

SIEMENS

Step7中有关时间和定时器的使用和例程 About Time AND Timer sample in STEP7

Getting Started

Edition (2007-9)

摘要

本文档主要介绍以下相关问题：

西门子S5TIMER和IEC TIMER的使用方法

西门子有关时间的OB块的使用方法

西门子有关时间的系统功能块的使用方法

关键词

定时器； IEC定时器；组织块；系统功能块

Key Words

S5Timer;Timer;OB Block; SFCs/SFBs

目 录

1. 定时器	4
2. IEC定时器 (SFB3、SFB4、SFB5)	6
2.1 SFB3 (TP)	7
2.2 SFB4 (TON)	7
2.3 SFB5 (TOF)	7
3. 系统时钟 (SFC0、SFC1、OB1)	8
3.1 设置系统时钟的两种办法	8
3.2 读取系统时钟	15
4. 循环中断 (OB35)、延迟中断 (OB20)、时间日期中断 (OB10)	16
4.1 循环中断 (OB35)	16
4.2 延迟中断 (OB20)	21
4.3 时间日期中断 (OB10)	22
5. 时钟寄存器 (Clock Memory)	27
6. FAQ:	30
6.1 如何从上位机或者触摸屏设定Time或S5Time值?	30
6.2 定时器时间太短不够用怎么办?	38
6.3 如何用Timer作为变量为FC/FB传递参数?	44
6.4 用IEC 定时器如何使用多重背景	48
6.5 Time to S5Time, S5Time to Time.....	52
6.5.1 Time to S5Time.....	52
6.5.2 S5Time to Time.....	55
7 附录一推荐网址.....	57
7.1 西门子自动化与驱动产品的在线技术支持.....	57
7.1.1 如何获得西门子自动化与驱动产品的资料.....	58
7.1.2 需要设备选型及订货.....	58
7.1.3 西门子技术支持热线	58
7.1.4 西门子自动化产品的其它网站.....	58

前言:

在Step7中提供了比较丰富的与时间有关的元素例如定时器、系统时钟以及相关的组织块和系统功能块等，非常方便。但因为大家对其缺乏必要的了解，所以在使用中总会出现这样那样的问题。为了帮助大家比较系统的了解这个有关时间的主题我们在这个文档中分别做以介绍和提供一简单例程。

重要提示:

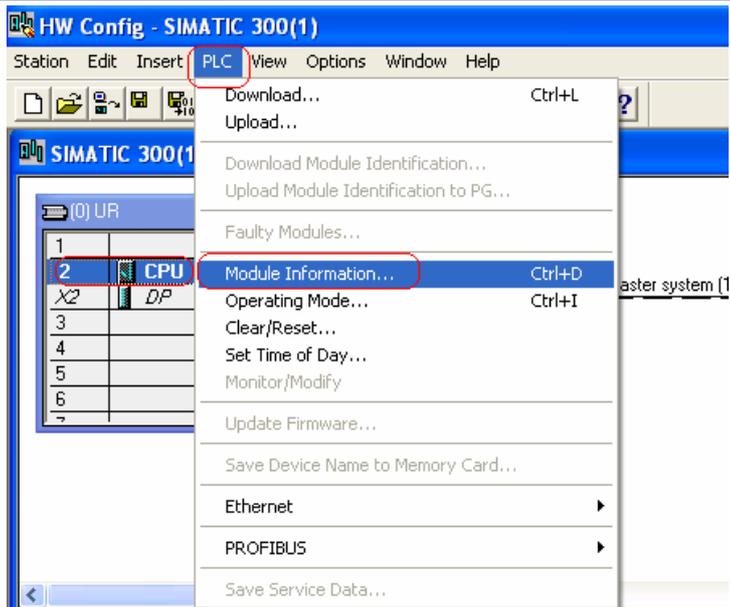
- 由于所附例程是免费的，任何用户可以免费复制或传播此程序例子。程序的作者对此程序不承担任何功能性或兼容性的责任，使用者风险自负
- 西门子不提供此程序例子的错误更改或者热线支持。
- 注意所附例程只注重功能的实现，即非唯一的方法也没有考虑限制范围，仅作参考。

1. 定时器

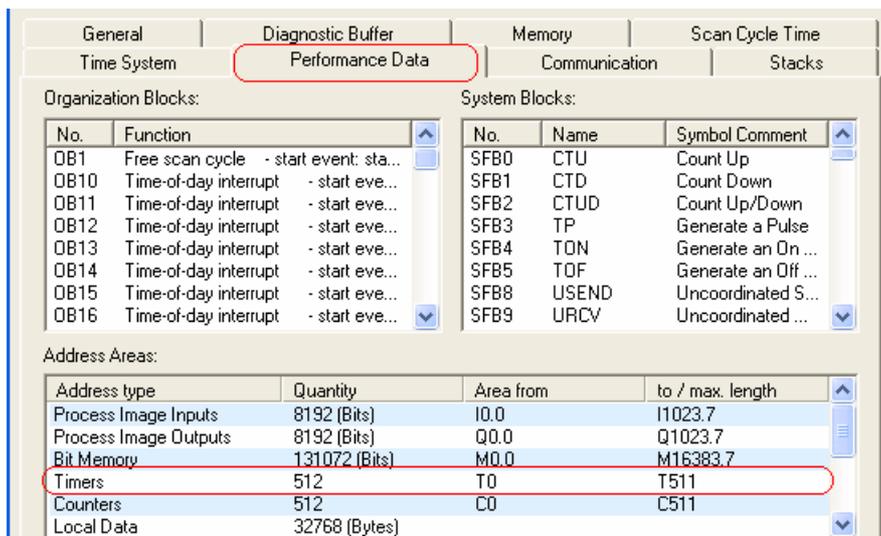
西门子 PLC 定时器最小时间单元为 10ms，可用的定时器指令列表如下:

S_PULSE	脉冲定时器
S_PEXT	扩展脉冲定时器
S_ODT	接通延迟定时器
S_ODTS	保持接通延迟定时器
S_OFFDT	断开延迟定时器
---(SP)	脉冲定时器线圈
---(SE)	扩展脉冲定时器线圈
---(SD)	接通延迟定时器线圈
---(SS)	保持接通延迟定时器线圈
---(SF)	断开延迟定时器线圈

西门子 PLC 定时器有数量限制，如果定时器不够用，可以使用 IEC 定时器，具体描述参见下面有关 IEC 定时器的描述。不同型号的 CPU 定时器的数量是不同的，可以在线从 CPU 的属性中查看，点击 CPU ，然后选择 PLC -> Module Information...，画面如下:



点击完成后，选择 Performance Data 按钮，查看 Timers 数量，画面如下：



对于不同定时器的编程使用可参见该指令的在线帮助，打开任一程序块，选中 Timers 中的指令，以 SODT 为例，然后按键盘的 F1 键，即为该指令的在线帮助，画面如下：

LAD/STL/FBD - Help on Ladder Logic

File Edit Insert File Edit Bookmark Options Help

Contents Index Back Print <<

S_ODT On-Delay S5 Timer

Symbol

English

German

Parameter	Parameter	Data Type	Memory Area	Description
English	German			
T no.	T-Nr.	TIMER	T	Timer identification number; range depends on CPU
S	S	BOOL	I, Q, M, L, D	Start input
TV	TW	S5TIME	I, Q, M, L, D	Preset time value
R	R	BOOL	I, Q, M, L, D	Reset input
BI	DUAL	WORD	I, Q, M, L, D	Remaining time value, integer format

在该在线帮助中找到 Example 部分的例程和描述，可以参考，画面如下：

Example

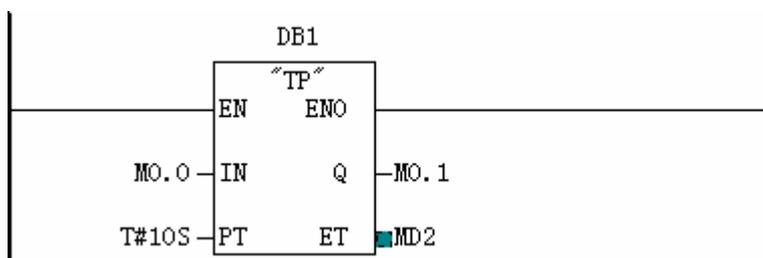
If the signal state of I0.0 changes from "0" to "1" (positive edge in RLO), the timer T5 will be started. If the time of two seconds elapses and the signal state at input I0.0 is still "1", the output Q4.0 will be "1". If the signal state of I0.0 changes from "1" to "0", the timer is stopped and Q4.0 will be "0" (if the signal state of I0.1 changes from "0" to "1", the time is reset regardless of whether the timer is running or not).

2. IEC 定时器（SFB3、SFB4、SFB5）

IEC 定时器没有数量限制，IEC 定时器通过系统功能块 SFB3（TP），SFB4（TON），SFB5（TOF）来实现，分别对应定时器指令的脉冲定时器，接通延迟定时器 and 断开延迟定时器，具体功能块的参数和使用方法可参考在线帮助。

2. 1 SFB3 (TP)

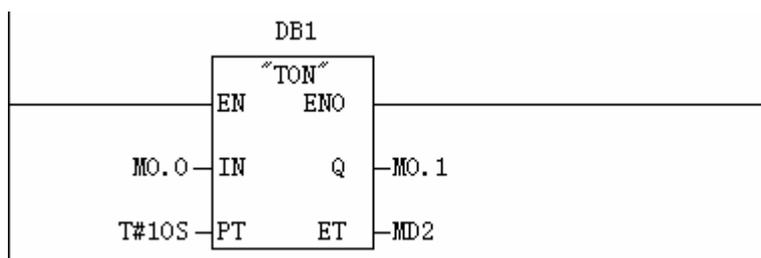
简单的程序如下：



DB1 为 SFB3 的背景数据块，当 MO.0 从 0 变为 1 定时器启动，10 秒钟内 MO.1 输出为 1，时间到后 MO.1 为 0，产生一个 10 秒的脉冲。如果在 10 秒钟内 MO.0 从 1 变为 0，则 MO.1 的输出不变，从 MD2 可以读出定时器已运行的时间。

2. 2 SFB4 (TON)

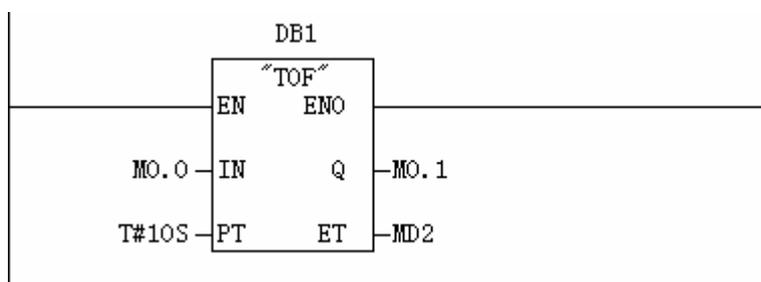
简单的程序如下：



DB1 为 SFB4 的背景数据块，当 MO.0 从 0 变为 1 定时器启动，定时时间到（10 秒钟）MO.1 输出为 1。如果定时时间未到 MO.0 从 1 变为 0，则定时器复位，从 MD2 可以读出定时器已运行的时间。

2. 3 SFB5 (TOF)

简单的程序如下：

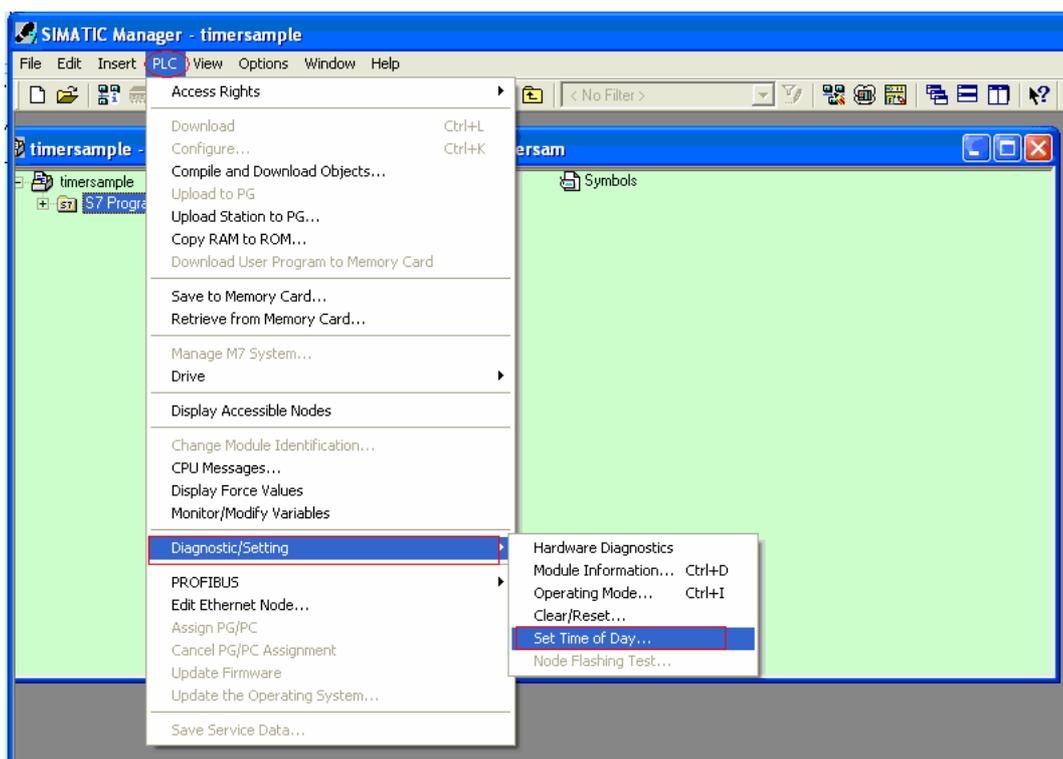


DB1 为 SFB5 的背景数据块，当 MO.0 为 1，则 MO.1 输出为 1，当 MO.0 从 1 变为 0 定时器启动，定时时间到（10 秒钟）MO.1 输出为 0。如果定时时间未到 MO.0 从 0 变为 1，则定时器复位，从 MD2 可以读出定时器已运行的时间。

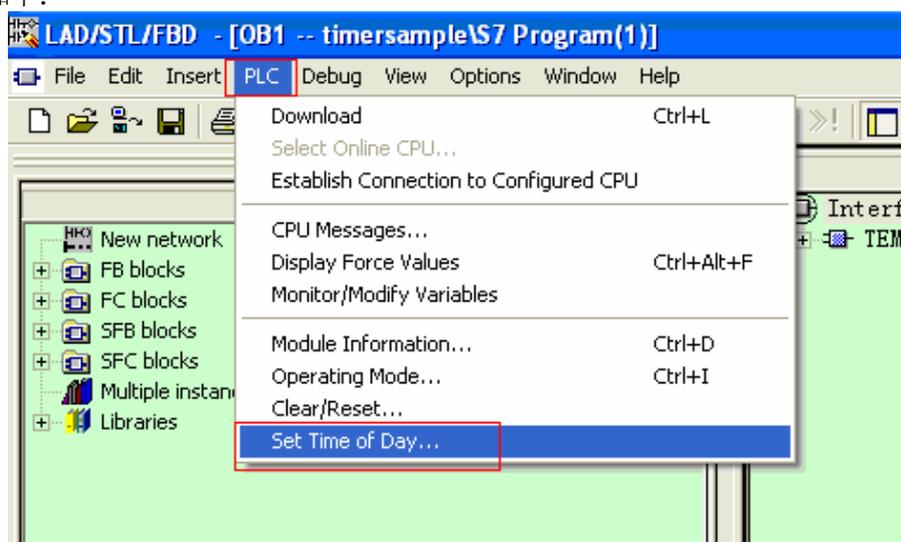
3. 系统时钟（SFC0、SFC1、OB1）

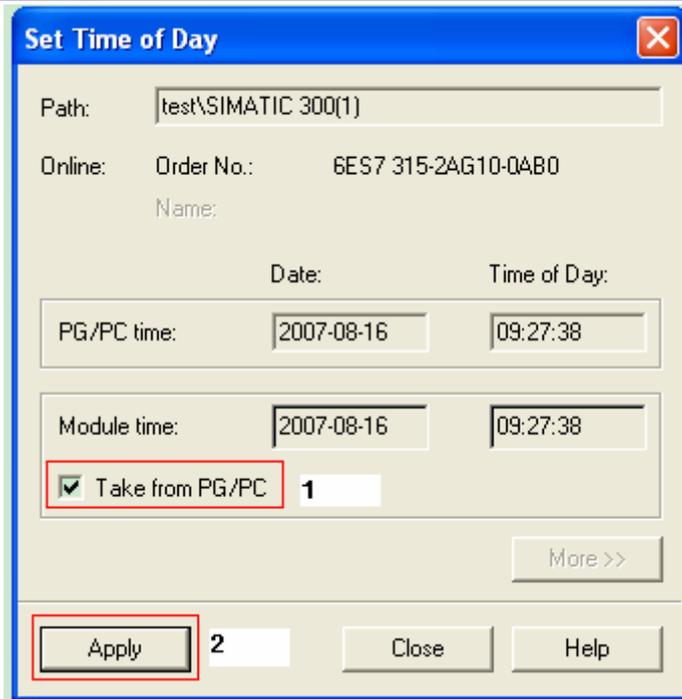
3.1 设置系统时钟的两种办法

- 1) 直接使用 STEP7 软件中的相关指令在联机的情况下直接设定系统时钟。如下图所示，选中项目中的站，使用 SIMATIC Manager→Diagnostic/setting→Set Time of Day 指令设置系统时钟：



也可打开程序块或硬件组态，使用 PLC→Set Time of Day 指令来在线设置系统时钟，画面如下：

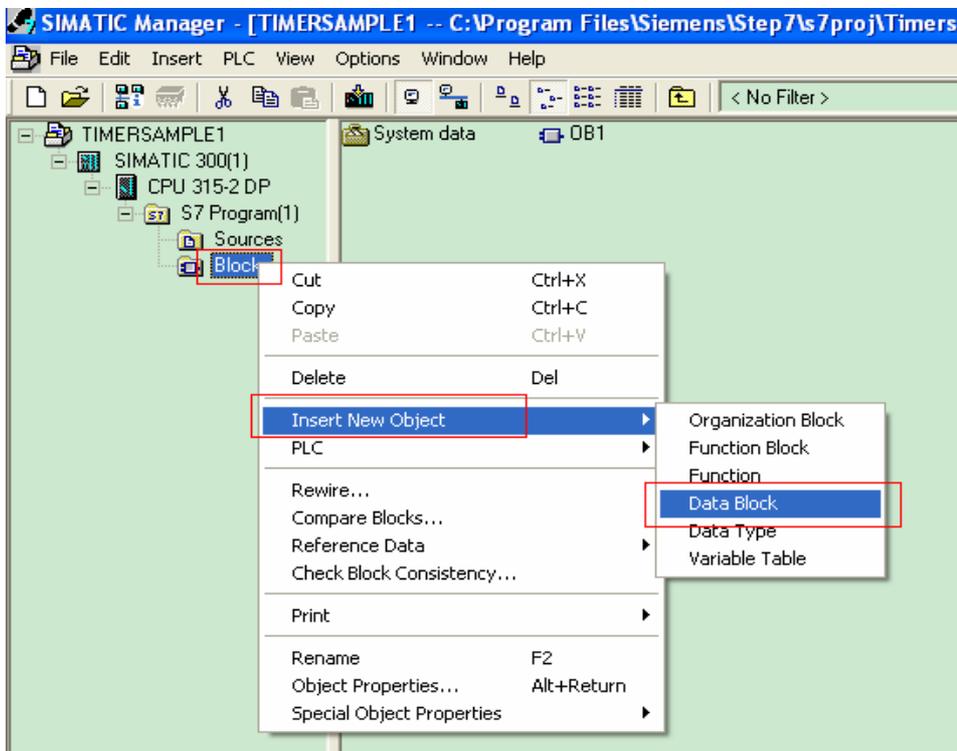




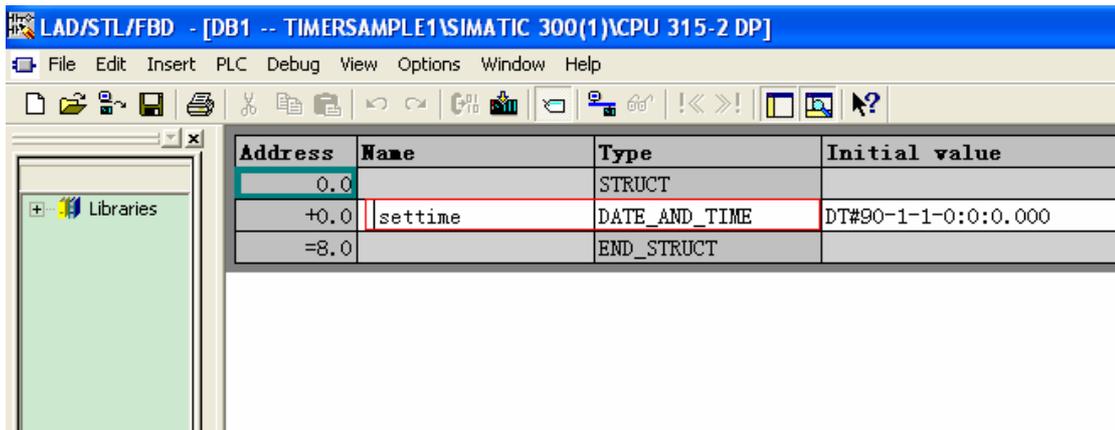
勾选“Take from PG/PC”，使用计算机时钟同步 PLC 时钟，然后点击“Apply”按钮完成。

2) 使用 SFC0 来设置系统时钟

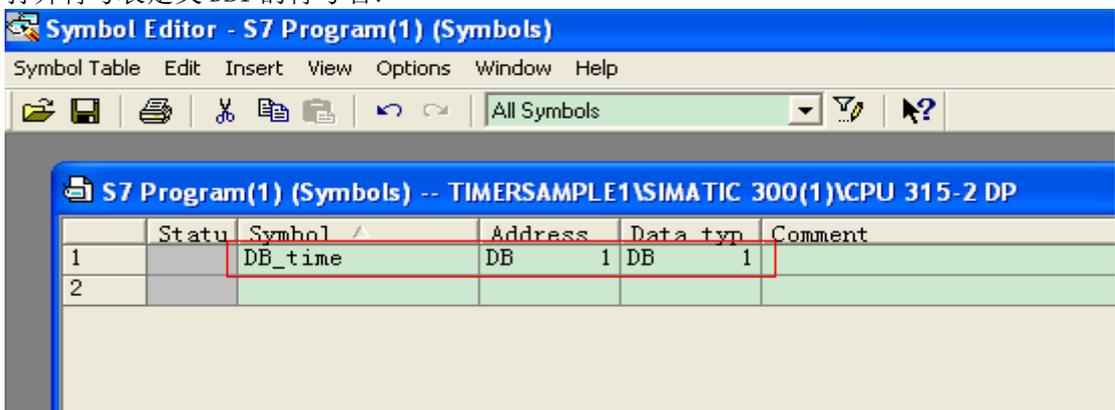
创建一个 DB 块 DB1



打开 DB1 块定义一个 DATE_AND_TIME 的变量如图所示：



打开符号表定义 DB1 的符号名：



这里先介绍一下 DATE_AND_TIME 变量的格式，其由八个字节组成分别代表年、月、日、时、分、秒、毫秒，最后一个字节 0-3 位代表星期 4-7 位表示毫秒，是以 BCD 码表示的。

DATE AND TIME

- Daylight saving time
- DB
- DB Register
- Deactivating
 - Test using Program Status
 - Time-of-Day Interrupt
- Debugging STL Source Files
- Declaration View of Data Blocks
- Declaring Local Variables
 - DB for the Sample Industrial Blending Process
- Declaring Parameters
 - FC for the Sample Industrial Blending Process
- Declaring Local Data
- Default Settings for the LAD/STL/FBD Program Editor
- Defective
 - CPU Operating Mode
- Defining
- Defining Logic Blocks
 - Individual Symbols
 - Symbols when Programming
 - the Trigger for Modifying Variables
 - the Trigger for Monitoring Variables
- Delayed Processing of Interrupts and Asynchronous
 - Example
- Delaying
 - Start Events
- Deleting
 - Deleting a Display Device
 - Deleting a language
 - Deleting a Project
 - Deleting Associated Values
 - Deleting in the RAM of the Programmable Controller
 - Deleting One or More Connections
 - Deleting Part of a Project
 - Deleting Sync Domains and Removing Nodes
 - STEP 7 Objects
 - a Force Job
 - in the Integrated EPROM
 - Library
 - Objects
 - on the Programmable Controller
 - Part of a Library
 - 57 Blocks on the Programmable Controller
 - Symbol Rows
- Describing the Individual Functional Areas
- Describing the Operator Console
 - Example of Industrial Blending Process
- Describing the Required Operator Displays and Controls
- Detailed Comparison
- Detectable Errors
- Determining the Cause of a STOP
- Determining the Communication Capabilities from the...

The following table shows the contents of the bytes that contain the date and time information for the example Thursday, December 25, 1993, at 8:12 and 34,567 seconds in the morning.

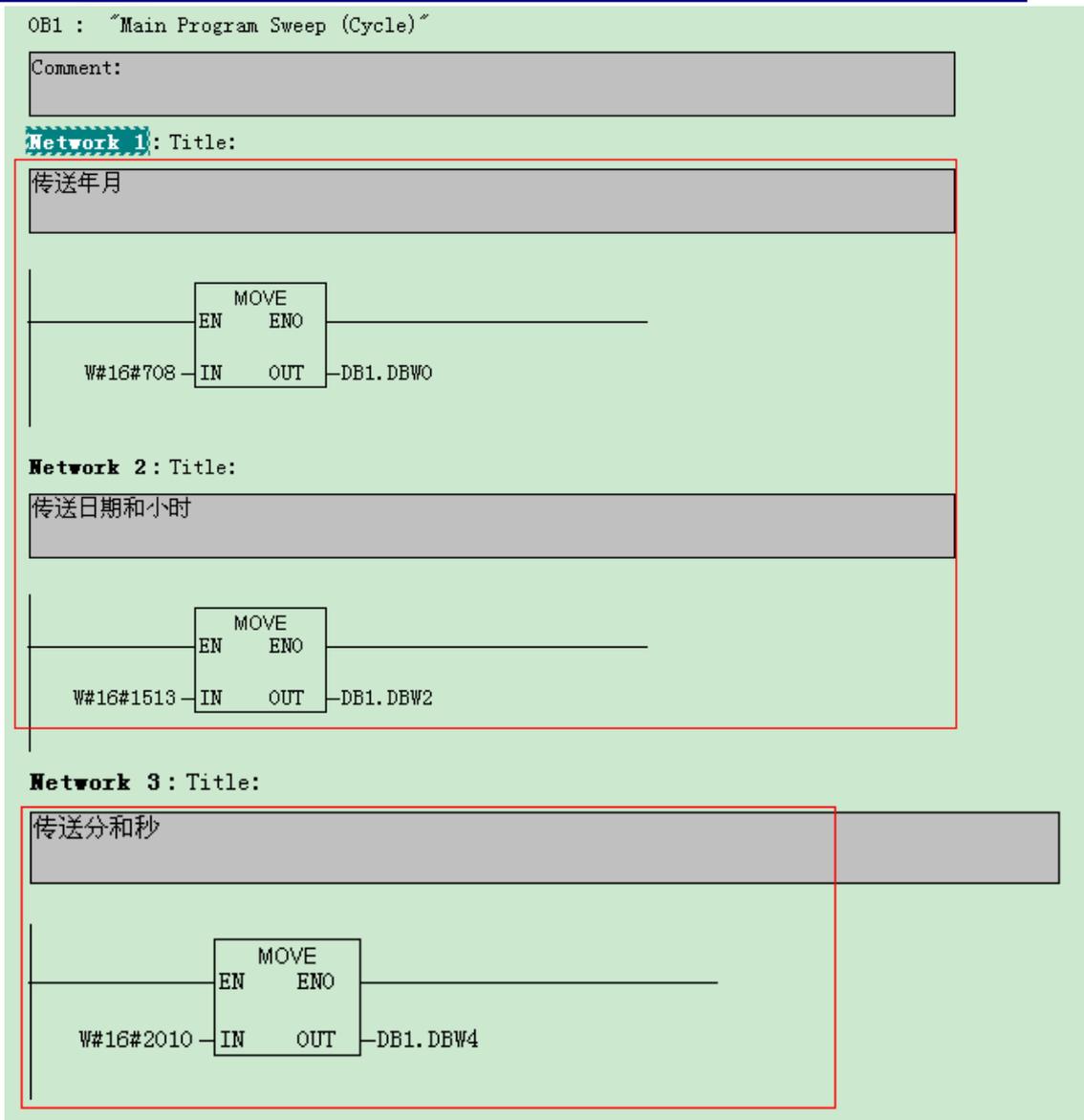
Byte	Contents	Example
0	Year	B#16#93
1	Month	B#16#12
2	Day	B#16#25
3	Hour	B#16#08
4	Minute	B#16#12
5	Second	B#16#34
6	Two most significant digits of MSEC	B#16#56
7 (4MSB)	Two least significant digits of MSEC	B#16#7
7 (4LSB)	Day of week 1 = Sunday 2 = Monday ... 7 = Saturday	B#16#_5

The permitted range for the data type DATE_AND_TIME is:

- min.: DT#1990-1-1-0:0:0
- max.: DT#2089-12-31-23:59:59

	Possible Value Range	BCD Code
Year	1990 - 1999	90 - 99
	2000 - 2089	00 - 89
Month	1 - 12	01 - 12
Day	1 - 31	01 - 31
Hour	00 - 23	00 - 23
Minute	00 - 59	00 - 59
Second	00 - 59	00 - 59
Millisecond	0 - 999	000 - 999
Day of week	Sunday - Saturday	1 - 7

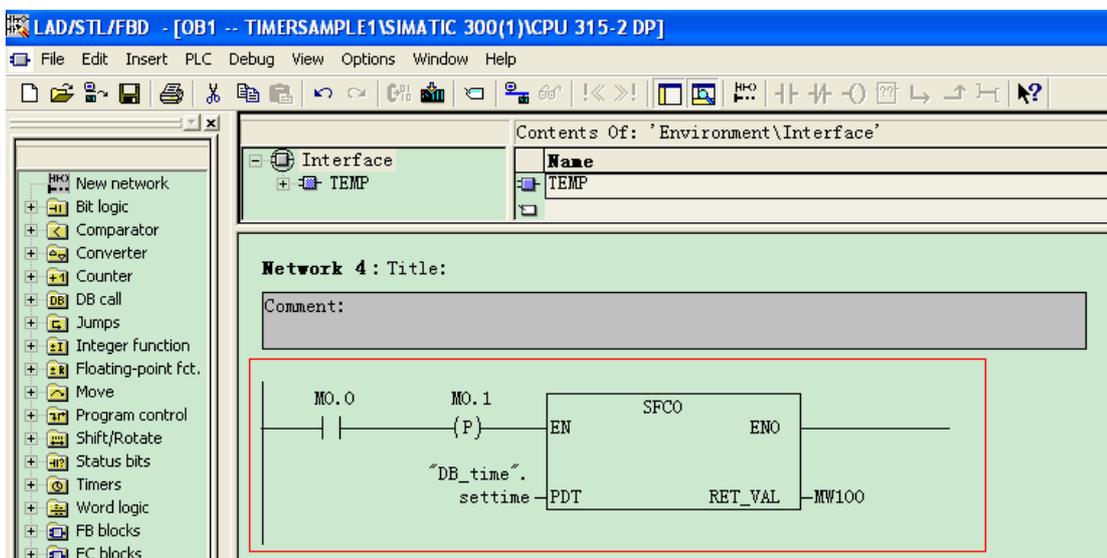
然后打开 OB1, 首先将需要设定的时间以 16 进制 BCD 码的形式赋值给定义的 DATE_AND_TIME 变量的各个字节, 最后一个字节不需设定系统会自己计算并赋值, 例如设定的时间为 07 年 8 月 15 日 13 点 20 分 10 秒。



在程序运行后打开监视和修改变量表即可观察到最后一个字节 DB1.DBW7 的低 4 位已经被系统自动计算为 4 即星期三。

	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBB 0		HEX	B#16#07	B#16#07
2	DB1.DBB 1		HEX	B#16#08	B#16#08
3	DB1.DBB 2		HEX	B#16#15	B#16#15
4	DB1.DBB 3		HEX	B#16#13	B#16#13
5	DB1.DBB 4		HEX	B#16#20	B#16#20
6	DB1.DBB 5		HEX	B#16#10	B#16#10
7	DB1.DBB 6		HEX	B#16#00	
8	DB1.DBB 7		HEX	B#16#04	
9					

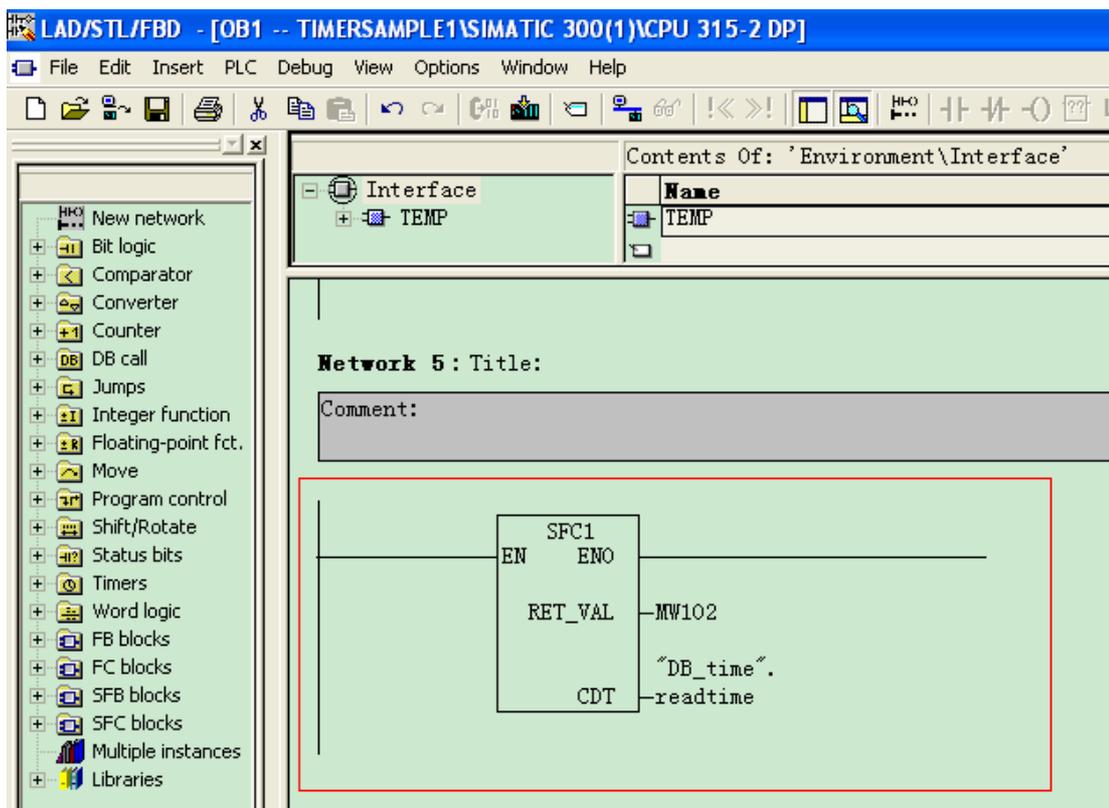
在程序中调用 SFC0, 将存放设定时间的 DATE_AND_TIME 变量以符号名的方式赋给 SFC0 的 PDT 形参, 返回变量赋值给 MW100, 这样当 M0.0 由 0 到 1 时 SFC0 被执行。



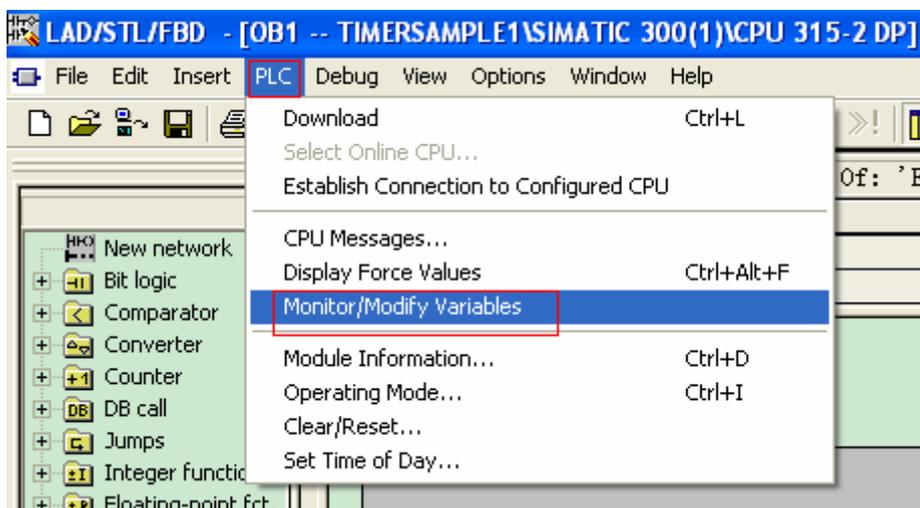
为了观察系统时间是否被正确设定, 我们在 DB1 中再定义一个 DATE_AND_TIME 的变量如图所示:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	settime	DATE_AND_TIME	DT#90-1-1-0:0:0.000	存放想设定的系统时间
+8.0	readtime	DATE_AND_TIME	DT#90-1-1-0:0:0.000	存放读取的系统时间
=16.0		END_STRUCT		

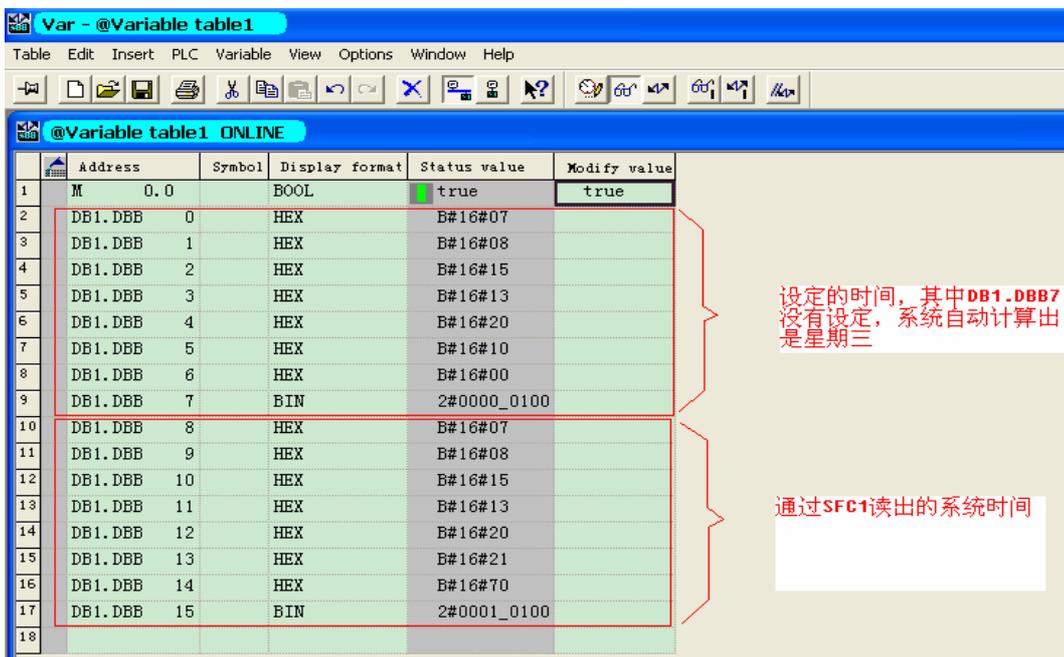
在 OB1 中调用 SFC1 读取系统时钟并将系统时间传送给” DB_time”.readtime 变量:



打开 Monitor/Modify Variables 表:



添加变量，通过  将 M0.0 的状态改为 true，然后通过  观察变量状态



Address	Symbol	Display format	Status value	Modify value
M 0.0		BOOL	true	true
DB1.DBB 0		HEX	B#16#07	
DB1.DBB 1		HEX	B#16#08	
DB1.DBB 2		HEX	B#16#15	
DB1.DBB 3		HEX	B#16#13	
DB1.DBB 4		HEX	B#16#20	
DB1.DBB 5		HEX	B#16#10	
DB1.DBB 6		HEX	B#16#00	
DB1.DBB 7		BIN	2#0000_0100	
DB1.DBB 8		HEX	B#16#07	
DB1.DBB 9		HEX	B#16#08	
DB1.DBB 10		HEX	B#16#15	
DB1.DBB 11		HEX	B#16#13	
DB1.DBB 12		HEX	B#16#20	
DB1.DBB 13		HEX	B#16#21	
DB1.DBB 14		HEX	B#16#70	
DB1.DBB 15		BIN	2#0001_0100	

注：DB1.DBB 7 的值为 2#0000_0100，表示星期三。

注：DB1.DBB 8-15 的值为通过 SFC1 读出的系统时间。

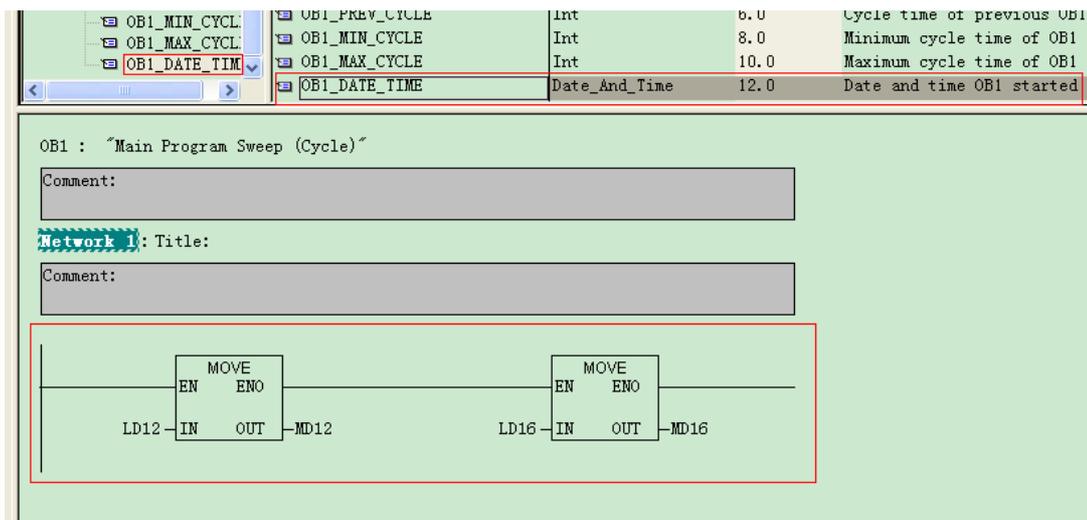
程序见所附例程 TIMERSAMPLE1

3.2 读取系统时钟

我们可以通过使用 SFC1 或 OB1 中的临时变量 OB1_DATE_TIME 来读取系统时钟。

- 1) 使用 SFC1 的方法前面已经介绍不再重复。
- 2) 使用 OB1 中的临时变量 OB1_DATE_TIME 来读取系统时钟

OB1 的每个循环周期都读取系统时钟并存放在 OB1_DATE_TIME 临时变量中，如下图所示我们将 LB12 开始的 8 个字节赋值给 MB12 开始的 8 个字节。



通过监视变量表即可看到实时的系统时钟。

The screenshot shows the 'Var - @Variable table1' window in SIMATIC Manager. The table displays the following data:

	Address	Symbol	Display format	Status value	Modify value
1	MB 12		HEX	B#16#07	
2	MB 13		HEX	B#16#08	
3	MB 14		HEX	B#16#15	
4	MB 15		HEX	B#16#14	
5	MB 16		HEX	B#16#30	
6	MB 17		HEX	B#16#05	
7	MB 18		HEX	B#16#94	
8	MB 19		BIN	2#0100_0100	
9					

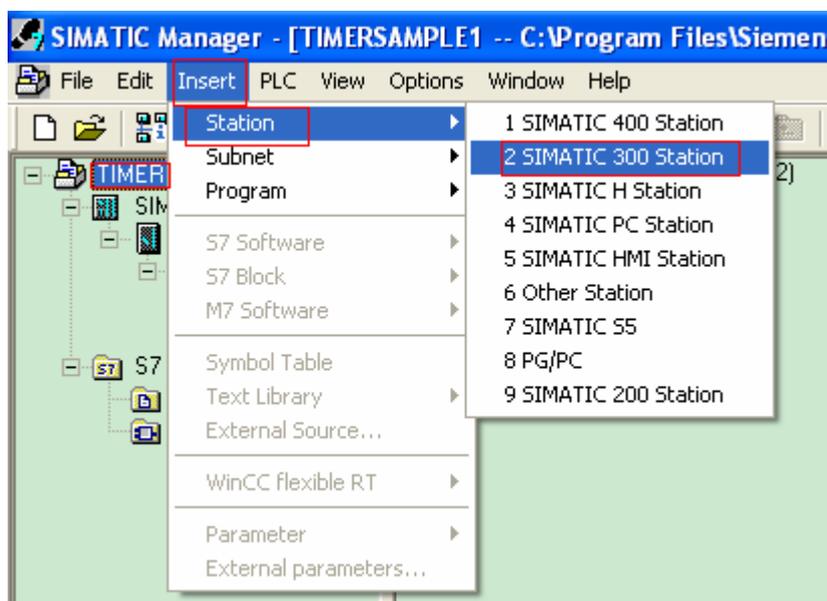
程序见所附例程 TIMERSAMPLE1

4. 循环中断（OB35）、延迟中断（OB20）、时间日期中断（OB10）

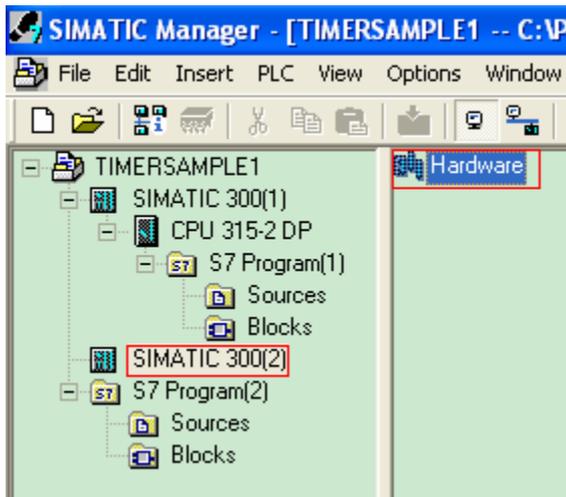
4.1 循环中断（OB35）

具体步骤如下：

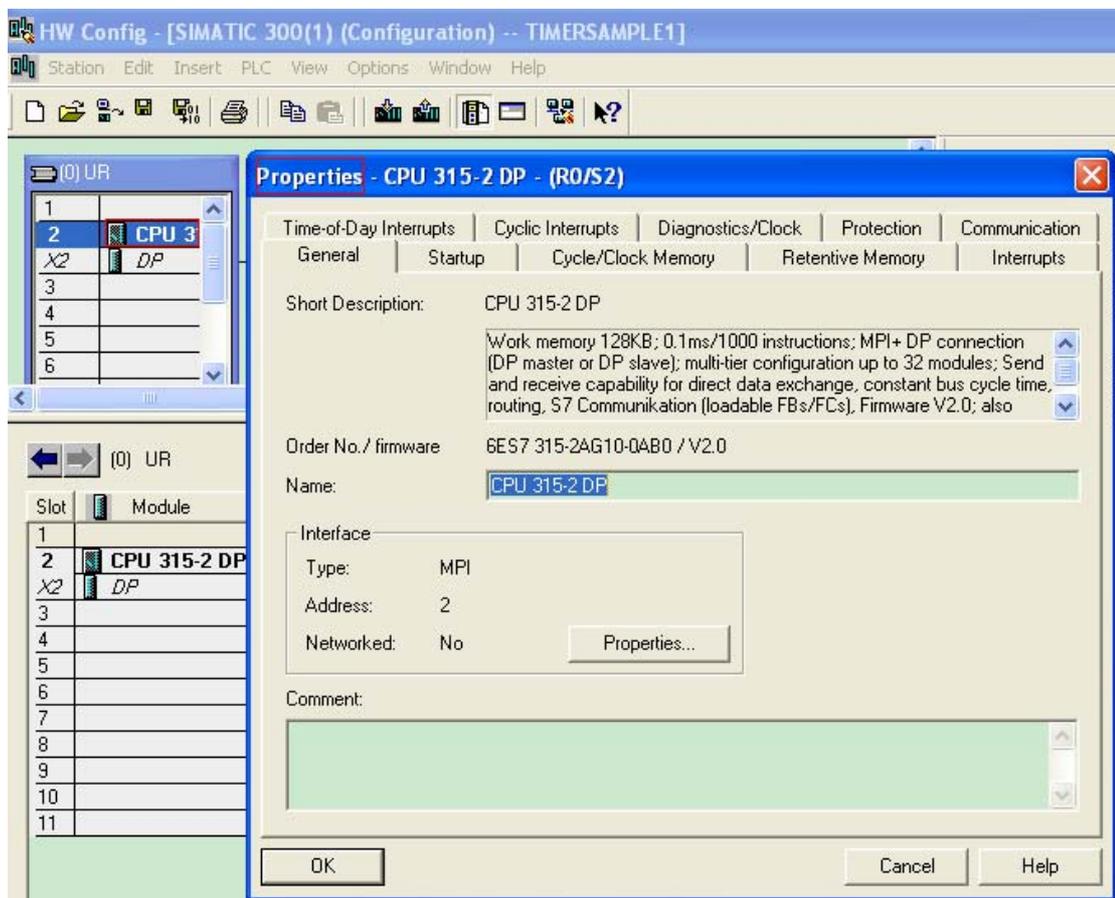
1) 创建一个项目并插入一个 300 站



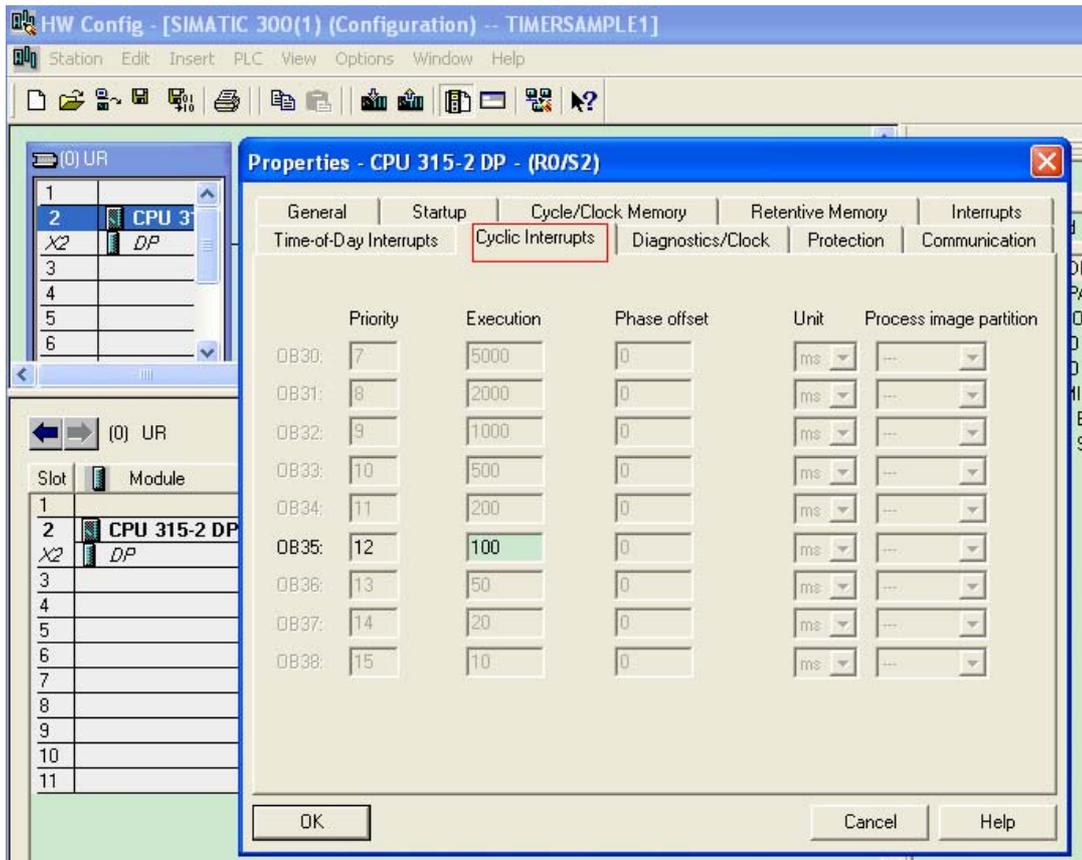
2) 双击打开 300 站，再双击 Hardware



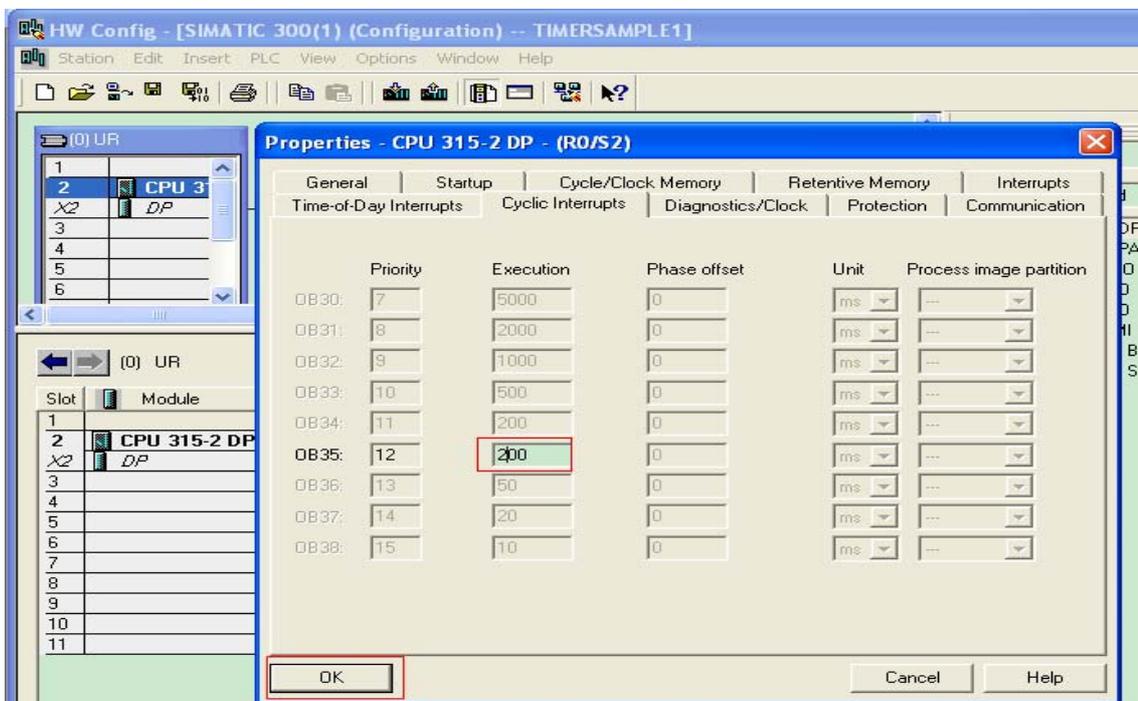
3) 配置 300 站并双击 300CPU 打开 Properties 窗口。



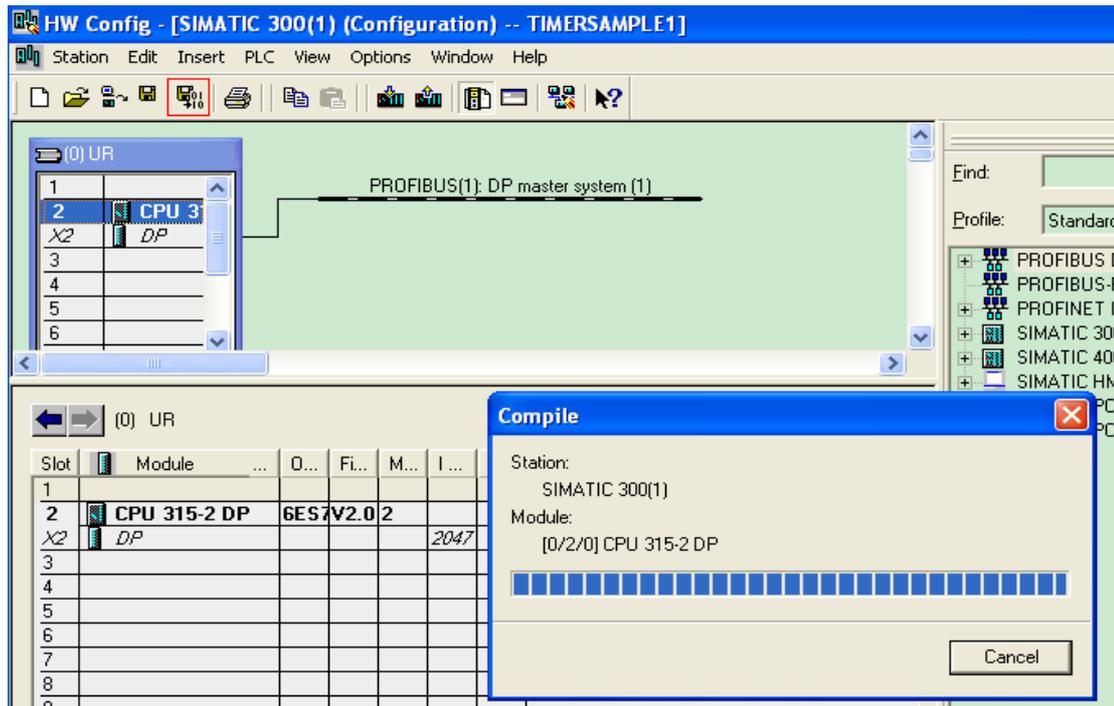
4) 选择 Cycle interrupts.



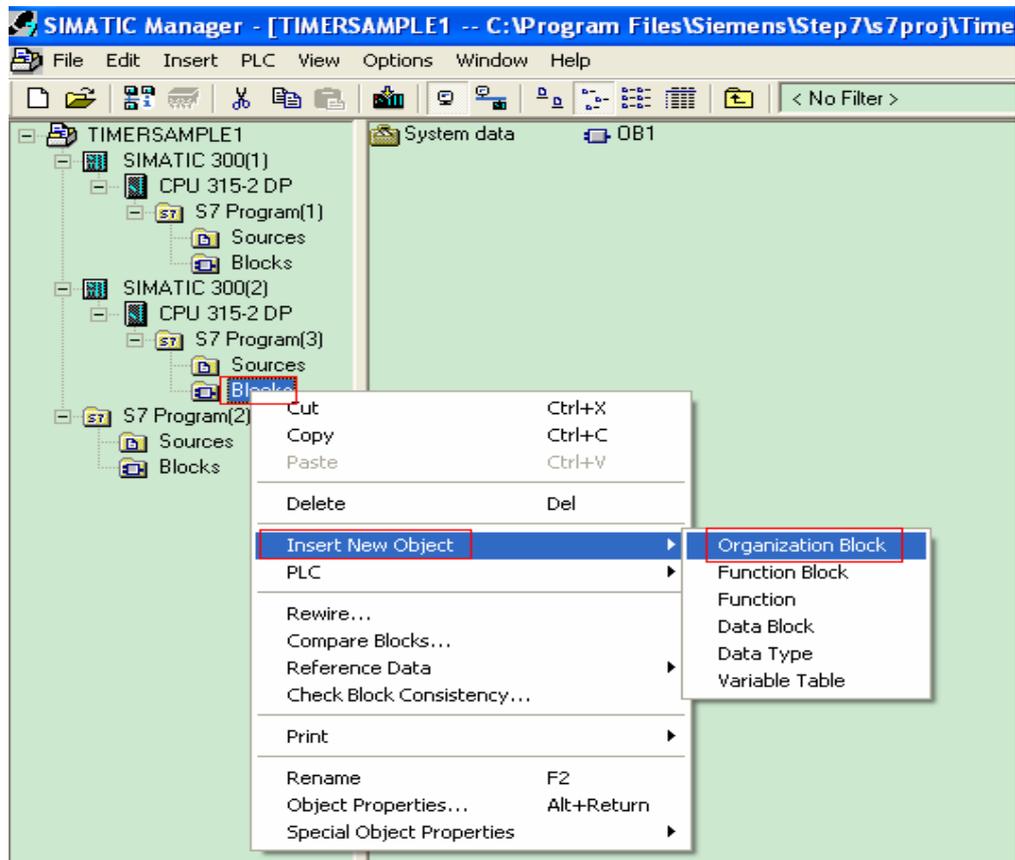
5) 将 OB35 对应的时间修改为需要的时间并点击 OK 确认。

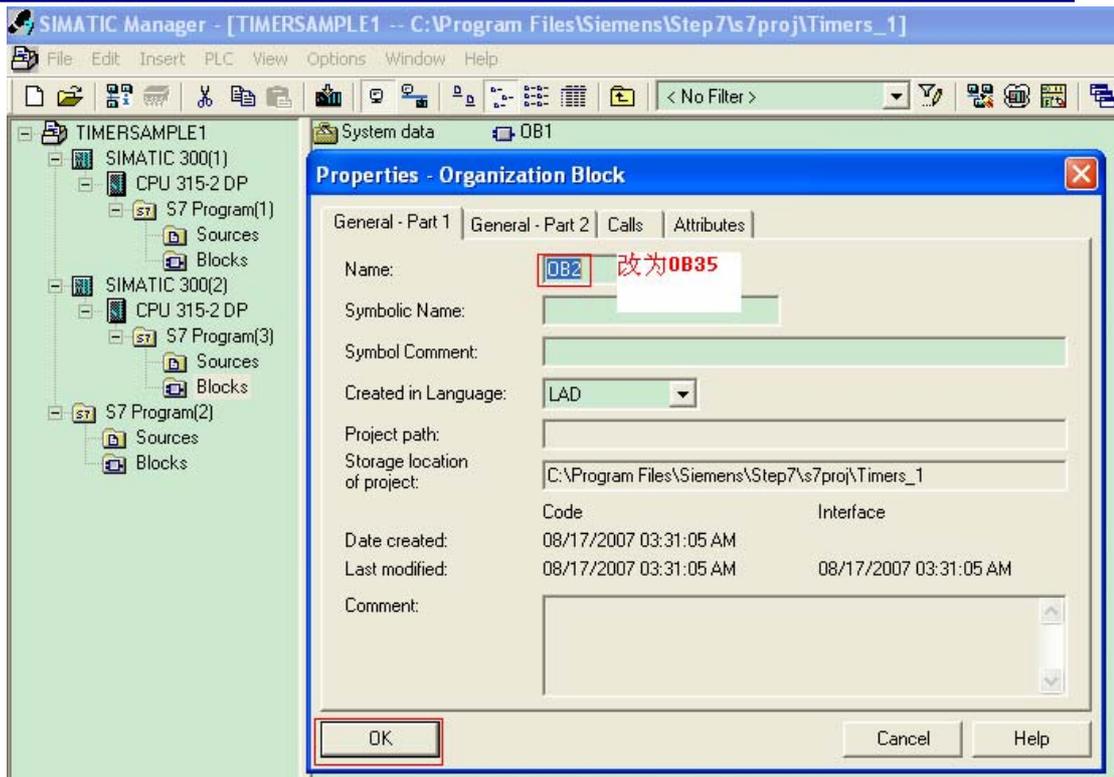


6) 编译保存

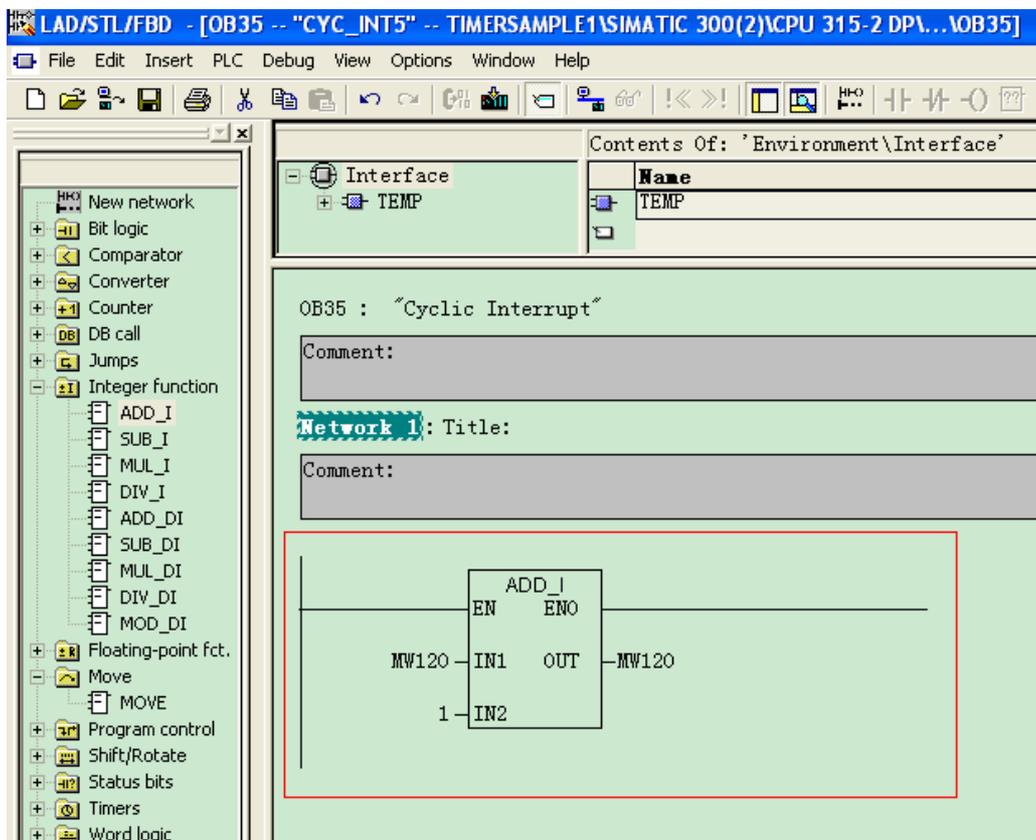


7) 通过 BLOCKS->Insert New Object->Organization Block 插入 OB35





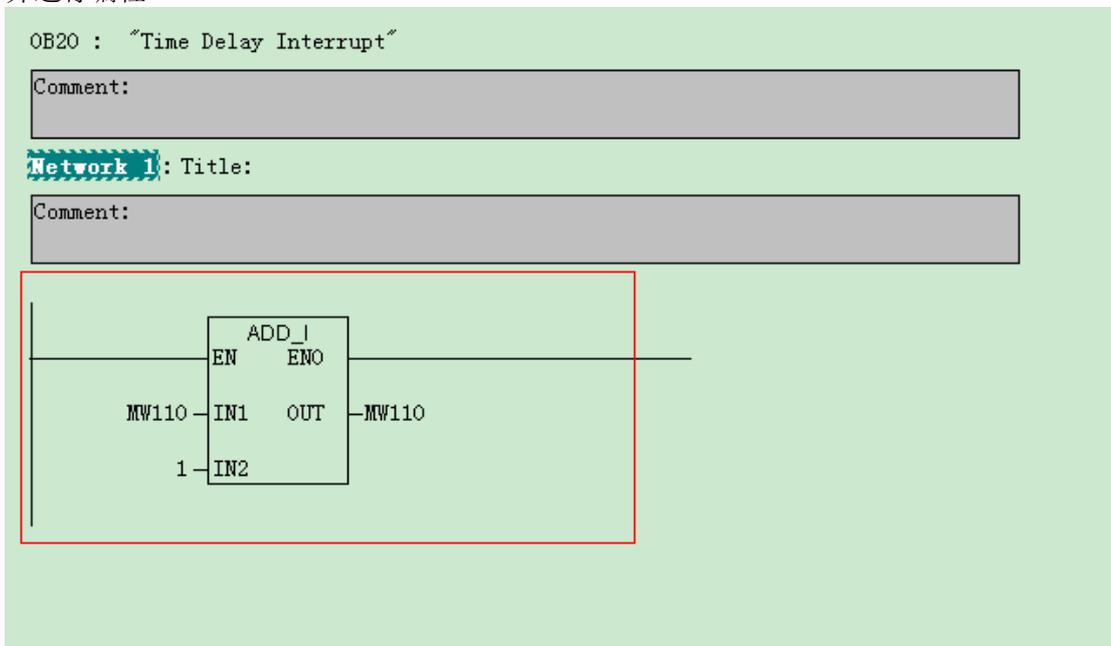
8) 双击打开 OB35 进行编程即可。



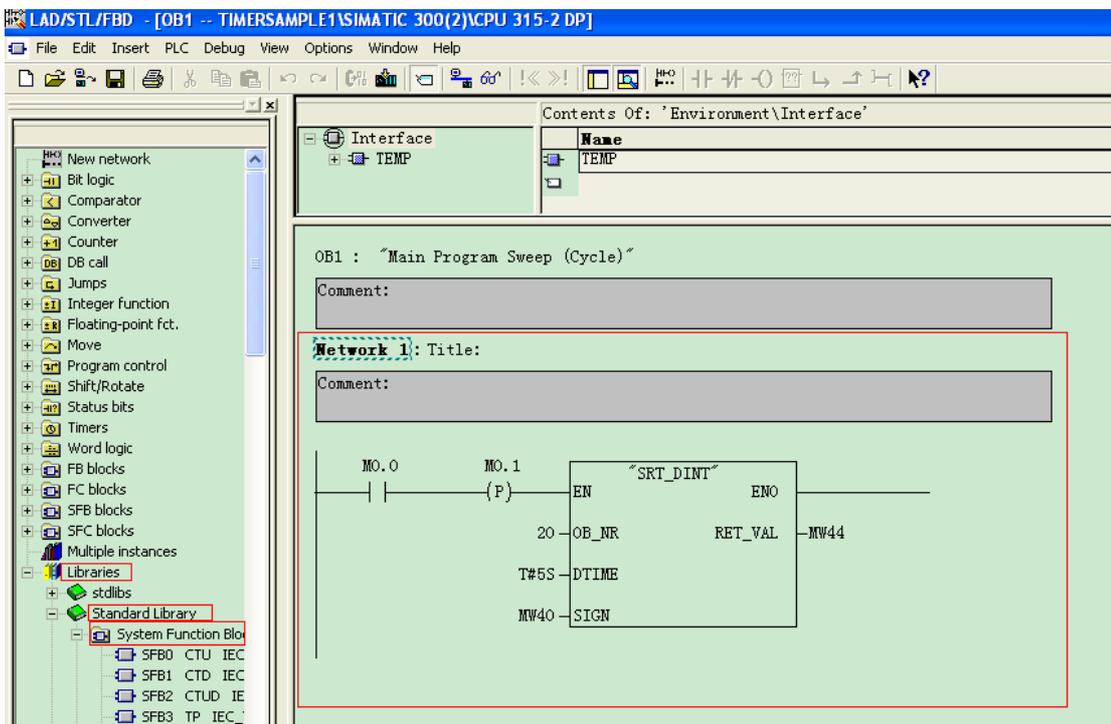
这样 OB35 每隔 200ms 执行一次，MW120 值加 1。

4.2 延迟中断 (OB20)

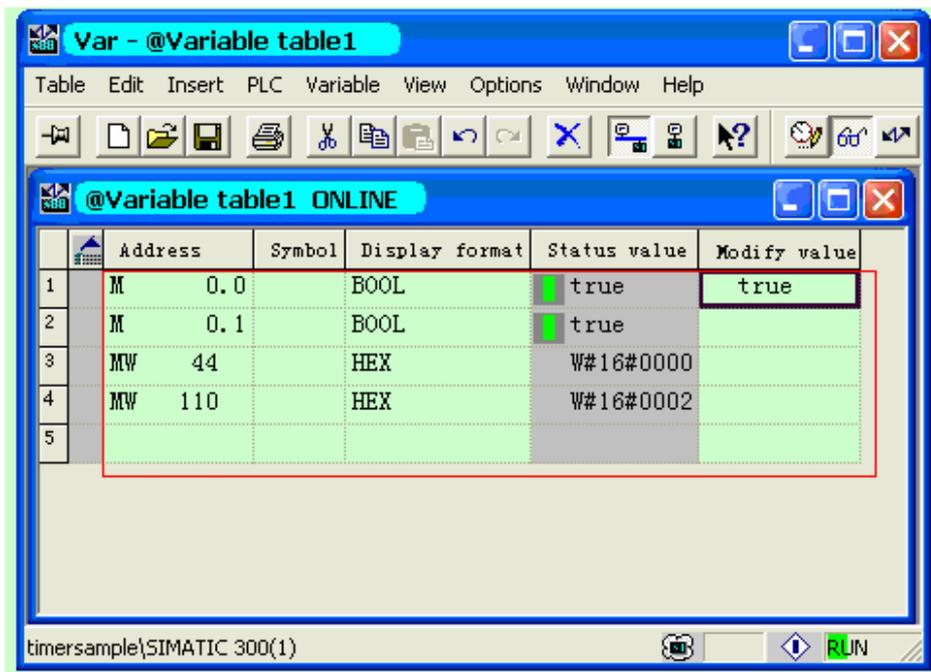
1) 通过 BLOCKS->Insert New Object->Organization Block 插入 OB20, 双击打开 OB20 并进行编程。



2) 打开 OB1, 调用 SFC32.



如图所示每当 M0.0 由 0 变 1 时, 经过 5S 后 OB20 被执行, MW110 执行了加 1 操作。

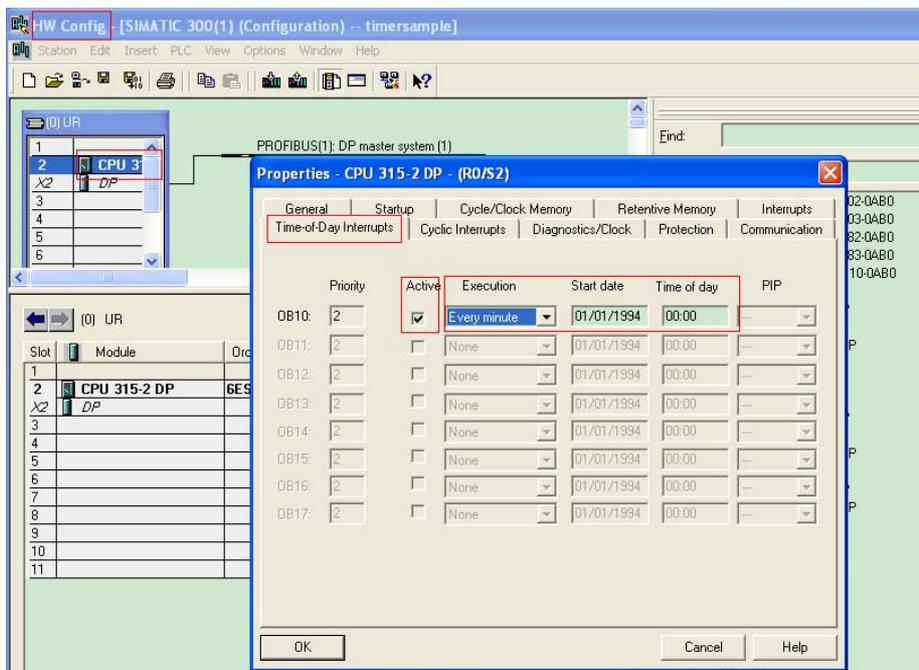


程序见所附例程 TIMERSAMPLE1

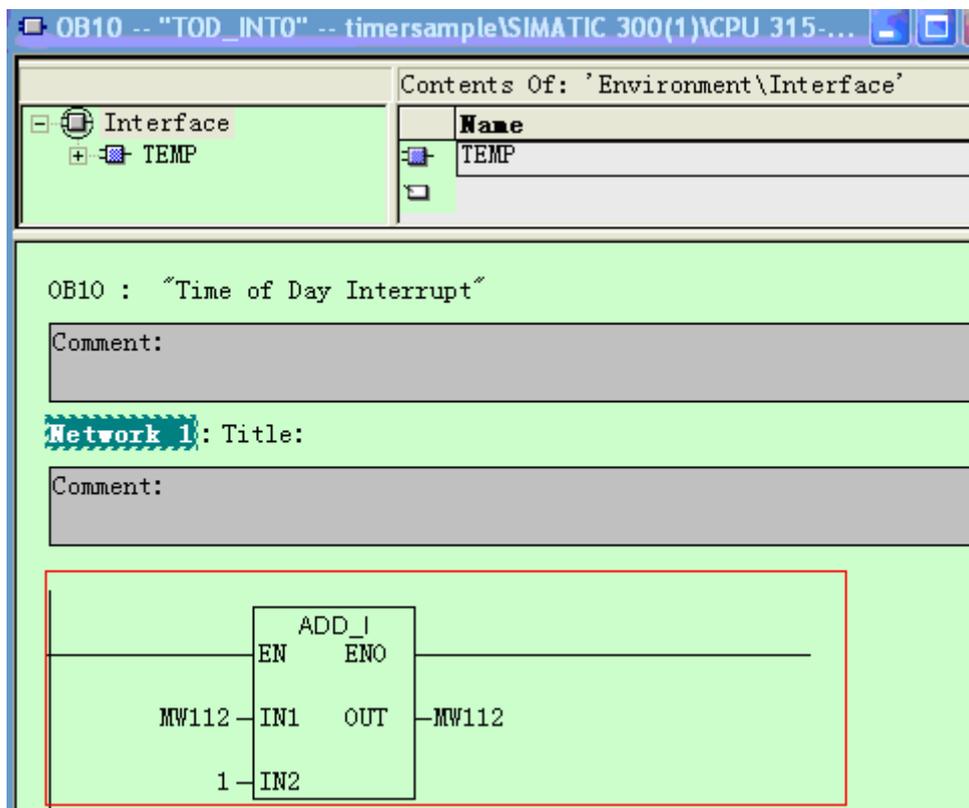
4. 3 时间日期中断 (OB10)

OB10 的执行可由 3 种办法来设定:

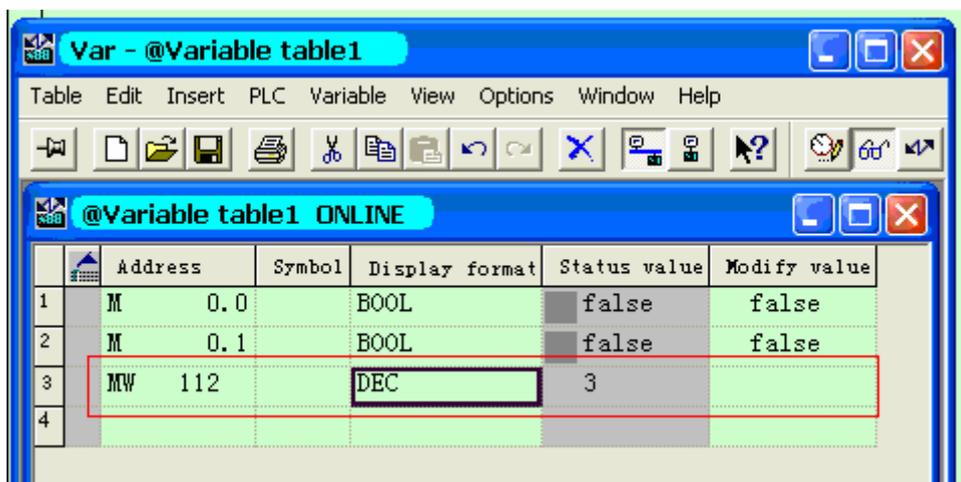
- 1) 方法一: 如下图所示在 HW CONFIG 窗口下双击 300 的 CPU 打开 Properties, 设定开始执行的时间和执行频率, 然后选中 Active 下的复选框, 点击 OK.



通过 BLOCKS->Insert New Object->Organization Block 插入 OB10, 双击打开 OB10 并进行编程.

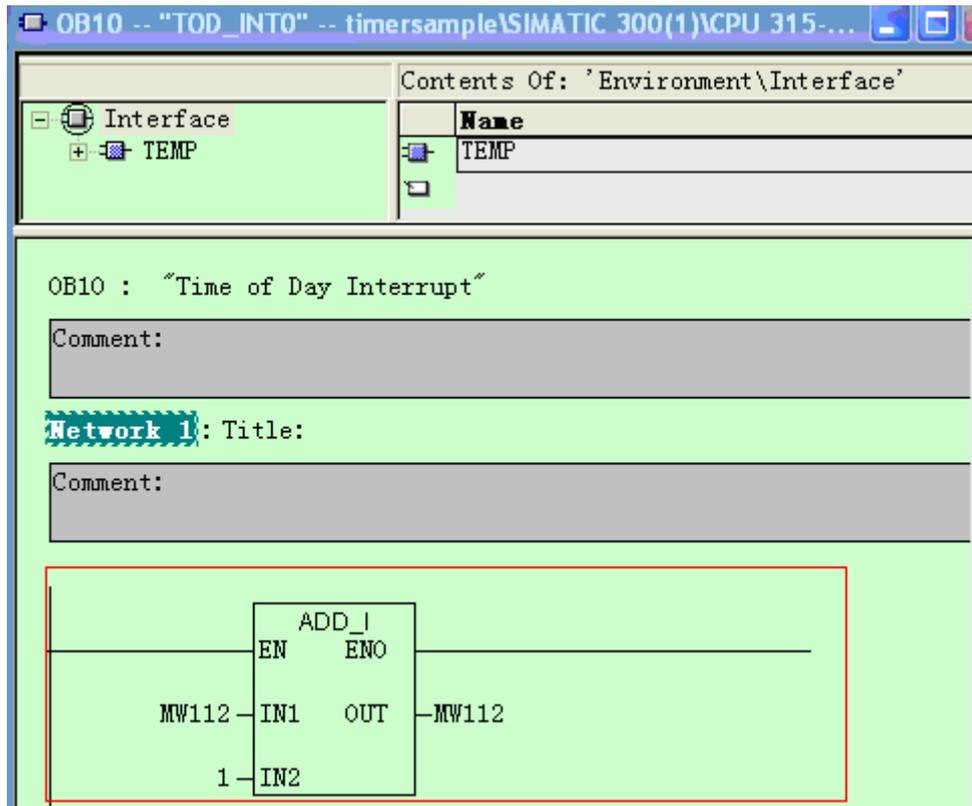


当程序下载到 PLC 中后 OB10 将按设定执行, 通过监视变量表可观察 OB10 的执行情况。



2) 方法二: 如下图所示在 HW CONFIG 窗口下双击 CPU300 打开 Properties, 设定开始执行的时间和执行频率, 不选中 Active 下的复选框, 点击 OK.

编程 OB10:

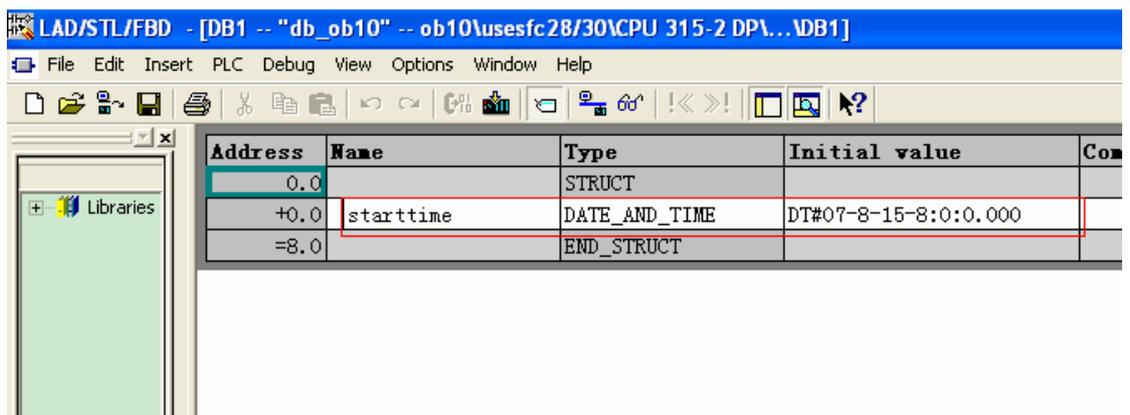


编译下载即可。

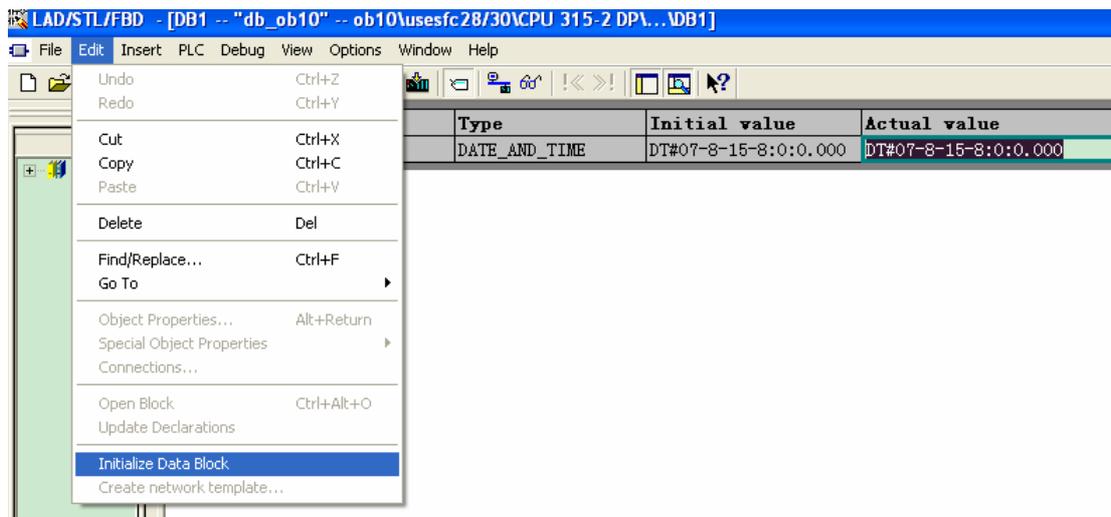
注意:SFC30 只需执行一次,即 CPU 停机后再启动不需要再次调用 SFC30。

- 3) 方法三: 在 HW Config 中不作配置,调用 SFC28 和 SFC30

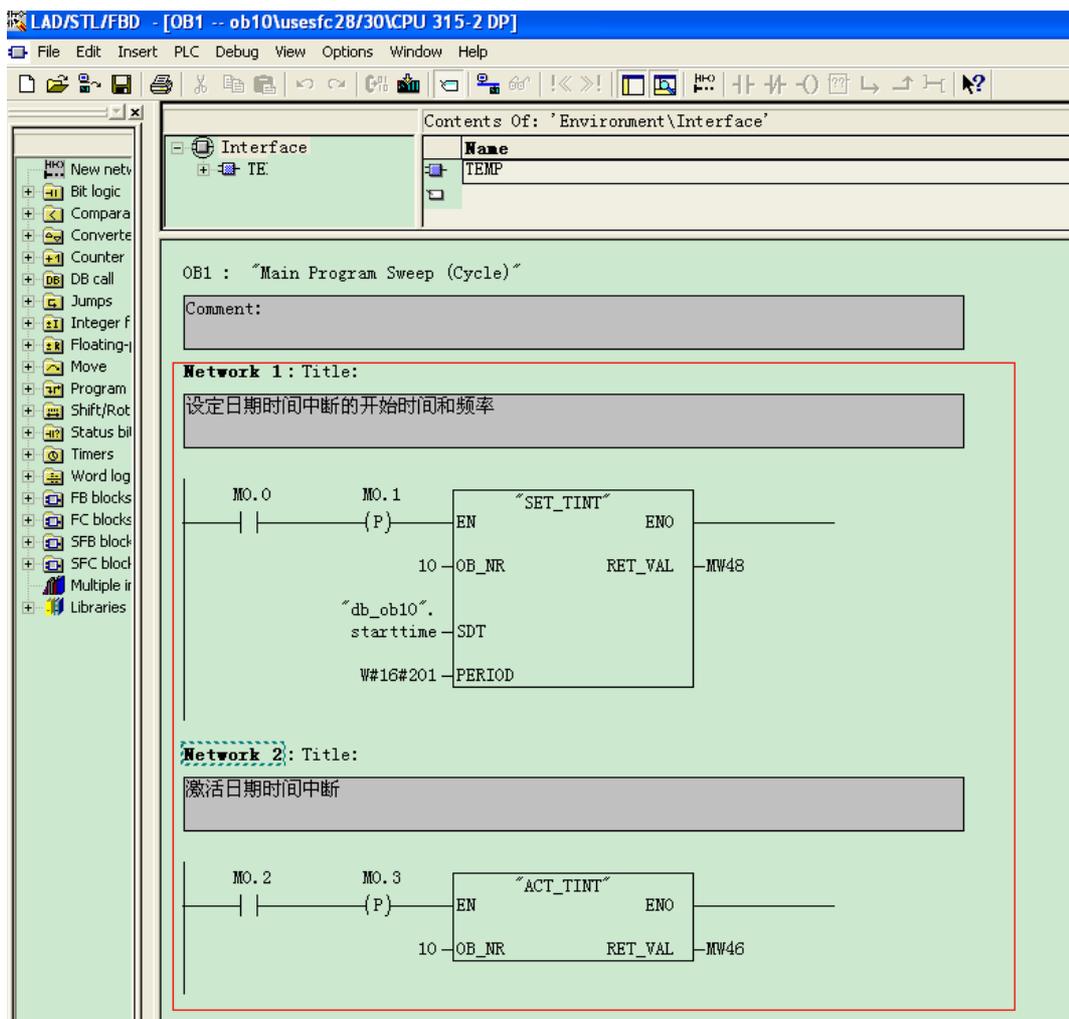
在 DB1 中定义一个 DATE AND TIME 的变量



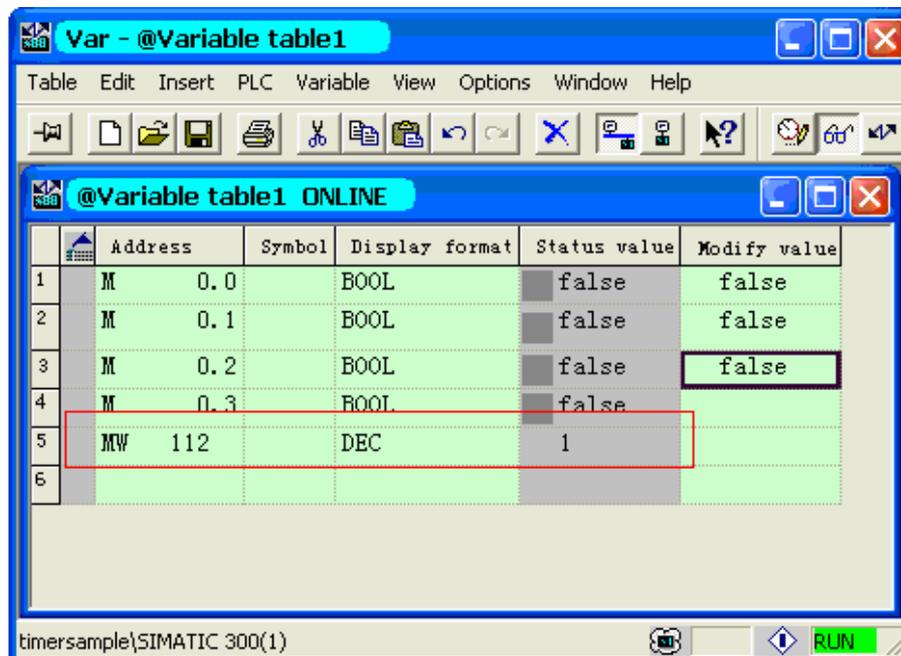
初始化数据块，将初值作为设定值：



调用 SFC28 和 SFC30



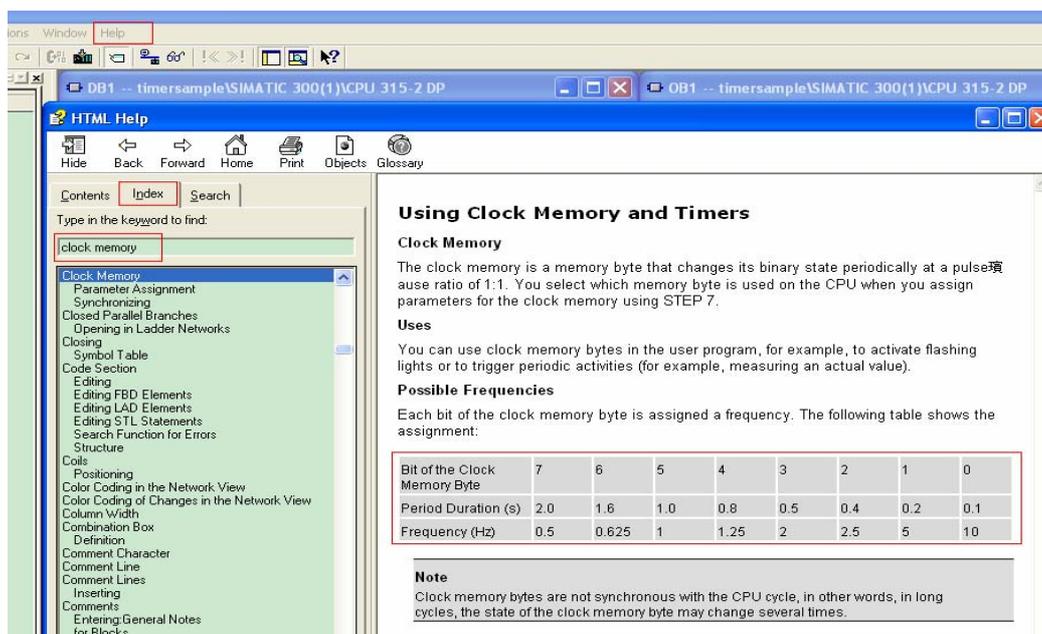
当 M0.0 由 0 变 1 时 SFC28 被调用, M0.2 由 0 变 1 时 SFC30 被调用, OB10 将按设定方式执行。



同样 SFC28 也只需执行一次即可。参见所附例程 ob10。

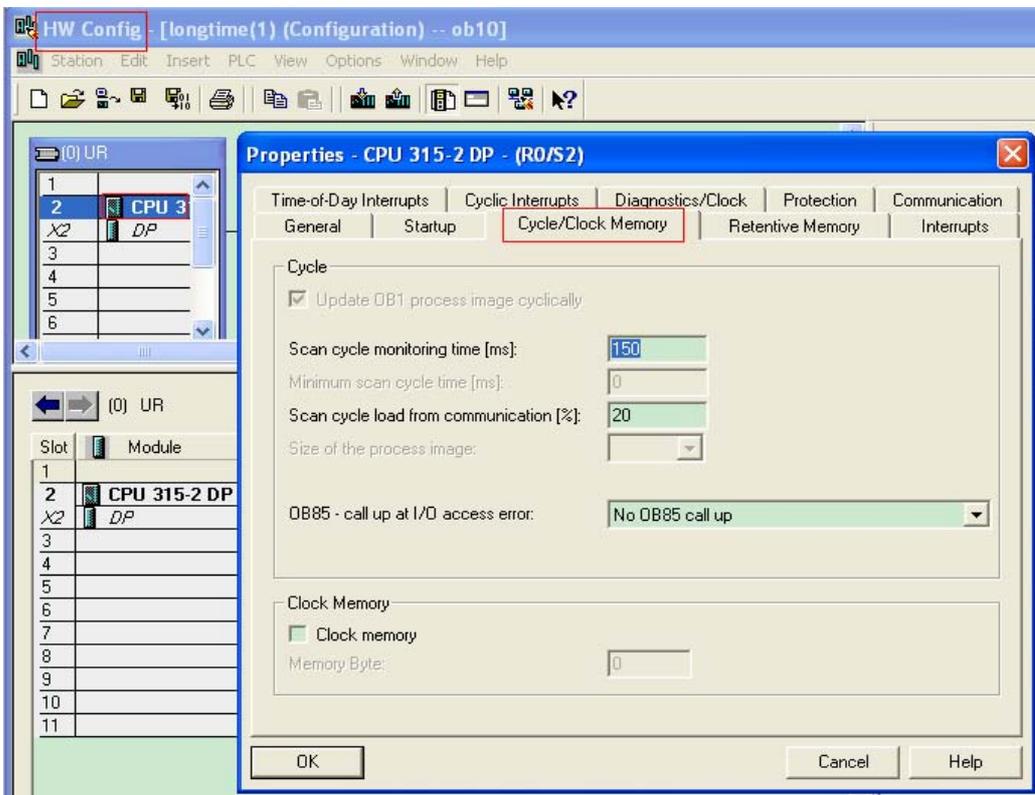
5. 时钟寄存器 (Clock Memory)

西门子 300 和 400 的 PLC 提供一个 Clock memory 的字节, 每一位以不同的频率执行 0 到 1 的变化, 用户可根据需要使用, 对于具体描述可通过 HELP 菜单打开如下图所示的帮助界面。

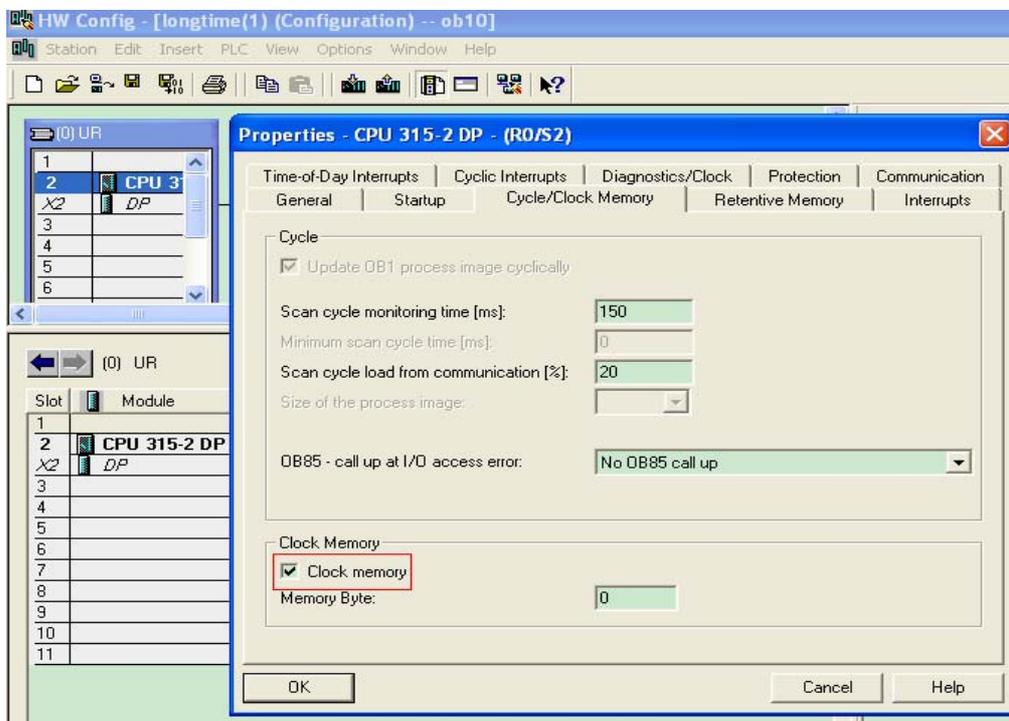


设定方法如下:

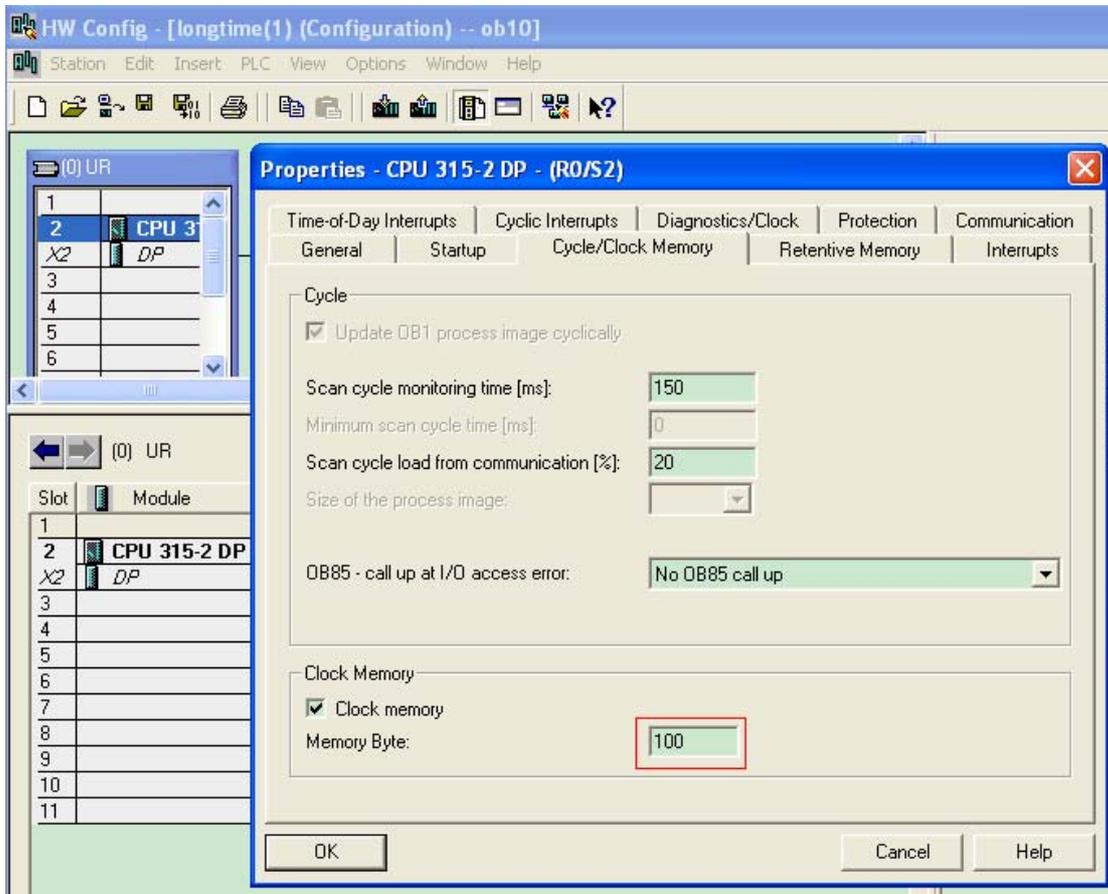
打开 HW CONFIG 界面, 双击 CPU300 打开 Properties 窗口选择 Cycle/Clock memory 选项



选中如下图所示 Clock memory 的复选框



在 Memory byte 对话框中输入需要使用的 Memory byte 号，如输入 100 则 MB100 被设定为 Memory byte。



M100.0–M100.7 将按不同的频率变化

Each bit of the clock memory byte is assigned a frequency. The following table shows the assignment:

Bit of the Clock Memory Byte	7	6	5	4	3	2	1	0
Period Duration (s)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Frequency (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

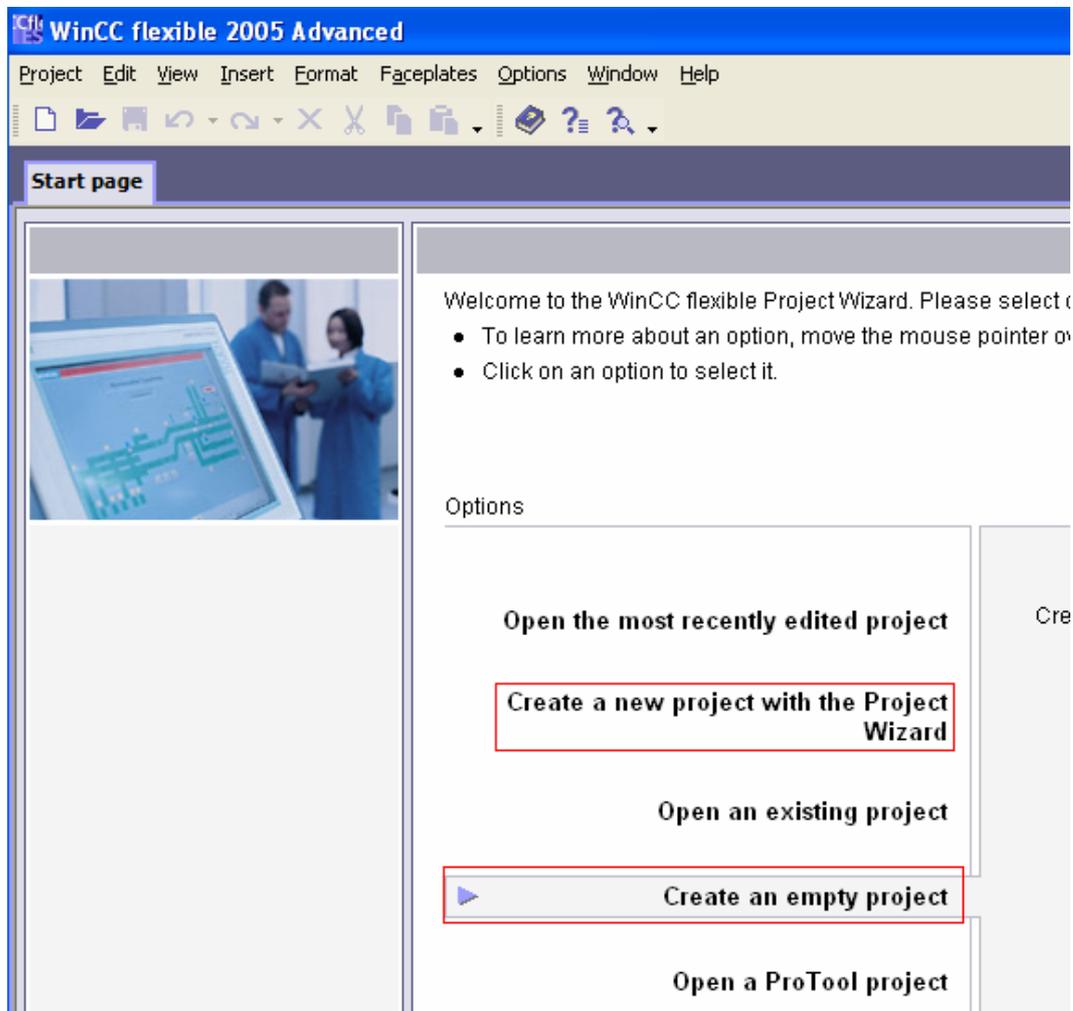
Note

Clock memory bytes are not synchronous with the CPU cycle, in other words, in long cycles, the state of the clock memory byte may change several times.

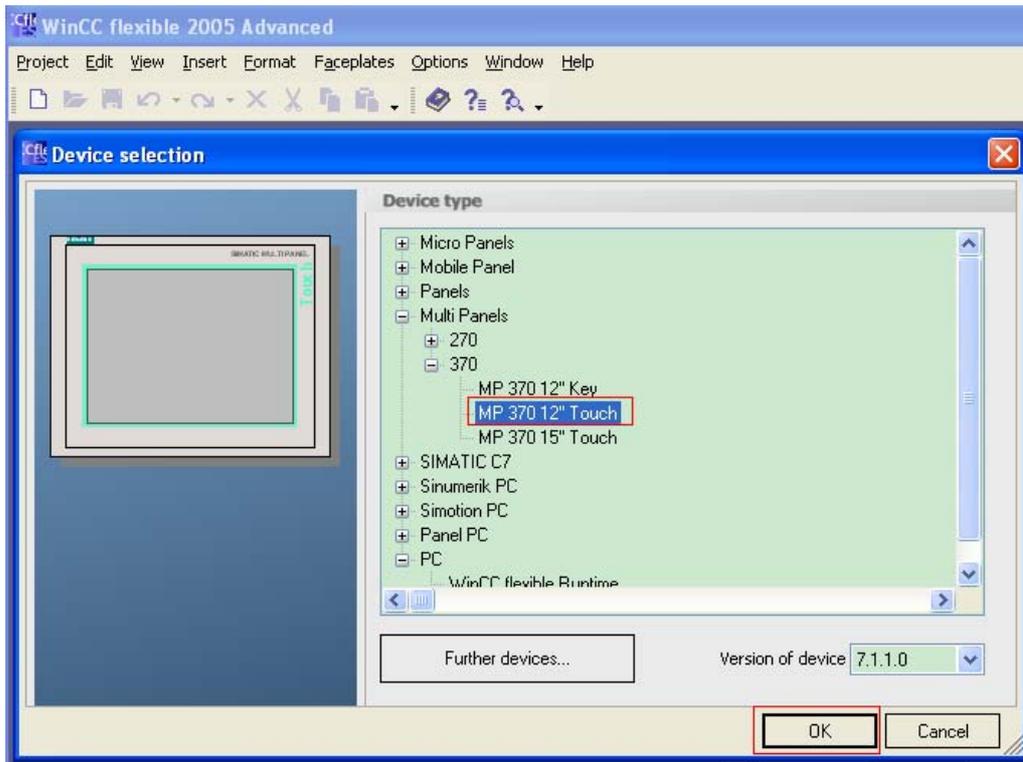
6. FAQ:

6. 1 如何从上位机或者触摸屏设定 Time 或 S5Time 值?

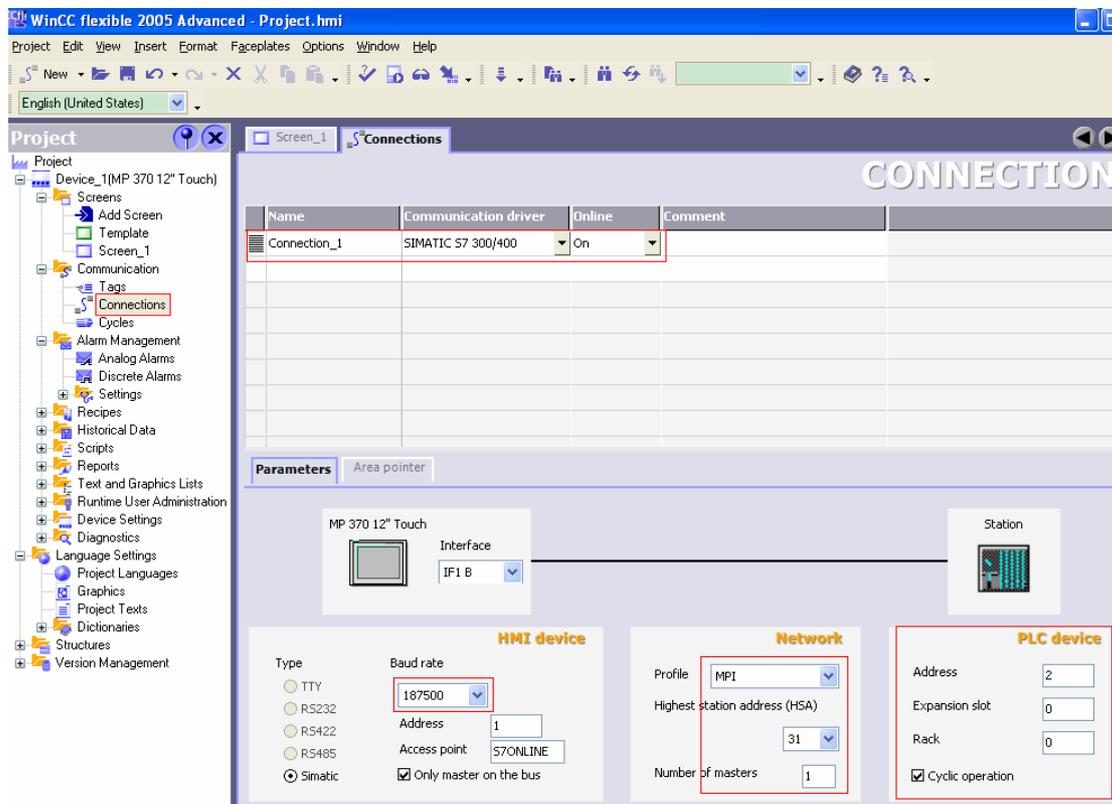
首先必须在上位机上安装 WINCC 或 wincc flexible, 然后打开软件如 wincc flexible 利用向导或手动创建一个项目:



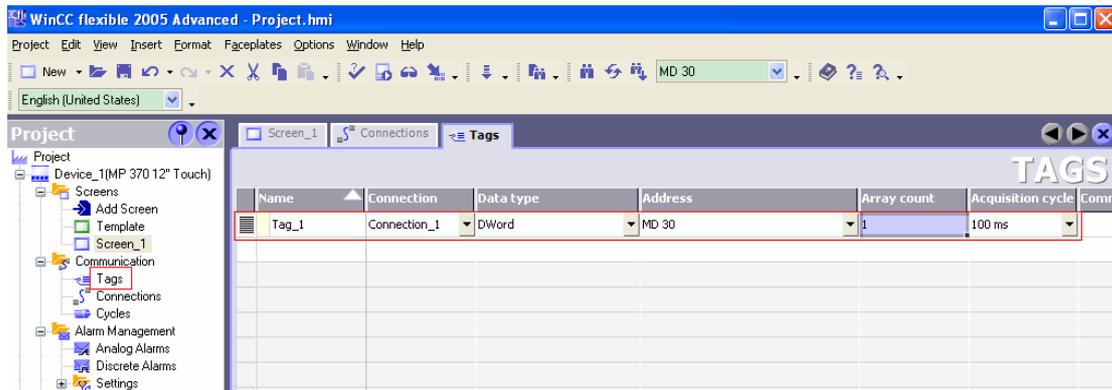
选择设备:



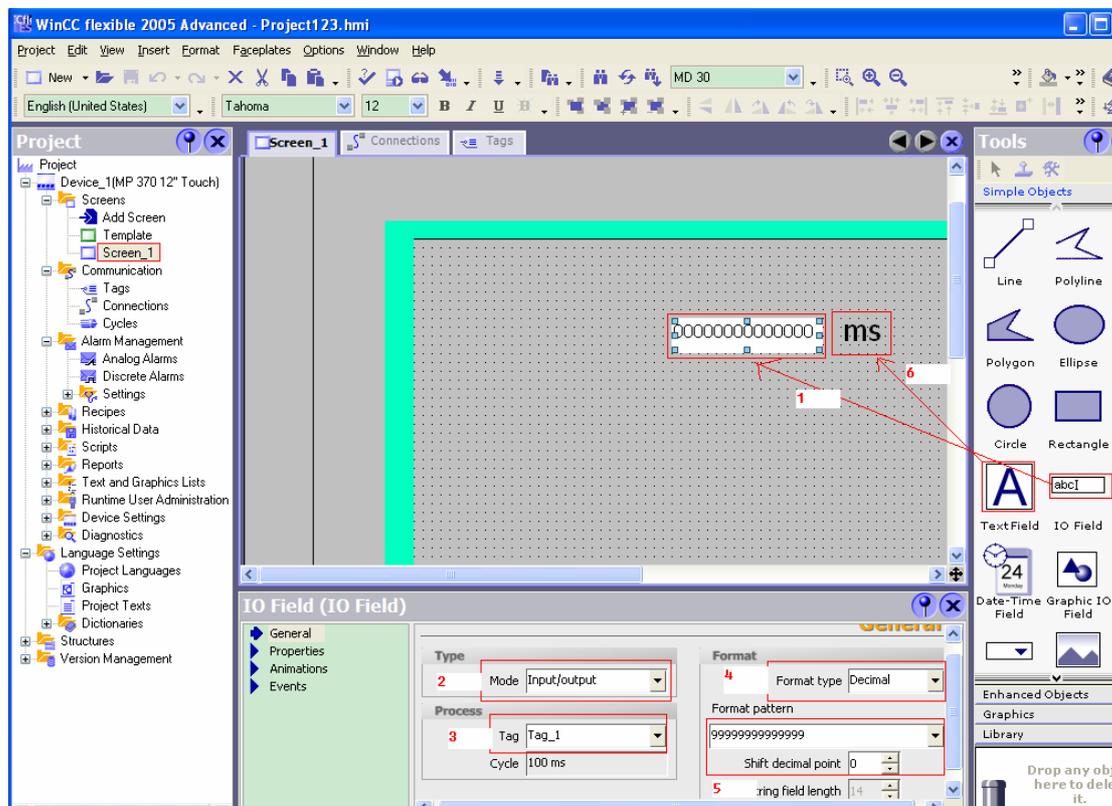
双击 connections, 在打开的页面中定义连接并定义相关的通讯参数。



双击 Tags，在打开的页面中定义 Tag：



打开画面，插入一个 INPUT/OUTPUT 域并选择对应格式。最后插入文本输入域。



完善项目并下载。此处是以毫秒为例，客户当然可以定义自己的时基，然后在 PLC 程序中换算即可，例如此处定义为秒则在 PLC 程序中可对对应变量乘以 1000 转化成毫秒再送给 TIME 变量即可

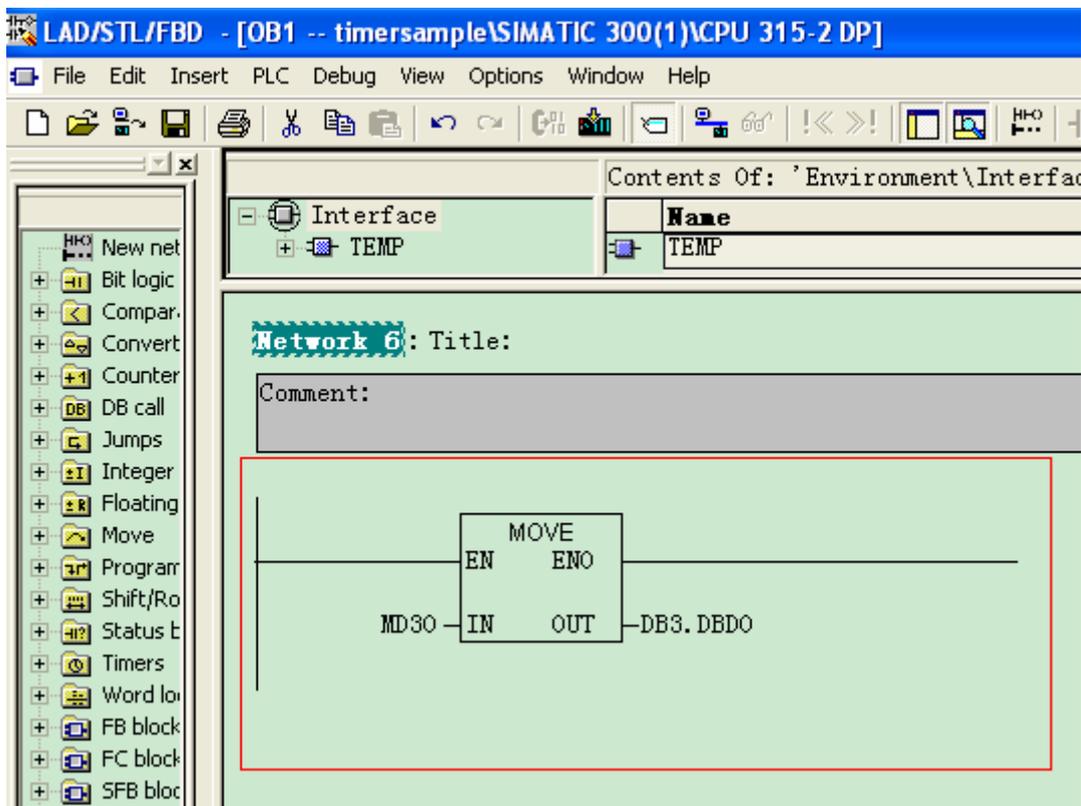
在 PLC 侧：

设定对应的系统参数（MPI 地址和波特率等），然后可创建 TIME 变量。

插入 DB 块并定义变量：

Address	Name	Type	Initial value	Comme
0.0		STRUCT		
+0.0	timer1	TIME	T#0MS	
=4.0		END_STRUCT		

打开 OB1 输入如下指令，MD30 为上位机写入的数据，DB3.DBDO 为 PLC 中定义的数据，也可以不经过转换直接用 MD30 作为 Time 变量的输入值，也就是说 FB41 的 TI 和 TD 可以用 MD30，也可以用 DB3.DBDO。



然后我们即可将 DB3.DBDO 赋给相应的 IEC 定时器或功能块的 Time 变量作为设定值，如 FB41 的 TI 和 TD。

```

OB35 : "Cyclic Interrupt"
Comment:
Network 1: Title:
Comment:
CALL "CONT_C" , DB41
COM_RST :=
MAN_ON :=
PVPER_ON:=
P_SEL :=
I_SEL :=
INT_HOLD:=
I_ITL_ON:=
D_SEL :=
CYCLE :=DB3.DBDO
SP_INT :=
PV_IN :=
PV_PER :=
MAN :=
GAIN :=
TI :=
TD :=
TM_LAG :=
DEADB_W :=
LMN_HLM :=
LMN_LLM :=
PV_FAC :=
PV_OFF :=
LMN_FAC :=
LMN_OFF :=
I_ITLVAL:=
DISV :=
LMN :=
LMN_PER :=
QLMN_HLM:=
QLMN_LLM:=
LMN_P :=
LMN_I :=
LMN_D :=

```

如需将其上位机的设定赋值给 S5TIME 变量时我们需要将设定值进行转换。首先我们先介绍一下 S5TIME 的数据格式，在 STEP7 的 Help 下拉菜单中点击 Contents，然后利用索引 Index 搜索关于相应的数据类型我们能找到如下解释：



Upper Limits for Entering Timers

Note the following upper limits for entering timers:
 Example: W#16#9999 (maximum value in BCD format)

Examples:

Address	Monitor Format	Enter	Modify Value Display	Explanation
T 1	SIMATIC_TIME	137	S5TIME#130MS	Conversion to milliseconds
MW4	SIMATIC_TIME	137	S5TIME#890MS	Representation in BCD format possible
MW4	HEX	137	W#16#0089	Representation in BCD format possible
MW6	HEX	157	W#16#009D	Representation in BCD format not possible, therefore the monitor format SIMATIC_TIME cannot be selected

Note

- You can enter timers in millisecond steps but the value entered is adapted to the time frame. The size of the time frame depends on the size of the time value entered (137 becomes 130 ms; the 7 ms were rounded down).
- The modify values for addresses of the data type WORD, for example, IW1, are converted to BCD format. Not every bit pattern is a valid BCD number, however. If the entry cannot be represented as SIMATIC_TIME for an address of the data type WORD, the application reverts automatically to the default format (here: HEX, see Select Monitor Format, Default Command (View Menu)) so that the value entered can be displayed.

BCD Format for Variables in the SIMATIC_TIME Format

Values of variables in the SIMATIC_TIME format are entered in BCD format. The 16 bits have the following significance:

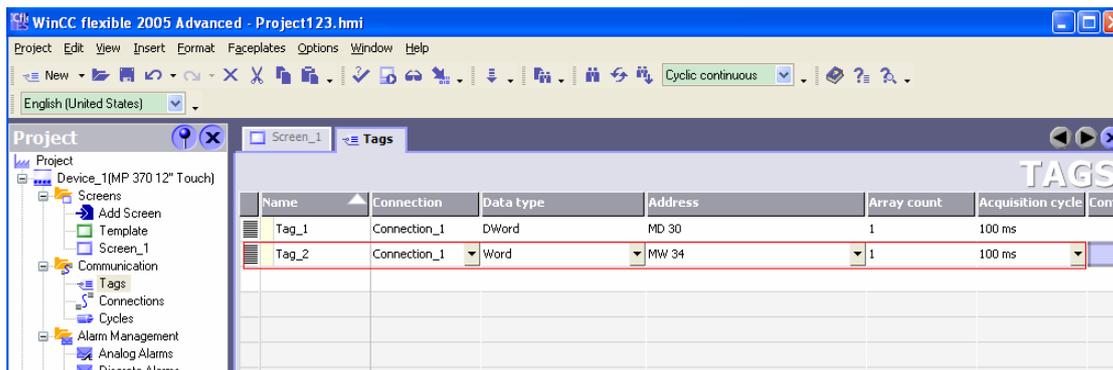
| 0 0 x x | h h h h | t t t t | u u u u |

- Bits 15 and 14 are always zero.
- Bits 13 and 12 (marked with xx) set the multiplier for bits 0 to 11:

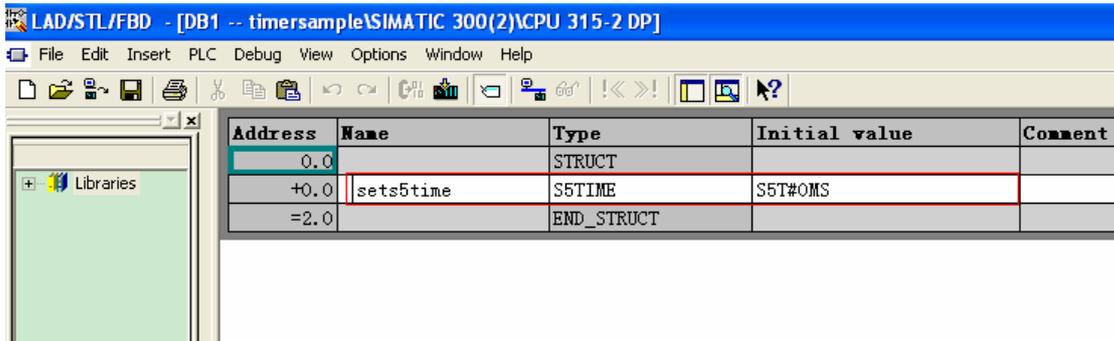
 - 00 => multiplier 10 milliseconds
 - 01 => multiplier 100 milliseconds
 - 10 => multiplier 1 second
 - 11 => multiplier 10 seconds

- Bits 11 to 8 hundreds (hhhh)
- Bits 7 to 4 tens (tttt)
- Bits 3 to 0 units (uuuu)

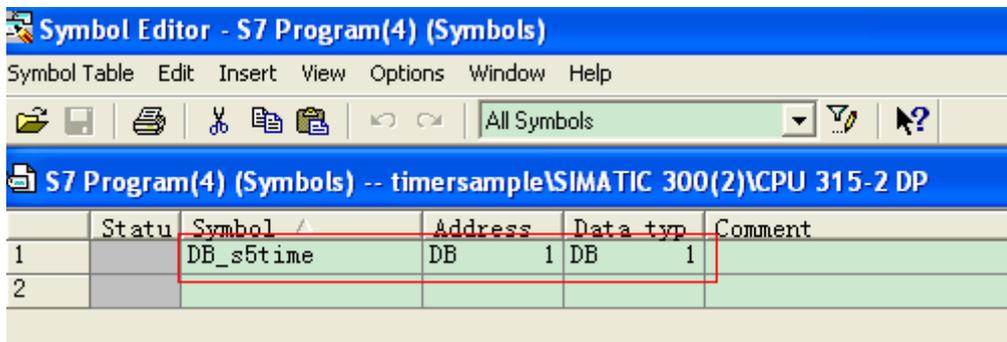
参照上面的方法我们在上位机定义一个 WORD（或 INT）类型的变量（例如 MW34）



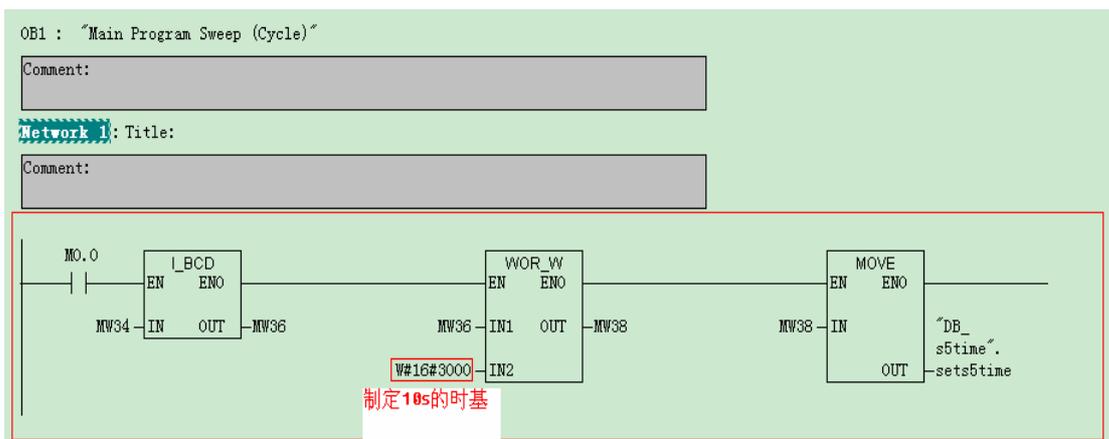
在 STEP7 中我们首先定义一个 DB 块并定义一个 S5TIME 数据类型的变量：



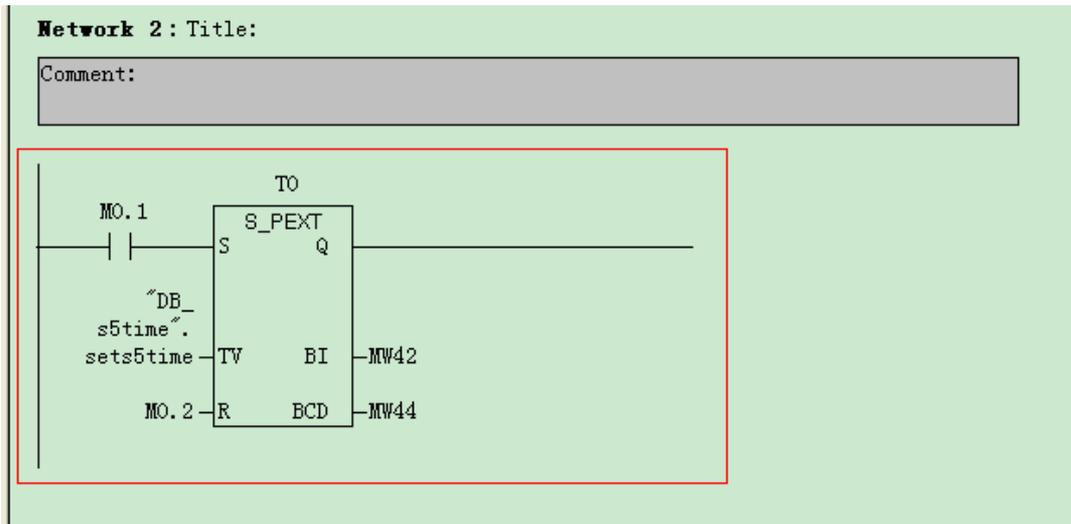
打开符号表编辑符号：



打开 OB1, 用 I_BCD 指令将整数转换为 BCD 格式, 需要注意写入的整数最大值为 999, 否则 I_BCD 会出错, 然后将其与时基做或运算 (例程中 W#16#3000 表示时基位 10 秒, 时间值为 0, 因为最高 4 位为 0011, 如果是 W#16#2000 表示时基位 1 秒, 时间值为 0, 因为最高 4 位为 0010), 并通过 MOVE 指令送入定义的 S5TIME 数据类型的变量中



编辑调用指令:



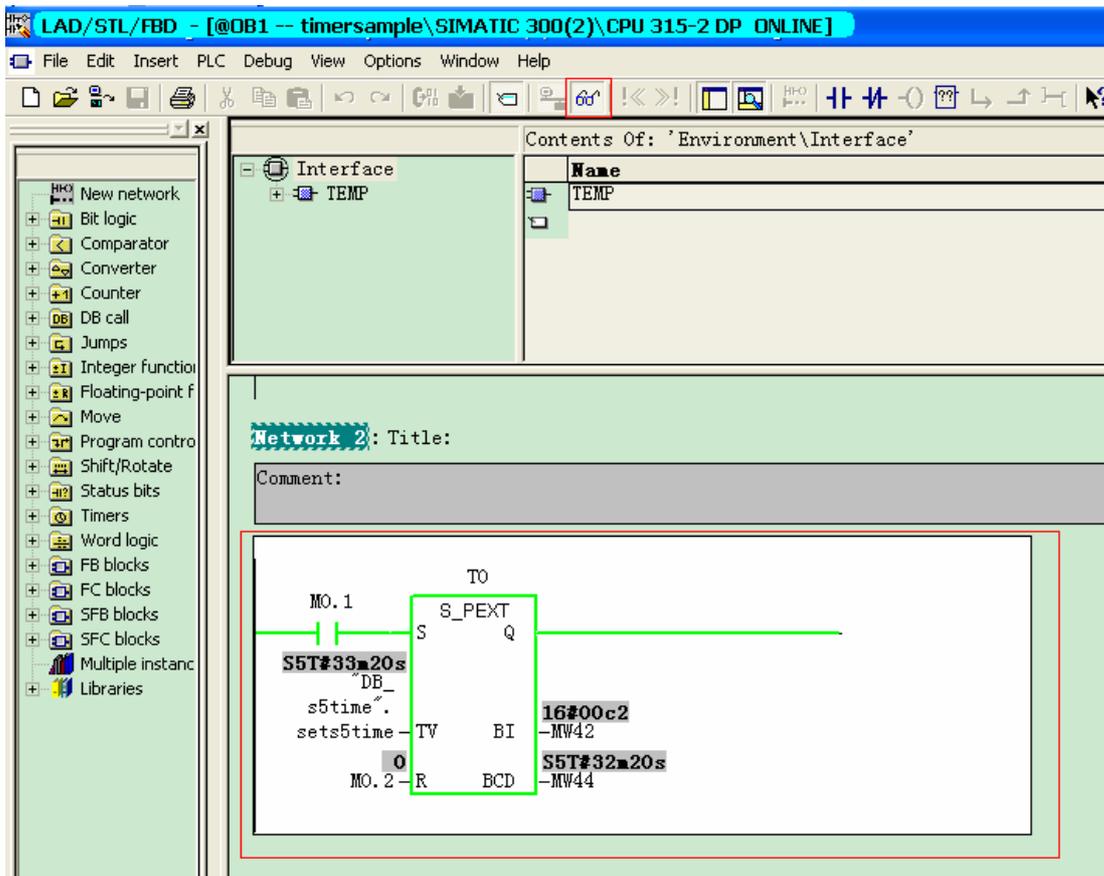
打开变量表观察，当 MO.0 为 1 时 MW34 中的值已经转化为 S5TIME 的格式并存储到定义的 DB1.DBW0 中:

Address	Symbol	Display format	Status value	Modify value
1	M 0.0	BOOL	true	true
2	MW 34	DEC	200	200
3	DB1.DBW 0	"DB_s5 SIMATIC_TIME	S5T#33m20s0ms	
4	M 0.1	BOOL	false	false
5	T 0	SIMATIC_TIME	S5T#0ms	
6				

当 MO.1 从 0 变为 1，T0 按设定值运行:

Address	Symbol	Display format	Status value	Modify value
1	M 0.0	BOOL	true	true
2	MW 34	DEC	200	200
3	DB1.DBW 0	"DB_s5 SIMATIC_TIME	S5T#33m20s0ms	
4	M 0.1	BOOL	true	true
5	T 0	SIMATIC_TIME	S5T#33m10s0ms	
6				

也可通过在线的方法来观察程序



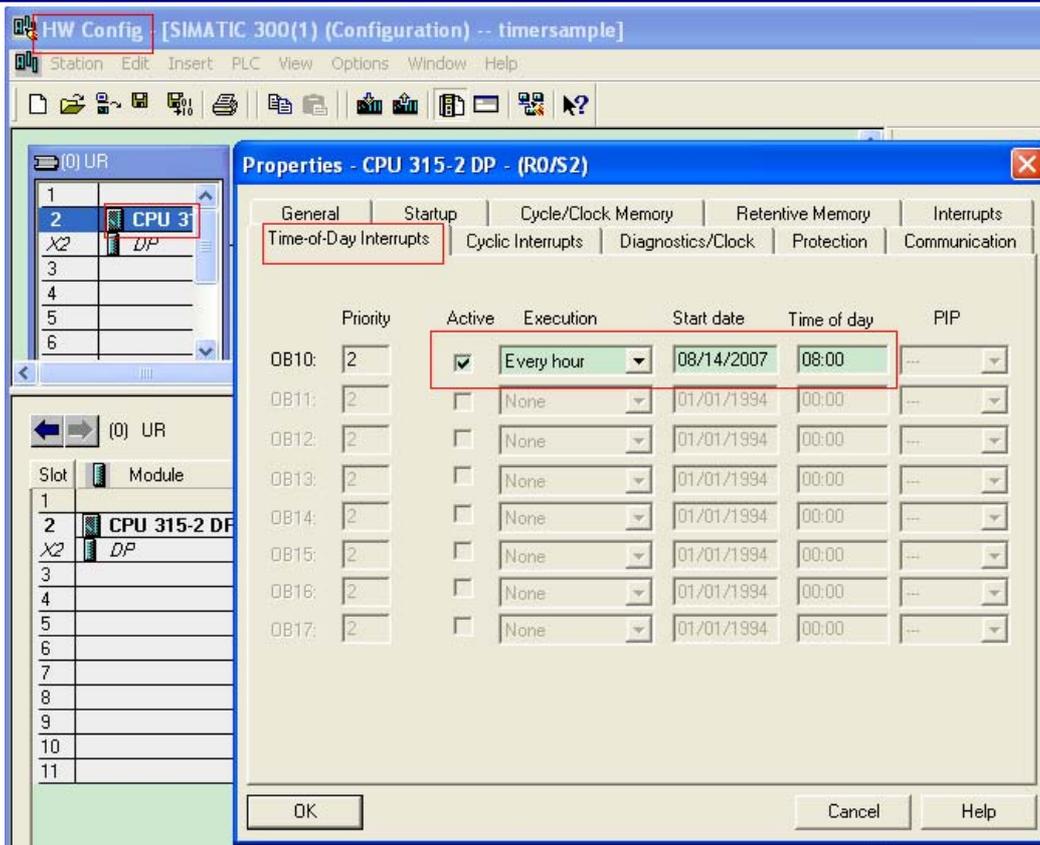
程序见例程 TIMERSAMPLE1

6.2 定时器时间太短不够用怎么办？

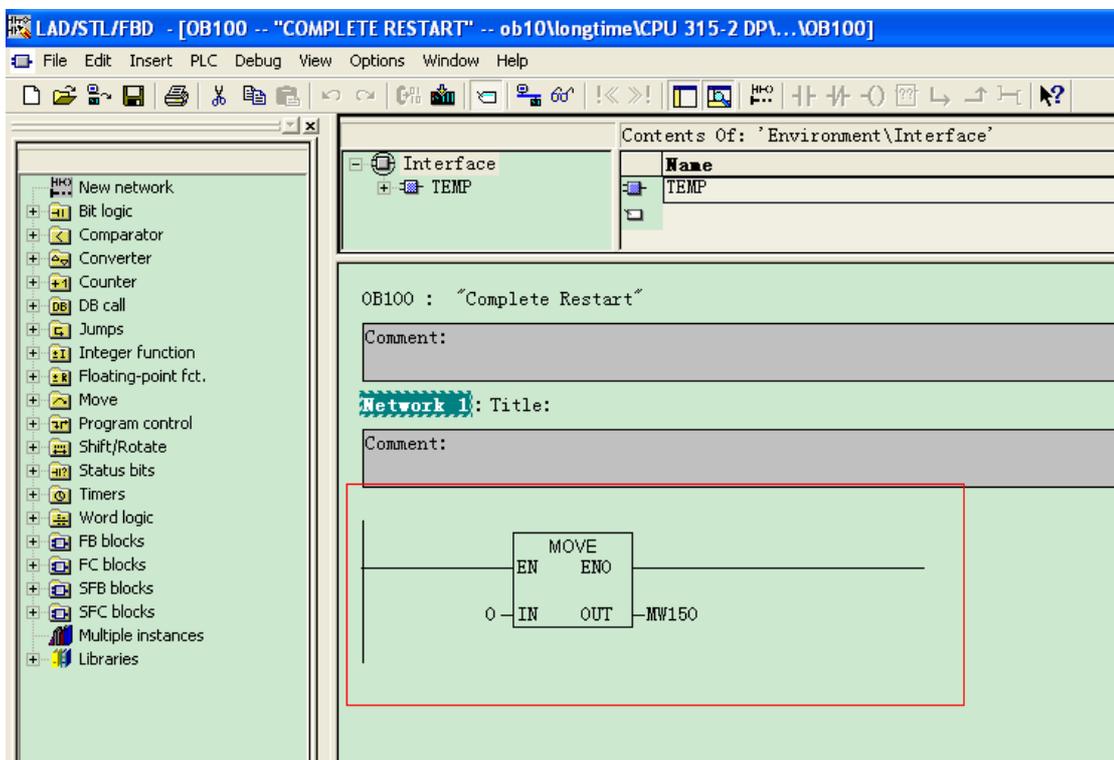
首先 S5Timer 的定时时间最大为：2H_46M_30S，而 IEC 定时器的定时时间最大为：24D_20H_31M_23S_648MS。如果需要更大时间的定时设置可通过编程来实现。

- 1) 如果有规律的可使用 OB35 或 OB10 编程来实现，例如某车间需计算 8 小时的班产量和日产量，我们可以如下操作：

首先在 cpu 属性中定义 OB10 为每小时中断一次；



然后在 OB100 中初始化计数变量如 mw150 为 0



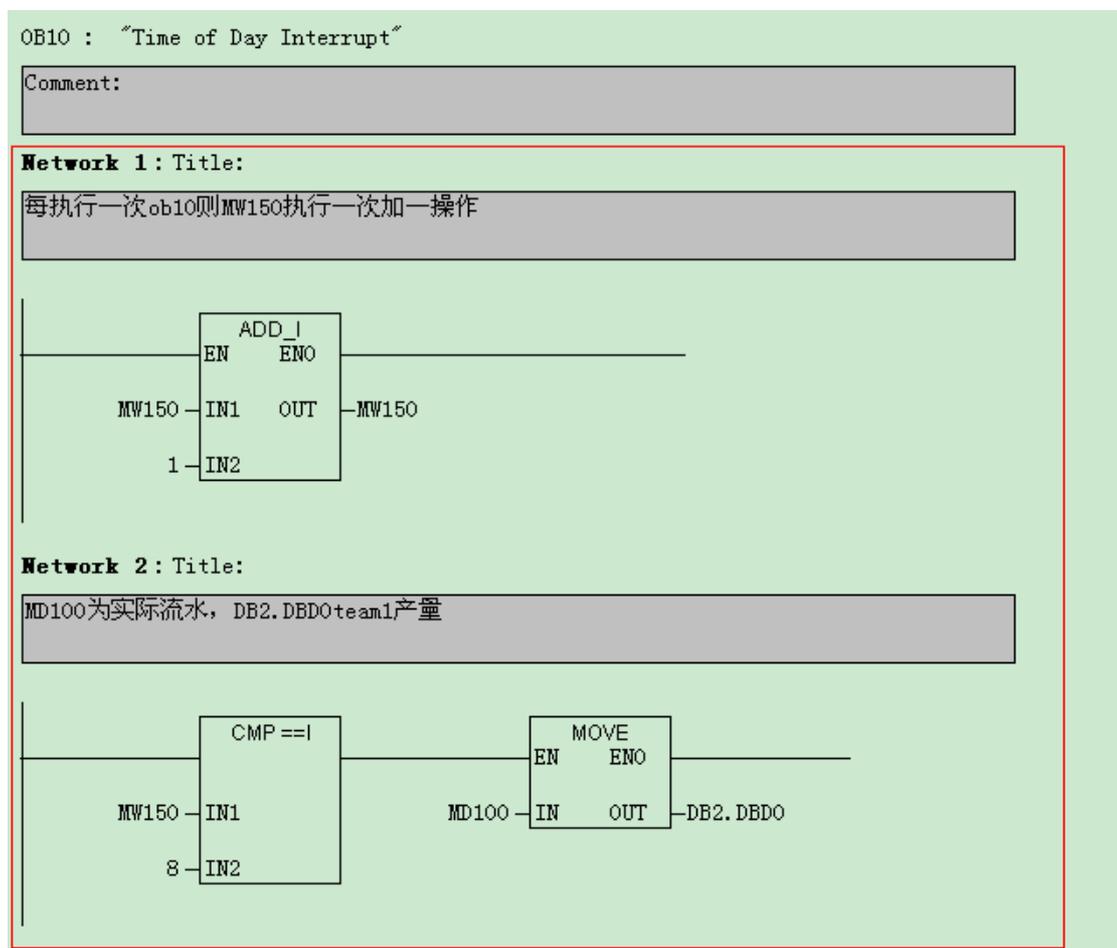
创建 DB 定义变量:

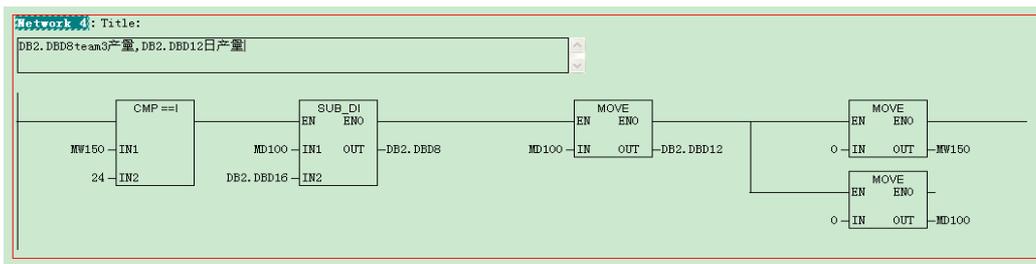
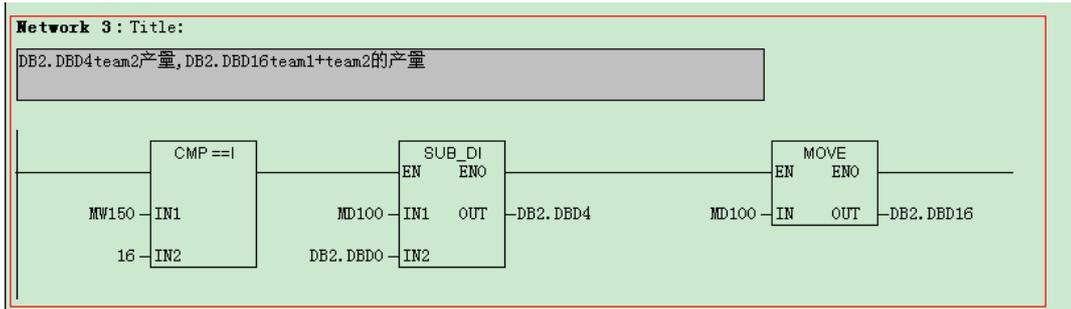
LAD/STL/FBD - [DB2 -- timersample\SIMATIC 300(1)\CPU 315-2 DP]

File Edit Insert PLC Debug View Options Window Help

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	team1	DINT	L#0	一班产量
+4.0	team2	DINT	L#0	二班产量
+8.0	team3	DINT	L#0	三班产量
+12.0	all	DINT	L#0	日三班产量
+16.0	temp	DINT	L#0	一班和二班产量和
=20.0		END_STRUCT		

打开 OB10 编制程序:





注意所附例程中没用编写计数程序（MW150）需用户自己编写。

2) 作为无固定规律的定时我们可以利用读取系统时钟并与设定时间来比较完成。

首先创建 DB 块并定义两个 DATE AND TIME 格式的变量

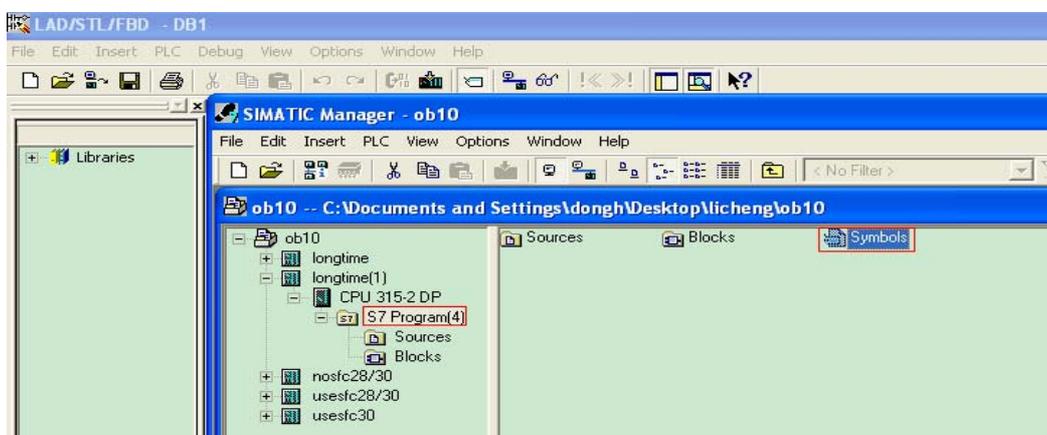
LAD/STL/FBD - DB1

File Edit Insert PLC Debug View Options Window Help

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	readtime	DATE_AND_TIME	DT#90-1-1-0:0:0.000	
+8.0	comparetime	DATE_AND_TIME	DT#07-8-15-0:0:0.000	
=16.0		END_STRUCT		

CPU 系统时钟的设定可参照前面 3.1 节介绍的方法，此处不再介绍。

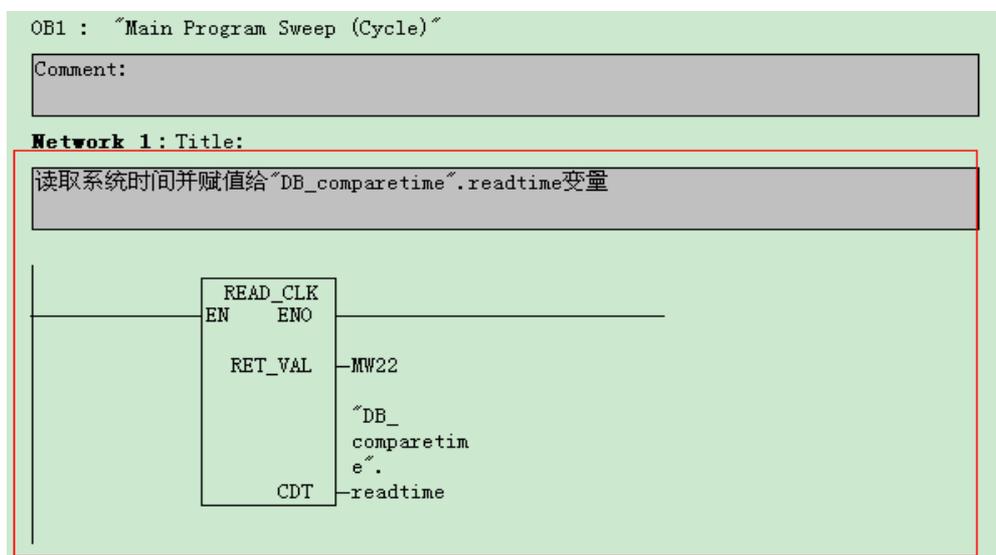
打开符号表



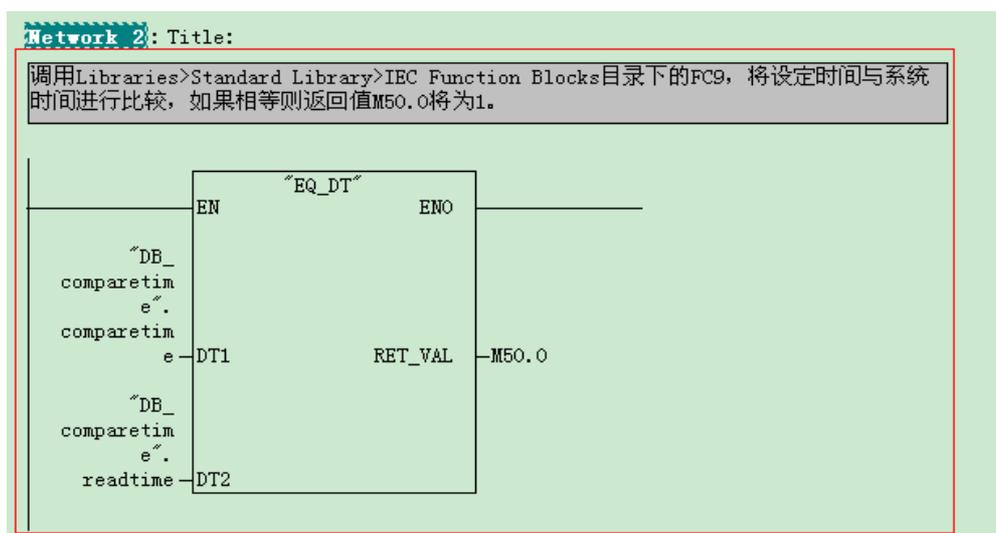
编辑符号

	Statu	Symbol	Address	Data typ	Comment
1		COMPLETE_RESTART	OB 100	OB 100	Complete Restart
2		EQ_DT	FC 9	FC 9	Equal DT
3		READ_CLK	SFC 1	SFC 1	Read System Clock
4		TOD_INT0	OB 10	OB 10	Time of Day Interrupt 0
5		DB_comparetime	DB 1	DB 1	存放读取的系统时间和用于比较的时间
6					

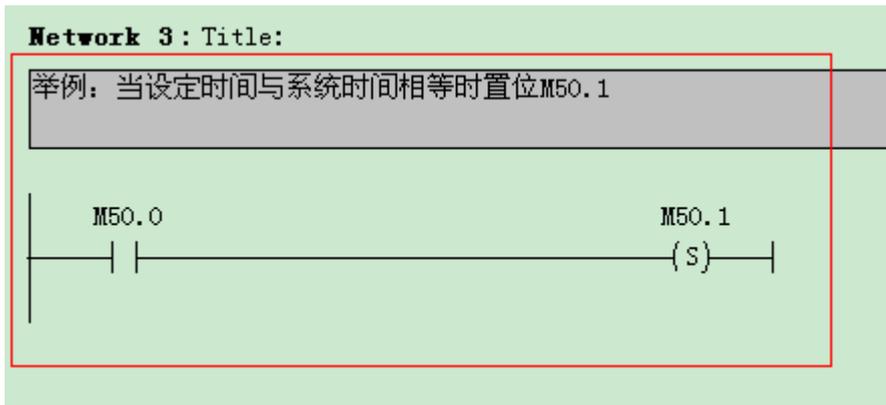
在 OB1 中编程，首先调用 SFC1



在 Libraries>Standard Library>IEC Function Blocks 目录下调用 FC9（时间日期比较）：



为方便观察我们定义一个置位操作：



打开监视和修改变量表：

Var - @Variable table1

Table Edit Insert PLC Variable View Options Window Help

@Variable table1 ONLINE

	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBB 0		HEX	B#16#07	
2	DB1.DBB 1		HEX	B#16#08	
3	DB1.DBB 2		HEX	B#16#16	
4	DB1.DBB 3		HEX	B#16#09	
5	DB1.DBB 4		HEX	B#16#07	
6	DB1.DBB 5		HEX	B#16#34	
7	DB1.DBB 6		HEX	B#16#17	
8	DB1.DBB 7		BIN	2#0101_0101	
9	DB1.DBB 8		HEX	B#16#07	B#16#07
10	DB1.DBB 9		HEX	B#16#08	B#16#08
11	DB1.DBB 10		HEX	B#16#16	B#16#16
12	DB1.DBB 11		HEX	B#16#09	B#16#09
13	DB1.DBB 12		HEX	B#16#04	B#16#04
14	DB1.DBB 13		HEX	B#16#00	B#16#00
15	DB1.DBB 14		HEX	B#16#00	B#16#00
16	DB1.DBB 15		BIN	2#0000_0101	2#0000_0101
17	M 50.0		BOOL	false	
18	M 50.1		BOOL	true	
19					

读取的系统时间

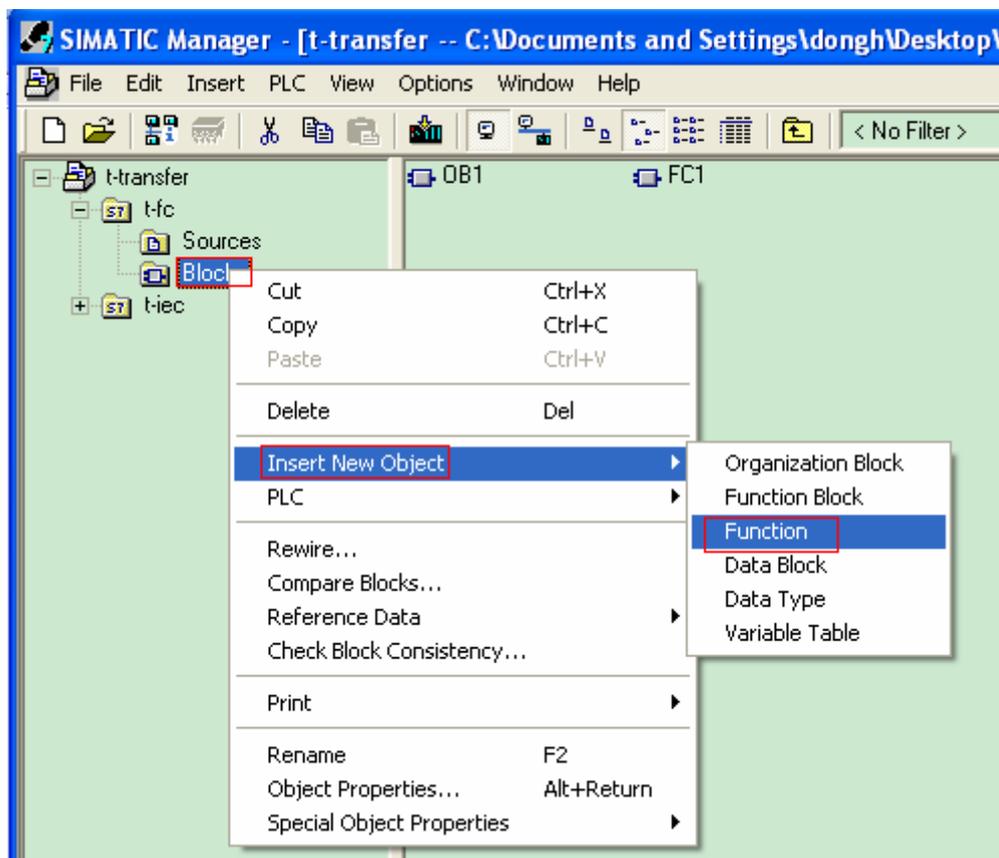
设定的比较时间

程序运行的结果

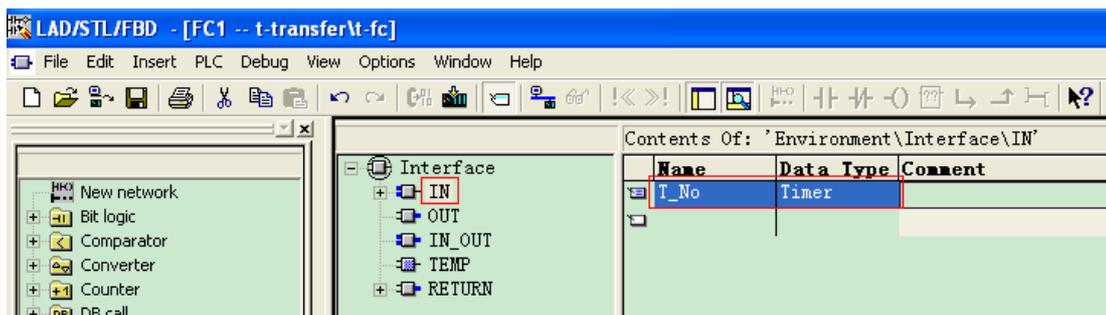
详见例程 OB10

6. 3 如何用 Timer 作为变量为 FC/FB 传递参数？

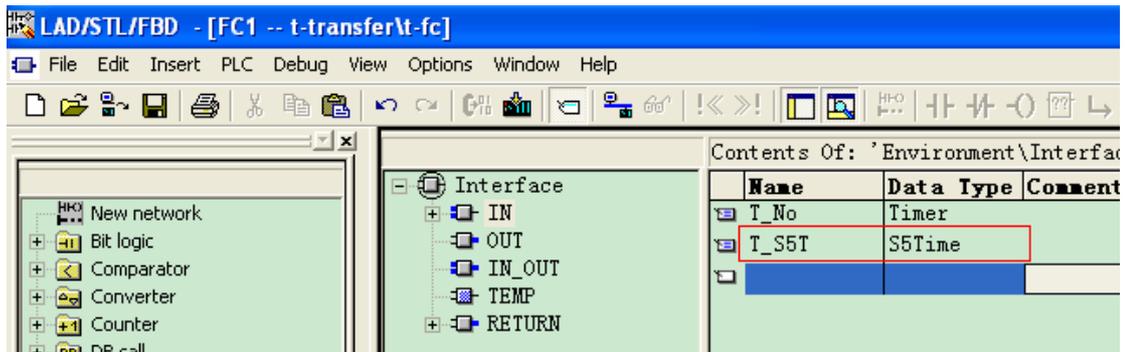
1) 创建一个 FC/FB (如 FC1)



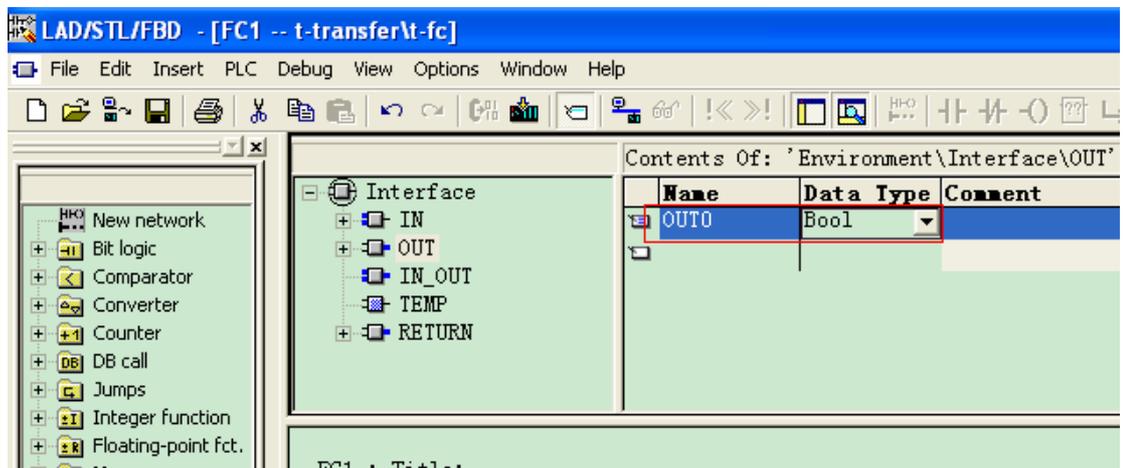
2) 双击打开 FC1 定义一个 IN 参数 Name 为 T_No, Data_Type 为 Timer 格式用于传递定时器号, 如下图所示:



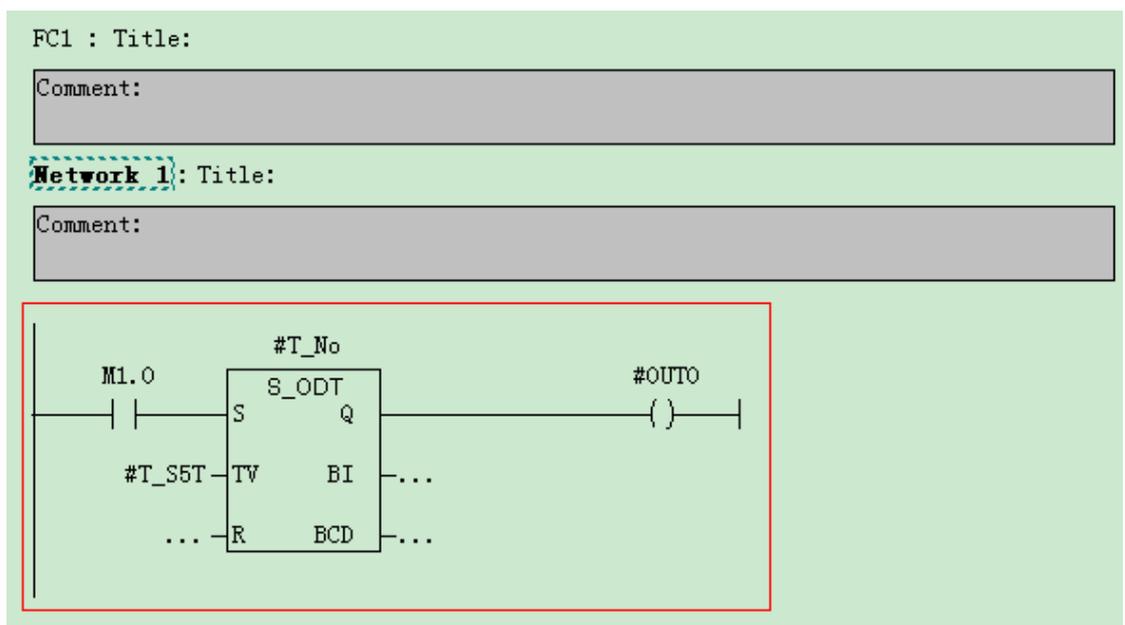
3) 再定义一个 IN 参数 Name 为 T_S5T, Data_Type 为 S5Time 格式用于传递定时时间:



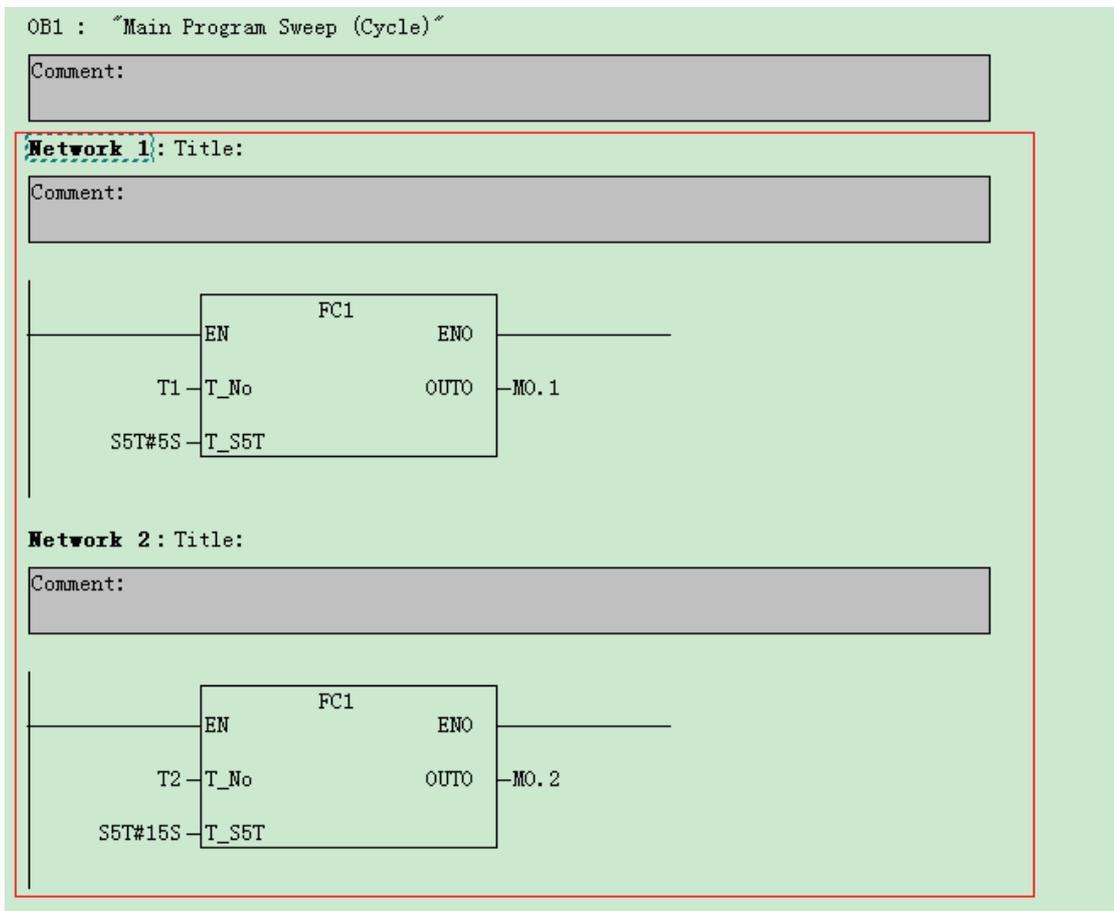
4) 在 OUT 参数中定义一个 BOOL 输出参数 OUT0:



5) 在 FC1 中编程调用 S_ODT 指令。



6) 在 OB1 中多次调用 FC1 并传递不同的定时器号和定时值。



7) 在变量表中观察可见，当 M1.0 变为 1 时 T1 和 T2 分别以设定的时间运行互不干涉。
定时器运行时间小于 5S 时：

	Address	Symbol	Display format	Status value	Modify value
1	M 1.0		BOOL	true	true
2	T 1		SIMATIC_TIME	S5T#3s770ms	
3	M 0.1		BOOL	false	
4	T 2		SIMATIC_TIME	S5T#13s800ms	
5	M 0.2		BOOL	false	
6					

定时器运行时间大于 5S 并小于 15S 时:

	Address	Symbol	Display format	Status value	Modify value
1	M 1.0		BOOL	true	true
2	T 1		SIMATIC_TIME	S5T#0ms	
3	M 0.1		BOOL	true	
4	T 2		SIMATIC_TIME	S5T#9s500ms	
5	M 0.2		BOOL	false	
6					

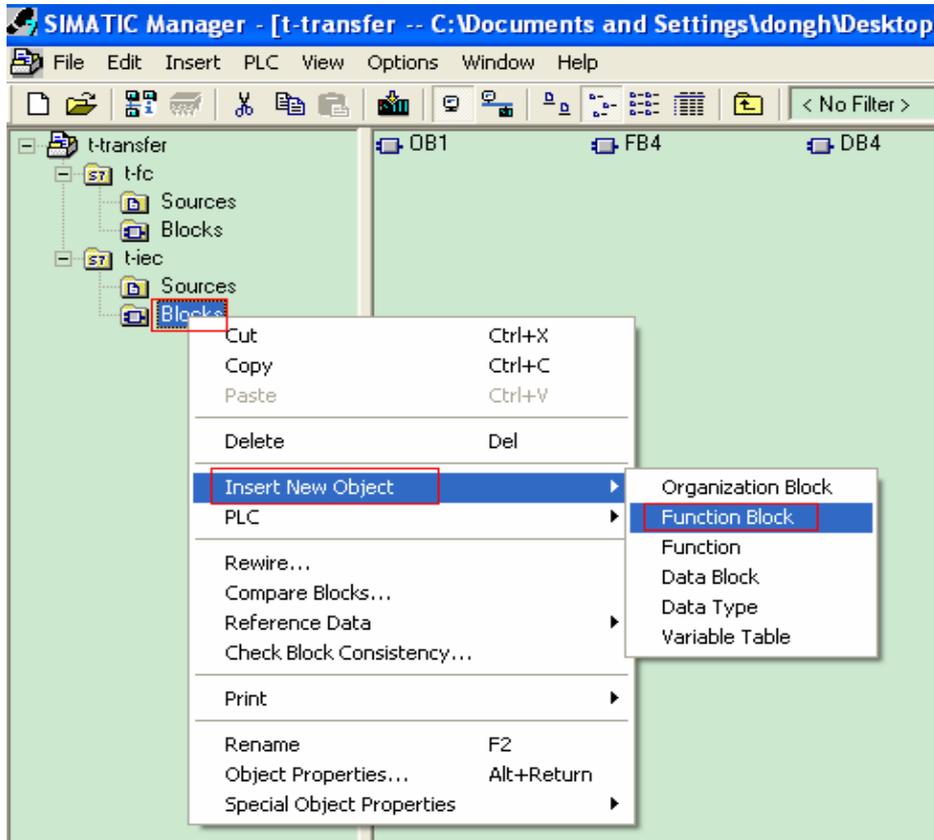
定时器运行时间大于 15S 时:

	Address	Symbol	Display format	Status value	Modify value
1	M 1.0		BOOL	true	true
2	T 1		SIMATIC_TIME	S5T#0ms	
3	M 0.1		BOOL	true	
4	T 2		SIMATIC_TIME	S5T#0ms	
5	M 0.2		BOOL	true	
6					

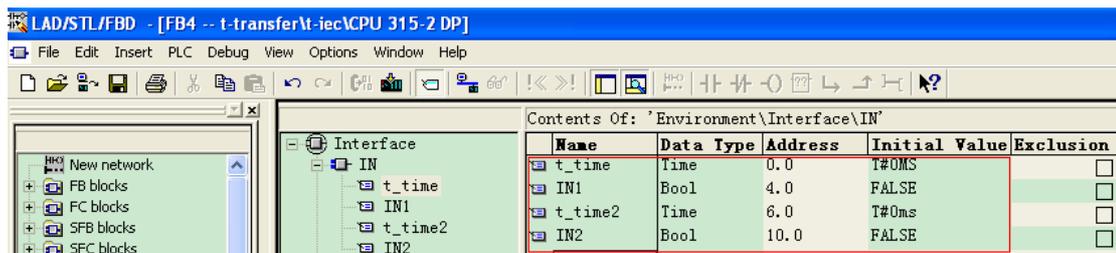
具体程序见例子 t_transfer

6. 4 用 IEC 定时器如何使用多重背景

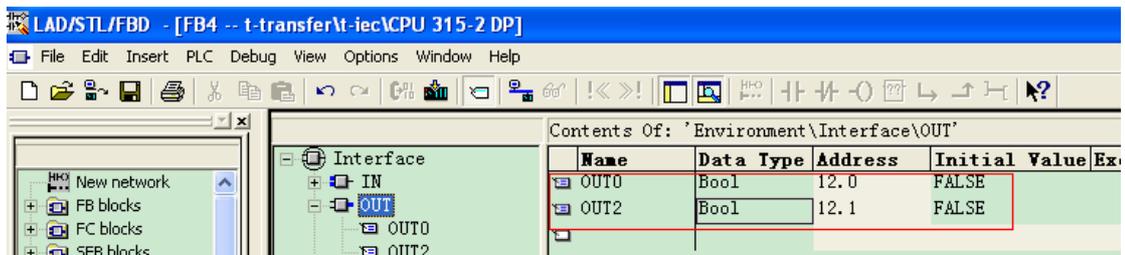
1) 创建一个 FB (如 FB4)



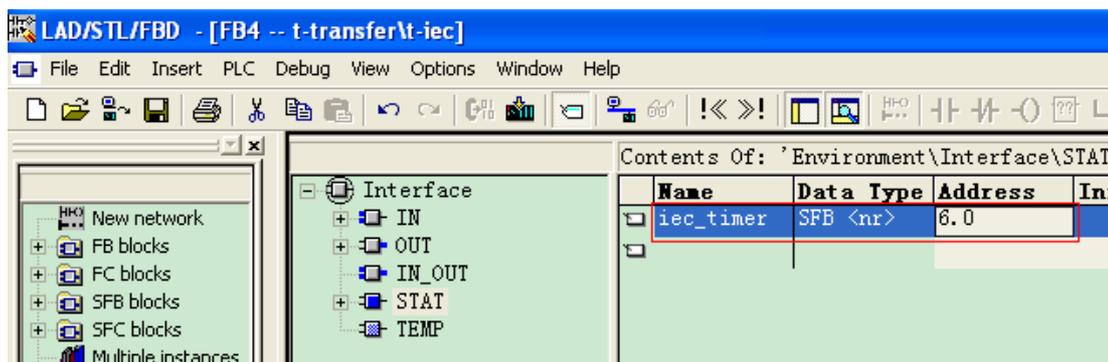
2) 打开 FB4 定义输入参数 (如 t_time 和 t_time2 为 TIME 格式的输入形参, 可根据实际情况来定义输入/出参数)



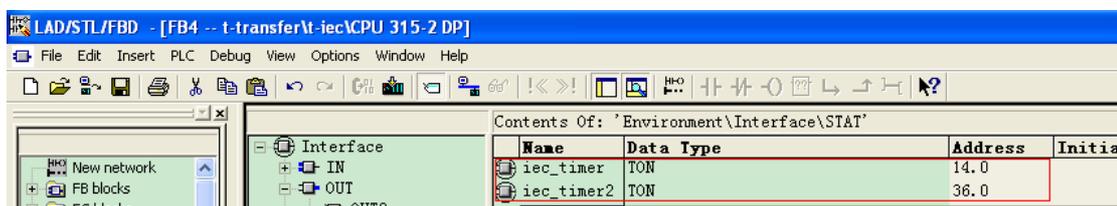
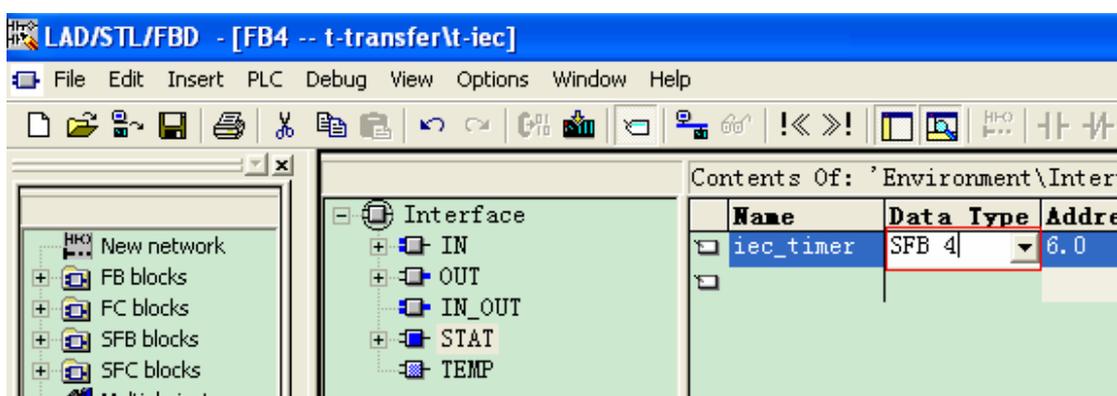
定义输出参数



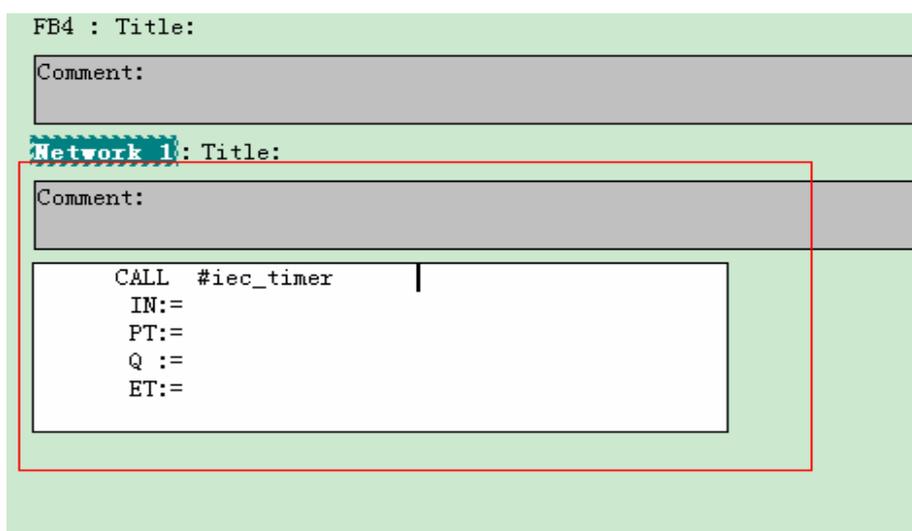
3) 定义两个 IEC 定时器的静态变量（如 iec_timer 和 iec_timer2）



将 SFB<nr>修改为 SFB4:



4) 选择 STL 的编程方式，以符号名的形式调用 SFB4:



FB4 : Title:

Comment:

Network 1 : Title:

Comment:

```
CALL #iec_timer
IN:=#IN1
PT:=#t_time
Q :=#OUT0
ET:=
```

Network 2 : Title:

Comment:

```
CALL #iec_timer2
IN:=#IN2
PT:=#t_time2
Q :=#OUT2
ET:=
```

- 5) 创建需要调用 FB4 的 FC 或 FB 或 OB 块，在此以 OB1 为例，在 OB1 中调用 FB4, 并创建背景数据块 DB4

OB1 : "Main Program Sweep (Cycle)"

Comment:

Network 1 : Title:

Comment:

```
CALL FB 4 , DB4
t_time :=T#7S
IN1 :=M0.0
t_time2:=T#15S
IN2 :=M0.1
OUT0 :=Q1.0
OUT2 :=Q1.1
```

6) 可打开背景数据块观察, iec_timer 和 iec_timer2(SFB4) 需要的相应参数已经存在其中。

Address	Declaration	Name	Type	Initial value	
1	0.0	in	t_time	TIME	T#0MS
2	4.0	in	IN1	BOOL	FALSE
3	6.0	in	t_time2	TIME	T#0MS
4	10.0	in	IN2	BOOL	FALSE
5	12.0	out	OUT0	BOOL	FALSE
6	12.1	out	OUT2	BOOL	FALSE
7	14.0	stat:in	iec_timer.IN	BOOL	FALSE
8	16.0	stat:in	iec_timer.PT	TIME	T#0MS
9	20.0	stat:out	iec_timer.Q	BOOL	FALSE
10	22.0	stat:out	iec_timer.ET	TIME	T#0MS
11	26.0	stat	iec_timer.STATE	BYTE	B#16#0
12	28.0	stat	iec_timer.STIME	TIME	T#0MS
13	32.0	stat	iec_timer.ATIME	TIME	T#0MS
14	36.0	stat:in	iec_timer2.IN	BOOL	FALSE
15	38.0	stat:in	iec_timer2.PT	TIME	T#0MS
16	42.0	stat:out	iec_timer2.Q	BOOL	FALSE
17	44.0	stat:out	iec_timer2.ET	TIME	T#0MS
18	48.0	stat	iec_timer2.STATE	BYTE	B#16#0
19	50.0	stat	iec_timer2.STIME	TIME	T#0MS
20	54.0	stat	iec_timer2.ATIME	TIME	T#0MS

7) 打开监视和修改变量表:

将 M0.0 置 1, SFB4 运行时间小于 7s 时:

Address	Symbol	Display format	Status value	Modify value
M 0.0		BOOL	true	true
Q 1.0		BOOL	false	
DB4.DBD 22		TIME	T#459ms	调用的第一个SFB4的已运行时间
M 0.1		BOOL	true	true
Q 1.1		BOOL	false	
DB4.DBD 44		TIME	T#459ms	调用的第二个SFB4的已运行时间

当 SFB4 运行时间大于 7s 小于 15s 时: Q1.0 变为 1 定时 7s 的定时器已经停止计时, 而定时 15 的定时器依然继续互相并不干扰。

Address	Symbol	Display format	Status value	Modify value
M 0.0		BOOL	true	true
Q 1.0		BOOL	true	
DB4.DBD 22		TIME	T#7s	
M 0.1		BOOL	true	true
Q 1.1		BOOL	false	
DB4.DBD 44		TIME	T#9s823ms	

当 SFB4 运行时间大于 15s 时：Q1.0 和 Q1.1 都变为 1 定时器停止计时。

var1 -- @t-transfer\t-iec\CPU 315-2 DP\S7_Program(3) ONLINE						
	Address	Symbol	Display format	Status value	Modify value	
1	M 0.0		BOOL	true	true	
2	Q 1.0		BOOL	true		
3	DB4.DBD 22		TIME	T#7s		
4	M 0.1		BOOL	true	true	
5	Q 1.1		BOOL	true		
6	DB4.DBD 44		TIME	T#15s		
7						

当 M0.0 置 1M0.1 为 0 时，第一个 SFB4(iec_timer)运行而第二个 SFB4(iec_timer2)不运行，二者互相并不干扰：

var1 -- @t-transfer\t-iec\CPU 315-2 DP\S7_Program(3) ONLINE						
	Address	Symbol	Display format	Status value	Modify value	
1	M 0.0		BOOL	true	true	
2	Q 1.0		BOOL	false		
3	DB4.DBD 22		TIME	T#2s689ms		
4	M 0.1		BOOL	false	false	
5	Q 1.1		BOOL	false		
6	DB4.DBD 44		TIME	T#0ms		
7						

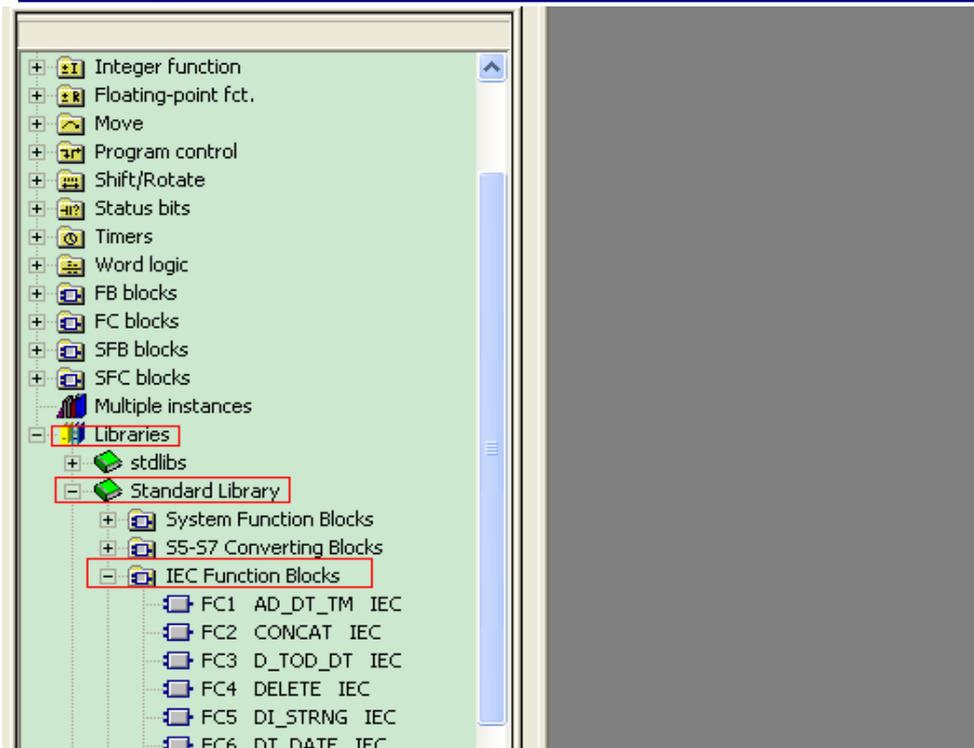
具体程序见例子 t_transfer

6. 5 Time to S5Time, S5Time to Time

6. 5. 1 Time to S5Time

使用 FC40 可将 Time 转换为 S5Time 格式：

FC40 的位置在 Libraries>Standard Library>IEC function block 目录下。



在 STEP7 项目中定义数据块 DB1, 并定义三个变量: DB1.DBW0 为 INT 类型, DB1.DBD2 为 DINT 类型, DB1.DBD6 为 TIME 类型, DB1.DBW10 为 S5TIME 类型。

LAD/STL/FBD - [DB1 -- "DB_1 to s5t" -- timertest\time and s5time...DB1]

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	SET_TV	INT	0	上位机设定的时间
+2.0	SET_DITV	DINT	L#0	将上位机设定的时间转换为双整数
+6.0	TV_time	TIME	T#0MS	以TIME格式存储设定时间
+10.0	TV_S5TIME	S5TIME	S5T#0MS	转换后的S5TIME时间
=12.0		END_STRUCT		

打开符号表定义符号:

Symbol Editor - time and s5time (Symbols)

Symbol Table Edit Insert View Options Window Help

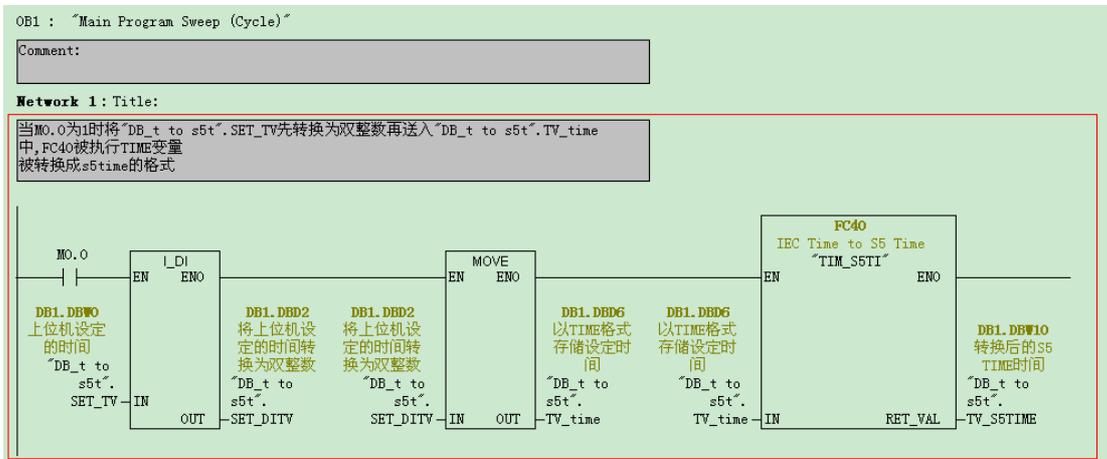
All Symbols

time and s5time (Symbols) -- timertest

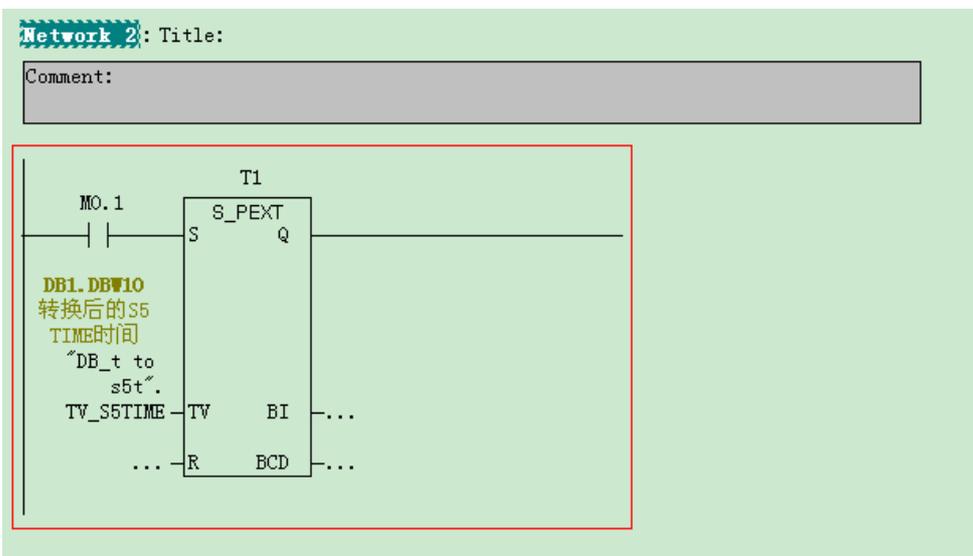
	Statu	Symbol	Address	Data typ	Comment
1		S5TI_TIM	FC 33	FC 33	S5 Time to IEC Time
2		TIM_S5TI	FC 40	FC 40	IEC Time to S5 Time
3		var1	VAT 1		
4		var2	VAT 2		
5		DB_t to s5t	DB 1	DB 1	
6					

打开 OB1 编程:

调用 FC40



调用定时器指令:



打开变量表, 为 DB1.DBW0 赋值并分别让 MO.0 和 MO.1 为 TRUE:

Var - var1

Table Edit Insert PLC Variable View Options Window Help

var1 -- @timertest\time and s5time ONLINE

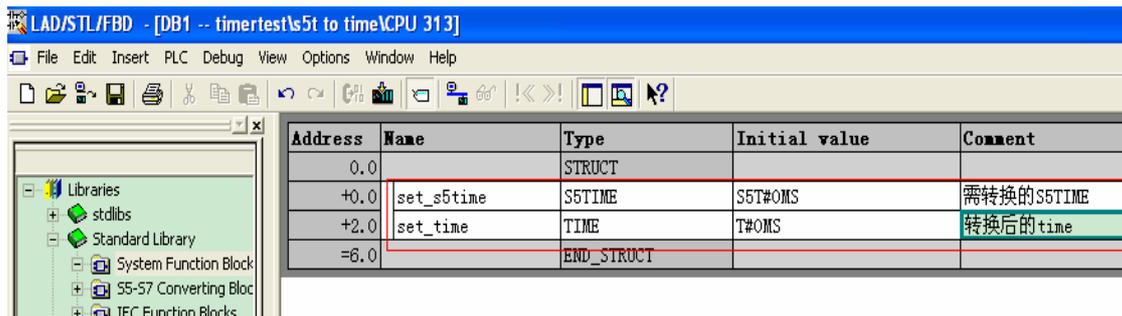
	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	true	true
2	M 0.1		BOOL	true	true
3	DB1.DBW 0	"DB_t to s5t".SET_TV	DEC	2450	2450
4	DB1.DBD 2	"DB_t to s5t".SET_DITV	DEC	L#2450	
5	DB1.DBD 6	"DB_t to s5t".TV_time	TIME	T#2s450ms	
6	DB1.DBW 10	"DB_t to s5t".TV_S5TIME	SIMATIC_TIME	S5T#2s450ms	
7	T 1		SIMATIC_TIME	S5T#910ms	
8					

详见附例程：timertest，注意所附例程只注重功能的实现，即非唯一的方法也没有考虑限制范围，仅做参考。

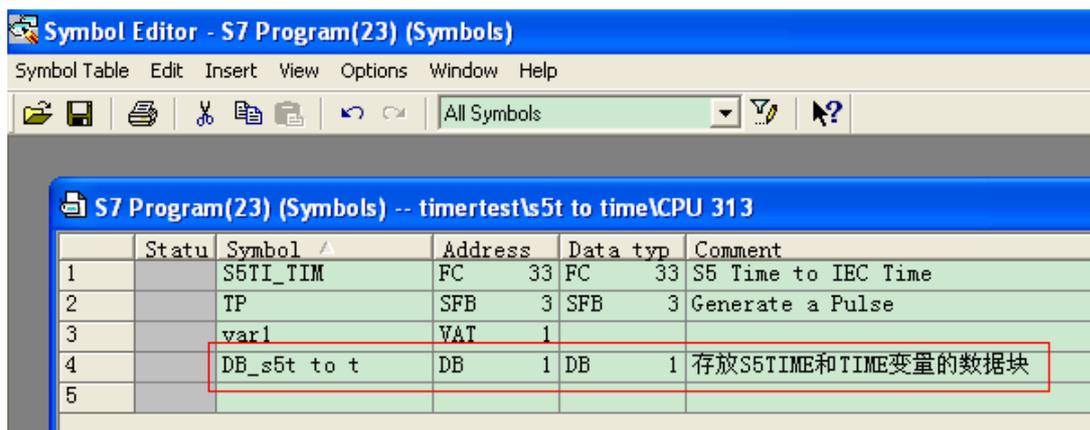
6. 5. 2 S5Time to Time

使用 FC33（在 Libraries>Standard Library>IEC function block 目录下）可将 S5Time 转换为 Time 格式。

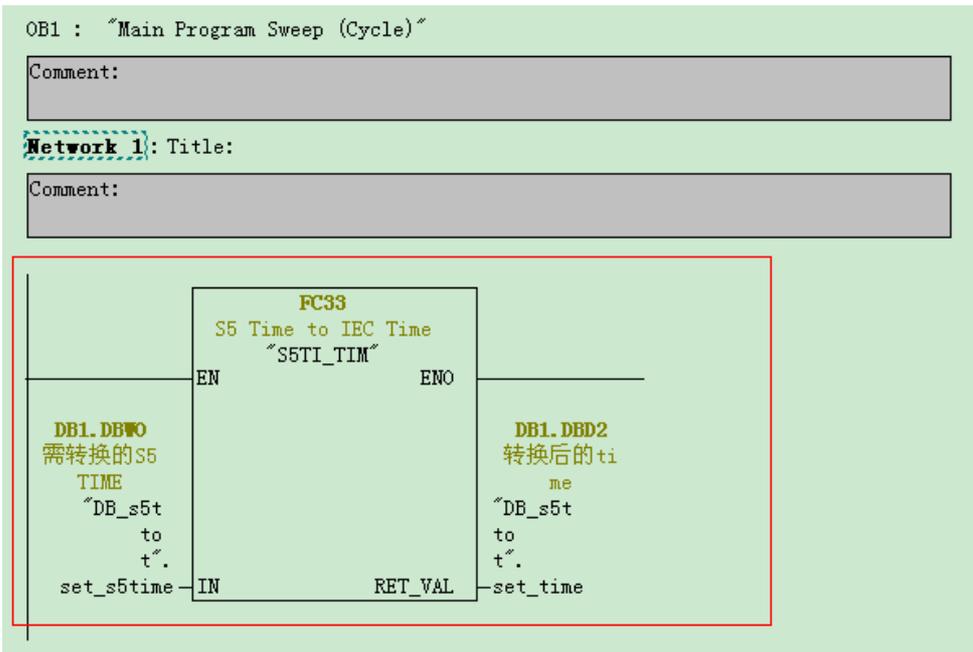
插入 DB 块，打开并定义两个 S5Time 转换为 Time 格式的变量：



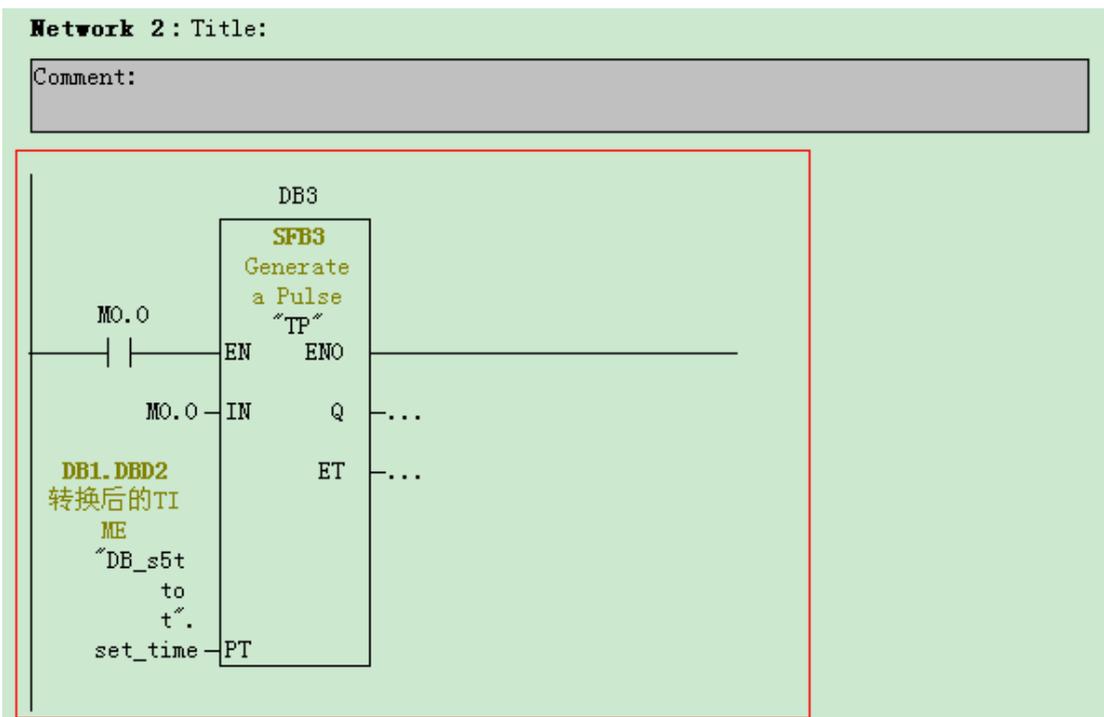
打开符号表定义符号：



在 OB1 中调用 FC33:



举例调用 IEC 定时器（SFB3 其中 DB3 为其背景数据块）:



打开 DB3，其结构如下：

	Address	Declarat	Name	Type	Initial value	Actual value	Co
1	0.0	in	IN	BOOL	FALSE	FALSE	
2	2.0	in	PT	TIME	T#0MS	T#0MS	
3	6.0	out	Q	BOOL	FALSE	FALSE	
4	8.0	out	ET	TIME	T#0MS	T#0MS	
5	12.0	stat	STATE	BYTE	B#16#0	B#16#0	
6	14.0	stat	STIME	TIME	T#0MS	T#0MS	
7	18.0	stat	ATIME	TIME	T#0MS	T#0MS	

打开变量表，为 DB1.DBW0 赋值，可见 FC33 执行后 DB1.DB2 和 DB3.DB2 中均为正确的 TIME 数值：

	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBW 0	"DB_s5t to t".set_s5time	SIMATIC_TIME	S5T#3s840ms	S5T#3s840ms
2	DB1.DB2 2	"DB_s5t to t".set_time	TIME	T#3s840ms	
3	M 0.0		BOOL	false	
4	DB3.DB2 2		TIME	T#0ms	
5	DB3.DB2 8		TIME	T#0ms	

将 M0.0 赋值为 TRUE，SFB3 按设定运行：

	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBW 0	"DB_s5t to t".set_s5time	SIMATIC_TIME	S5T#3s840ms	S5T#3s840ms
2	DB1.DB2 2	"DB_s5t to t".set_time	TIME	T#3s840ms	
3	M 0.0		BOOL	true	true
4	DB3.DB2 2		TIME	T#3s840ms	SFB3的设定运行时间
5	DB3.DB2 8		TIME	T#2s452ms	SFB3的已运行时间

详见附例程：timertest，

重要提示：

- 由于所附例程是免费的，任何用户可以免费复制或传播此程序例子。程序的作者对此程序不承担任何功能性或兼容性的责任，使用者风险自负
- 西门子不提供此程序例子的错误更改或者热线支持。
- 注意所附例程只注重功能的实现，即非唯一的方法也没有考虑限制范围，仅做参考。

7 附录—推荐网址

7.1 西门子自动化与驱动产品的在线技术支持

建议您访问西门子（中国）有限公司自动化与驱动集团 客户服务与支持中心：

网站首页: <http://www.ad.siemens.com.cn/Service/>您可以在检索窗口中键入STEP7, MMC或 Compatibility等关键字, 获取相关信息, 或通过点击以下链接下载获取更多信息

专家推荐精品文档: <http://www.ad.siemens.com.cn/Service/recommend.asp>

AS常问问题: <http://support.automation.siemens.com/CN/view/zh/10805055/133000>

AS更新信息: <http://support.automation.siemens.com/CN/view/zh/10805055/133400>

“找答案” AS版区:

<http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1027>

7. 1. 1 如何获得西门子自动化与驱动产品的资料

首先, 建议您通过Siemens A&D的网站搜索并下载。 www.ad.siemens.com.cn

您还可以致电 010-64763726 索取资料。

另外, 还有大量的手册可以通过分销商订购, 订货方式和其它产品一样。

7. 1. 2 需要设备选型及订货

如需设备选型及订货, 请联系西门子公司销售当地西门子分销商。分销商联系方式可致电 010-64731919 获得。

7. 1. 3 西门子技术支持热线

如有无法自行解决的技术问题, 请拨打西门子技术支持热线 400-810-4288 或 010-64719990 进行问题登记或在我们的技术支持与服务网站

<http://www.ad.siemens.com.cn/service/>点击“技术问题提交”进行问题登记, 等待西门子技术支持工程师回复。我们会在周一至周五 8:15 至 17:15(节假日除外)对您的技术问题进行解答。请注意在登记问题时尽量准确地描述所使用产品的类型, 以便尽快得到负责该产品的工程师的帮助。

技术支持传真: 010-64719991。

技术支持邮箱: adscs.china@siemens.com 或 4008104288.cn@siemens.com。

7. 1. 4 西门子自动化产品的其它网站

http://www.ad.siemens.com.cn/products/as/s7_200/ 提供S7-200 PLC相关知识及软件下载。

<https://mall.automation.siemens.com/CN/guest/>查找西门子自动化与驱动的所有产品订货号、图片、及技术参数, 或在技术支持与服务网站<http://www.ad.siemens.com.cn/service/>通过点击“A&D Mall” 进入该网站进行查询。