU 当作模块插拔故障来处理，因此需调用 OB83。

如图 1 所示当程序中没有插入 OB83 时电源模块掉电，CPU 会停机。查看 Diagnostic Buffer 中显示的信息是模块插拔故障导致停机。

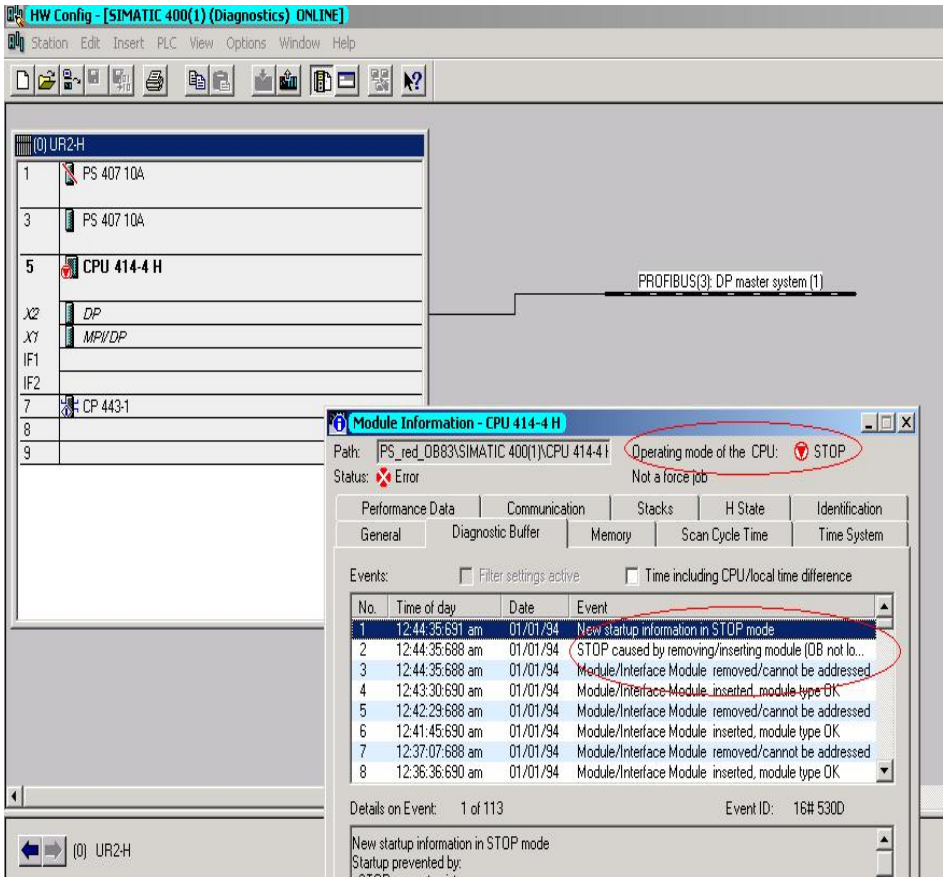


图 1

如图 2 所示当程序中没有插入 OB83 时电源模块掉电后恢复，CPU 停机不恢复。查看 Diagnostic Buffer 中显示的信息是模块插入恢复。

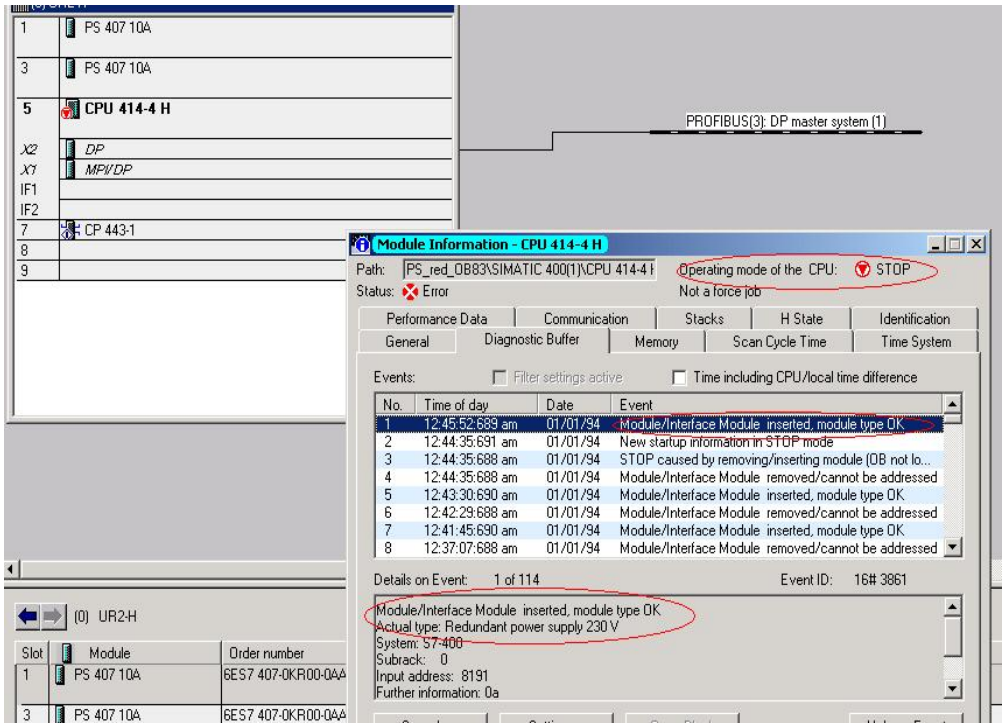


图 2

如图 3 所示当程序中插入 OB83 时电源模块掉电，CPU 不会停机。查看 Diagnostic Buffer 中显示的信息是模块拔除故障调用 OB83。

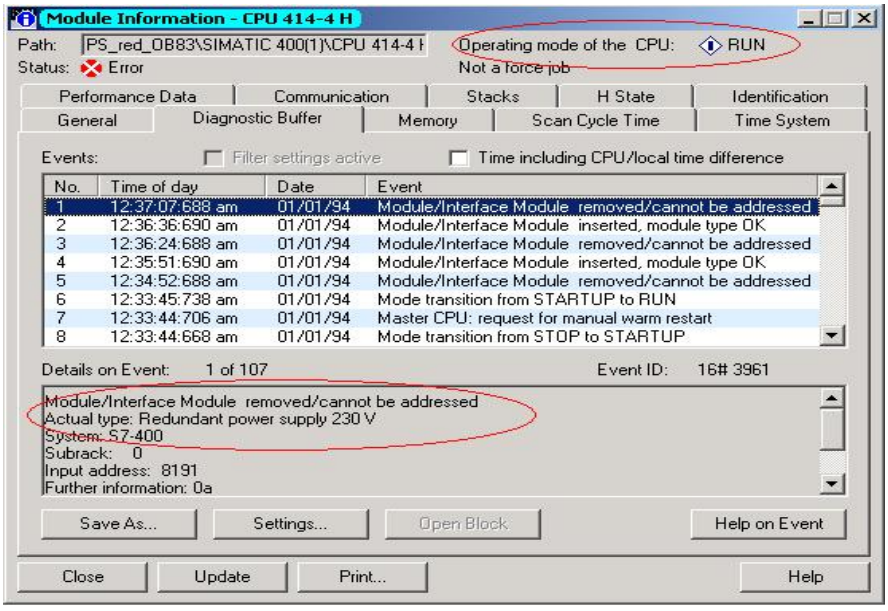


图 3

如图 4 所示当程序中插入 OB83 时电源模块掉电后恢复，CPU 不停机外部故障灯恢复。查看 Diagnostic Buffer 中显示的信息是模块插入恢复。

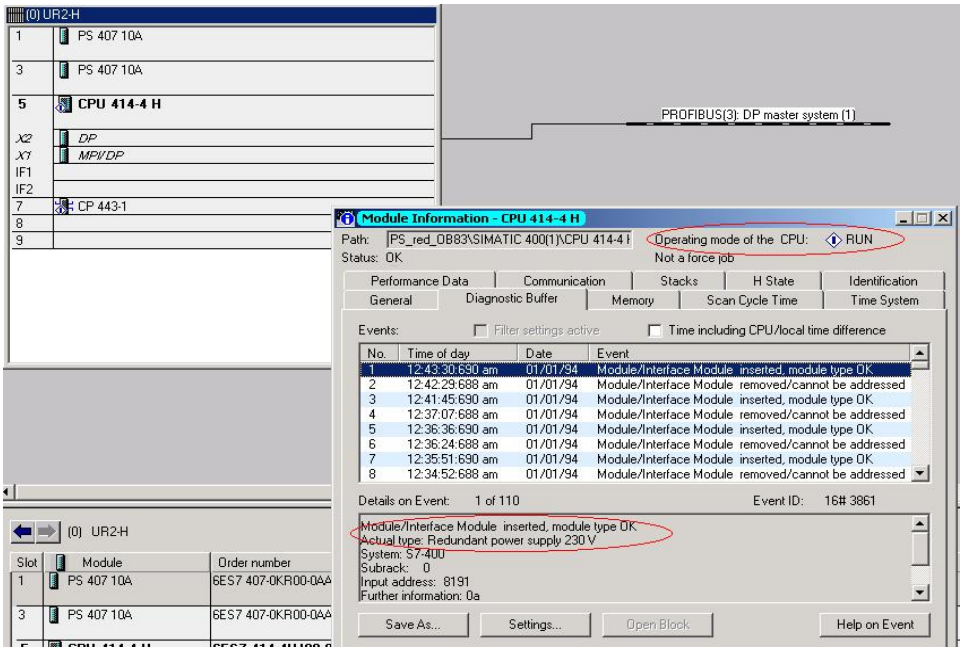

(0) UR2-H
1 PS 407 10A
3 PS 407 10A
5 CPU 414-4 H
X2 DP
X7 MPI/DP
IF1
IF2
7 CP 443-1
8
9
PROFIBUS(3): DP master system (1)
Module Information - CPU 414-4 H
Path: PS_red_0B83\SIMATIC 400(1)\CPU 414-4 H
Operating mode of the CPU: RUN
Status: OK
Not a force job
Performance Data
Communication
Stacks
H State
Identification
General
Diagnostic Buffer
Memory
Scan Cycle Time
Time System
Events:
Filter settings active
Time including CPU/local time difference
No. Time of day Date Event
1 12:43:30:690 am 01/01/94 Module/Interface Module inserted, module type OK
2 12:42:29:688 am 01/01/94 Module/Interface Module removed/cannot be addressed
3 12:41:45:690 am 01/01/94 Module/Interface Module inserted, module type OK
4 12:37:07:688 am 01/01/94 Module/Interface Module removed/cannot be addressed
5 12:36:36:690 am 01/01/94 Module/Interface Module inserted, module type OK
6 12:36:24:688 am 01/01/94 Module/Interface Module removed/cannot be addressed
7 12:35:51:690 am 01/01/94 Module/Interface Module inserted, module type OK
8 12:34:52:688 am 01/01/94 Module/Interface Module removed/cannot be addressed
Details on Event: 1 of 110
Event ID: 16# 3861
Module/Interface Module inserted, module type OK
Actual type: Redundant power supply 230 V
System: S7-400
Subrack: 0
Input address: 8191
Further information: 0a
Save As...
Settings...
Open Block
Help on Event
(0) UR2-H
Slot Module Order number
1 PS 407 10A 6ES7 407-0KR00-0AA
3 PS 407 10A 6ES7 407-0KR00-0AA

图 4