

# SIEMENS

## SIMATIC TDC 用户指导手册

SIMATIC TDC User Guide

**User Guide**

**Edition (2006-1)**

**SIEMENS A&D CS**

目录

第一章SIMATIC TDC控制系统 .....	4
1.1 TDC概述和特点 .....	4
1.2 TDC的应用领域 .....	6
1.3 TDC的设计思想和优点 .....	8
第二章 TDC系统的硬件和软件 .....	9
2.1 硬件架构 .....	9
2.2 组态举例 .....	15
2.3 图形化组态 .....	18
2.4 技术数据 .....	21
第三章TDC系统的组态 .....	27
3.1 软件要求 .....	27
3.2 描述和使用数据传输 .....	27
3.2.1 数据的连续性 .....	27
3.2.2 数据传输在同一CPU中同一任务中 .....	28
3.2.3 数据传输在同一CPU中不同任务中 .....	28
3.2.4 同一机架上数据传输在不同CPU的循环任务中 .....	29
3.2.5 数据传输在不同CPU的中断任务中 .....	30
3.2.6 减少死区时间 .....	31
3.3 CPU同步的意义和应用 .....	33
3.3.1 时间同步 .....	34
3.3.2 使本身的基本时钟与一个主CPU的基本时钟同步 .....	34
3.3.3 使本身的基本时钟与一个主CPU的中断任务时钟同步 .....	36
3.3.4 使本身的中断任务时钟与一个主CPU的中断任务同步 .....	39
3.3.5 多站的同步 .....	40
3.3.6 同步失败的响应 .....	40
3.4 精确计算处理器的利用率 .....	42
3.5 CPU的循环任务的工作方式 .....	43
3.6 数据传输模式 .....	44
3.6.1 握手数据传输模式 .....	44
3.6.2 刷新数据传输模式 .....	44
3.6.3 选择数据传输模式 .....	45

3.6.4 复合数据传输模式.....	46
第四章 通讯组态.....	47
4.1 CPU本地通讯.....	47
4.2 CPU-CPU之间通讯.....	61
4.3 Rack-Rack之间通讯 .....	65
4.4 MPI通讯.....	76
4.4 MPI通讯.....	76
4.4.1 CP50MO与WinCC通讯.....	76
4.4.2 CP50MO与OP27 通讯 .....	94
4.5 PROFIBUS-DP通讯.....	109
4.5.1 CP50MO作为主站 .....	110
4.5.2 CP50MO作为从站 .....	120
4.5.3 CP50MO同时作为主站和从站 .....	124
4.5.4 Shared Input方式.....	135
4.5.5 CP50MO与MM440 的通讯 .....	143
4.6 TCP/IP 通讯.....	155
4.6.1 CP51M1 与WinCC的标准通讯.....	156
4.6.2 CP51M1 与WinCC的PMC通讯.....	170
4.6.3 CP51M1 与CP343-1 通讯（TCP/UDP） .....	201
第五章 I/O组态 .....	217
5.1 数字量的读入/输出.....	217
5.2 模拟量的读入/输出.....	221
5.3 检测增量型编码器的输入 .....	224
5.4 检测绝对值型编码器的输入 .....	228
附录 .....	232
A.在COM PROFIBUS中安装新的.gsd文件.....	232
B.使用Symtrace-D7 可视化过程数据和逻辑分析.....	235

## 第一章 SIMATIC TDC 控制系统

### 1.1 TDC 概述和特点

如果您是一家工厂建设单位或工程单位，需要为工厂运营单位开发高性能的自动化解决方案，像金属冶炼、加工或输配电等领域。

而您正在寻找一种模块化的自动化系统，其性能允许大型系统的闭环控制和计算、开环控制和监视、信号传送和记录以及通讯。

同时在使用全集成自动化系列技术和驱动系统时，您还想降低成本，提高生产效率和竞争能力。

通过 SIMATIC TDC®（SIMATIC Technology and Drive Control），西门子可向您提供一种适用的多处理器自动化系统，尤其是驱动、能源和过程工程领域中的大型工厂。

SIMATIC TDC 擅长解决处理复杂的驱动、控制和通讯任务，在单一平台上拥有最大数量的框架和最短的循环周期，是对 SIMATIC S7 理想的扩充。SIMATIC TDC 是一种集成在 SIMATIC 中的工艺和驱动自动化系统，其组态和编程也可使用 SIMATIC 工具进行，因此也是全集成自动化理念中的一部分。

SIMATIC TDC 由一个或多个模板机架组成，其中可以插入所需模板。多处理器运行方式可以实现性能的几乎无限制扩展。使用 SIMATIC TDC，可以工厂操作员的指令变得高效、简洁而经济。

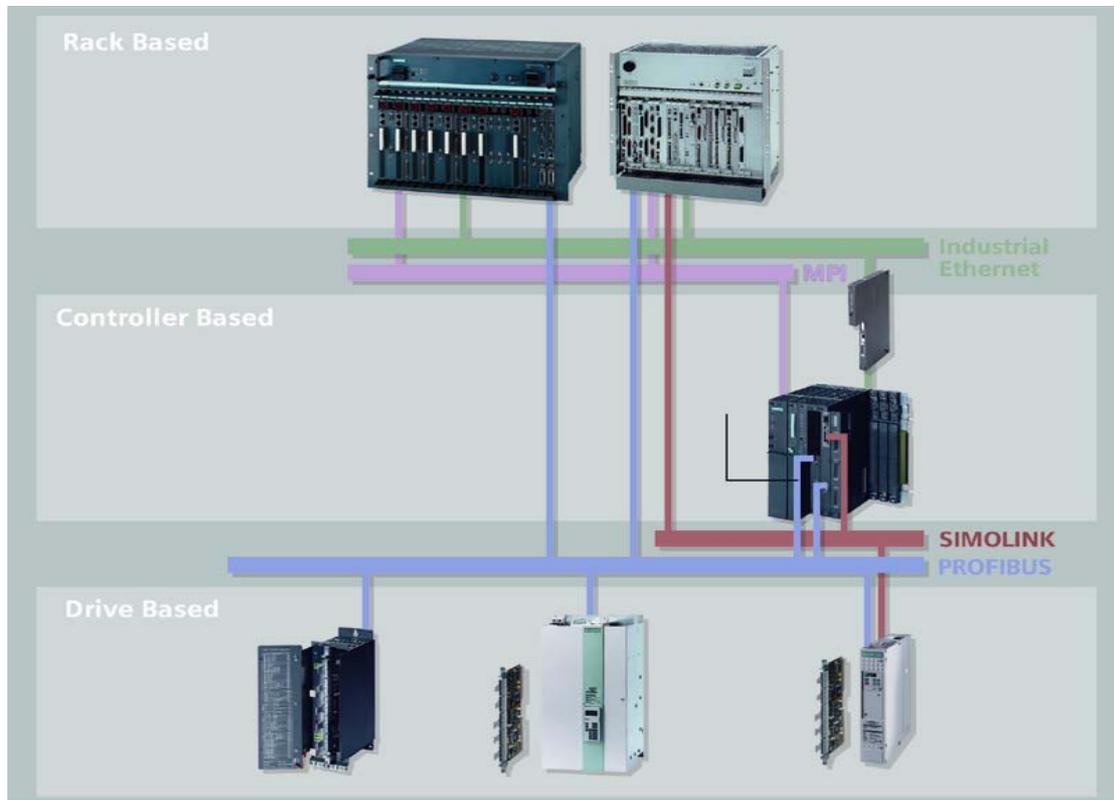
SIMATIC TDC 还是一种可靠而面向未来的系统，不但可以提高生产质量，而且还可以提高生产效率。

这一方面，SIMATIC TDC 得益于连续不断的研发，另一方面是长期可用性的保证，包括备件。

SIMATIC TDC的突出特性：

- 模块化的系统结构，硬件可扩展
- 采样时间间隔短，可达100ms，特别适用动态控制任务
- 中央处理器采用64位结构，具有最大性能

- 同步多处理器运行，每个机架最多可有20个CPU
- 可最多同步耦合44个机架
- 使用STEP7组态工具进行图形化组态：连续功能图(CFC)和顺序功能图(SFC)



## 1.2 TDC 的应用领域

SIMATIC TDC 的潜在用户不只是那些工厂建设单位,而且还包括为工厂运营单位开发自动化解决方案的工程与组态部门,尤其是以下领域:

- 金属冶炼
- 金属加工
- 配电
- 输电

SIMATIC TDC 可用于:

- 驱动系统的闭环控制(转矩、转速、位置、角度/ 角度偏差和速度),尤其是几个驱动系统协同运行或驱动系统之间的关系较复杂时
- 调节几个不同的物理变量(例如张力、压力等)
- 计算工艺/ 过程变量(例如温度)

SIMATIC TDC 可缩短运算循环(例如设定点 $<1\text{ ms}$ ),具有功能裕量,可提供优秀的灵活性。

SIMATIC TDC 的典型应用如下:

- 金属冶炼和金属加工:

拉丝设备、拉伸机、矫直机、弯曲机、拉直机、压机、圆弧形设备、轧机、锻造设备、剪切机和卷取机

- 用于长距离输电的高压直流输送系统,例如船用电缆
- 用于稳定电力输送的无功补偿系统,例如电容器单元、电容器组





### 1.3 TDC 的设计思想和优点

借助于 SIMATIC TDC 的高性能，可以实现极短的循环时间 (100  $\mu$ s)，因此具有较高的输出性能，显著提高生产效率。进而提高经济性能和竞争能力。

通过应用 SIMATIC TDC，在改进功能性的同时，高压直流输电系统(HVDC) 的空间要求可降低 50%。

使用 SIMATIC TDC，可以在单一平台上实现开环和闭环控制功能。由于部件数量大大减少，这可显著降低系统购置成本以及备件库存成本。

通过使用经过广泛应用、测试的标准 SIMATIC 工具，还可大大降低工程与组态成本。

- 用于硬件组态的 STEP 7®
- 用于图形化组态的 CFC (工艺图)和SFC (流程控制图)

另外，由于可以继续使用现有程序结构甚至是自有软件模块，用户已有经验可继续应用。

同时标准 SIMATIC 组态工具的使用，还可大大降低维护和培训成本。丰富的诊断功能可显著提高工厂的可用性水平。

标准的一致性使用，例如通讯和HMI 标准，也使得 SIMATIC TDC 的使用非常简便：

- PROFIBUS DP 和工业以太网
- SIMATIC WinCC® 和 SIMATIC 操作员面板

总之，SIMATIC TDC 具有以下优点：

- 凭籍其最大的运算能力，可显著提高生产效率和竞争能力
- 降低购置成本
- 减少工程成本
- 全球通用标准

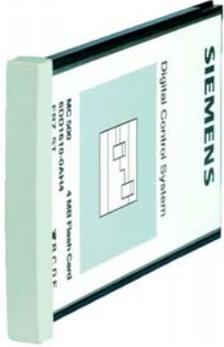
## 第二章 TDC 系统的硬件和软件

### 2.1 硬件架构

SIMATIC TDC 是一种模块化的多处理器系统，由一个或多个机架组成。在机架上可以安装 CPU、I/O 模板和通讯模板。下表介绍了在 TDC 系统中使用的各个组件。

	<p><b>机架 UR5213</b></p> <p>电磁屏蔽 19" 机架 UR5213 允许硬件扩展，具有较高的性能裕量。它适用于墙壁安装和箱柜安装，配装有一个具有冷却和内部监控功能的集成电源。</p> <p>总共有 21 个槽位用于扩展模板，并可通过 64 位背板总线连接。对于较高的性能要求，在一个机架内可最多有 20 个同步 CPU 模板多处理器运行，可有 44 个机架相互连接在一起。</p>
	<p><b>中央处理单元 CPU551</b></p> <p>中央处理单元 CPU551 适用于具有较高运算要求的开环和闭环控制任务。它采用 64 位 RISC 技术设计，拥有一个 32 MB 的用户存储器和插入式存储器模块。</p> <p>CPU 可保证严格根据可调的采样时间间隔 (100 <math>\mu</math>s) 进行循环处理。对于每个循环，操作系统本身只需要 25...50 s 的循环时间。这就意味着运算时间非常短，例如每个 PI 控制器大约为 1...3 s。另外，CPU 还集成有 4 个具有报警能力的数字量输入和一个诊断接口。</p>

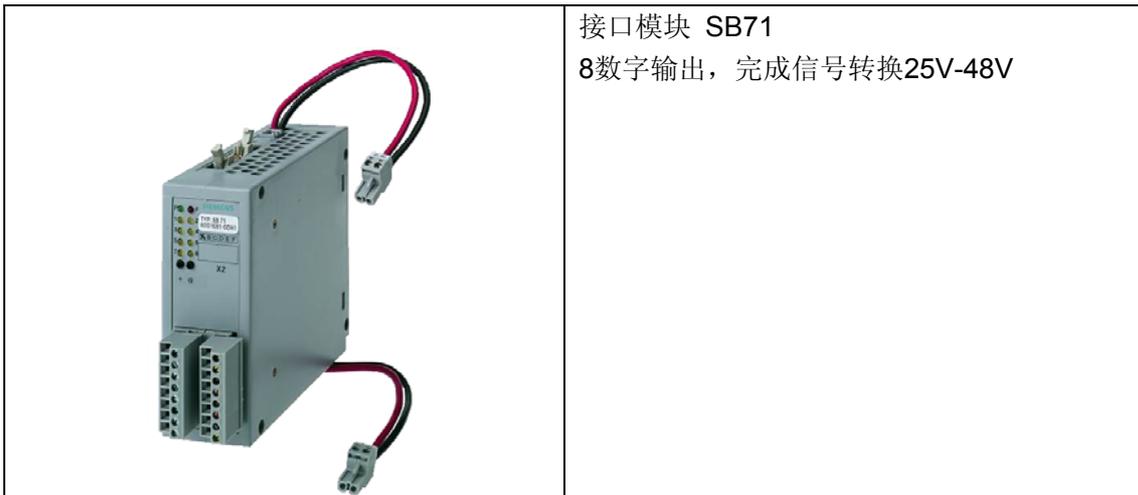
	<p><b>I/O 模板 SM500</b></p> <p>SM500 I/O 模板提供有丰富的选项，用于连接分布式I/O。除了16点二进制输入/输出以外，它还具有8点模拟量输入/输出以及4点积分型模拟量输入。另外，还可连接4个增量位置编码器和4个绝对值编码器。使用6个 LED 指示模板的运行状态。</p>
 <p style="text-align: center;">CP50MO</p>	<p><b>通讯模板 CP50MO</b></p> <p>通讯模板 可提供高性能的通讯，用于</p> <ul style="list-style-type: none"> <li>• 调试</li> <li>• 过程控制</li> <li>• HMI</li> </ul> <p>它们可以控制功能强大的协议</p> <ul style="list-style-type: none"> <li>• MPI</li> <li>• PROFIBUS DP</li> </ul> <p>还为各种不同的 CPU，提供有一个1 MB 的通讯存储器。</p>
 <p style="display: flex; justify-content: space-around;"> <span>CP5100</span> <span>CP51M1</span> </p>	<p><b>通讯模板 CP51M1, CP5100</b></p> <p>通讯模板 CP51M1 和 CP5100 可提供高性能的通讯，用于</p> <ul style="list-style-type: none"> <li>• 调试 (CP51M1)</li> <li>• 过程控制</li> <li>• HMI</li> </ul> <p>它们可以控制功能强大的协议</p> <ul style="list-style-type: none"> <li>• TCP/IP</li> <li>• UDP</li> <li>• ISO on TCP(CP51M1)</li> <li>• S7(CP51M1)</li> </ul> <p>CP5100需要组态在Slot18~21。CP51M1最多组态4块在机架上。</p>

 <p>CP52M0      CP52IO      CP52A0</p>	<p><b>全局数据存储器 GDM</b></p> <p>通过全局数据存储器 (GDM) ， 安装有 CP52x0 的大量机架相互之间可以进行通讯， 运算能力几乎可以无限制扩展。通过光纤电缆和共享存储器， 可最多有 44 个机架联网。</p> <p>除了几个机架之间的通讯之外， 使用 GDM， 还可实现同步（ 采样时间、 时钟时间） 和报警功能。刷新时间&lt;1ms。</p> <p>只有1个CP52M0在其中一个机架上且位于 Slot1。 CP52IO与CP52M0位于同一个机架可组态在Slot2~12。</p>
	<p><b>插槽盖板 SR51</b></p> <p>如果机架中存在空余插槽， 那么就可以利用插槽盖板。这样有利于系统冷却和保证系统的 EMC特性。</p>
	<p><b>程序存储器模块MC500, MC510,MC521</b></p> <p>用于存储由用户组态的处理器程序， 硬件组态和CFC</p>

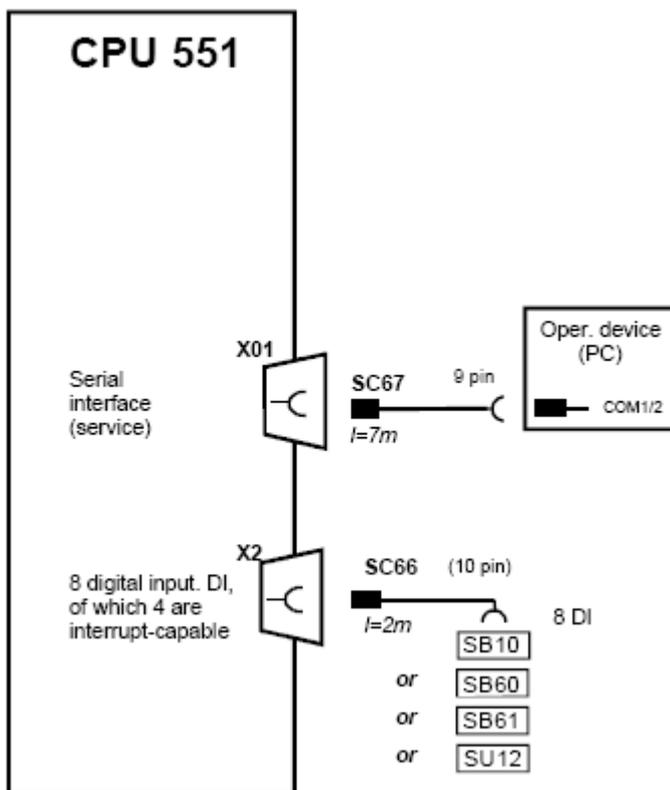
 A coiled grey cable with a green multi-pin connector on one end and a white multi-pin connector on the other.	<p>接口电缆 SC62 用于连接SM500模板</p>
 A coiled light green cable with a green multi-pin connector on one end and a green multi-pin connector on the other.	<p>接口电缆 SC63 用于连接SM500模板的数字输入</p>
 A coiled light green cable with a green multi-pin connector on one end and a white multi-pin connector on the other.	<p>接口电缆 SC66 用于连接CPU模板上的数字输入模块</p>
 A coiled light green cable with a green multi-pin connector on one end and a white multi-pin connector on the other.	<p>接口电缆 SC67 用于CPU模板与PG/PC的连接</p>

 A grey Siemens SU12 interface module with 10 screw terminals on the top and a connector on the bottom.	<p>接口模块 SU12 电气上是一一对一的连接，没有信号转换 10个螺钉型端子</p>
 A green Siemens SU13 interface module with 50 screw terminals on the front.	<p>接口模块 SU13 电气上是一一对一的连接，没有信号转换 50个螺钉型端子</p>
 A grey Siemens SB10 interface module with 8 digital I/O channels, an LED display, and a connector with red and black wires.	<p>接口模块 SB10 电气上是一一对一的连接，没有信号转换 8个数字型I/O，LED显示</p>

 A Siemens SB60 interface module, a grey rectangular device with a terminal block on the front and a connector on the top. It has a label with technical specifications and a small LED display.	<p>接口模块 SB60 电气上隔离，完成信号转换230V-24V 8数字输入，LED显示</p>
 A Siemens SB61 interface module, similar in design to the SB60 but with a different internal configuration. It features a terminal block and a connector.	<p>接口模块 SB61 电气上隔离，完成信号转换48V-24V 8数字输入，LED显示</p>
 A Siemens SB70 interface module, designed for signal conversion from 24V to 230V. It has a terminal block and a connector.	<p>接口模块 SB70 电气上隔离，完成信号转换24V-230V 8数字输出，LED显示</p>



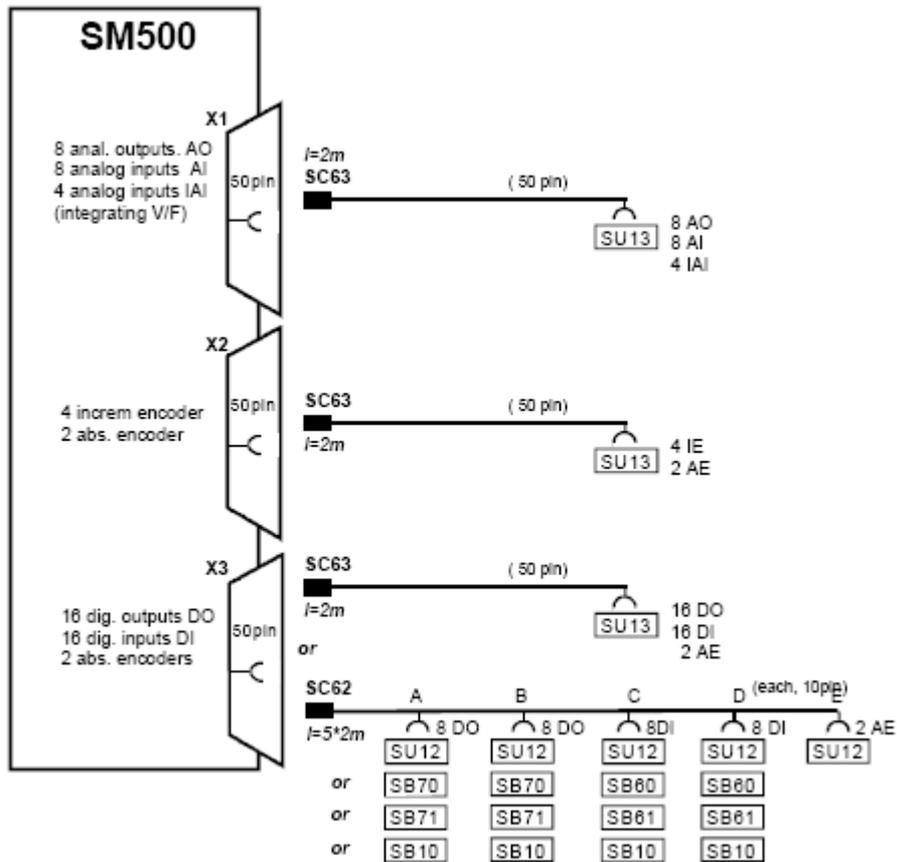
## 2.2 组态举例



例子1

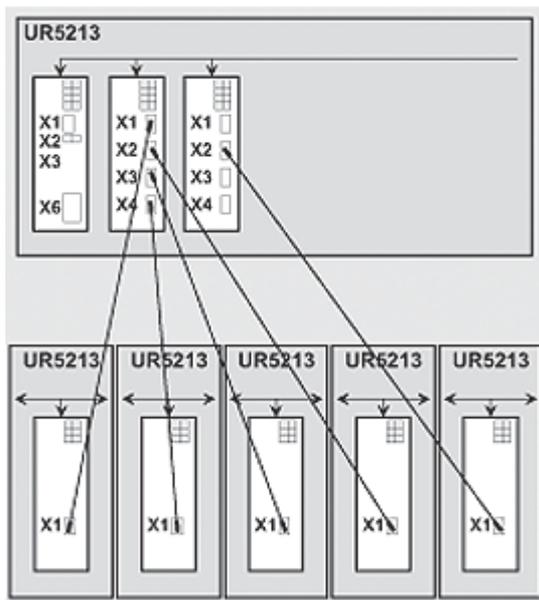
例子1中通过使用SC67把CPU551的串口与PC相连，通过SC66把CPU551的8DI输入口与

接口模块SB10或SB60或SB61或SU12相连。



例子2

例子2中I/O模板SM500分别通过连接电缆SC63或SC62与相应的接口模块相连。

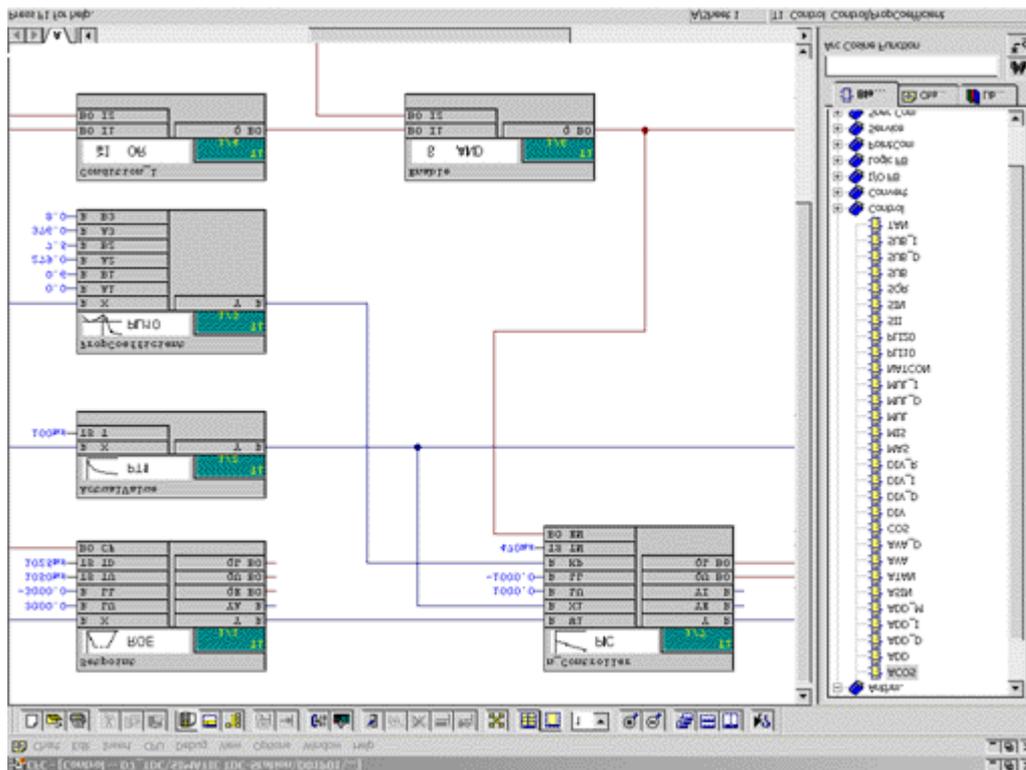


例子3

例子3中最上方UR5213作为GDM机架，CP52M0组态在最左侧插槽，其他插槽组态CP52A0。

对于其他机架只需要组态一块CP52A0，通过光缆，即可以组成多机架的组态连接。





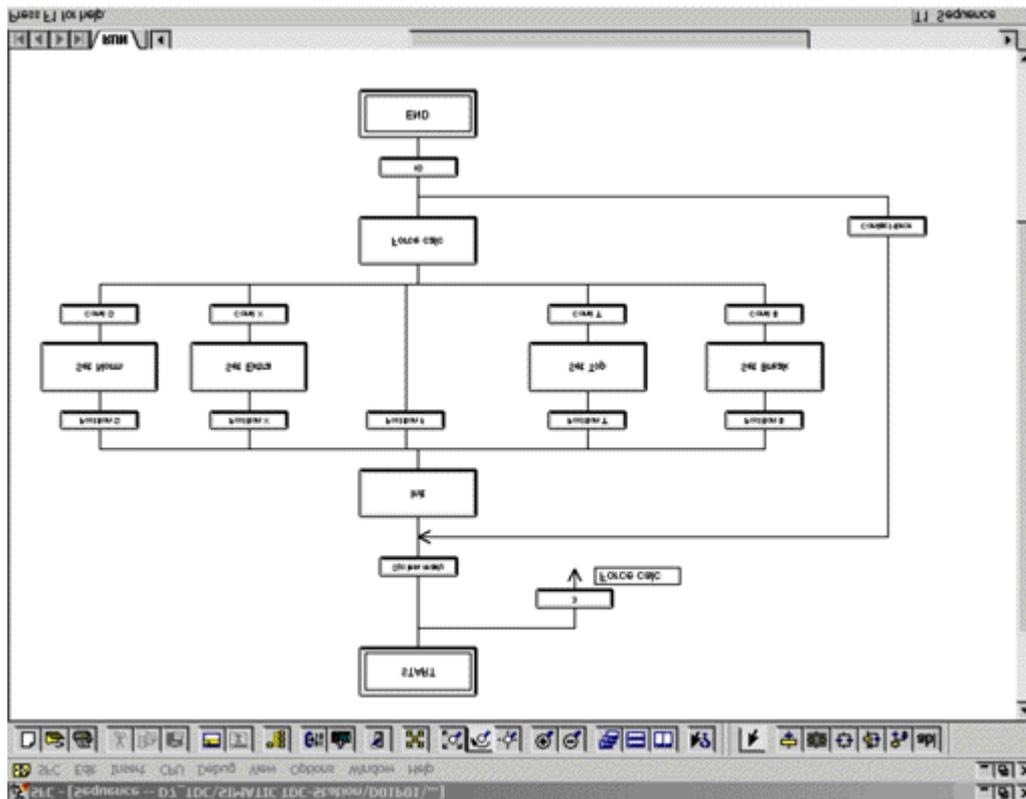
### 连续功能图 (CFC) 中库中功能块的图形化连接

通过简单的编译，即可由此生成具有最佳运行时间的机器代码。CFC图可很容易地进行复制和修改。功能强大的在线功能可显著缩短调试时间。在 CFC 中，可以在线插入和删除程序块，创建、修改或删除功能块之间的连接。因此可以在线优化用户程序，避免费时的编译和装载过程。

使用 CFC 图，可以对用户程序进行自动而简便的图形化归档，以及非常简便的程序维护。另外，CFC 图还可移植到其它硬件平台，例如 FM458-1 DP 应用模板。

对于特殊任务还提供有其它工具：

- 使用顺序功能图 (SFC)，可图形化生成顺序控制
- 从 C 资源，功能块生成器可创建一个可执行的功能块，并可在CFC 中调用。由于算法被隐藏起来，可以进行程序移植以及专用技术的保护。
- 使用样条曲线编辑器，可以对电子凸轮盘进行组态。



使用顺序功能图 (SFC) 进行顺序控制器的图形化组态

## 2.4 技术数据

机架 UR5213	
输入电压范围	90 ~ 264 V, 47~ 63 Hz
外形尺寸 (宽×高×深)	482.6 x 354.9 x 343
重量	约20 kg
防护等级	IP20
工作温度范围	0 °C ~ +55 °C
订货号组	6DD1682-0CH0

中央处理单元 CPU551	
数字量输入	数量8 点输入, 其中4 点输入具有报警能力
隔离	通过可选接口模板实现
输入电压	
• 额定电压	24 V
• 对信号“0”	-1 V ~ +6 V
• 对信号“1”	+13.5 V ~ +33 V
输入电流	
• 对信号“0”	0 mA
• 对信号“1”	3 mA
• 延迟时间	100 s
实时时钟, 分辨率	0.1 ms
订货号组	6DD1600-0BA1
附件	
存储器卡 MC 521	2 MB 闪存
订货号组	6DD1 610-0AH3
存储器卡 MC 500	4 MB 闪存
订货号组	6DD1 610-0AH4
存储器卡 MC 510	8 MB 闪存
订货号组	6DD1 610-0AH6
功能块库	D7-SYS
订货号组	6DD1 801-5DA.
功能块生成器	D7-FB-Gen
订货号组	6DD1 805-5D..
SIMATIC 手册汇编订货号组	6ES7998-8XC01-8Y..

I/O 模板 SM500	
模拟量输出	
数量	8
设计	输出接地
电流隔离	-
输出电压范围	-10 V ~ +10 V
输出电流	10 mA
分辨率	12 位
每个通道的转换时间	4 s
精度:	
• 最大微分线性度误差 (单一性保证)	± 1 LSB
• 最大放大误差	± 0.3 %
• 最大偏移误差	± 24 LSB
转换速率	约 3.5 V/ s 1)
电压输出:	
• 短路保护, 机架接地	是
• 短路电流约	100 mA 1)
模拟量输入	
数量	8
设计	差分输入
电流隔离	-
输入电压范围	-10 V ~ +10V
分辨率	12 位
每个通道的最大转换时间	约 20 s 1)
精度:	
• 最大微分 线性度误差	1 LSB (没有偏差代码)
• 最大放大误差	± 0.3%
• 最大偏移误差	± 5 LSB
输入电阻	20 k
输入滤波器	34 kHz
极性反转保护	是, 正弦差分输入
集成模拟量输入 (U/f)	
数量	4

设计	差分输入
电流隔离	-
输入电压范围	-10 V ~ +10 V
分辨率	取决于积分时间，例如 15 位积分时间为 4 ms
最大积分/ 采样时间	可组态
精度:	
• 最大放大误差	1%
• 最大积分线性度误差	0.05%
• 最大偏移误差	2 LSB( 软件补偿)
输入电阻	470 k
输入滤波器	2 kHz
极性反转保护	是， 正弦差分输入
数据输出	
数量	16
电流隔离	通过可选接口模板实现
外部电源	
• 额定电压	24 V
• 允许范围	20 V ~ 30 V
• 瞬时	35 V, 最大 0.5 s 时
• 最大空载电流	40mA
输出电压范围:	
• 信号“0”， 最大	3 V
• 信号“1”， 最小	外部电源-2.5 V
输出电流:	
• 信号“0”， 最小	- 20 A
• 对信号“1”	50 mA
• 额定值	
• 最大允许范围	100 mA
延迟时间	100 s
阻性负载输出的最大工作频率	6 kHz
短路保护	
• 机架接地	是
• 外部电源	-
最大短路电流	250 mA
输出总电流 ( 最高 60 °C )	16 x 50 mA
感性截止电压限制外部电源电压	+1 V
数字量输入	DI
数量	16
隔离	通过可选接口模板实现

输入电压:	
• 额定电压	24 V
• 对信号“0”	-1 V ~ +6 V
• 对信号“1”	+13.5 V ~ +33 V
输入电流	
• 对信号“0”	0 mA
• 对信号“1”	3 mA
延迟时间	100 s
增量编码器输入数量	4
可连接类型	增量式位置编码器。90°跟踪相位，前后跟踪
设计	差分输入，在 15 V (HTL)和 5 V (TTL) 编码器信号之间 可切换
跟踪信号	跟踪 A、B，带或不带指针信号前 (FW) 后 (BW) 跟踪
跟踪信号的最小相位差	200 ns
输出脉冲频率 (跟踪频率)	1 MHz, 2.5 MHz
输入电压	
<b>15 V 编码器:</b>	
• 允许范围	-30 V ~ +30 V
• 对信号“0”	-30 V ~ +4 V
• 对信号“1”	+8 V ~ +30 V
<b>5 V 编码器:</b>	
• 允许范围	-7 V ~ +7 V
• 对信号“0”	-7 V ~ -0.7 V
• 对信号“1”	+1.5 V ~ +7 V
输入电流	
• 15 V 编码器:	5.0 mA (典型/绝对值)
• 5 V 编码器:	1.5 mA (典型/绝对值)
控制输出未用	控制输入如数字量输入参数
报警复位输出	
• 短路保护，机架接地	是
• 外部电源	-
• 最大短路电流	20 mA
报警输入:	
• 输入电压 (允许范围)	0 V ~ 5 V
• 信号“0”，最大	< 0.5 V
• 信号“1”，最小	> 2.0 V
输入电流	

•“0”信号	- 2.8 mA
•“1”信号	1.6 mA
编码器电源	
数量	1
隔离	-
典型输出电压	13.5 V
最大输出电流	150 mA, 短路保护, 机架接地, 短路电流 250 mA1)
绝对值编码器输入	
数量	4
设计	差分输入, RS485
可连接类型	单个或多个编码器
协议	SSI, EnDat
数据格式	格雷码; 二进制码
数据流向	
• 单向	SSI
• 双向	EnDat
数据位	SSI: 13+ 奇偶校验位, 25+ 奇偶校验位, EnDat: 变量
最大脉冲频率	2 MHz, 取决于电缆长度
输入电压	
• 允许范围	RS 485
订货号组	6DD1640-0AH0

1) 取决于环境条件

通讯模板CP50MO	
用于 PROFIBUS DP/MPI 的连接	2 x RS 485
支持协议	ROFIBUS DP, MPI
波特率最大	12 Mbit/s (DP)
最大	187 kbit/s (MPI)
订货号组	6DD1661-0AD0
通讯模板 CP5100	
用于工业以太网的连接	1 x RJ 45
支持协议	TCP/IP, UDP
波特率	10/100 Mbit/s, 自适应
订货号组	6DD1661-0AE0
通讯模板 CP51M1	
用于工业以太网的连接	1 x RJ 45
支持协议	TCP/IP, UDP, ISO on TCP, S7
波特率	10/100 Mbit/s, 自适应

订货号组	6DD1661-0AE1
通讯模板 CP52M0	
全局数据存储器 GDM 的接口	是
数据输出	
数量	16
隔离	-
外部电源:	
• 额定电压	24 V
• 允许范围	20 V ~ 30 V
• 瞬时	35 V, 最大 0.5 s 时
• 最大空载电流	40 mA
输出电压范围:	
• 信号“0”, 最大	3 V
• 信号“1”, 最小	外部电源电压-2.5 V
输出电流:	
• 信号“0”, 最小	-20 A
• 对信号“1”	
• 额定值	50 mA
• 最大允许范围	100 mA
延迟时间	100 s
阻性负载输出的最大工作频率	6 kHz
短路保护	
• 机架接地	是
• 外部电源	-
最大短路电流	250 mA
输出总电流 (最高 60. C)	16 x 50mA
感性截止电压限制外部电源电压	+1 V
订货号组	6DD1660-0BF0
接口模板	CP5210
订货号组	6DD1660-0BG0
存取模板	CP52A0
订货号组	6DD1660-0BH0

## 第三章 TDC 系统的组态

### 3.1 软件要求

软件：

需要的软件包

- ①STEP7 V5.3 SP2
- ②CFC V6.1
- ③D7-SYS V6.2
- ④COM PROFIBUS V5.1

以上软件要按上述顺序准确地安装在您的 PG/PC 中，要求的操作系统为 Windows95/98/ME/NT4.0/2000/XP，STEP7、CFC 和 D7-SYS V6.2 需要安装授权。使用上述版本的软件编写本书的例子程序证明是有效的。

### 3.2 描述和使用数据传输

数据传输就是在不同功能块之间的数据交换。



图 3.2.两个任务之间的数据传输

#### 3.2.1 数据的连续性

SIMATIC TDC 保证在不同周期的循环任务中的数据传输所有的数据连续性。这就意味着在某一任务中的要传输的所有数据都是来自于这个任务的同一计算周期。在采样时间过程中计算任务中的数据，任务结束后数据被导出存储在 CPU 的缓存中。若某一任务开始时，需要的数据就会从缓存中导入。由此要保证不能同时读写数据缓存区。

有以下 4 种数据传输类型：

- 数据传输在同一 CPU 中同一任务中
- 数据传输在同一 CPU 中不同任务中
- 数据传输在不同 CPU 的循环任务中
- 数据传输在不同 CPU 的中断任务中

### 3.2.2 数据传输在同一 CPU 中同一任务中

在 TDC 系统中功能块的每一个输出都分配给一个存储区域。功能块被系统处理后，输出值保存在存储区域中。与功能块输出相连的功能块的输入就从该存储区域中取出存储的输出值。为了减少这一存取时间，那么系统会使用数据流的方式进行数据传输，也就是功能块的输出作为其他功能块的输入被首先计算。在同一任务中的 CFC 组态，只需要直接互连即可。

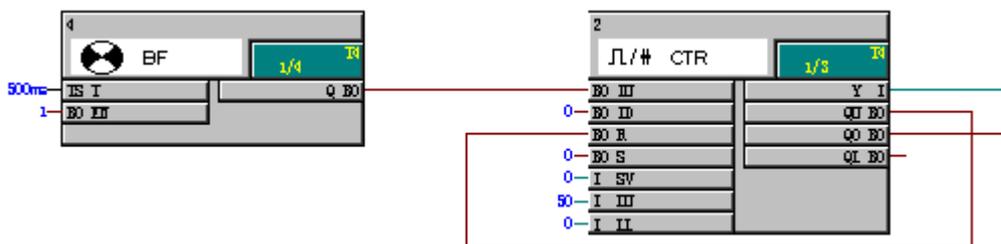


图 3.2.2.同一任务中的数据传输

在图中显示数据传输发生在同一任务周期 T4。

### 3.2.3 数据传输在同一 CPU 中不同任务中

数据传输在同一 CPU 中的不同任务是通过 CPU 上的一个缓存系统来实现的。特别对于，循环时间较快的循环任务发送数据给循环时间较慢的循环任务时，在循环时间较慢的循环任务可以观察到两种情况，一种是检测不到循环时间较快的循环任务的数据变化，另一种就是可以通过例如增加脉冲延时功能块等方法去检测到循环时间较快的循环任务延迟的数据变化。

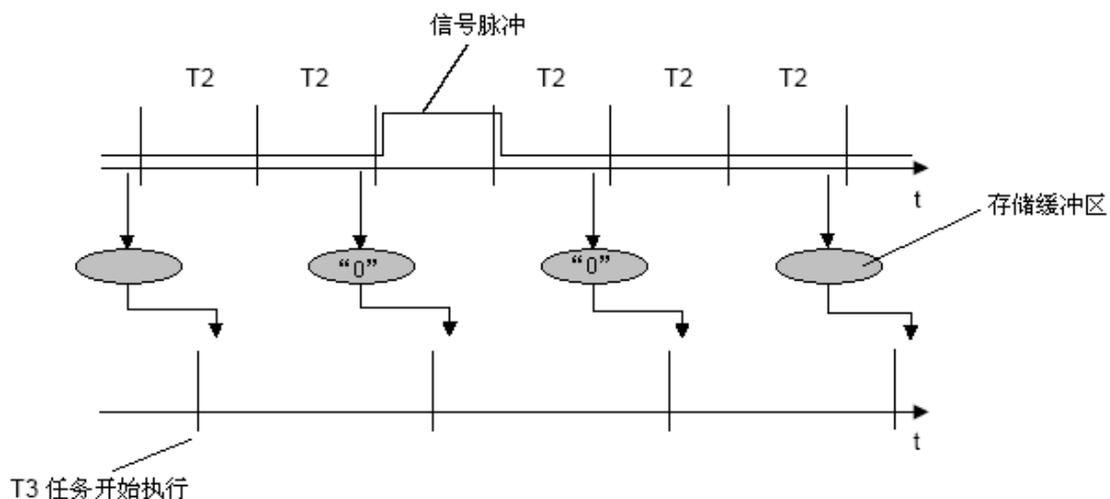


图 3.2.3a T2 周期的数据没有通过时间延时发送给 T3 周期

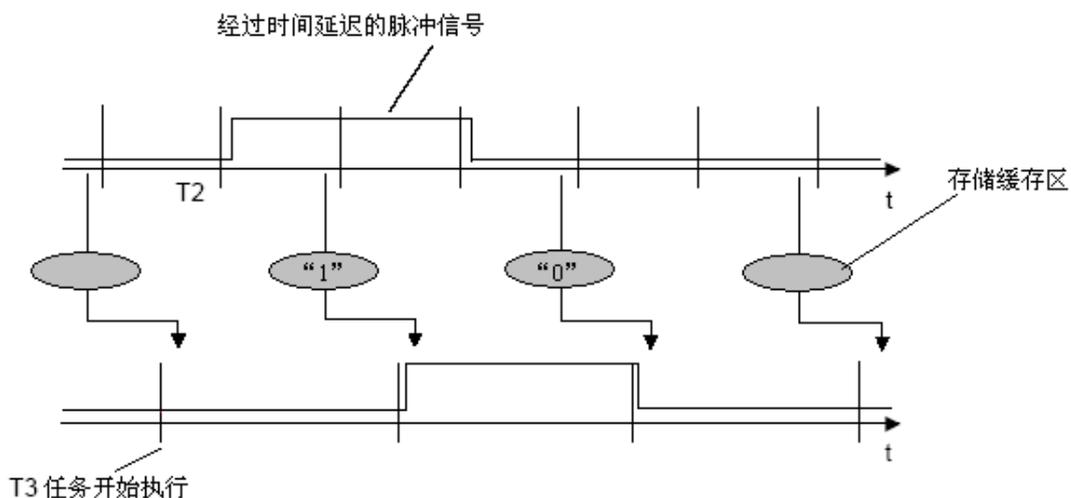


图 3.2.3b T2 周期的数据经过时间延时发送给 T3 周期

### 3.2.4 同一机架上数据传输在不同 CPU 的循环任务中

通过使用 CP50MO 和 CP51M1 通信缓冲模块，同一机架上的不同 CPU 之间可以进行数据传输。那么 \$ 信号就是在不同 CPU 中不同功能块之间的连接。组态 \$ 信号需要

- 信号名(Name)
- 类型(Standard 或 Fast)
- 总线路径(L bus)，TDC 只有 L bus

信号传输是否需要系统的连续性检查

- 需要连续性检查 (Standard)
- 不需要连续性检查 (Fast)

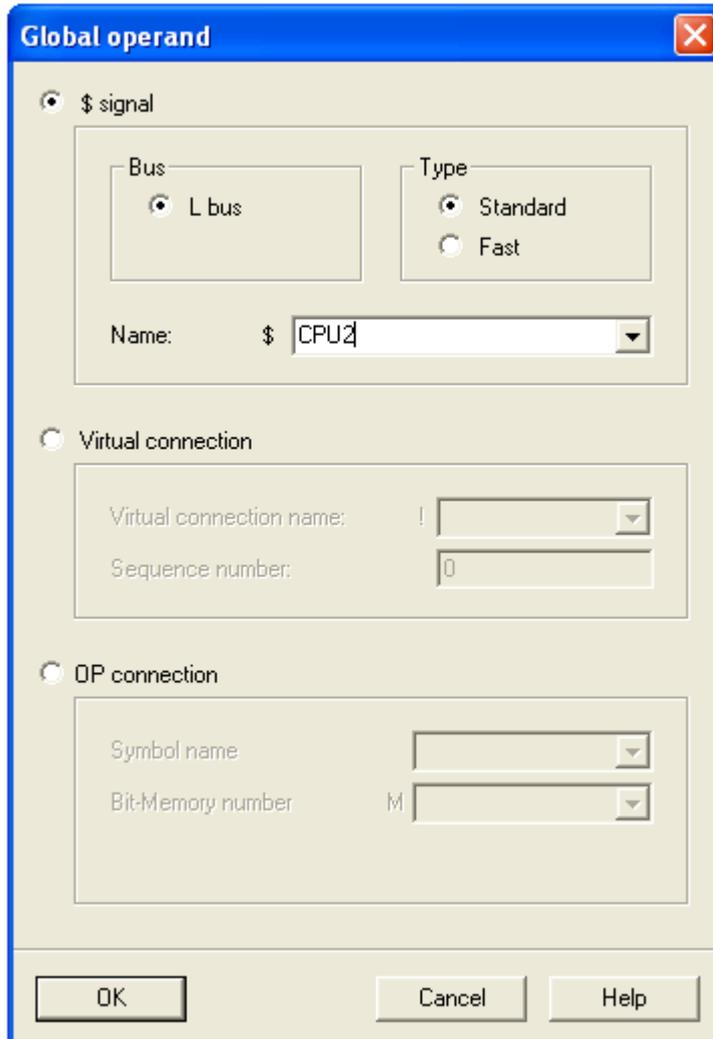


图 3.2.4 \$信号的组态

### 3.2.5 数据传输在不同 CPU 的中断任务中

如果数据的产生和使用在中断任务中，那么需要组态 **Fast\$**信号。这是由于中断时间可以发生在时间的任意时刻，所以为了减少数据传输丢失，不需要连续性检查来组态 **Fast\$**信号。

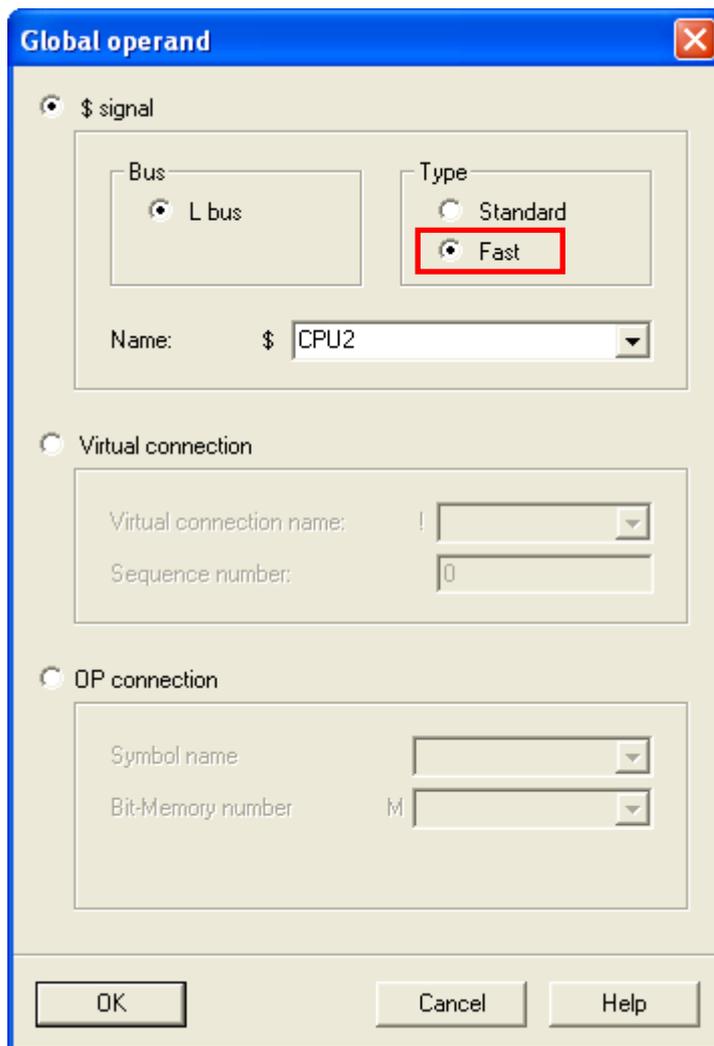


图 3.2.5 Fast \$信号的组态

### 3.2.6 减少死区时间

死区时间就是在同一时间任务中或相邻不同时间任务中，写数据缓存区和读该数据缓存区之间的时间间隔，死区时间是不可避免的，否则会产生同时读写数据缓冲区。为了减少缓冲时间，数据可以直接传送而不经数据缓冲区，从而逃避系统的数据连续性检查。

对于“数据传输在同一 CPU 中同一任务中”或“数据传输在同一 CPU 中不同任务中”可以使用假拟注释（Pseudo comment）@DATX 来完成同一 CPU 的任务互连。

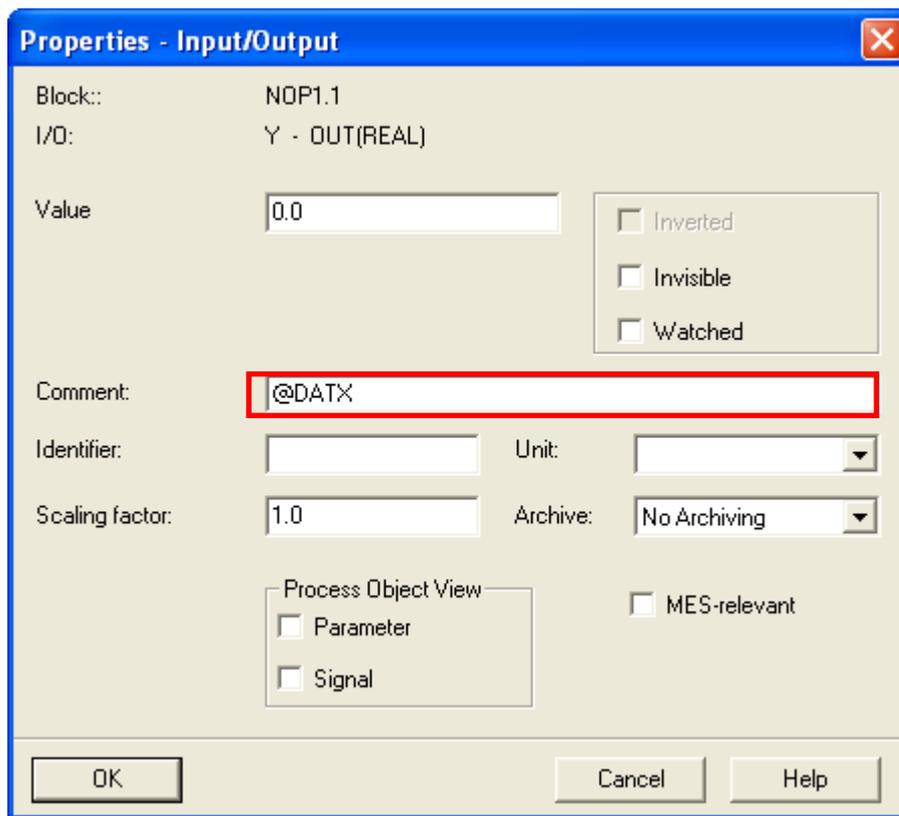


图 3.2.6a. @DATX 的组态

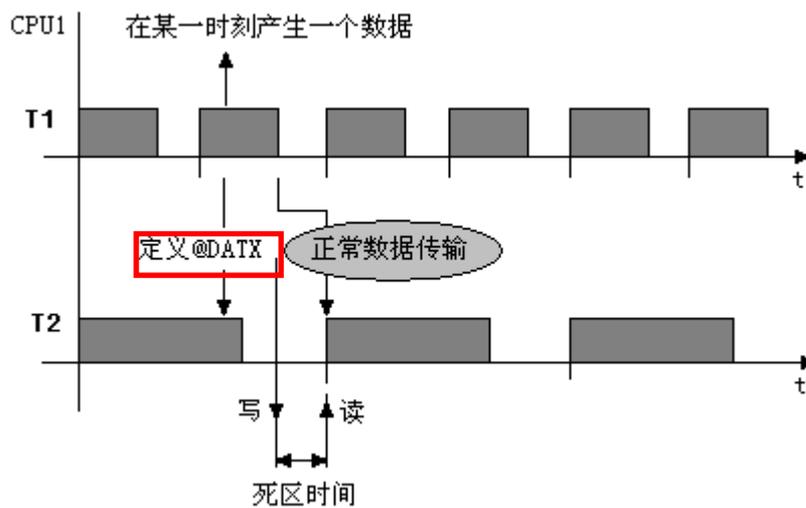


图 3.2.6b. 组态@DATX 的数据传输模式

对于“数据传输在不同 CPU 的循环任务中”和“数据传输在不同 CPU 的中断任务中”可以使用 Fast\$ 信号来完成不同 CPU 之间的任务互连。

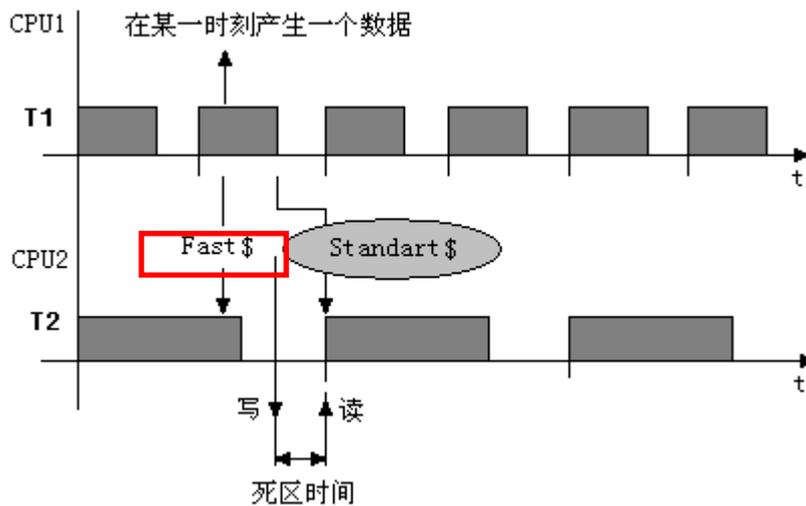


图 3.2.6c. Fast \$信号与 Standard \$在信号数据传输模式的比较

### 3.3 CPU 同步的意义和应用

组态 CPU 的同步在 Step7 中的 HWConfig 中完成。通过双击硬件组态中的 CPU，出现 CPU 属性的对话框。

TDC 提供下列同步机制：

- 时间同步
- 使本身的基本时钟与一个主 CPU 的基本时钟同步
- 使本身的基本时钟与一个主 CPU 的中断任务时钟同步
- 使本身的中断任务时钟与一个主 CPU 的中断任务同步
- 多站的同步
- 同步失败后的响应

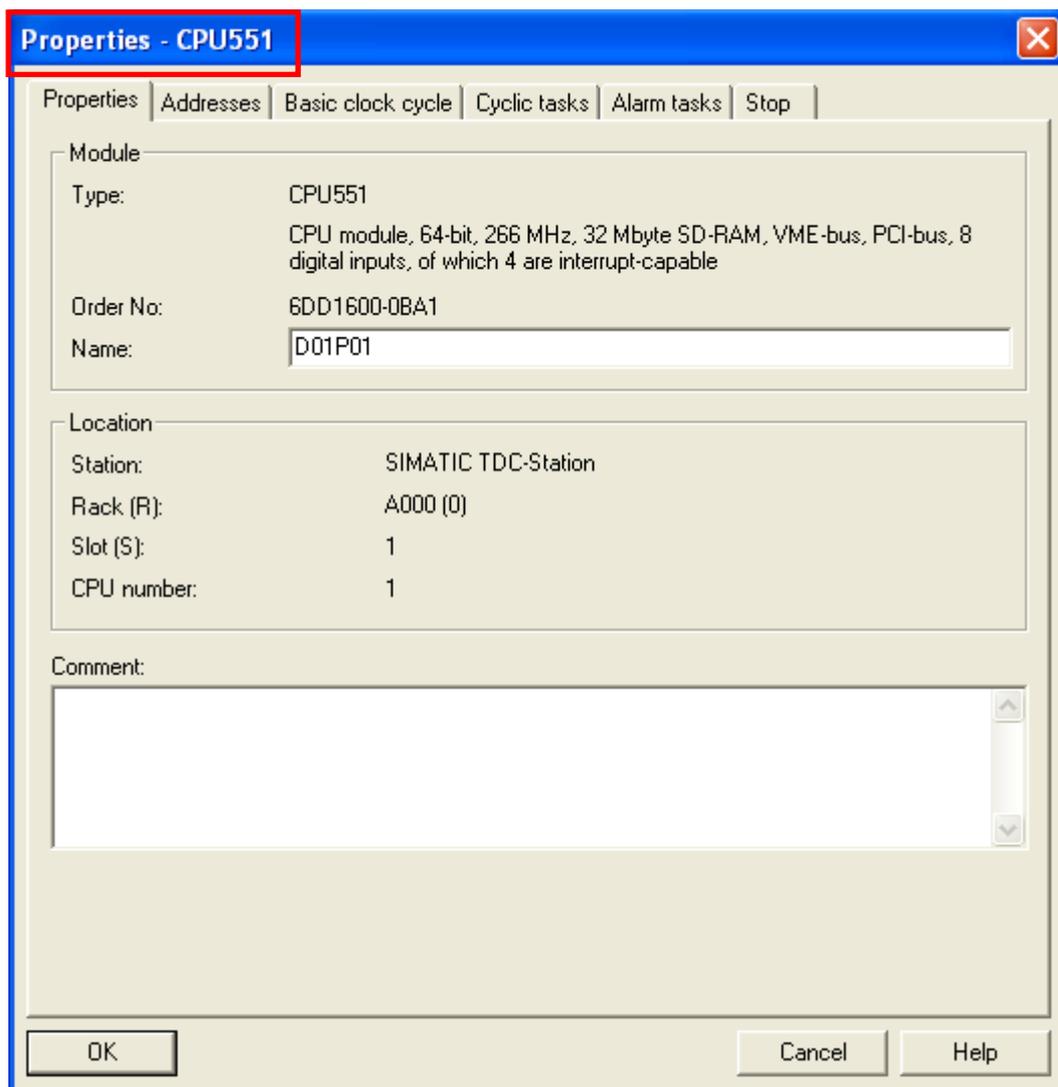


图 3.3a. CPU 的属性页

### 3.3.1 时间同步

在 TDC 的同一机架上的所有 CPU 的实时时钟都被插在 1 号槽的 CPU 所同步（即靠在最左面的 CPU）。这样会防止不同 CPU 之间的时钟漂移。这种同步是每隔 10 秒自动实现。

### 3.3.2 使本身的基本时钟与一个主 CPU 的基本时钟同步

一个 CPU 的基本时钟可以通过 L bus 传递，可以由其它 CPU 接收到。其它的 CPU 可以在同一机架上的 CPU，也可以是通过 GDM 耦合的其他机架上的 CPU。

组态的例子就是同一机架上的不同 CPU。

Slot	Name	Type	Order number
<b>1</b>	<b>D01P01</b>	<b>CPU551</b>	<b>6DD1600-0BA1</b>
1.1	D01_1	MC521	6DD1610-QAH3
1.2			
<b>2</b>	<b>D02P02</b>	<b>CPU551</b>	<b>6DD1600-0BA1</b>
2.1	D02_1	MC521	6DD1610-QAH3
2.2			
3			

图 3.3.2a. 硬件组态

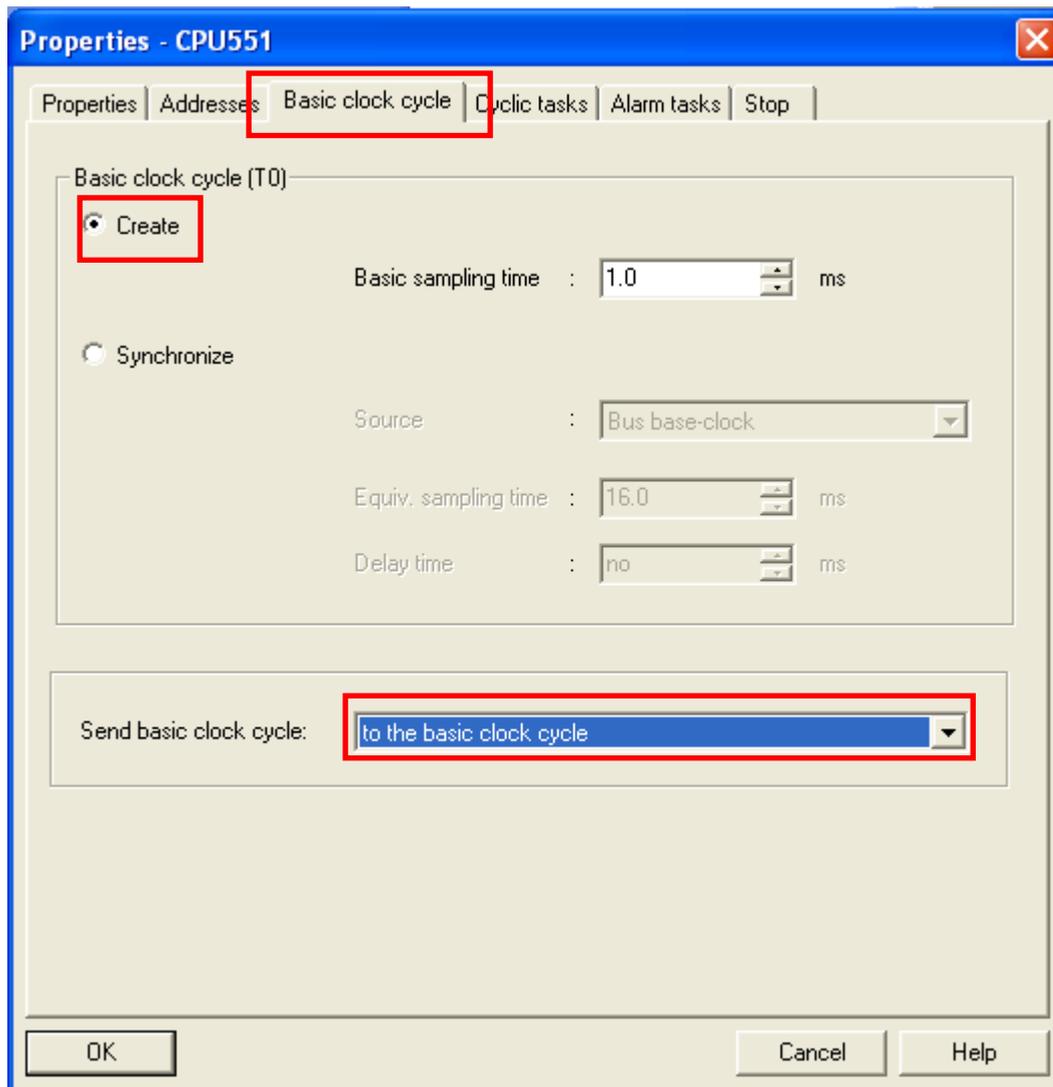


图 3.3.2b. 设置 Slot1CPU 发送同步时钟

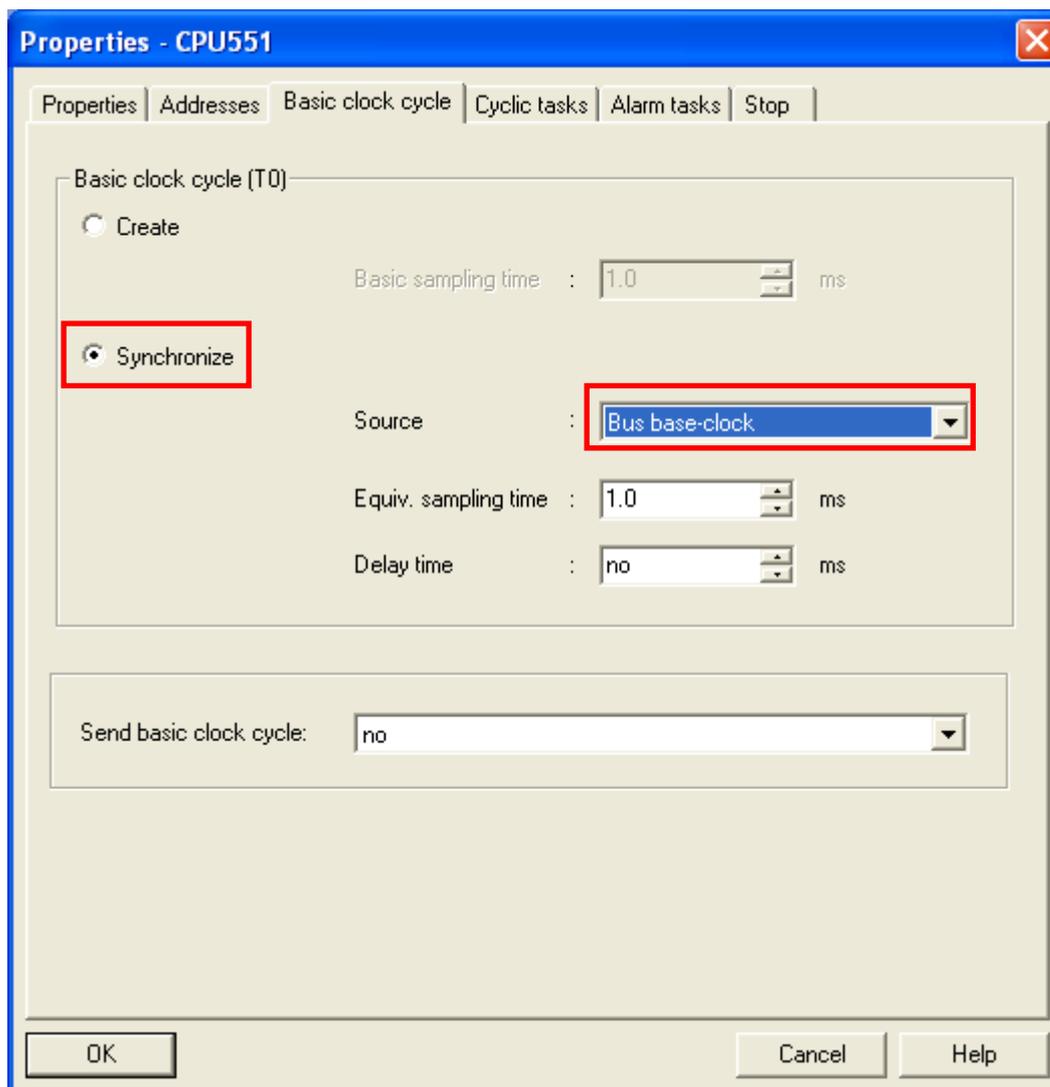


图 3.3.2c. 设置 Slot2 CPU 接收同步时钟作为基本时钟

### 3.3.3 使本身的基本时钟与一个主 CPU 的中断任务时钟同步

一个 CPU 的中断任务的开始和结束，是可以初始化 L bus 中断的。可以由其它 CPU 接收到并作为该 CPU 的基本时钟。其它的 CPU 可以在同一机架上的 CPU，也可以是通过 GDM 耦合的其他机架上的 CPU。

组态的例子就是同一机架上的不同 CPU。

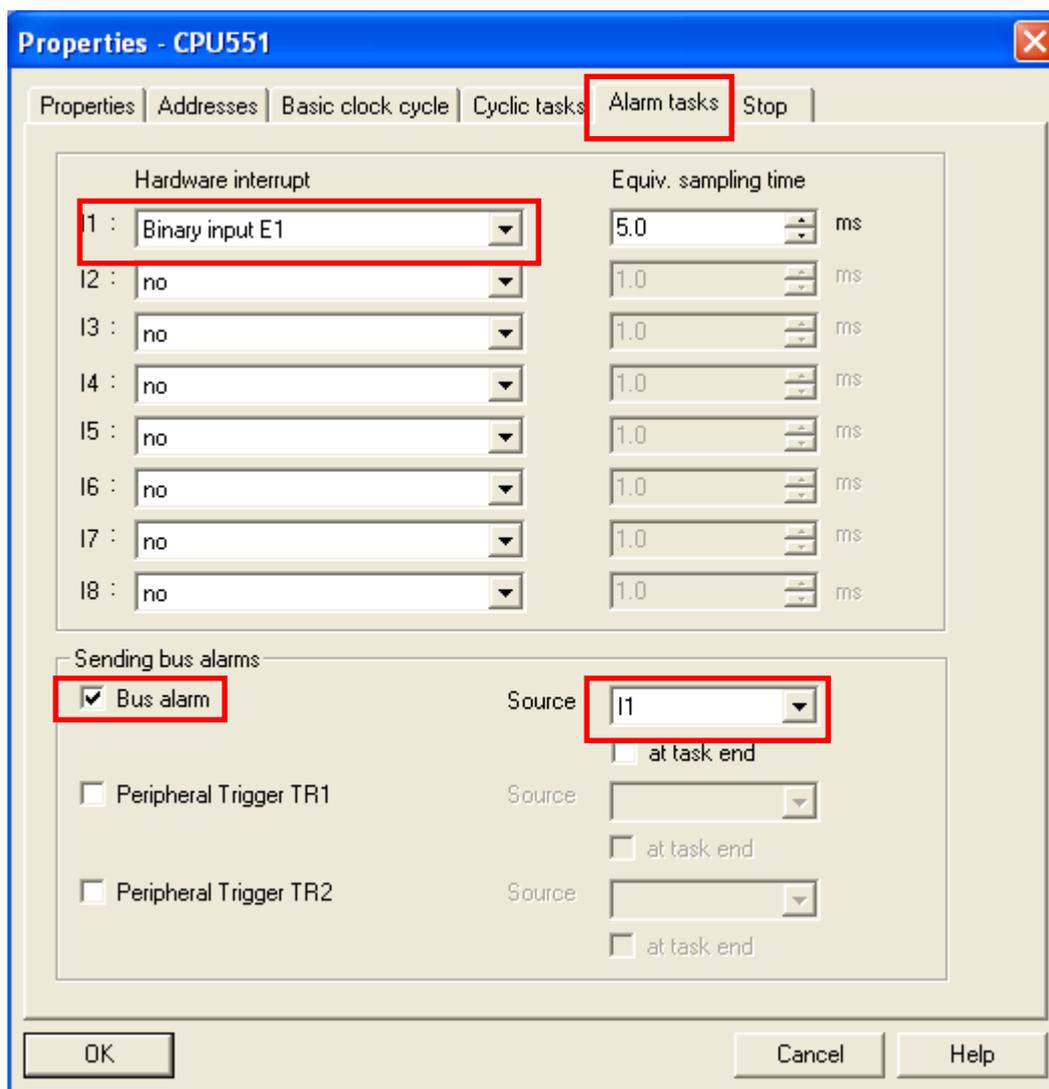


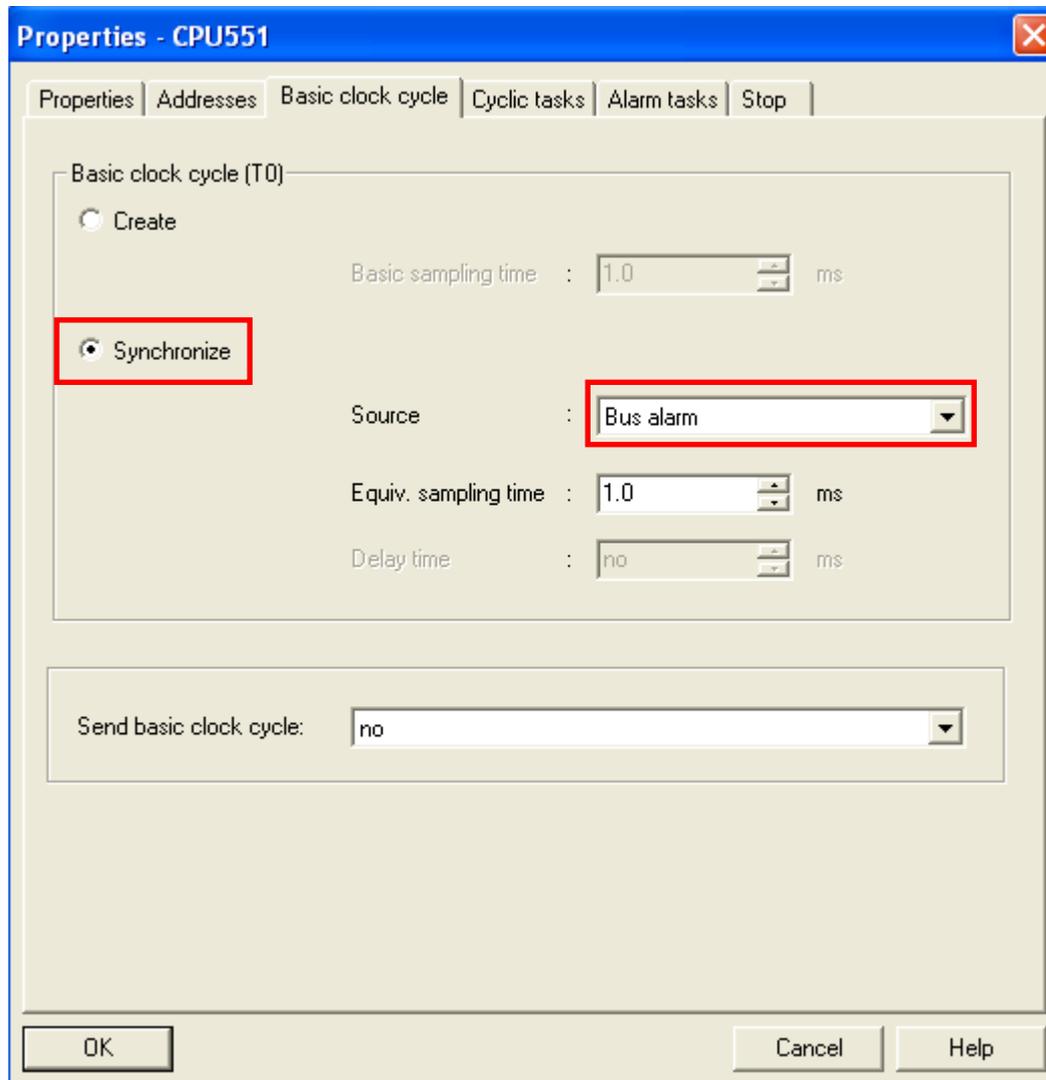
图 3.3.3a. 设置 Slot1CPU 发送中断任务时钟

CPU1 需要设置中断处理任务 **Hardware interrupt**，本例中通过二进制输入 **Binary Input E1** 实现中断任务，该 **BOOL** 量是 CPU 模板上带有中断能力的 4 个 **BOOL** 量输入信号，可以参考 CPU 模板技术数据。

**Equiv. sampling time** 是针对在中断任务中，如果存在与时间相关的功能块，例如积分功能块。那么该设置的时间就是用来计算积分的步长。在下图中为 **INT** 功能块的 **F1** 帮助。**TA** 就是组态的 **Equiv. sampling time**。

产生的中断通过 **L-Bus** 发送到另一个 **CPU2** 上，通过设置总线中断即 **Bus alarm**。中断源就是刚设置的 **I1** 即 **Binary Input E1**。如果没有选择 **at task end**，那么就是在二进制中断 **Binary**

Input E1 产生中断任务 I1 开始就立刻发送总线中断，否则在 I1 中断任务结束后，才发送总线中断。



LU if LL >= LU.

The discrete values are calculated using the following algorithm (TA is the sampling time in which the function block is configured):

$$Y_n = Y_{n-1} + \frac{TA}{TI} \cdot X_n$$

**Y<sub>n</sub>** Value from Y in sampling interval n

**Y<sub>n-1</sub>** Value from Y in sampling interval n-1

**X<sub>n</sub>** Value from X in sampling interval n

图 3.3.3b. 设置 Slot2 CPU 接收中断任务时钟作为基本时钟

### 3.3.4 使本身的中断任务时钟与一个主 CPU 的中断任务同步

通过初始化 L bus 中断，在主 CPU 的中断任务的开始和结束时来同步其它的 CPU 的中断任务。其它的 CPU 可以是在同一机架上的 CPU，也可以是通过 GDM 耦合的其他机架上的 CPU。

组态的例子就是同一机架上的不同 CPU。

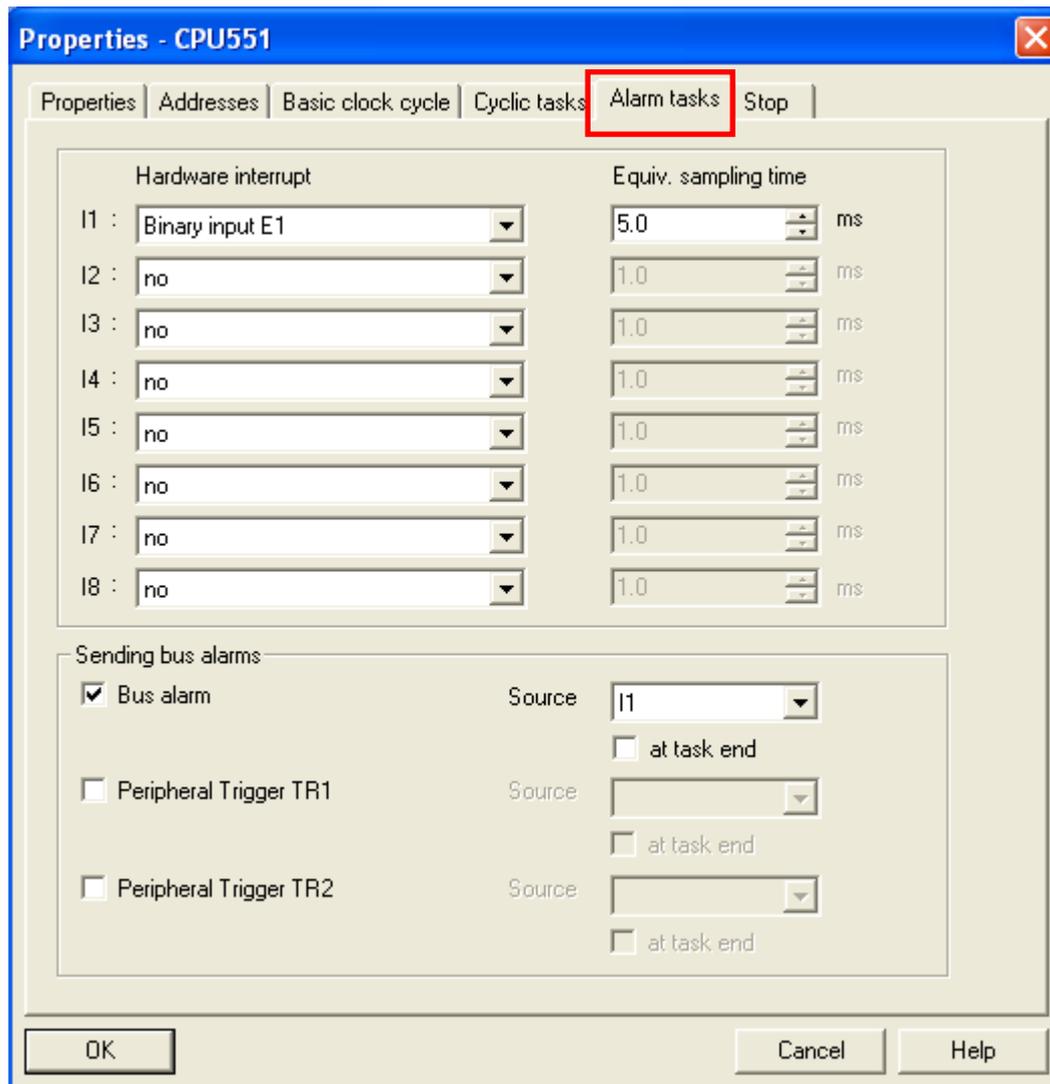


图 3.3.3a. 设置 Slot1 CPU 发送中断任务时钟

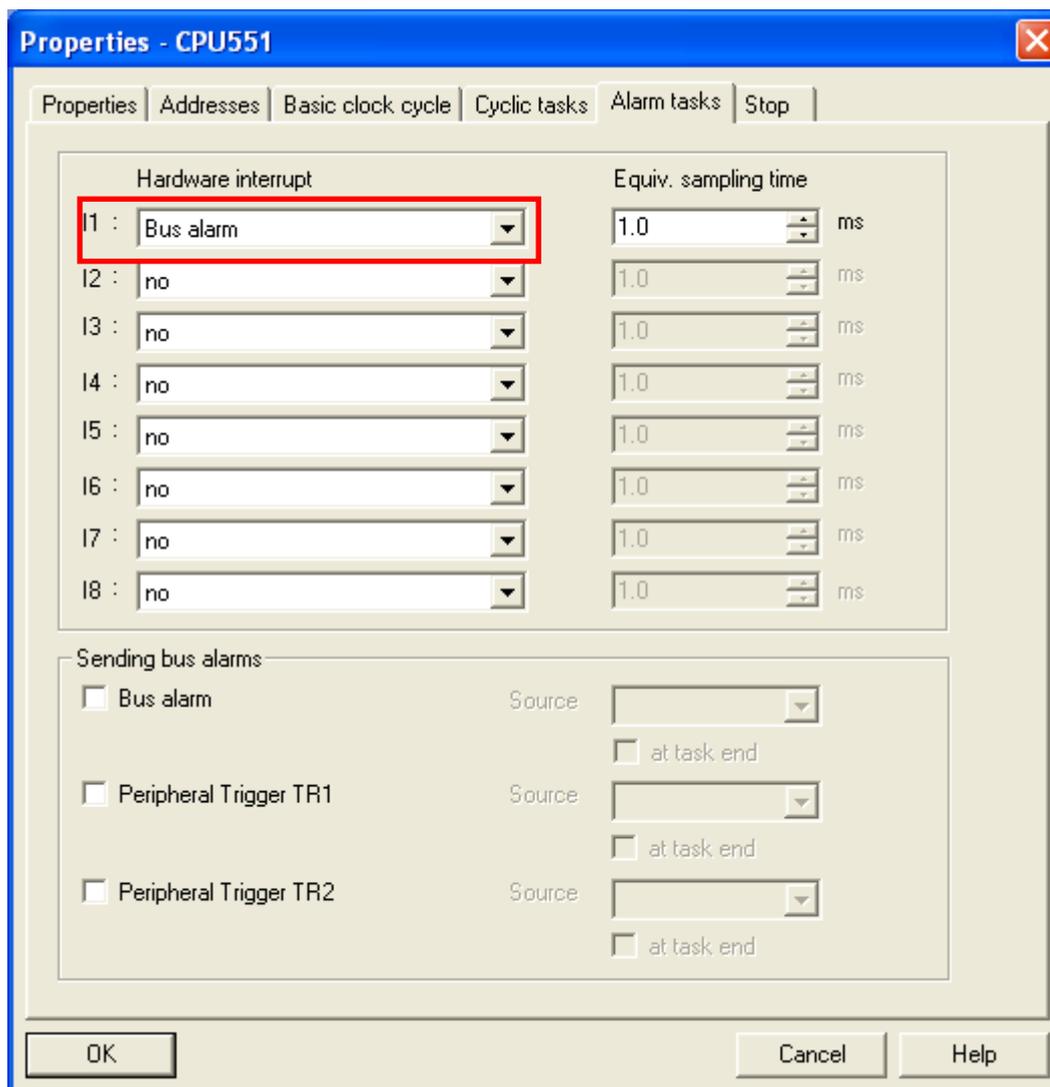


图 3.3.3b. 设置 Slot2 CPU 接收中断任务时钟

### 3.3.5 多站的同步

TDC多站之间通过全局数据存储器GDM安装有CP52M0, CP52IO。大量机架相互之间可以利用CP52A0与CP52IO相连, 实现多个机架的通讯, 运算能力几乎可以无限制扩展。通过同步基本时钟实现多机架的同步。同3.3.2节。

### 3.3.6 同步失败的响应

被同步的接收方通过自身的硬件定时器来监控基本同步时钟。如果发现在4个时钟内没有发现发送过来的时钟, 那么接收方就用本身硬件组态的时钟来产生基本时钟,这个时钟就是

Equiv. sampling time，需要注意的是这个Equiv. sampling time与Alarm tasks中的Equiv. sampling time是不同的，前者范围是从0.1ms-16.0ms；后者没有范围要求。两者的使用意义是不同的。请参考下图。这时CPU的7段数码管闪烁E。如果发送方时钟恢复，接受方可以使用功能块DTS再次与发送方时钟同步。

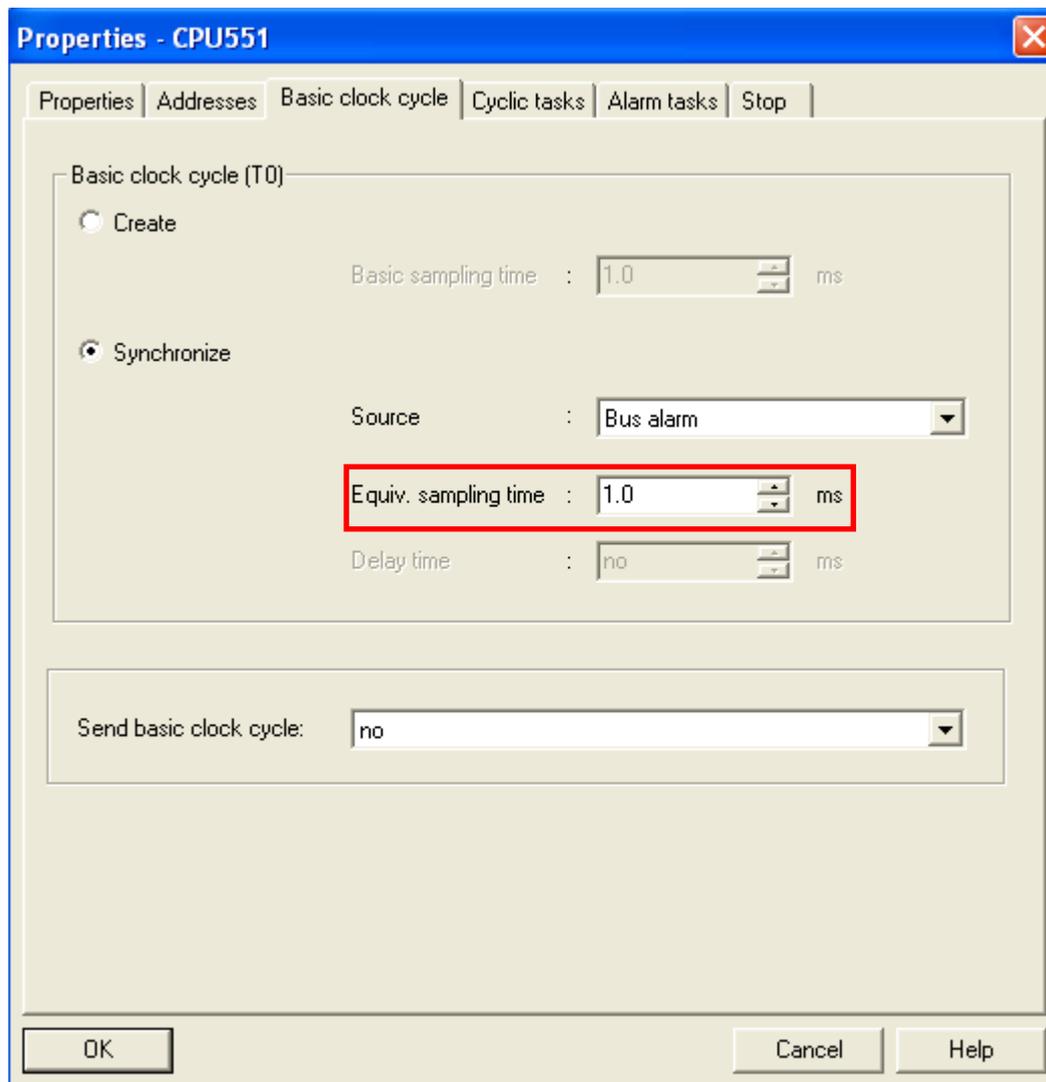


图 3.3.6a. Basic clock cycle 中的 Equiv. sampling time

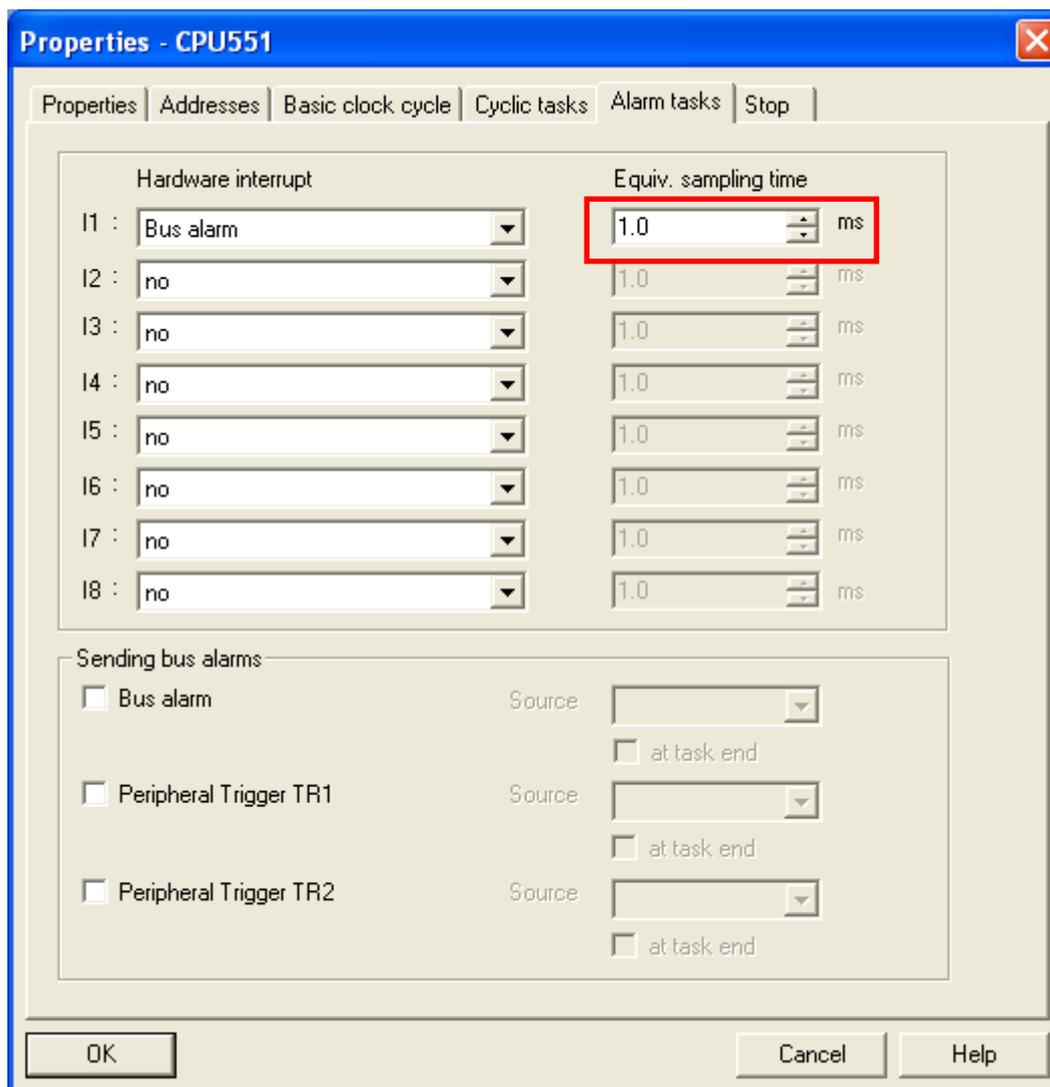


图 3.3.6b. Alarm tasks 中的 Equiv. sampling time

### 3.4 精确计算处理器的利用率

在程序中使用PSL ("Permanent System Load") 功能块，可以判断5个循环周期中的CPU的负载。该功能块可以组态在任意循环周期中。

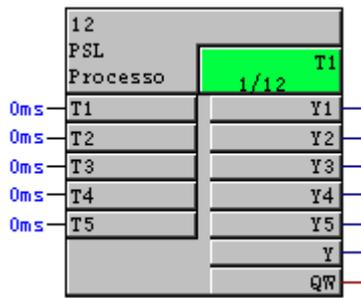


图3.4. 功能块PSL

Y1~Y5表示CPU中的不同任务周期的负载率，与T1~T5对应。正常时不会超过1.0（100%），如果超过1.0，那么就说明该CPU过载。

如果在输入引脚T1~T5处输入XXms，就可以利用该数字来模拟该任务周期增加的时间负载。这样就会在对应的Y1~Y5处，读到增加的负载值。

### 3.5 CPU 的循环任务的工作方式

如果任务需要的处理时间很短，那么这个任务可以在一个基本时钟周期完成，那么这就是第一个循环。如果这个任务不能在一个基本时钟周期内完成，那么它会在随后的基本时钟周期内完成。用短时采样时间处理的任务要在用长时采样时间处理任务完成之前处理完。也就是T1在T2前，T3在T4前等等。

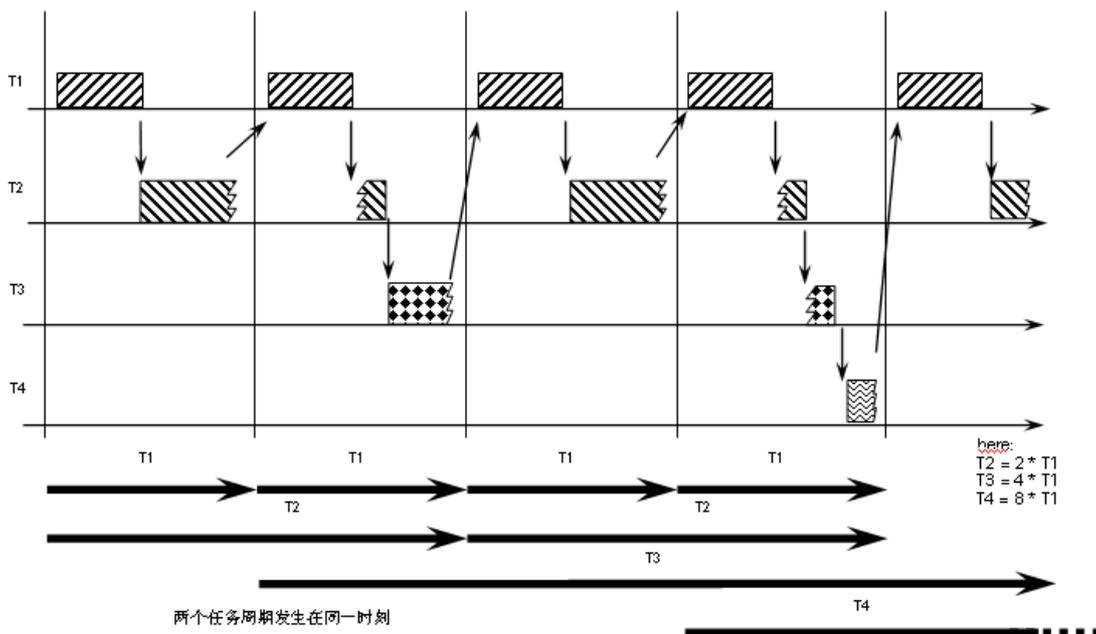


图3.5. CPU的循环任务

### 3.6 数据传输模式

根据不同网络的通讯要求，有以下 4 种不同的数据传输模式

以下 4 种数据传输模式：

- 握手（handshake）
- 刷新（refresh）
- 选择（select）
- 复合（multiple）

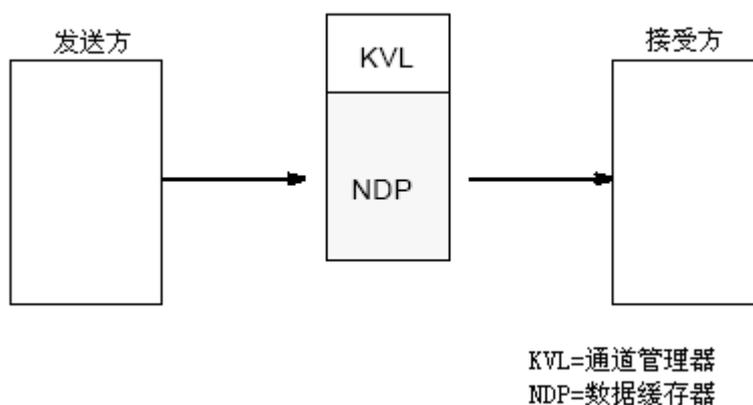
这 4 中数据传输模式指定在 CTV 或 CRV 功能块的 MOD 输入上。

#### 3.6.1 握手数据传输模式

在两种情况下使用握手传输模式

第一种情况是当数据被覆盖时，不会发生数据丢失；第二种情况是每一个发送方都会有一个接收方与其对应。

握手模式定义了一个连续的通道处理。具体的流程是接收方收到这个数据，通过给发送方发送确认信息表明接收数据完成，然后发送方在通道中放置一个新的数据。这个通道提供一个数据缓冲区用于数据传送。发送方发送数据和接收方接收数据是分别通过各自的一个操作循环来完成。

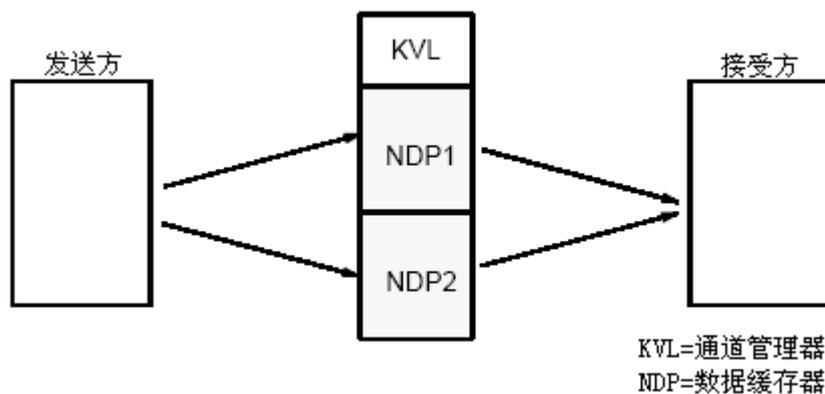


#### 3.6.2 刷新数据传输模式

在两种情况下使用刷新传输模式

第一种情况是接收者总是接收到最新的数据；第二种情况是每一个发送方都会有一个接收方与其对应。

刷新模式定义了数据覆盖的通道处理。具体的流程是发送方总是在通道中放置最新的数据，接收方总接收最新的数据而不需要给发送方提供应答信息。这个通道提供 2 个数据缓冲区用于数据传送，这两个数据缓冲区中相互交替存放发送方提供的最新数据。发送方通过设置标志来表明哪一个数据缓冲区中的数据是最新的。



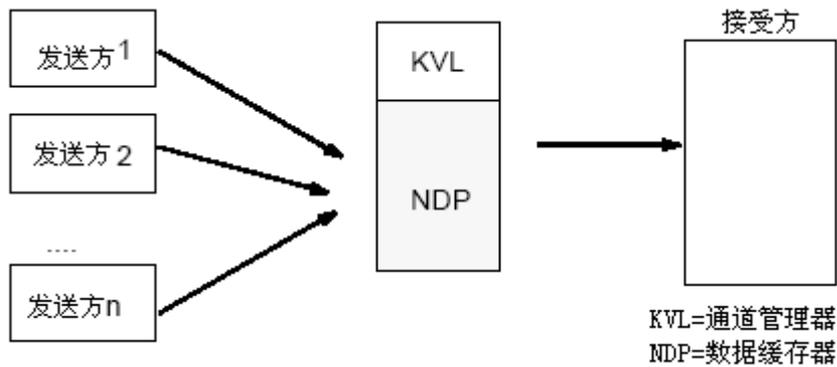
### 3.6.3 选择数据传输模式

在两种情况下使用选择传输模式

第一种情况是当数据被覆盖时，不会发生数据丢失；第二种情况是每一个接收方有多个发送方与其对应。

选择模式定义了一个连续的通道处理。具体的流程是接收方收到这个数据，通过给发送方发送确认信息表明接收数据完成，然后发送方在通道中放置一个新的数据。这个通道提供一个数据缓冲区用于数据传送。发送方发送数据和接收方接收数据是分别通过各自的一个操作循环来完成。这一点与握手模式相同。

所有组态的发送方都使用相同的数据缓冲区。如果不定义发送方之间的发送顺序，那么第一个发送方就首先发送。为了控制发送方的发送顺序，可以通过分时的给功能块EN设置为1。需要注意的是组态的发送方的时间任务周期一定要小于接收方的时间任务周期。



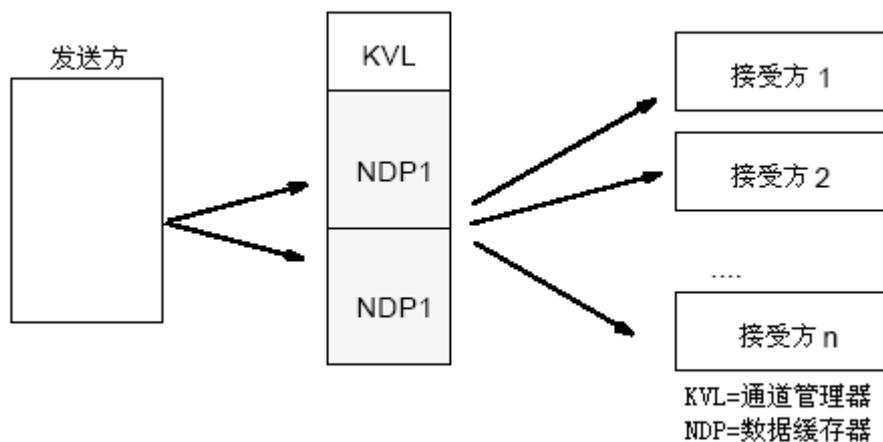
### 3.6.4 复合数据传输模式

在两种情况下使用复合传输模式

第一种情况是接收者总是接收到最新的数据；第二种情况是每一个发送方都会有多个接收方与其对应。

刷新模式定义了数据覆盖的通道处理。具体的流程是发送方总是在通道中放置最新的数据，接收方总接收最新的数据而不需要给发送方提供应答信息。这个通道提供 2 个数据缓冲区用于数据传送，这两个数据缓冲区中相互交替存放发送方提供的最新数据。发送方通过设置标志来表明哪一个数据缓冲区中的数据是最新的。

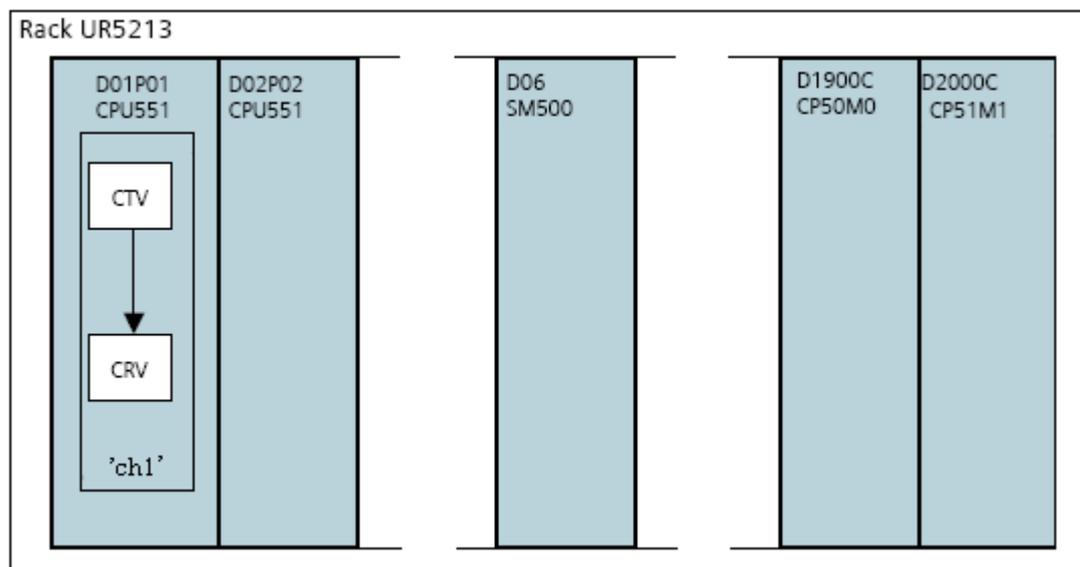
需要注意的是组态的接收方的时间任务周期一定要小于发送方的时间任务周期。



## 第四章 通讯组态

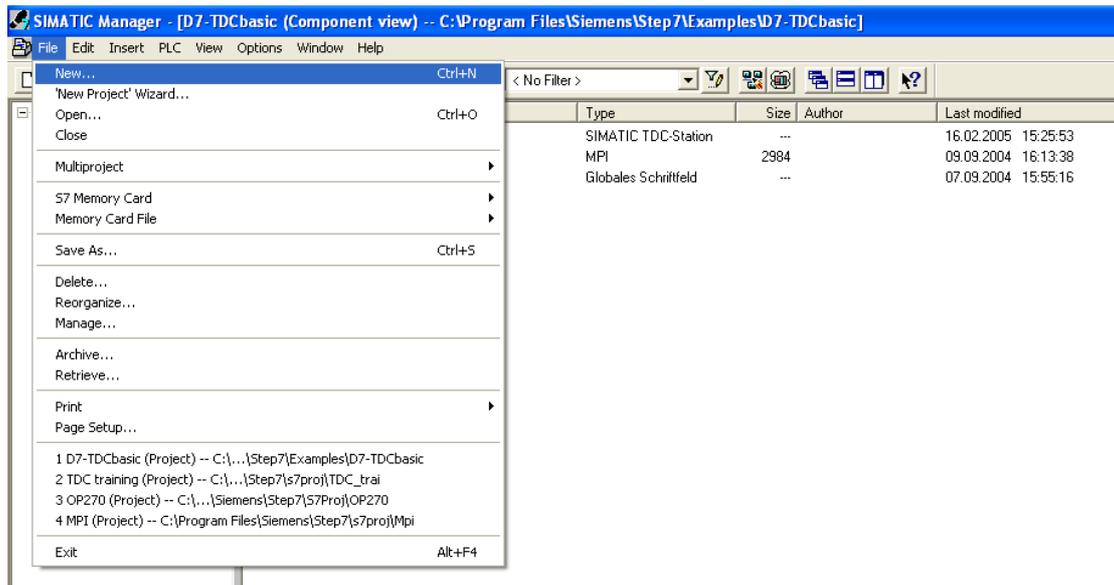
### 4.1 CPU 本地通讯

CPU 本地通讯不需要耦合模块，即数据缓冲区模块，例如 CP50MO，CP51M1。通过 @LOCAL 在本地 CPU 存储器上开辟数据通讯接口区，该接口区大小为 1Mbyte。过程数据通讯通过 CRV/CTV 功能块。主要功能用于 CPU 内部测试通讯的发送和接收，以及通过定义接口定制任务。

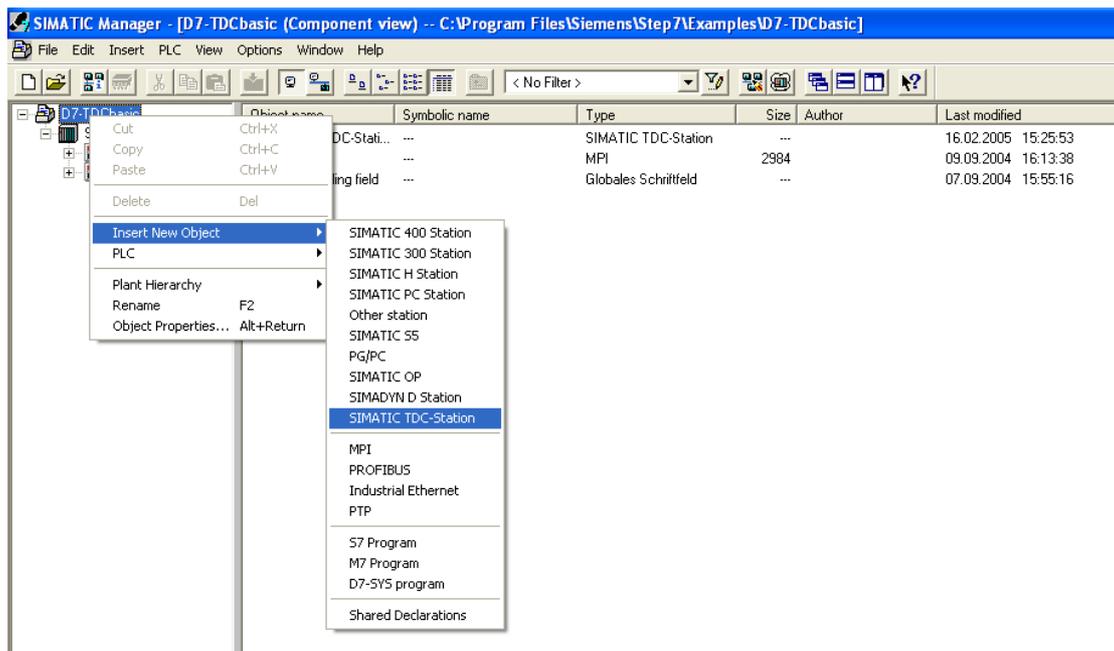


#### 4.1.1 在 SIMATIC Manager 中进行硬件组态

- 新建一个 TDC 项目。



- 插入一个 TDC 站。

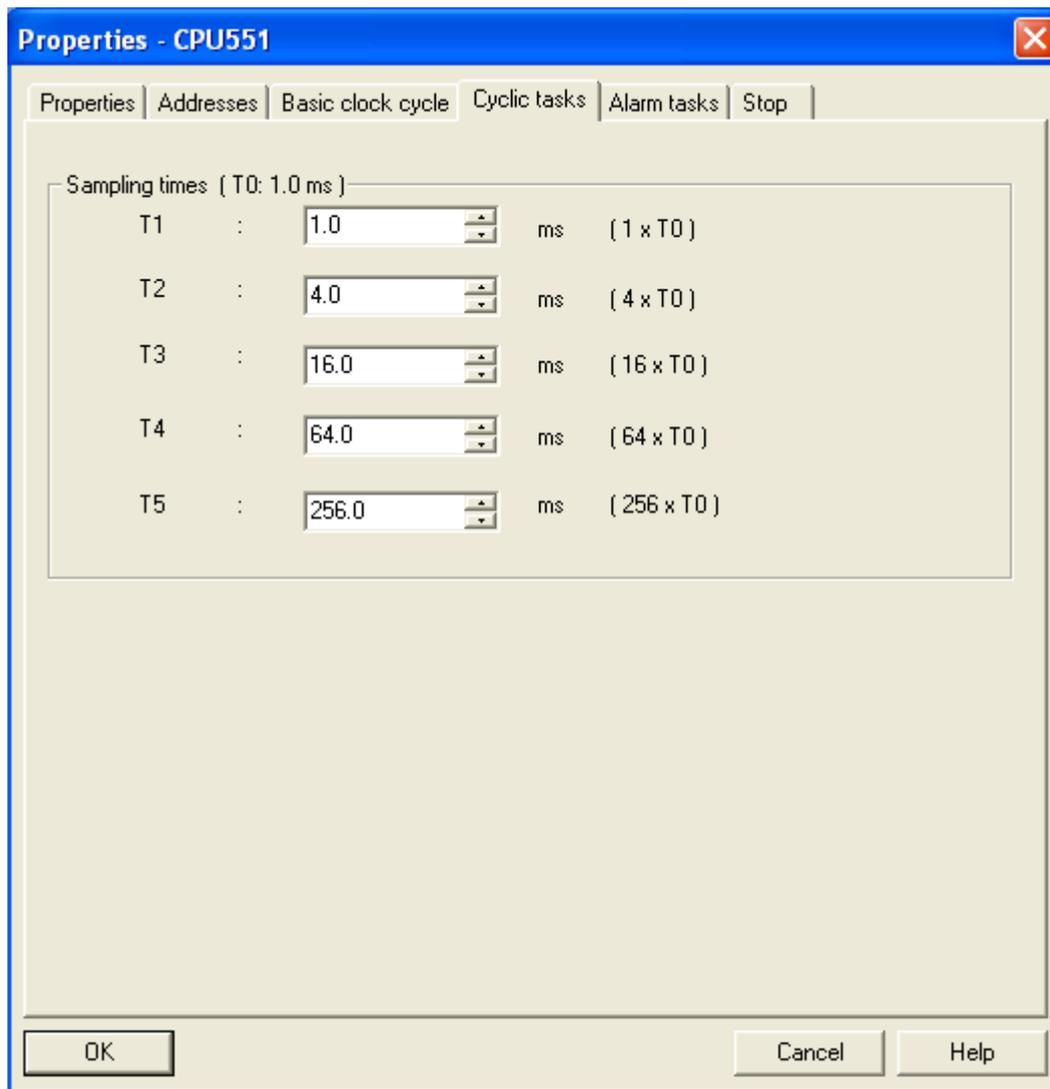


- 选择组态的硬件图标，双击打开。在 HW Config 中按照实际的机架组态所需的模板进行组态。

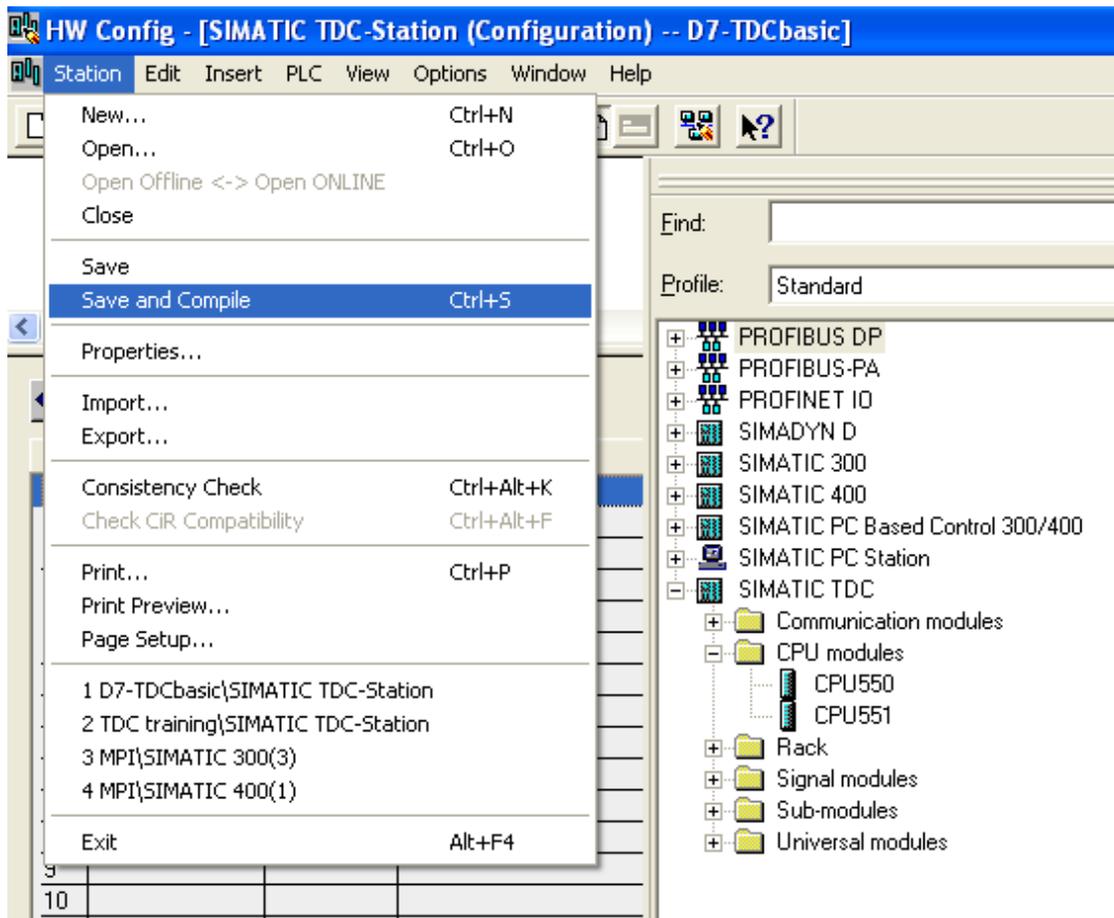
Slot	Name	Type	Order number
<b>1</b>	<b>D01P01</b>	<b>CPU551</b>	<b>6DD1600-0BA1</b>
1.1	D01_1	MC521	6DD1610-0AH3
1.2			
<b>2</b>	<b>D02P02</b>	<b>CPU551</b>	<b>6DD1600-0BA1</b>
2.1	D02_1	MC521	6DD1610-0AH3
2.2			
3			
4			
5			
6	D06	SM500	6DD1640-0AH0
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
<b>19</b>	<b>D1900C</b>	<b>CP50M0</b>	<b>6DD1661-0AD0</b>
X1	D19_1	MPI/DP	
X2	D19_2	MPI/DP	
<b>20</b>	<b>D2000C</b>	<b>CP51M1</b>	<b>6DD1661-0AE1</b>
X1	ind. Ethernet	ind. Ethernet	---
21			

6DD1610-0AH4  
Memory module: 4 Mbyte user program memory, 8 kbyte change memory

- 保持 CPU 等其他模板的默认属性

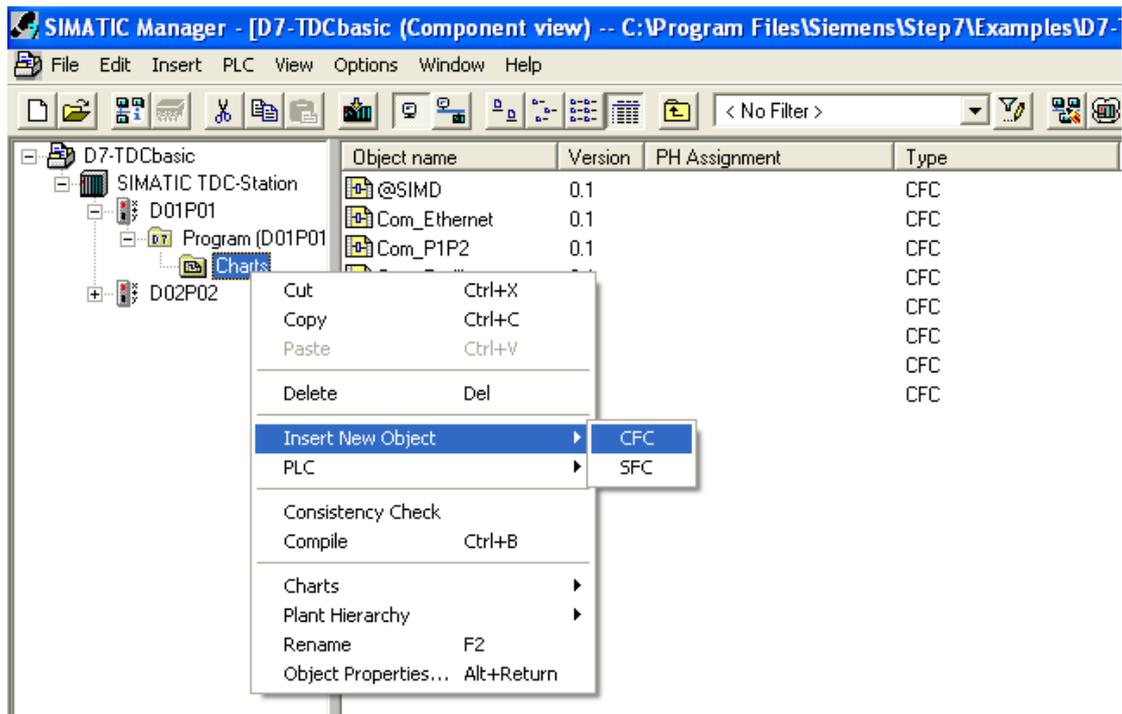


- 保存和编译 CPU 的硬件组态

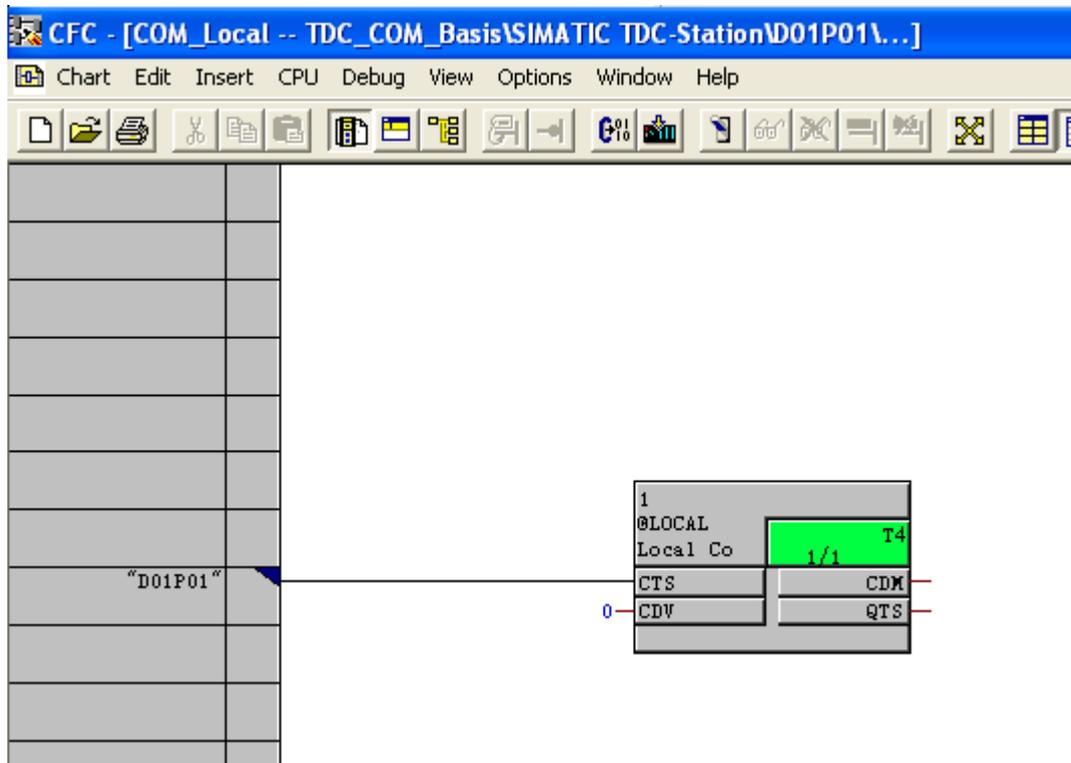


#### 4.1.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，实现 CPU-Local 通讯，右键点击 CPU1 的 Charts。插入新的 CFC 程序，命名为 Com\_Local，并通过双击打开。



- 插入@LOCAL 功能块到表中，连接在硬件组态中的硬件模板地址 D01P01。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。



- 分别插入 CTV 和 CRV 功能模块，插入 NOP1 功能块实现数据交换。

CTV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D01P01

AT—设置通道名称为 ch1

MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!CH11

CRV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D01P01

AT—设置通道名称为 ch1

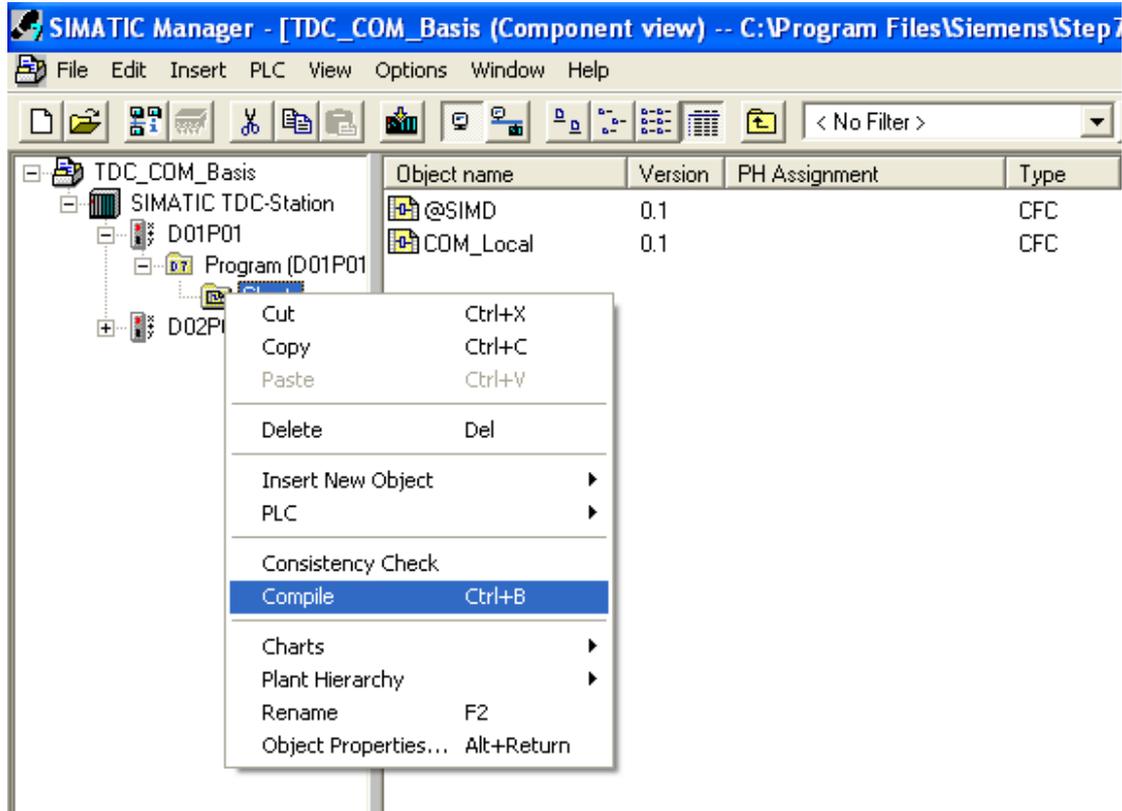
MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!CH12

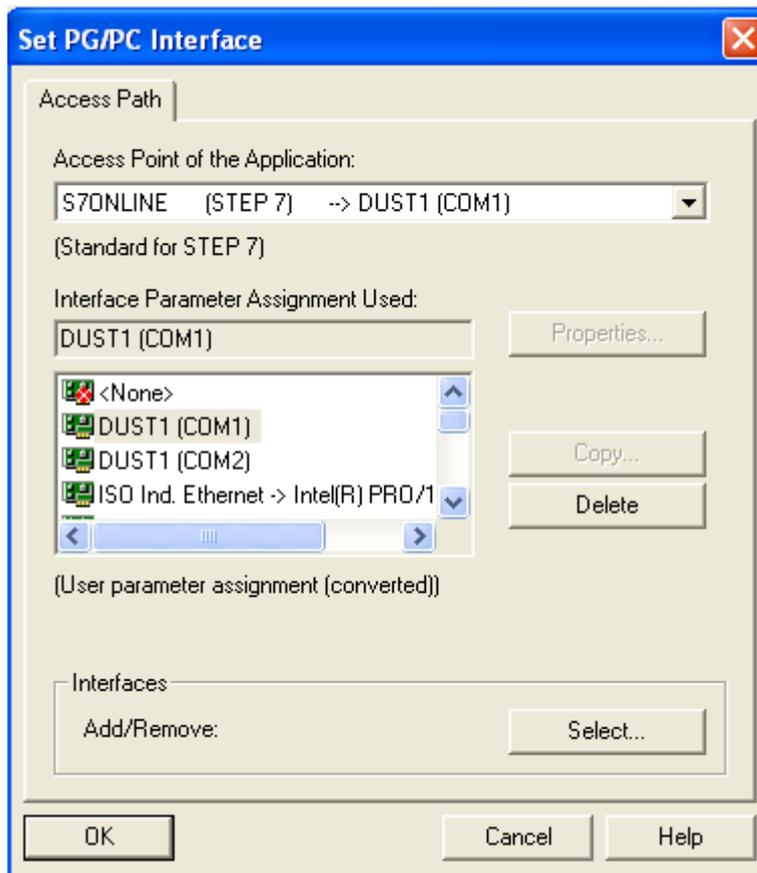
以上数据的发送和接收是定制的一个任务。



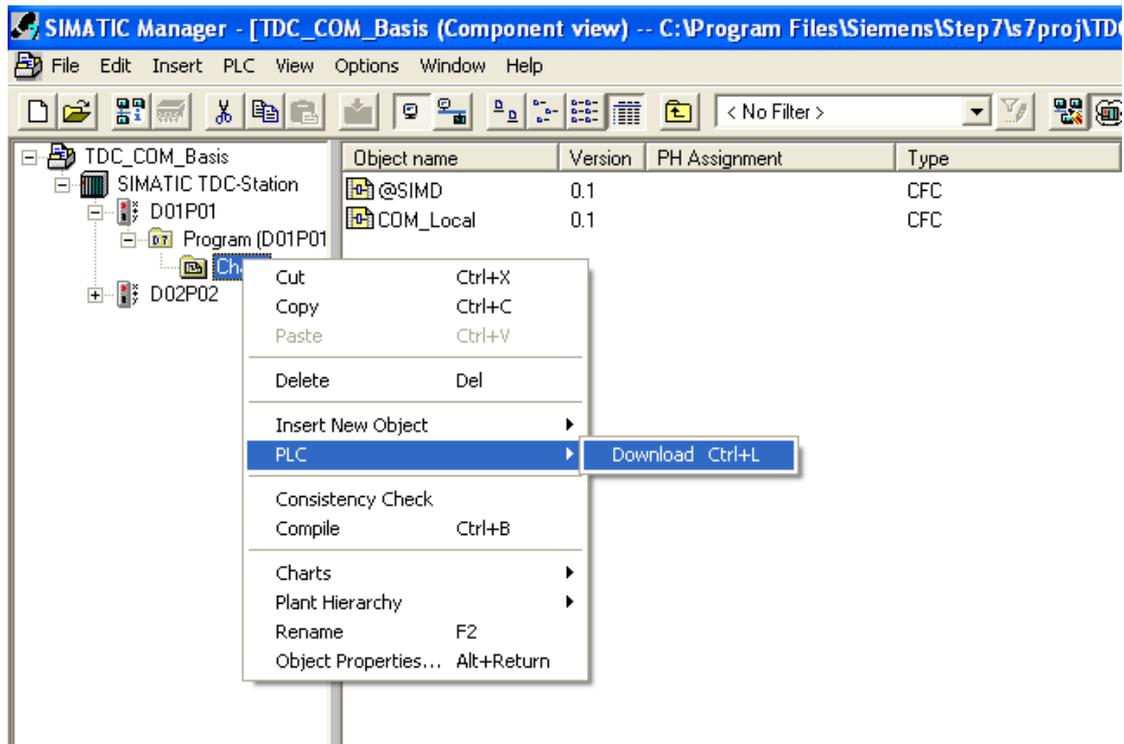
- 对所写的程序进行编译，以查证是否程序有错。



- 通过 SC67 电缆，连接于计算机的串口和 CPU551(slot1)的 X1 口。在控制面板中选择 Set PG/PC Interface，双击设置 PG/PC 接口，选择 DUST1(COM1)。



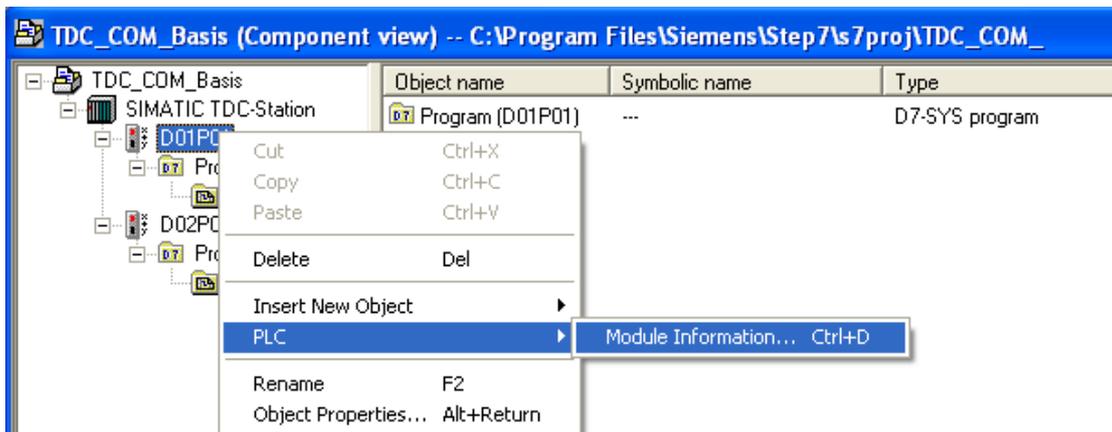
- 利用计算机的串口，通过 SC67 对程序进行下载。



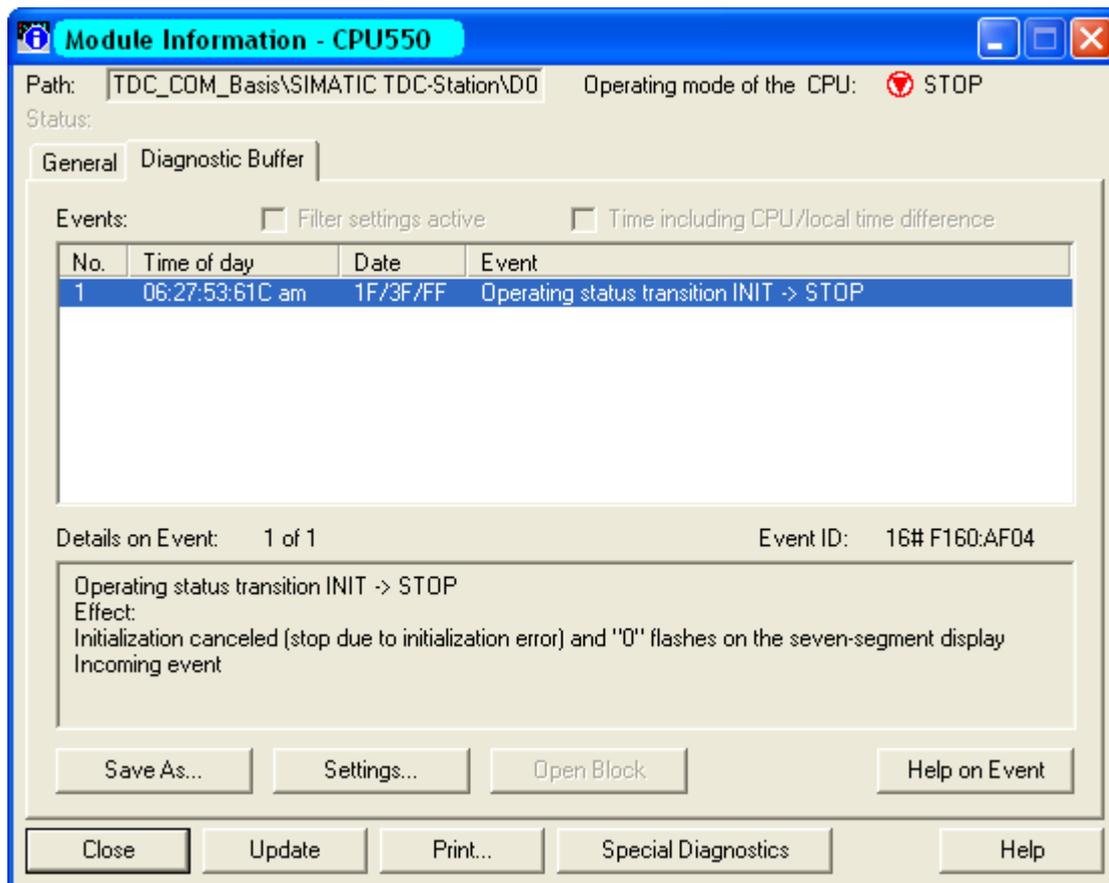
- 下载程序的设置，若是第一次下载系统组态参数和程序，通过串口电缆，并激活初始下载。



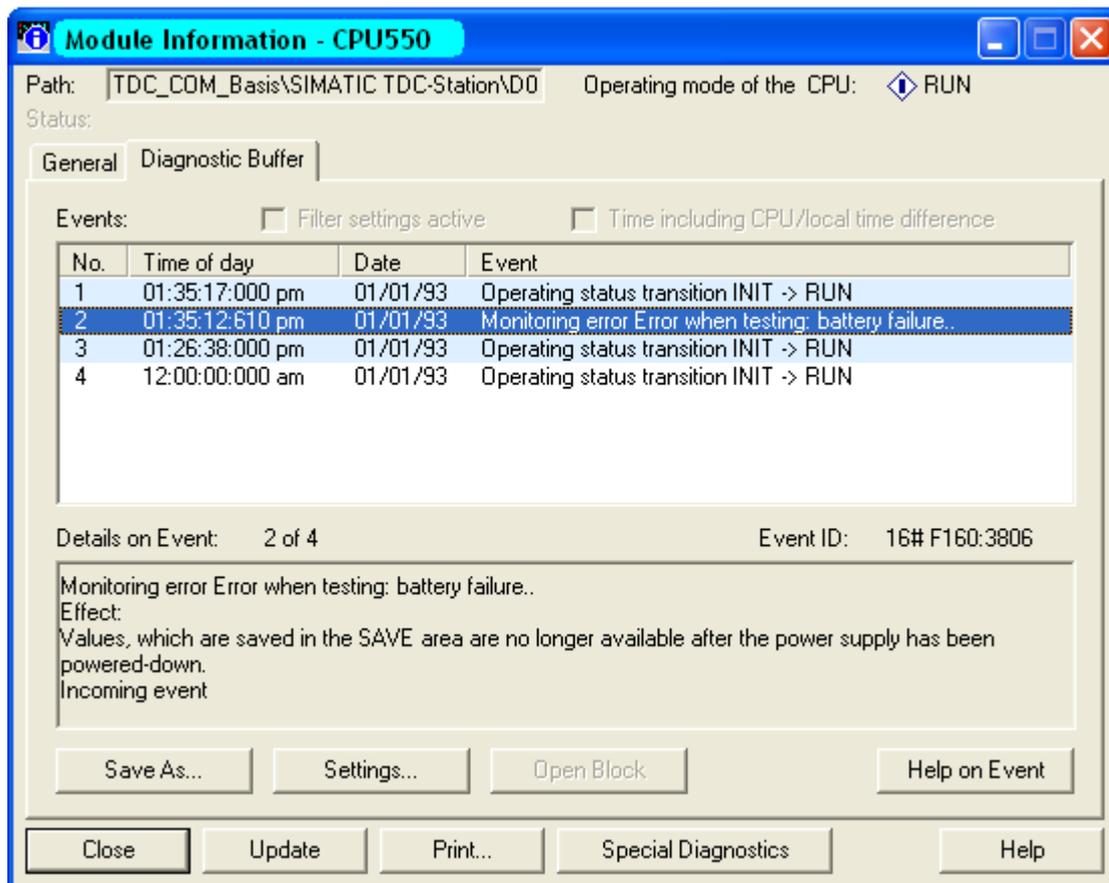
- 下载成功后，右键点击 D01P01，选择 PLC 中的 Module Information 来查看 TDC 的状态



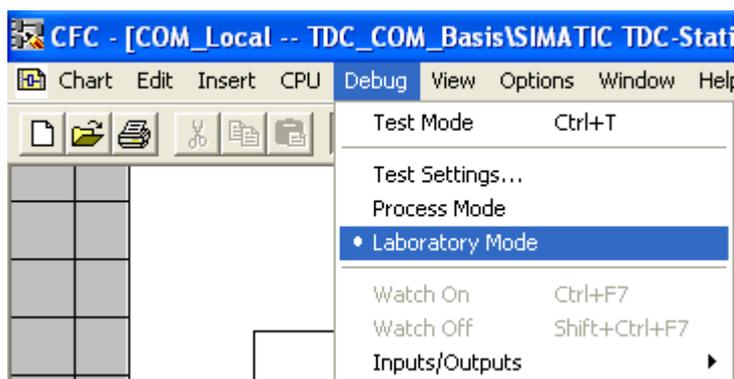
- TDC 中的 CPU 状态显示



- 需要注意的是对于新订购的 TDC 机架，是没有电池的，如果不安装电池，会出现监控故障，且在 CPU551（Slot1）上闪烁“b”，只需要安装两节 5 号额定电压为 1.5V 的电池即可。

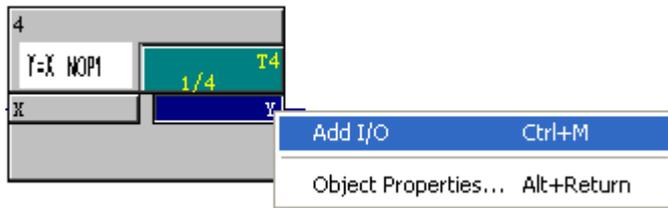


- 下载成功后，若系统参数组态和程序正确，此时 TDC 的 CPU551（Slot1）和 CPU551(Slot2)上分别显示为“1”和“2”。在 Com\_Local 的 CFC 程序中，选择 Debug 方式为 Laboratory Mode。

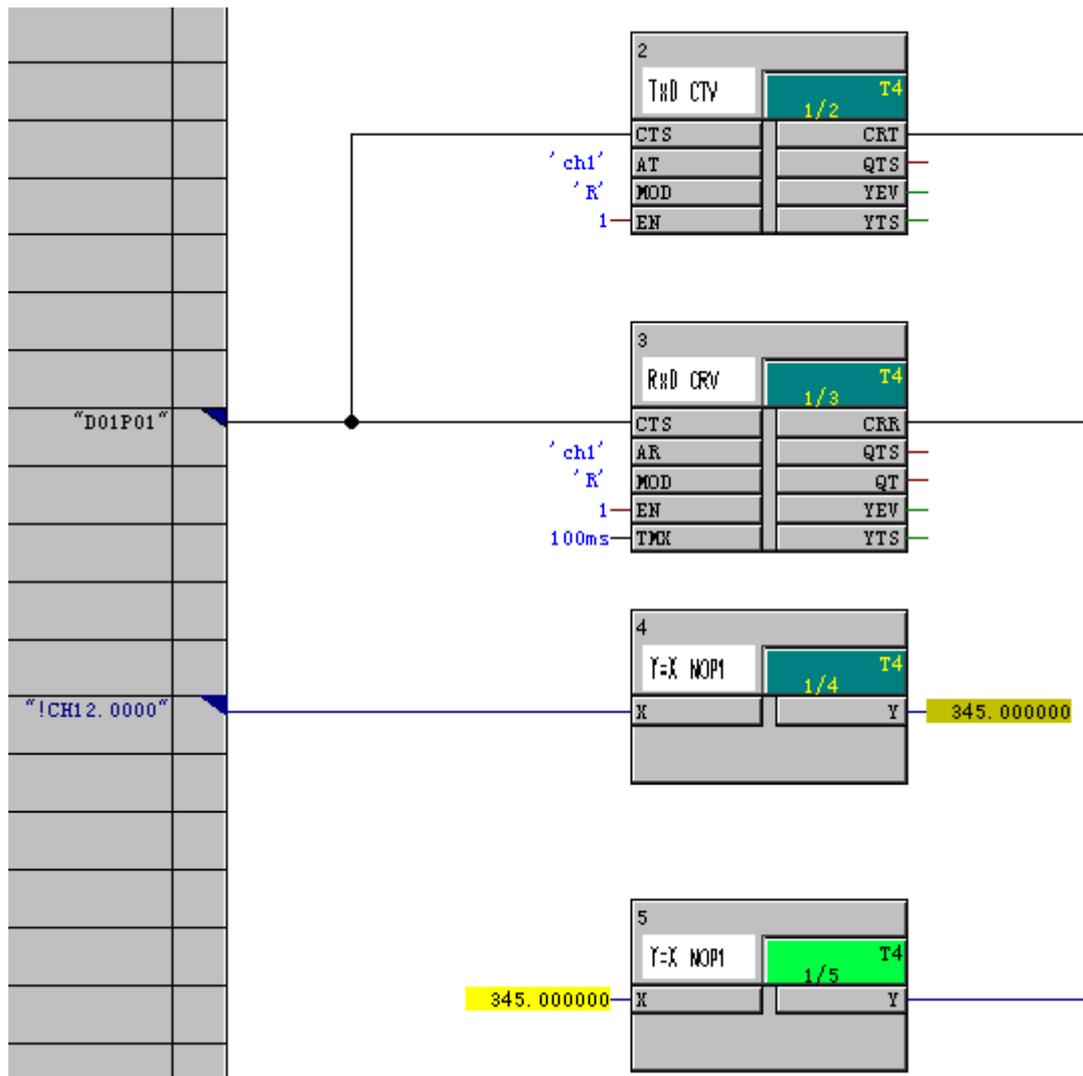


- 然后在工具栏中点击  图标，对程序在线观测。对所要观察的数据所在的数据块中点击右键，选择 Add I/O。即可在线观察数据。对于功能块的输入数据可以通过选

择 Object Properties... 来选择。

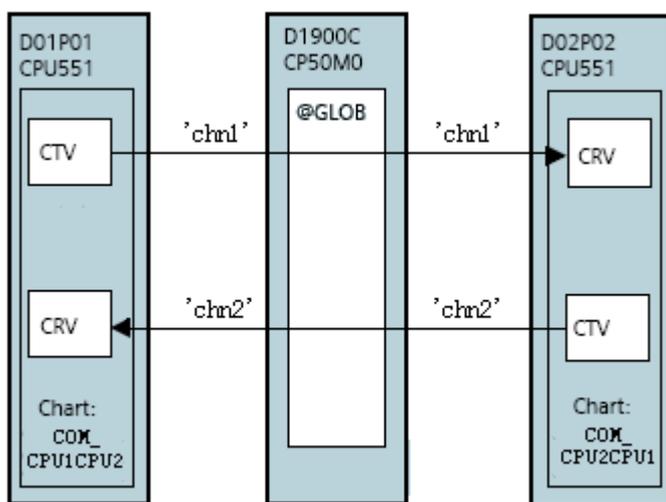


- 在本例中实现通道 ch1 的发送和接收数据，最终的结果见下图。



## 4.2 CPU-CPU 之间通讯

同一机架上的 CPU 与 CPU 之间的通讯，必须组态存储器模板 CP50MO 或 CP51M1。这样在存储器模板上可以提供 1Mbyte 的存储缓冲区。CPU 和 CPU 通讯，除了可以使用 \$-signals(在第三章介绍)来完成通讯；也可以使用@GLOB 功能块来实现 CPU 之间的通讯。@GLOB 功能块可以组态在任意 CPU 中。CPU-CPU 之间的通讯通过使用 CTV 和 CRV 功能块可以用于 CPU 之间大量数据的交换；也可以用于自定义接口的定制任务。

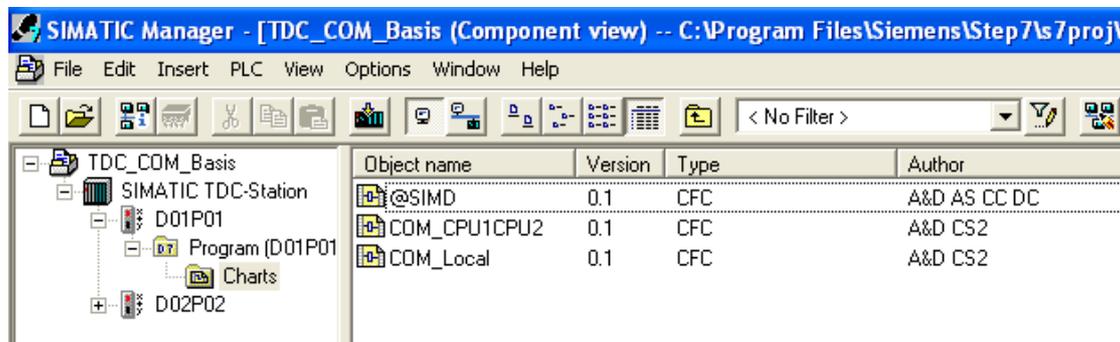


### 4.2.1 在 SIMATIC Manager 中进行硬件组态

- 硬件组态与 4.1 章的硬件组态相同。

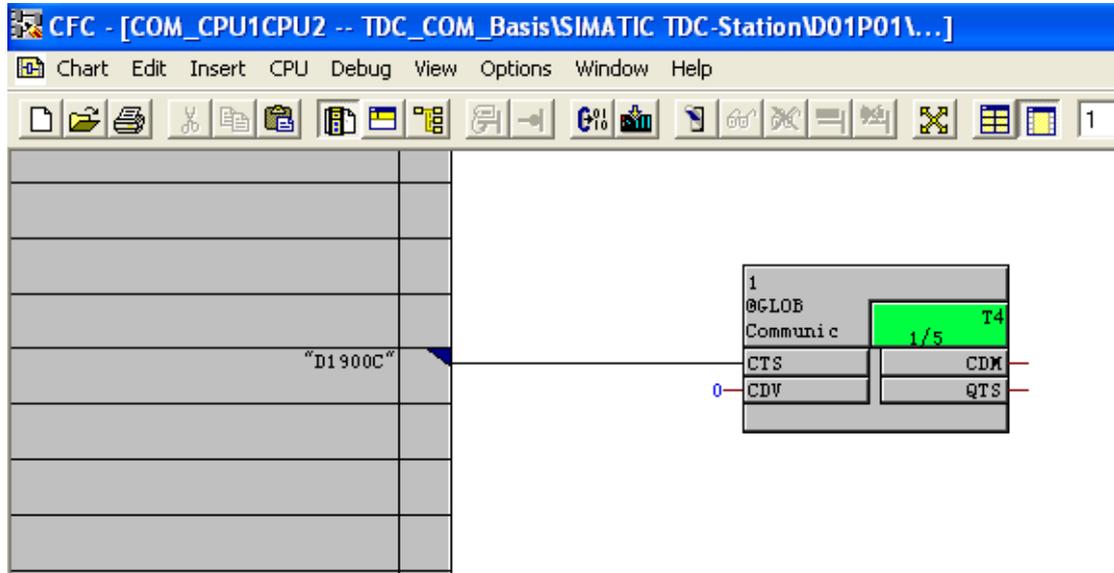
### 4.2.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 Com\_CPU1CPU2，并通过双击打开。



- 插入@GLOB 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C。即在

CP50MO 模板创建一个存储缓冲区。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。



- 分别插入 CTV 和 CRV 功能模块，插入 NOP1 功能块实现数据实数交换。

CTV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D1900C

AT—设置通道名称为 chn1

MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!CHN1

CRV 功能块的引脚说明：

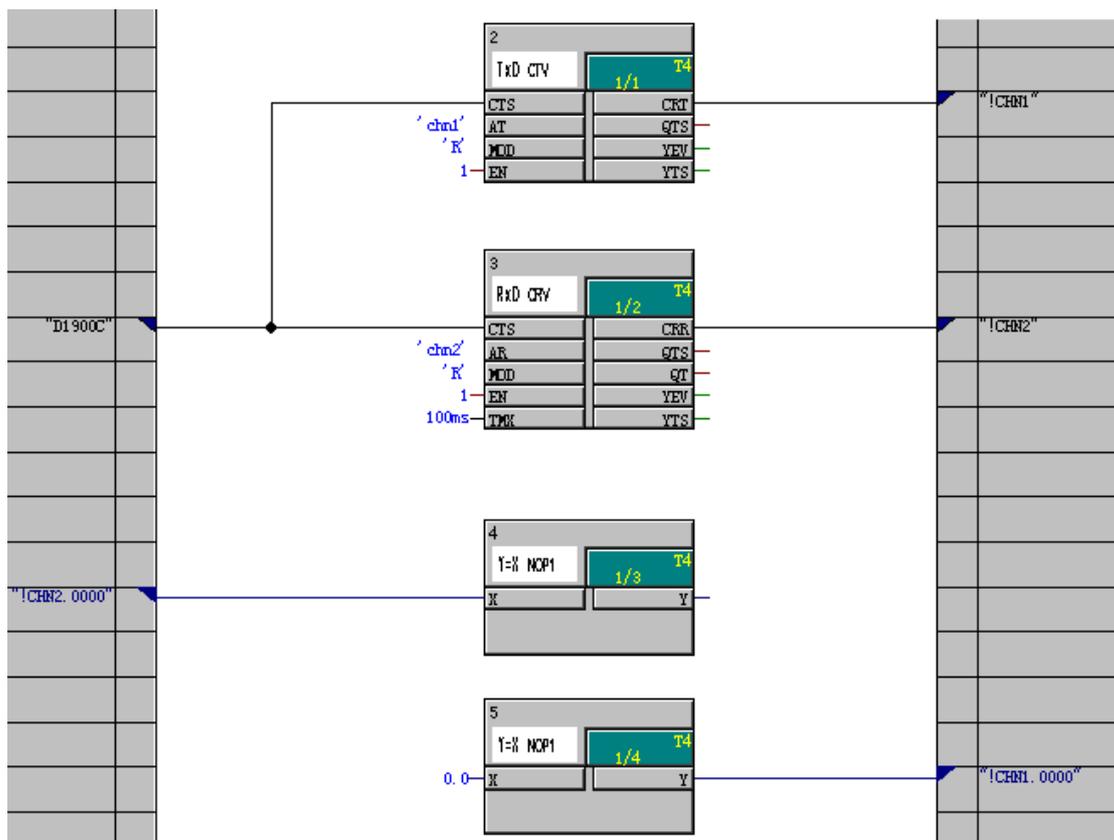
CTS—连接在硬件组态中的硬件模板地址 D1900C

AT—设置通道名称为 chn2

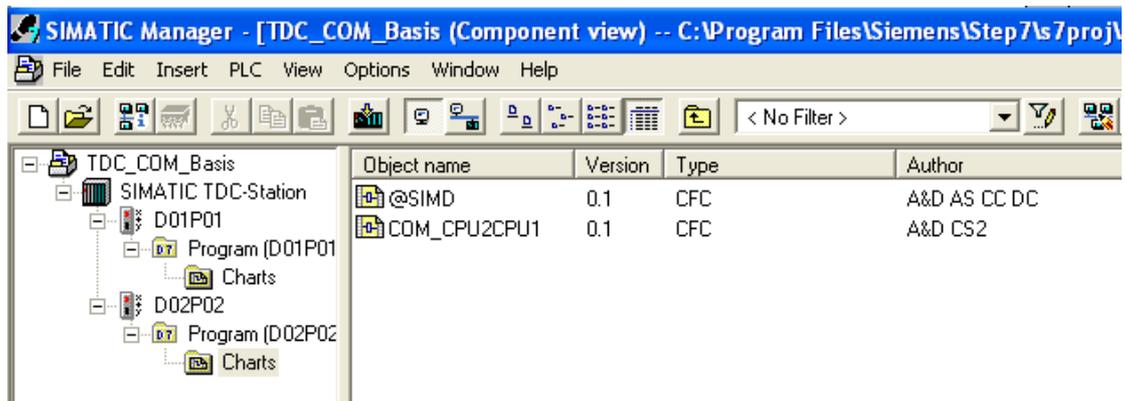
MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!CHN2

以上数据的发送和接收是定制的第一个任务。



- 在 CPU2:D02P02 中，在插入新的 CFC 程序，命名为 Com\_CPU2CPU1，并通过双击打开。



- 分别插入 CTV 和 CRV 功能模块，插入 NOP1 功能块实现数据交换。

CTV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D1900C

AT—设置通道名称为 chn2

MOD—定义为 R, Refresh 模式

CRT—定义虚拟连接名称为!CHN11

CRV 功能块的引脚说明:

CTS—连接在硬件组态中的硬件模板地址 D1900C

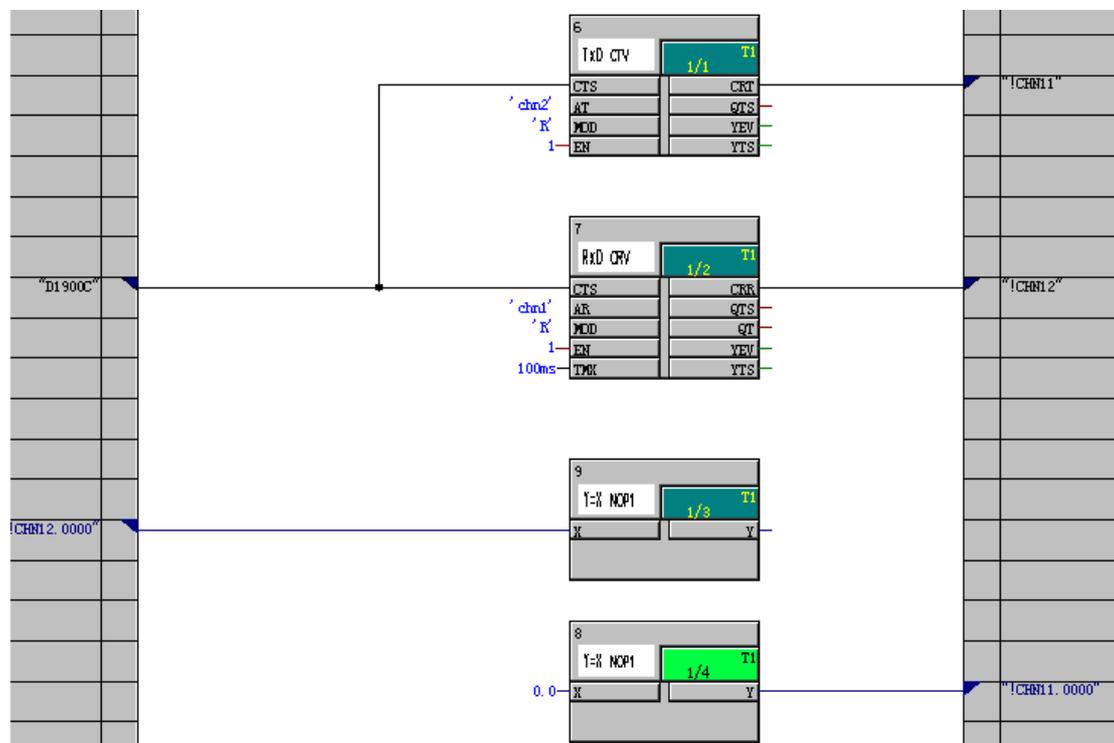
AT—设置通道名称为 chn1

MOD—定义为 R, Refresh 模式

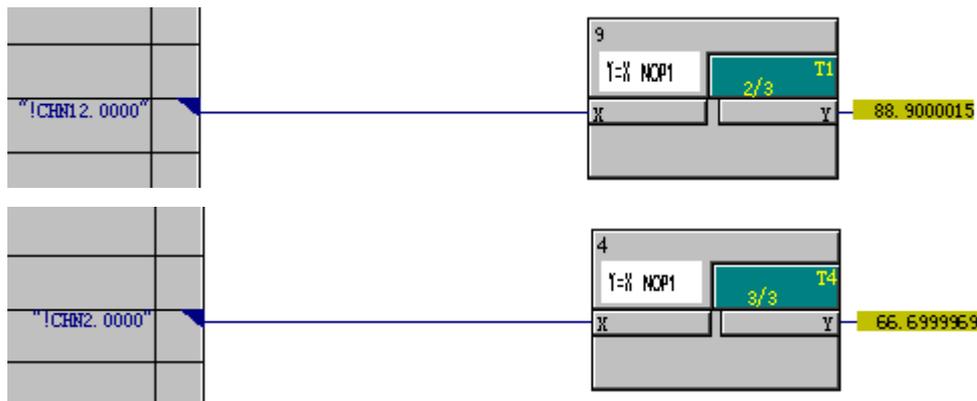
CRT—定义虚拟连接名称为!CHN12

以上数据的发送和接收是定制的第二任务。

注意: 第一个任务与第二个任务之间的发送与接收以及接收与发送的通道名称一定相同。



- 编译和下载程序方法参照 4.1 章通讯例子。
- 通过 SC67 分别插在 CPU551 (Slot1) 的 X1 和 CPU551 (Slot2) 的 X1, 来在线观察数据。对于 chn1 通道发送端发送数据 88.9。对于 chn2 通道发送数据 66.7。产看结果。



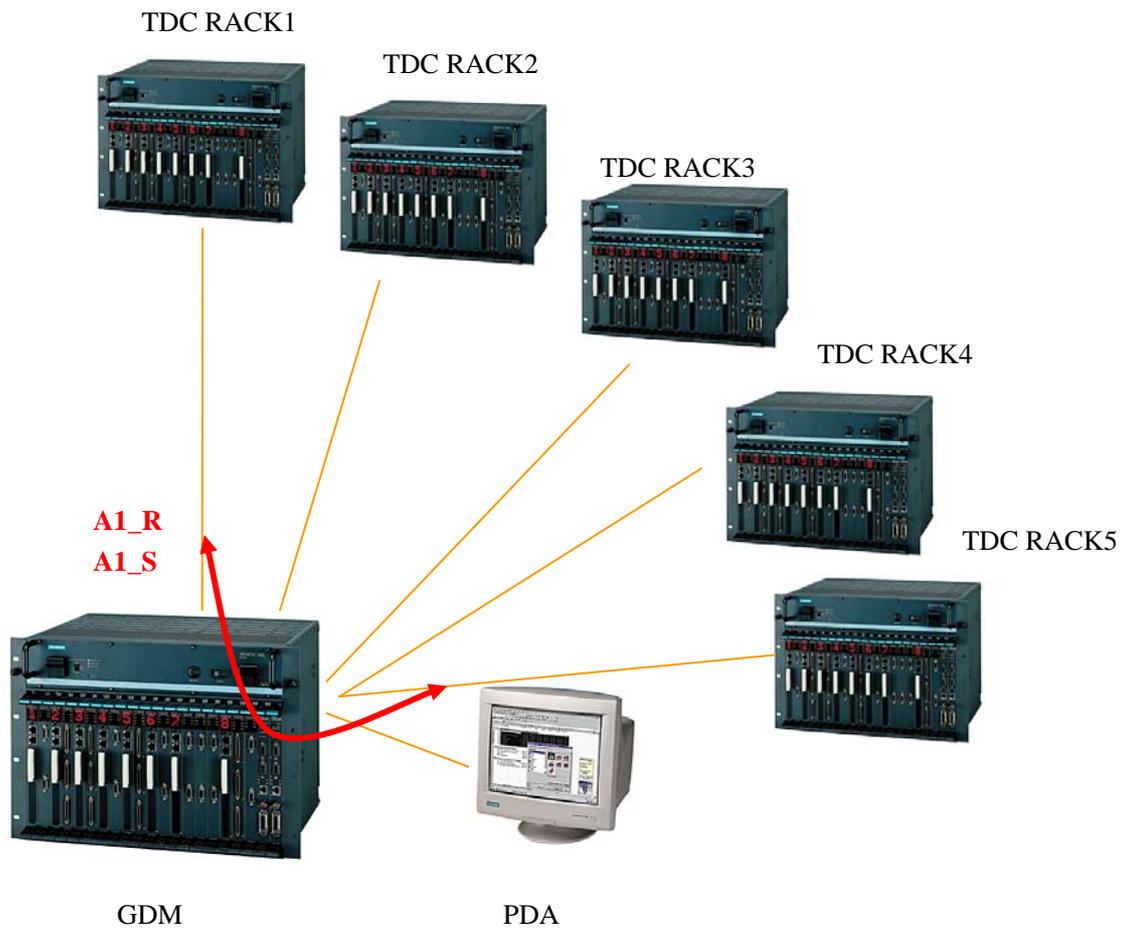
### 4.3 Rack-Rack 之间通讯

机架之间的数据交换可以通过GDM。高达44个机架可以通过中央存储器GDM实现耦合同步。这样系统可以扩展到最大，意味着有836CPU可以使用。GDM组态也需要UR5213来作为专用机架，CP52M0需要组态到该机架的第一号槽，CP52IO组态到第二槽到第十二槽的其余槽位。该硬件组态不需要在Step7中组态。

其余TDC机架，在硬件中需要组态CP52A0，该模板需要通过玻璃光纤与CP52IO相连。组成星型拓扑结构。光纤连接最大是200m。

CP52M0提供高达2Mbyte的数据通信缓冲区。

通过在每一个机架上组态@SRACK，来实现不同机架间的数据交换。所有机架可以被基本同步，可以一致的系统时钟，可以在任意时刻连接和断开一个TDC机架。



#### 4.3.1 在 SIMATIC Manager 中进行硬件组态

- 在各个 TDC 的硬件组态中插入 CP52A0 模板。



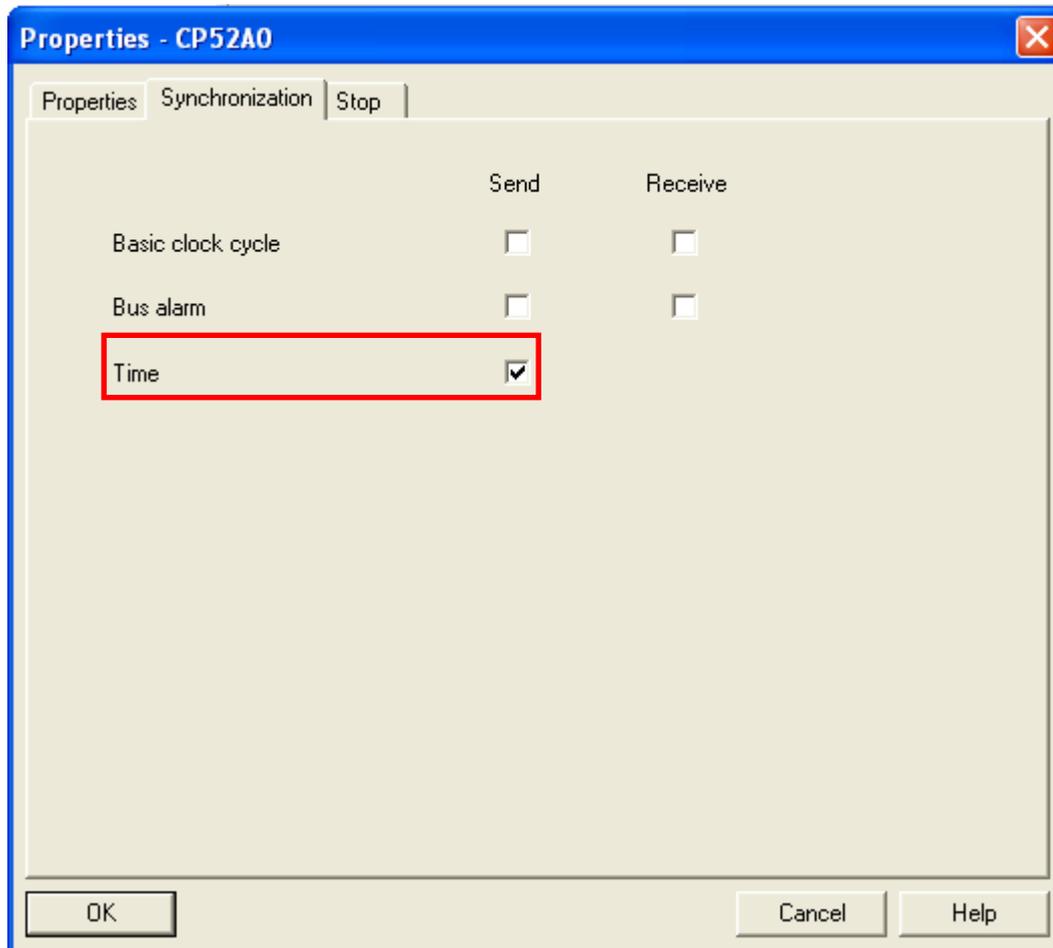
The image shows a software dialog box titled "Properties - UR5213". It contains the following fields and values:

- Rack Type:** UR5213  
Rack, 21 slots, 32-bit bus, fan, 115/230 V AC
- Order No.:** 6DD1682-0CH0
- Name:** A001 (highlighted in blue)
- Rack no.:** (empty dropdown menu)
- Comment:** (empty text area)

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

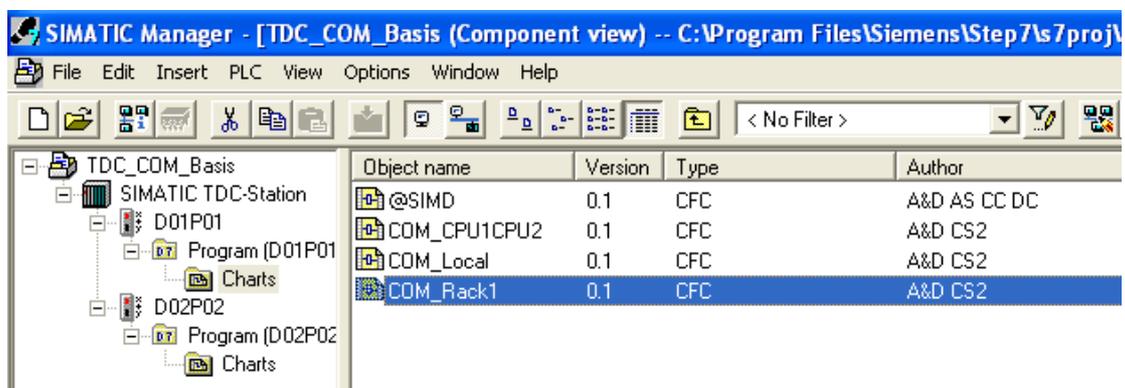
- 修改默认机架的名称为 A001

- 点击 TDC RACK1 硬件组态中 D1300B，显示组态画面。设置 TDC RACK1 为发送系统时间。这样各个机架间就同步的系统时间。对于基本时钟，总线中断请参考第三章相关内容。TDC RACK2 等其它机架就不必选中 Time 这个选项。

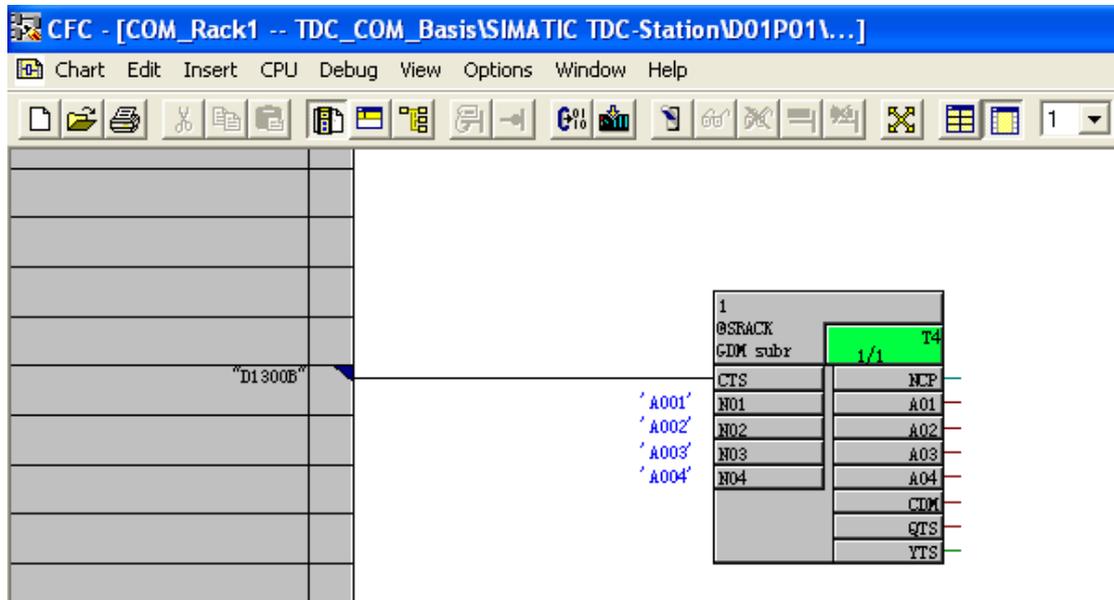


#### 4.3.2 在 SIMATIC Manager 中进行软件组态

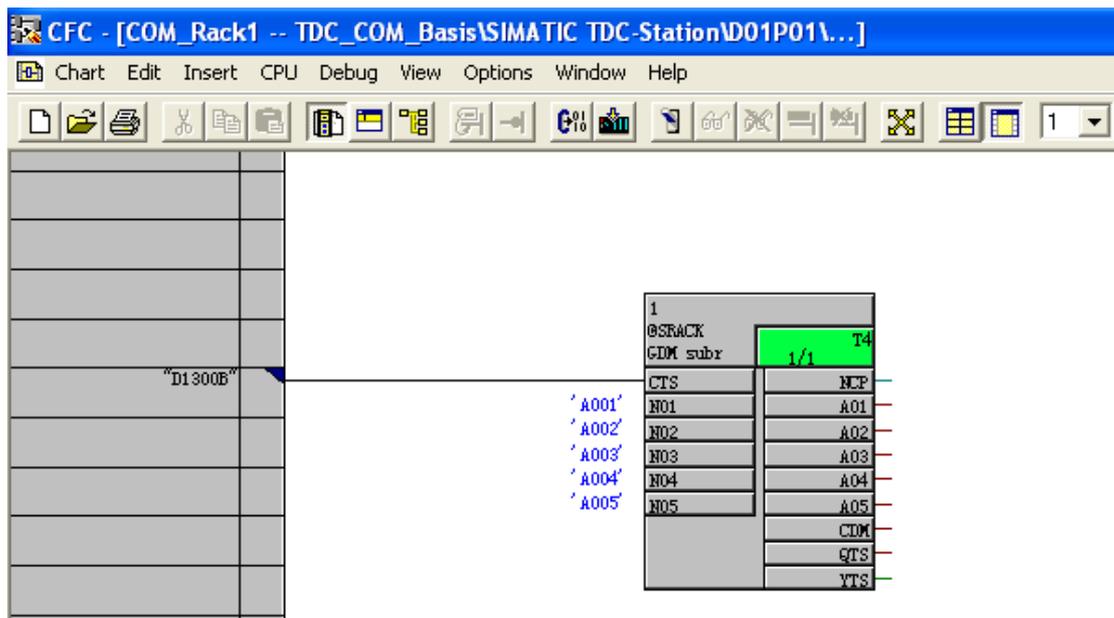
- 对于 TDC RACK1，新建 CFC，并命名为 COM\_Rack1，并双击打开。

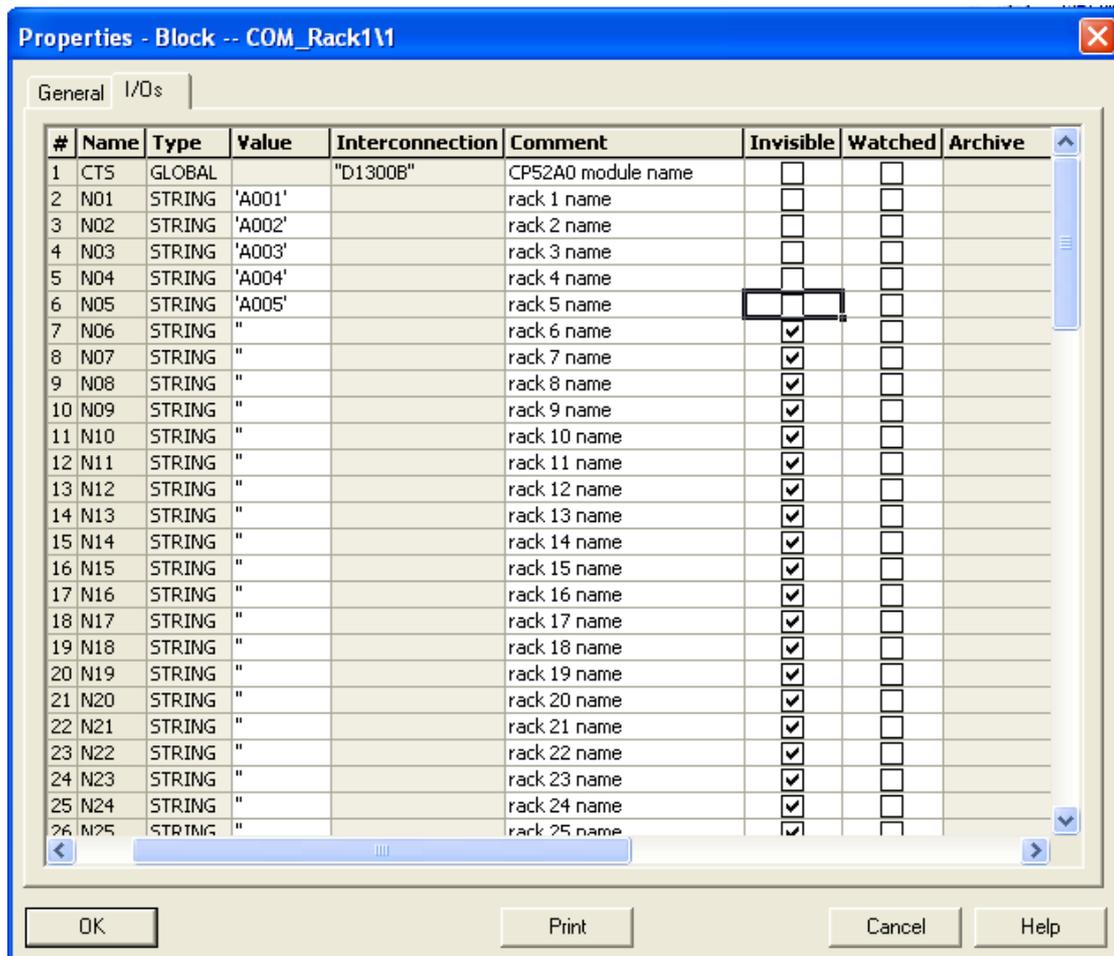


- 插入@SRACK 功能块到表中，连接在硬件组态中的硬件模板地址 D1300B。即利用 D1300B 模板作为接口通讯。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。



- 双击@SRACK 功能块，弹出该功能块的属性，选择 I/Os 界面。N01-N44 表示该机架的连接与之通讯其他机架的名称，在默认状态下 N05~N44 是默认不见的。A01~A44 表示对应的机架是否激活的状态。通过取消 N05, A05 的标记√，即可在功能块中显示。





- 分别插入 CTV 和 CRV 功能模块，插入 NOP1\_I 功能块实现与其它机架数据整数交换。

CTV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D1300B

AT—设置通道名称为 A1\_S

MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!A1S

CRV 功能块的引脚说明：

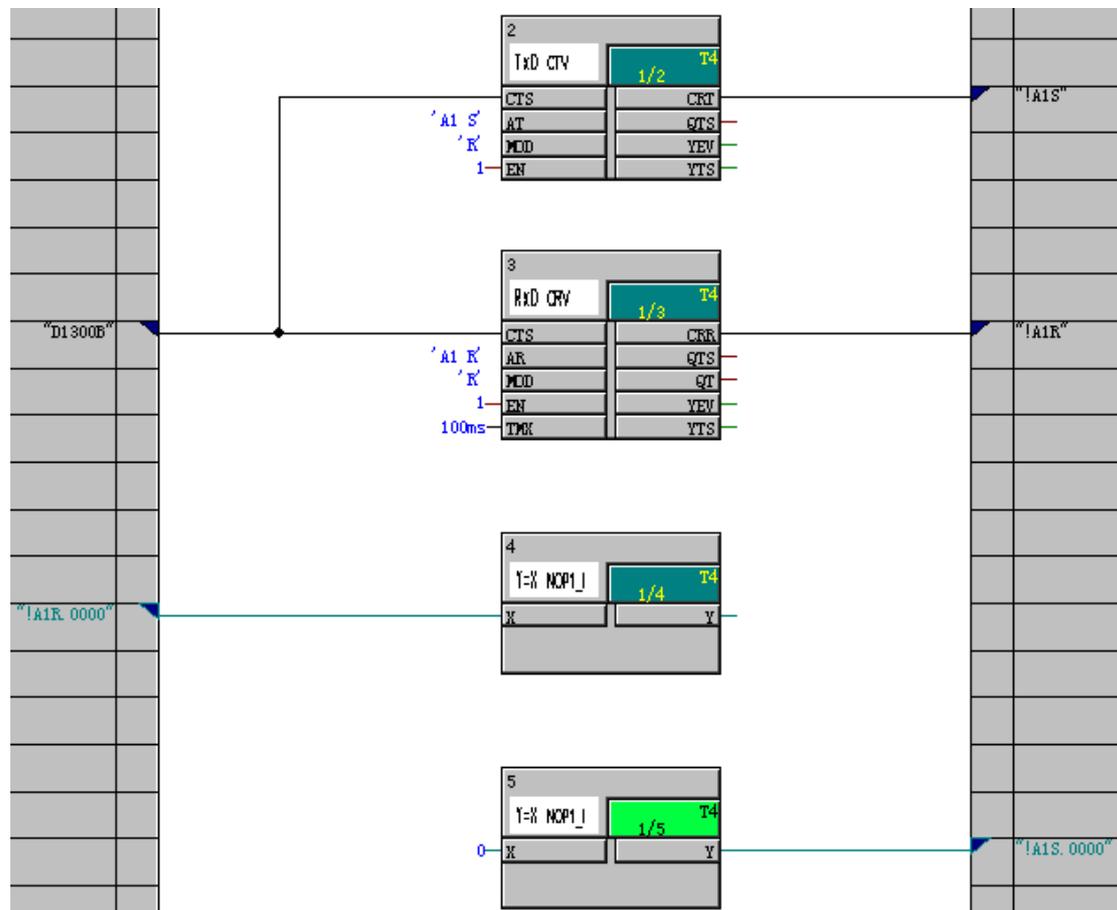
CTS—连接在硬件组态中的硬件模板地址 D1300B

AT—设置通道名称为 A1\_R

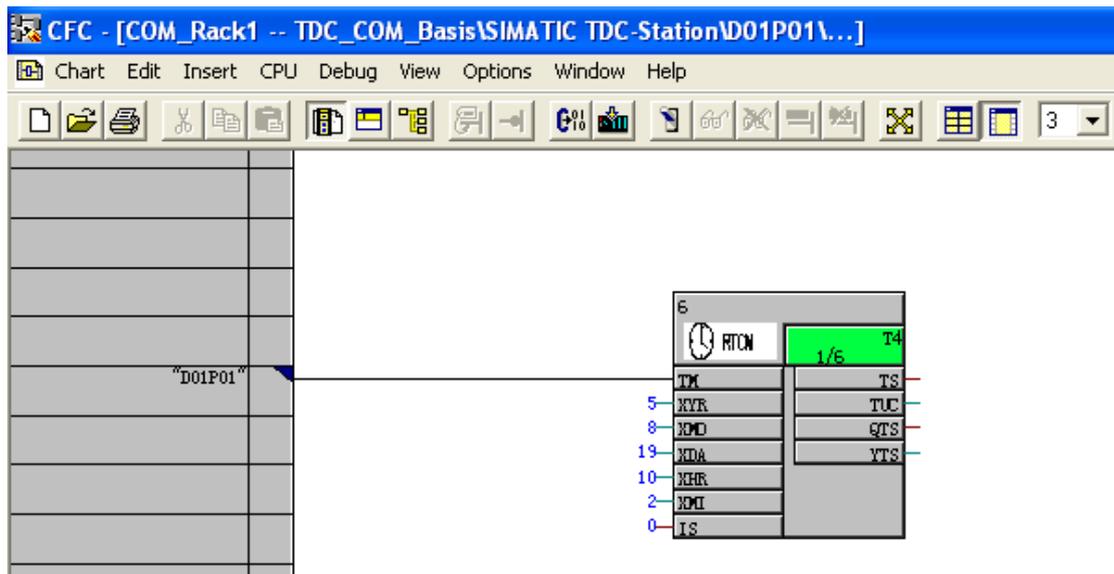
MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!A1R

以上数据的发送和接收是定制的一个任务。



- 在 TDC RACK1 中插入 RTCM 功能块，TM 连接硬件组态中的 CPU 模板。通过给 IS 一个上升沿，那么 05 年 8 月 19 日 10 点 02 分就设置给 D01P01 了。选择机架槽中最靠左侧的 CPU 作为时间源。



- 选择 TDC RACK5，注意需要修改默认机架的名称 A000 为 A005。新建 CFC，并命名为 COM\_Rack5，并双击打开。像 TDC RACK1 一样，在 TDC RACK5 中插入 @SRACK 功能块到表中，连接在硬件组态中的硬件模板地址 D1300B。即利用 D1300B 模板作为接口通讯。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。
- 分别插入 CTV 和 CRV 功能模块，插入 NOP1\_I 功能块实现与其它机架数据整数交换。

CTV 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D1300B

AT—设置通道名称为 A1\_R

MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!A5S

CRV 功能块的引脚说明：

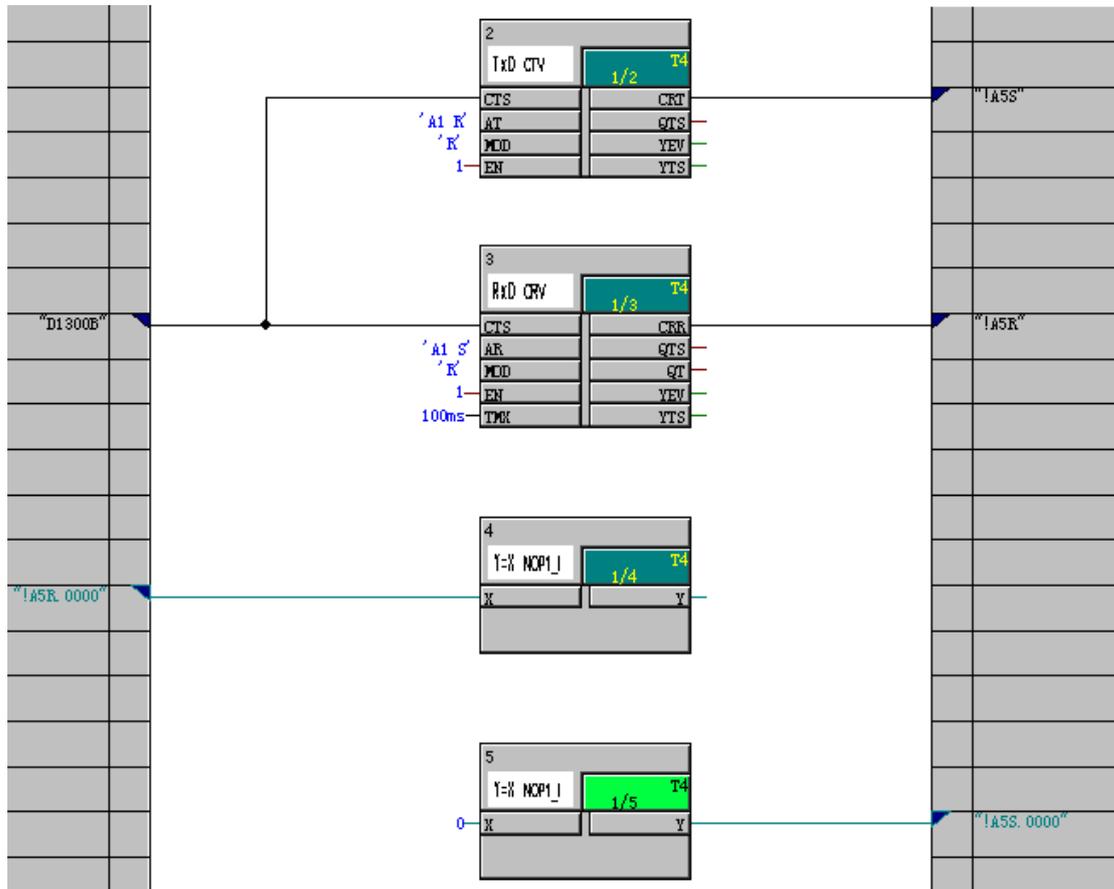
CTS—连接在硬件组态中的硬件模板地址 D1300B

AT—设置通道名称为 A1\_S

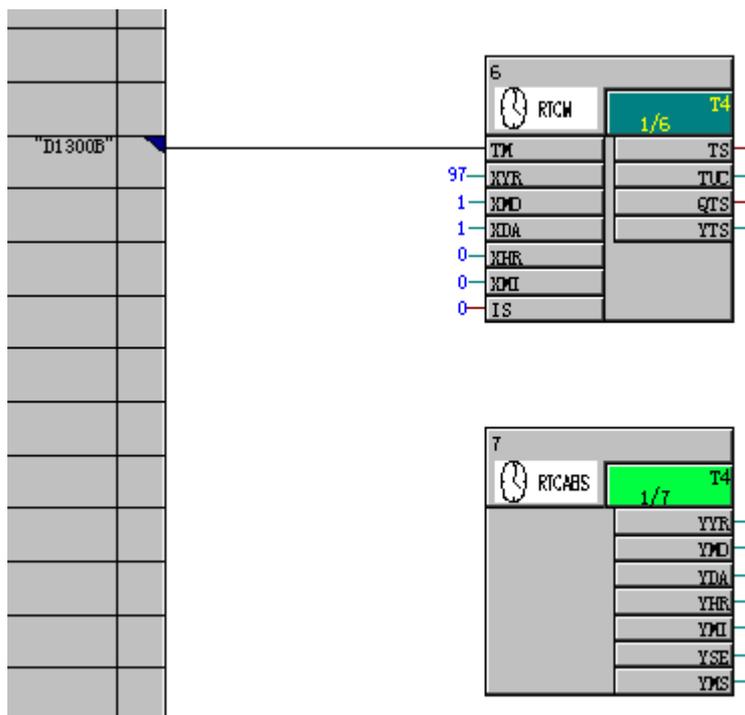
MOD—定义为 R，Refresh 模式

CRT—定义虚拟连接名称为!A5R

以上数据的发送和接收是定制的一个任务。



- 在 TDC RACK5 中，插入 RTCM 功能块，连接 TM 到 D1300B，意味着该机架的系统时间由 CP52A0 读入。插入 RTCABS 功能块，可以从中读出系统时间。



- 分别编译下载程序，由于 TDC RACK1 和 TDC RACK5 之间的发送和接收的通道名称相同，即 A1\_S, A1\_R。所以 TDC RACK1 和 TDC RACK5 即可以实现数据交换。这样若和其它机架通讯只需要对应的输入输出通道名称相同。

#### 4.4 MPI 通讯

MPI 通讯是 SIMATIC 产品的标准通讯协议，采用令牌传输方式，最高 32 个网络节点。CP50MO 提供 2 个 MPI 接口，PG 可以用 MPI 接口，对 CFC 程序进行测试和诊断 TDC 状态。该 MPI 接口也可以与 WinCC 以及操作员面板 OP/TP 进行通讯。

一个 CP50MO 模板的 MPI 接口，对应一个功能块 @MPI 来组态对应，该功能块用于初始化和监控 MPI 通讯。不需要使用功能块 CTV 和 CRV 来发送和接收数据。

可以组态多个 SER 服务功能块与一个 CPU 关联，用于 MPI 通讯服务。通过与 MPI 接口的耦合，PG/PC 可以通过 MPI 接口下载和调试 CFC。SER 功能块还初始化和处理显示设备（WinCC/OP/TP）的通讯连接。当用于连接显示设备时，参数 NOS 根据连接的显示设备的数量来设置即必须  $\geq 1$ ，否则 WinCC 等设备无法建立连接，也就无法显示。

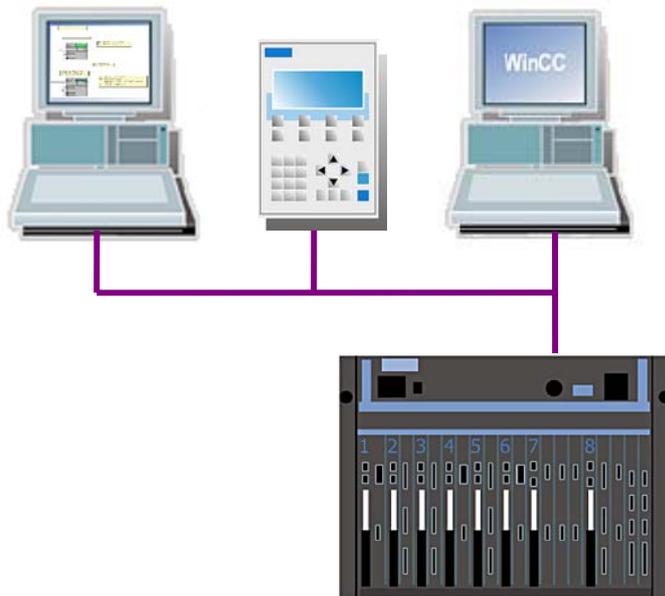
可以组态一个 S7OS 功能块用于 TDC 与显示设备通过 MPI 方式通讯（一个 CPU 需要一个 S7OS 功能块），用于处理和响应连接的 WinCC 或操作员面板 OP/TP。该功能块参数 NOS 默认为 1，应根据连接的显示设备的数量来设置。（可以设置为 0，通讯正常，但这时 CPU 会出现通讯故障，此时增大了 CPU 的通讯负荷）。

SER 功能块可以组态在任意循环时钟周期 T1~T5，但系统处理 SER 功能块的默认采样时间为 32ms。那么意味着该功能块会 32ms 来刷新在显示设备的数据。而 S7OS 的功能块可以组态在循环时钟周期 32ms~256ms，那么可以组态在采样周期长的循环时间中，这样会减少 CPU 的通讯负载。

这样 SER 功能块主要用于 PG 通过 MPI 网络连接和诊断 TDC；S7OS 功能块主要用于显示监控变量设备 WinCC 和 OP/TP 的连接。

使用 CP50MO 可以通过 MPI 网络仅能与 PG/PC，SIMATIC WinCC 和 SIMATIC OP/TP 通讯，不能和其他 SIMATIC 设备通讯。

##### 4.4.1 CP50MO 与 WinCC 通讯

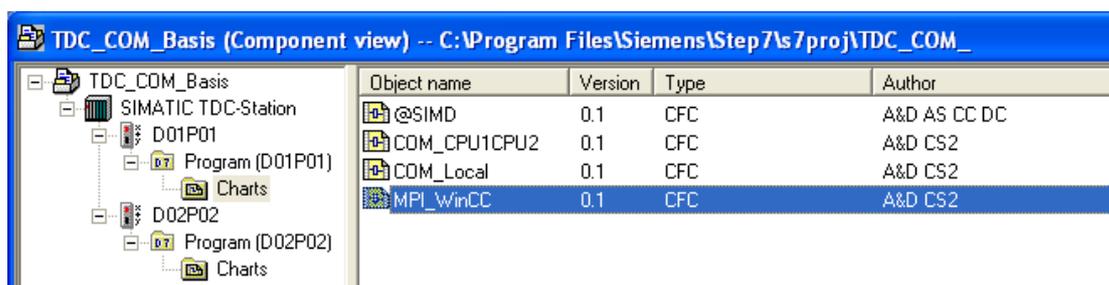


#### 4.4.1.1 在 SIMATIC Manager 中进行硬件组态

- 硬件组态与 4.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为 MPI 接口。MPI 默认的站地址为 2。
- 计算机 WinCC 操作员站使用标准的 MPI 电缆和 CP5511MPI 网卡。

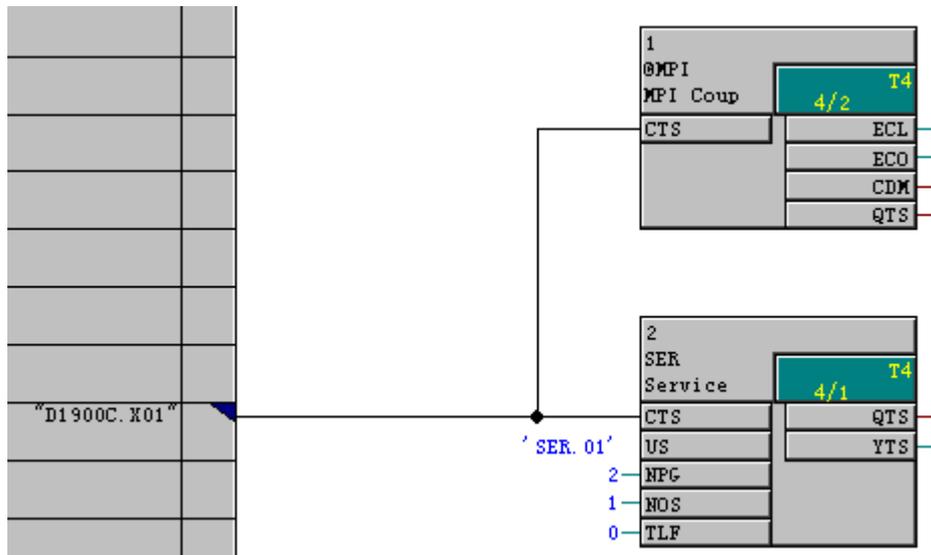
#### 4.4.1.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 MPI\_WinCC，并通过双击打开。



- 插入 @MPI 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C.X01。即利用 CP50MO 的 X1 作为与 WinCC 通讯的 MPI 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。插入 SER 功能块，可

以任意组态到任何一个时间周期。



SER 功能块的引脚说明：

CTS—连接硬件组态的模板地址为 D1900C.X01，即使用 CP50MO 的接口 X1 作为服务端口。

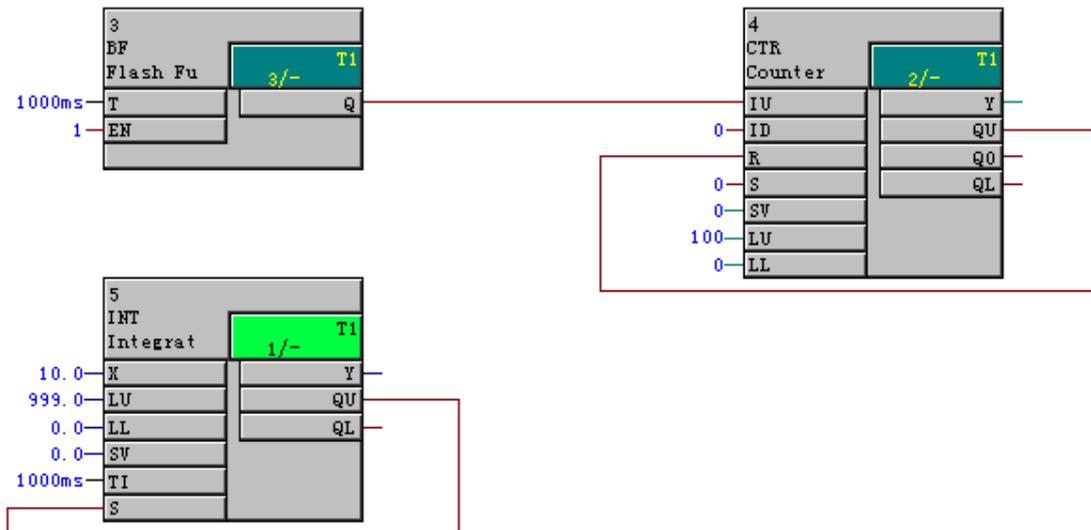
US—任意定义一个不大于 6 个字符长度的通道名，01 表示关联 CPU 的槽号地址。说明通过 SER 功能块，由 CPU1 来完成该接口通讯处理服务。

NPG—指定 PG 的个数，默认为 2。也就是可以通过 TDC 的 MPI 接口连接的 PG 的个数。即可以使用设定的 MPI 接口 X1 来下载和诊断 TDC。

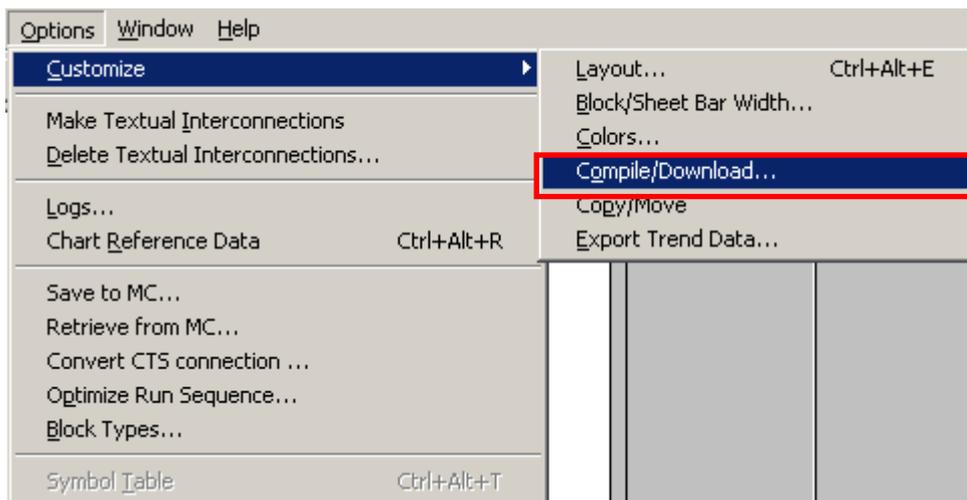
NOS—指定 OS 的个数，默认为 0。本例中为 1，也就是指定监控变量（C and M）的显示设备（例如 WinCC）的个数。即可以使用设定的 MPI 接口 X1 来实现 MPI 网络通讯。

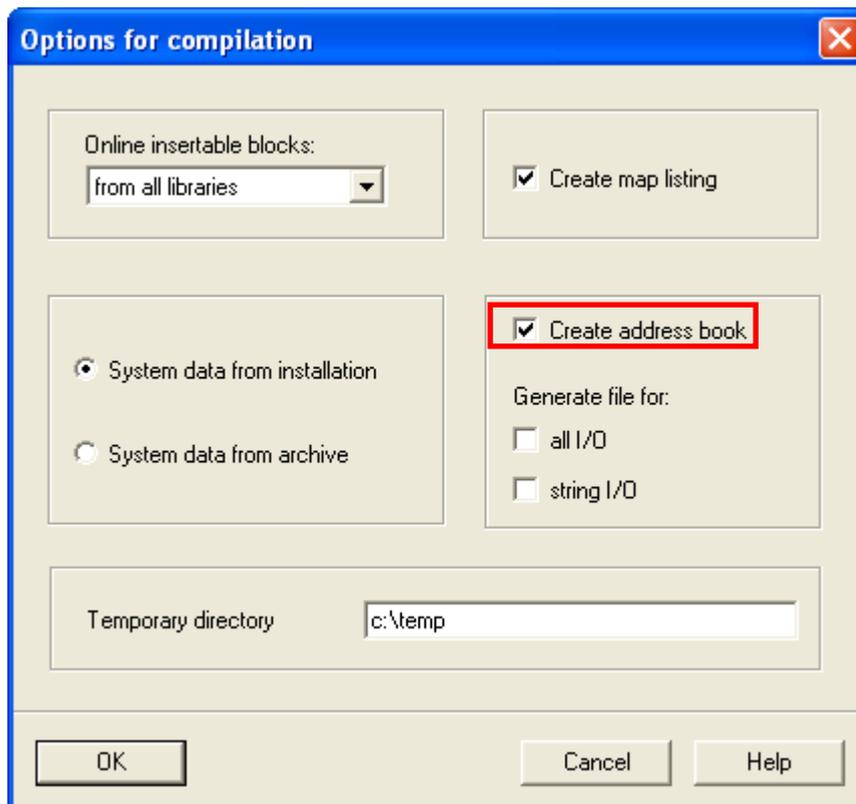
需要注意的是如果想通过 MPI 接口，对 CPU2(D02P02)下载时，需要在 CPU2 的 Charts 表中组态 SER 功能块，且 US 的 CPU 槽号地址应为 02。不要在 CPU1 中编写关于 CPU2 服务的 SER 功能块，否则会出现致命错误。

- 在 CFC 中定义两个功能，第一个功能通过 CTR 功能块累加 BF 发出的脉冲个数。累计脉冲数超过 100 时，CTR 功能块复位，然后有重新开始计数。另一个功能是利用积分功能块，实现自身累加，达到上限 999 时，复位，然后重新积分。

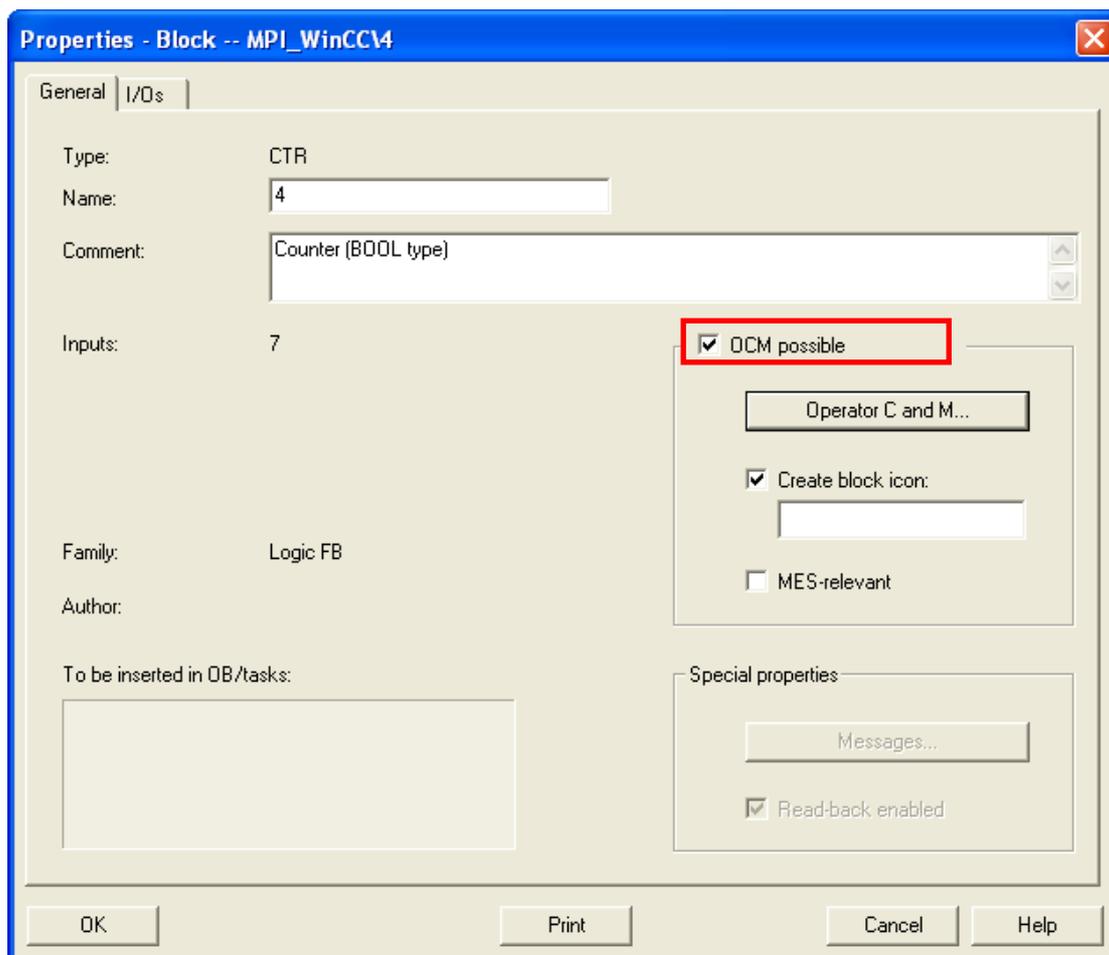


- 在 CFC 的菜单中，选择 Options 中的 Customize，选择 Compile/Download...，选中修改地址本选项 Create address book。

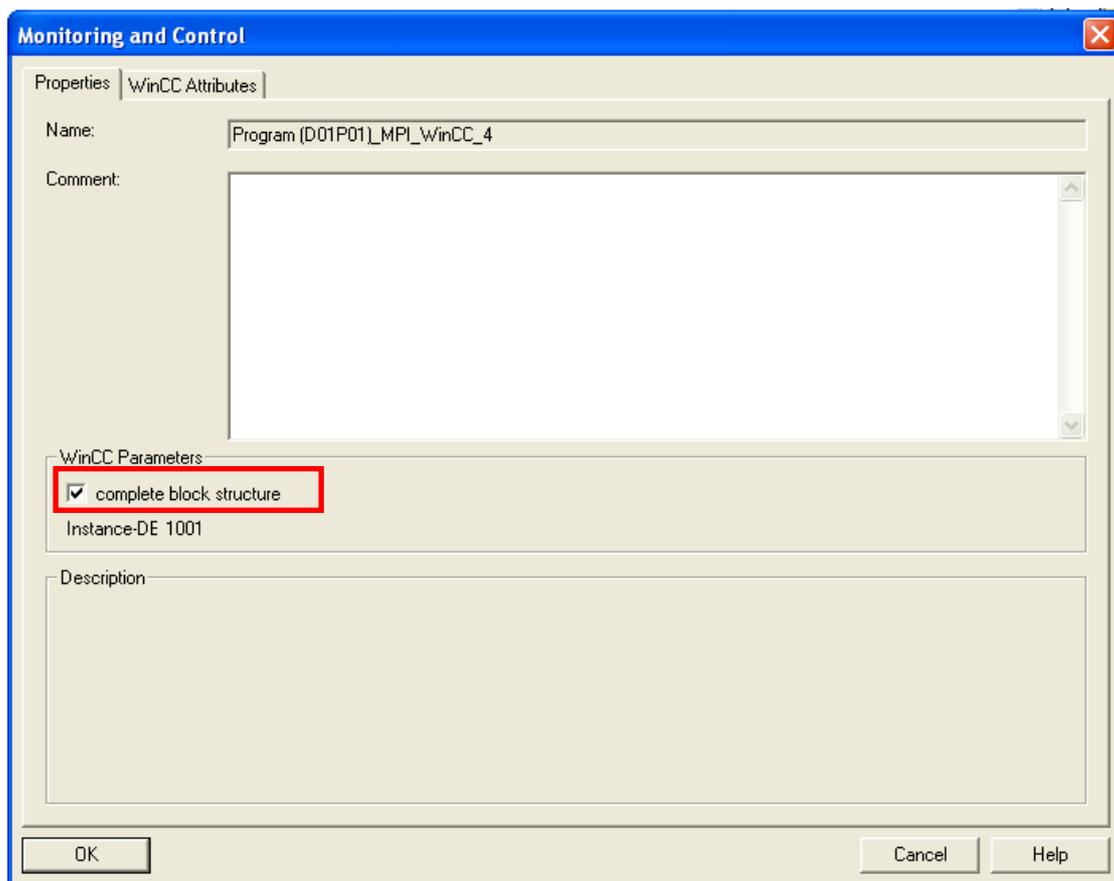




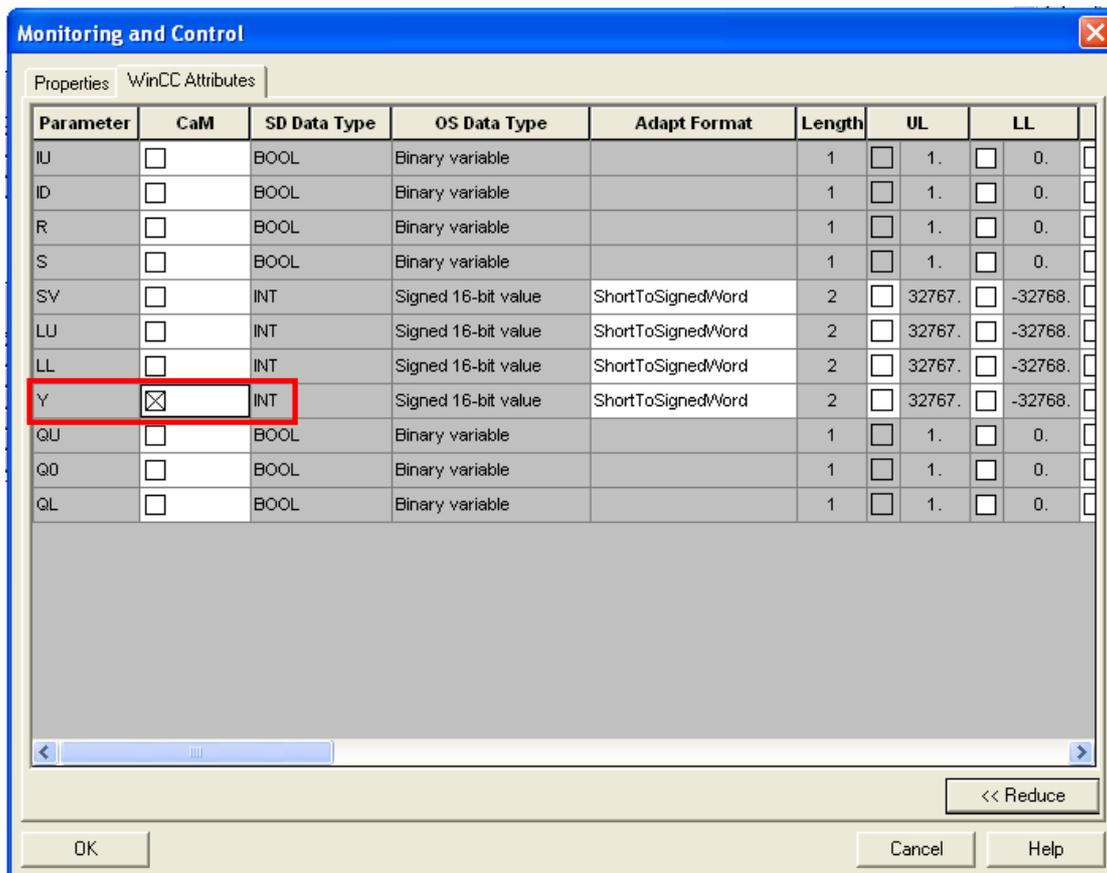
- 双击 CTR 功能块，打开 CTR 功能块的属性。选择 OCM possible。



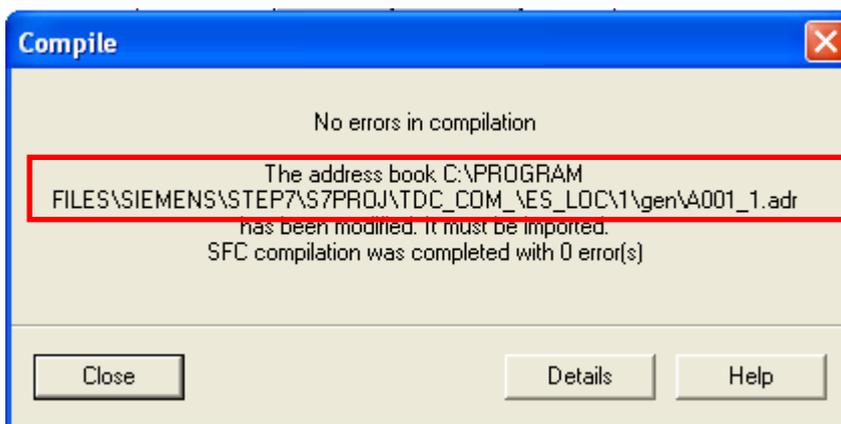
- 点击 Operator C and M...，打开 Monitoring and Control 属性。注意该功能块的映像区数据块为 1001。可以选中 complete block structure 选项，选中该选项，就是对该功能块的所有输入和输出进行监控。



- 在打开的 Monitoring and Control 属性中，切换到 WinCC Attributes 页面，选择要监控的变量。并且使用同样的方法选中 INT 功能块的输出 Y 作为监控变量。



- 设置完成后，对 CFC 进行编译，出现生成的地址本的路径。该地址本的文件名为 A001\_1.adr，A001 表示机架号，1 表示 CPU 槽号，adr 表示该文件是地址本的扩展文件名。

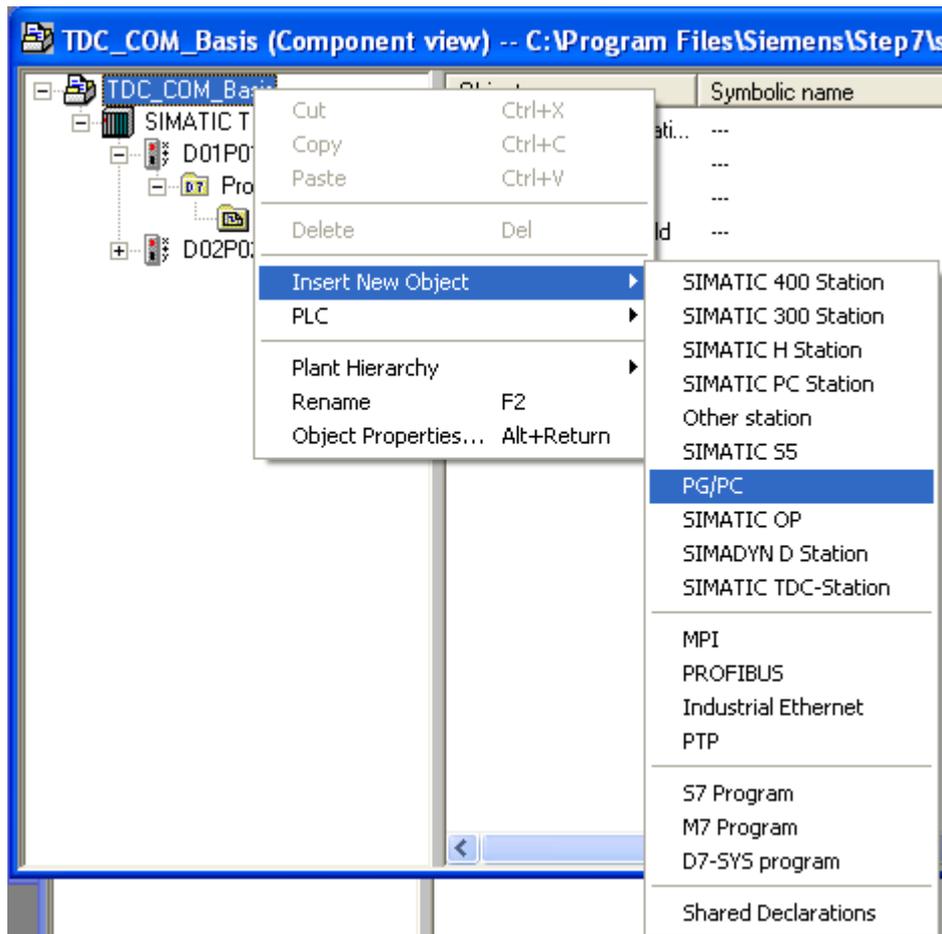


- 根据该路径，通过 WordPad 打开该文件。上面记录着 CPU1 即 D01P01 的周期任务(T1~T5)和中断任务(I1~I8)的设置信息。最重要的是提供了所要监控变量 Y(CTR)

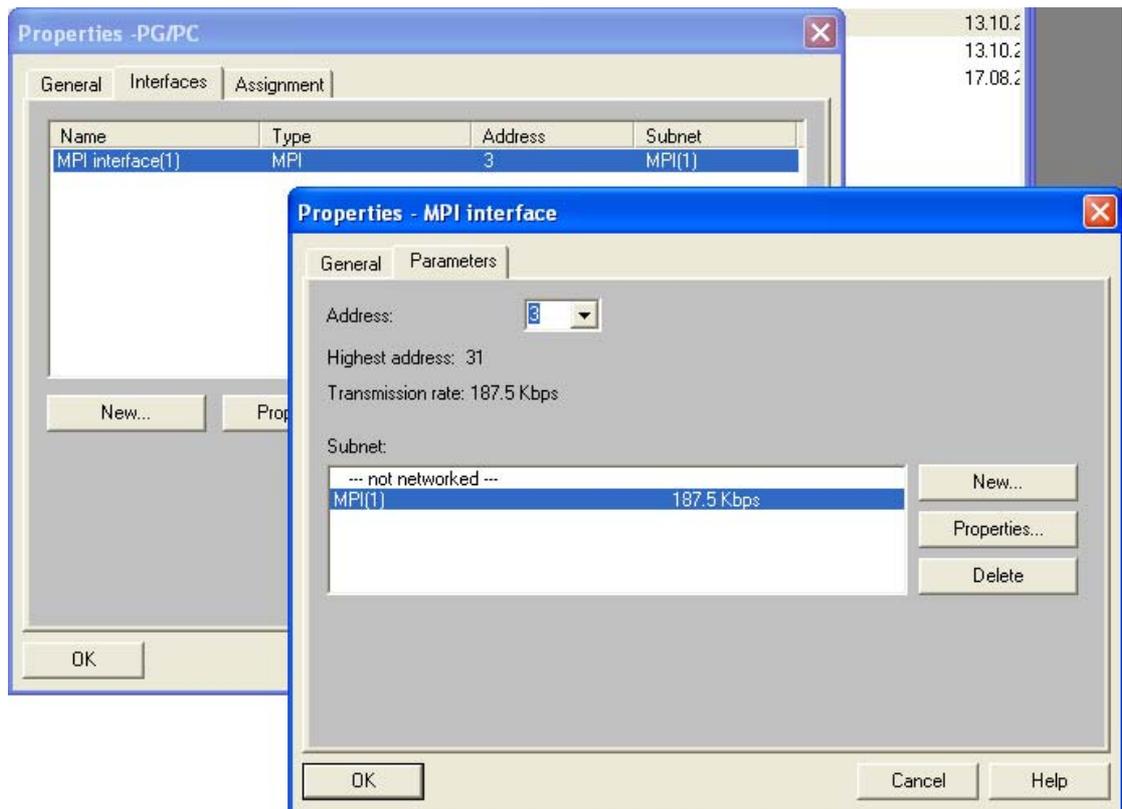
和 Y (INT) 所分别对应的数据功能块的代号 1001、1002, 以及对应该变量的偏移地址 24576。

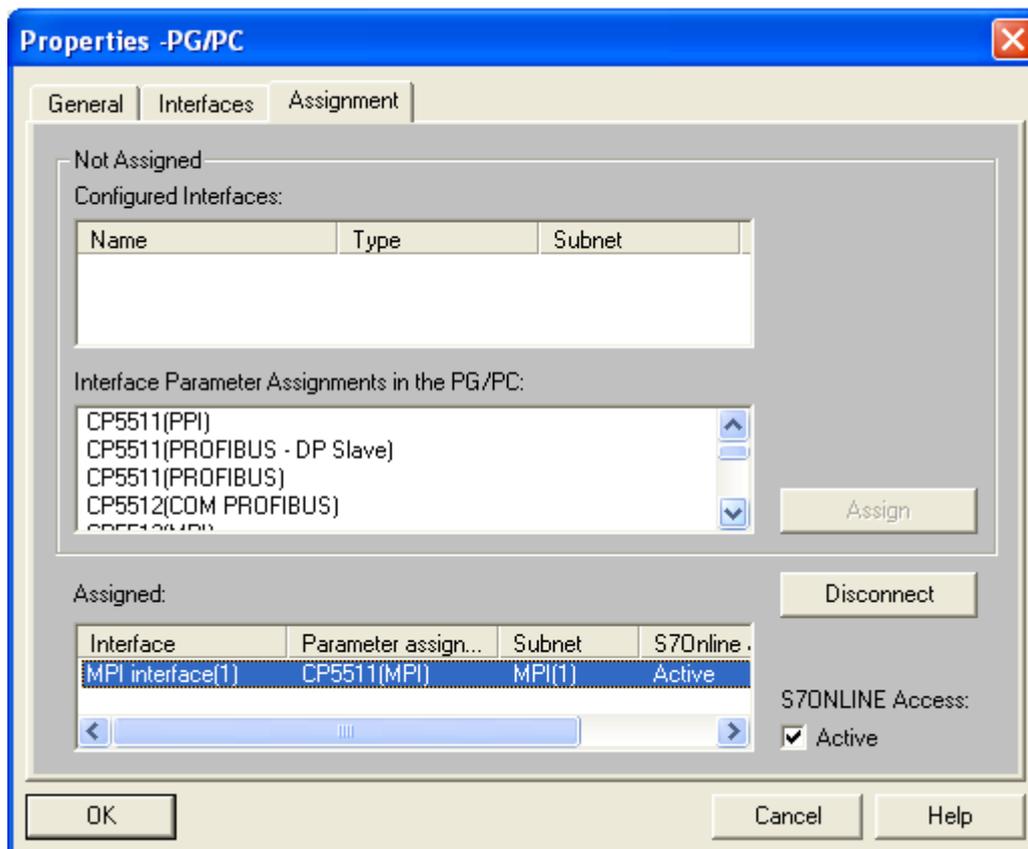
```
V 5.0.0
31.01.97
A001
D01P01
A001
13.10.2005:13:29
1.0
T1:1.0
T2:4.0
T3:16.0
T4:64.0
T5:256.0
I1:0
I2:0
I3:0
I4:0
I5:0
I6:0
I7:0
I8:0
FP:1
$:0
FP:MPI_WinCC:2:
FB:4:T1:1
Y :8091,1001,24576:I2::::A:::::N
FB:5:T1:1
Y :8091,1002,24576:NF:1.0::::A:::::F
```

- 下载 CFC。这时 CP50MO 的指示灯 H1, H3 灯绿常亮, H5 绿闪烁。
- 在 SIMATIC Manager 中, 插入 PG/PC 站。

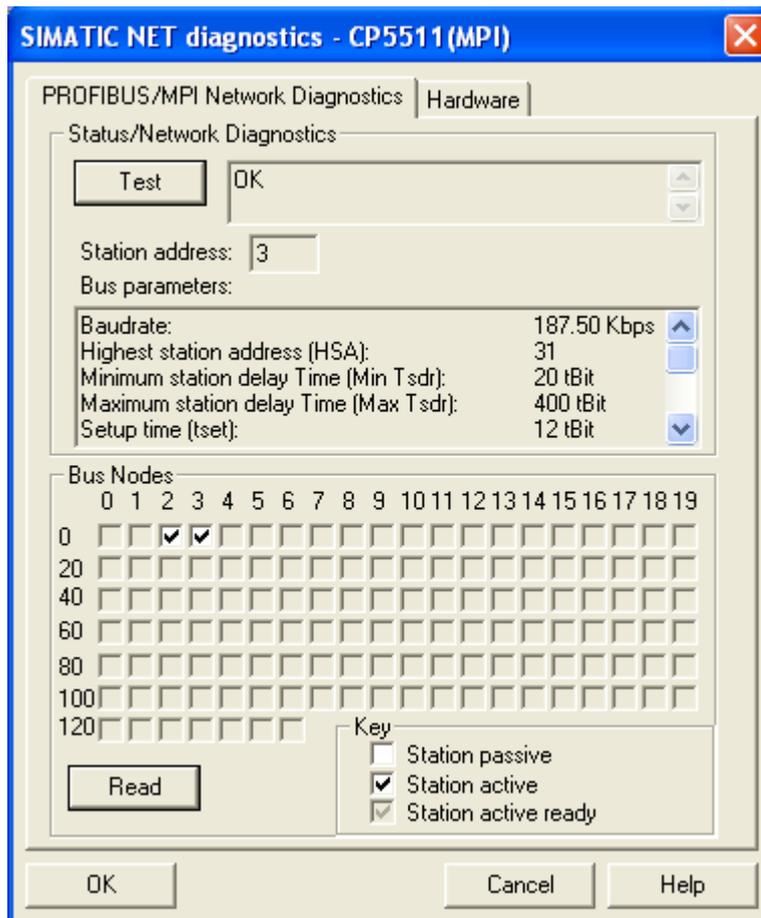


- 双击 PG/PC 图标，设置 MPI 的地址为 3。并且指定 PG/PC 的 MPI 接口为 CP5511(MPI)。

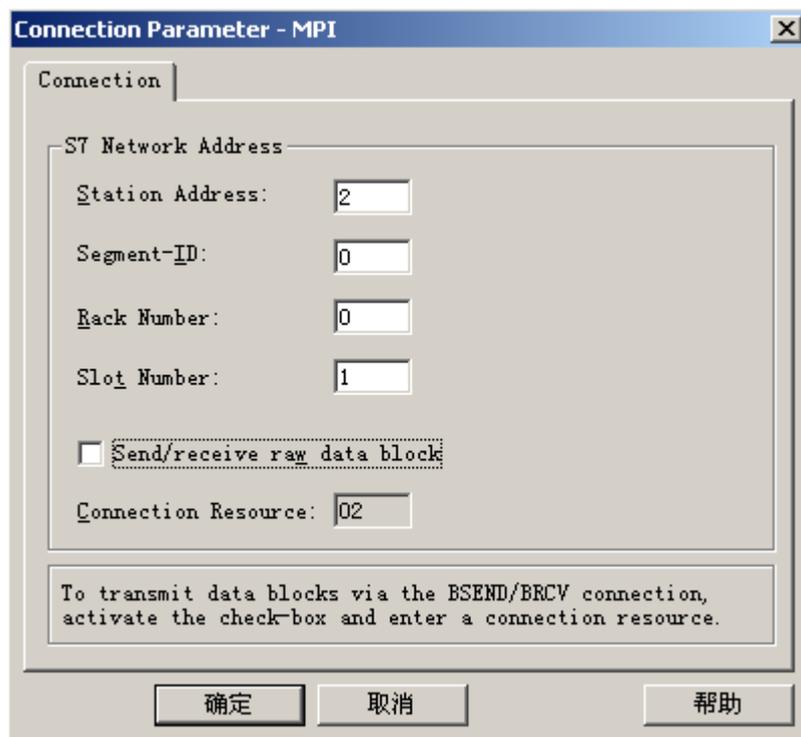




- 打开设置 PG/PC 接口，对 MPI 网络诊断。在本例中可以读到 MPI 的站地址为 2, 3。其中 2 为 TDC 的 CP50MO 的 X1 接口的 MPI 地址；3 为该 PG/PC 的 MPI 站地址。这时可以使用 MPI 接口，PG 对 CPU1 的程序进行下载和诊断。这种方法比使用 SC67 的串口电缆通讯要快的多。



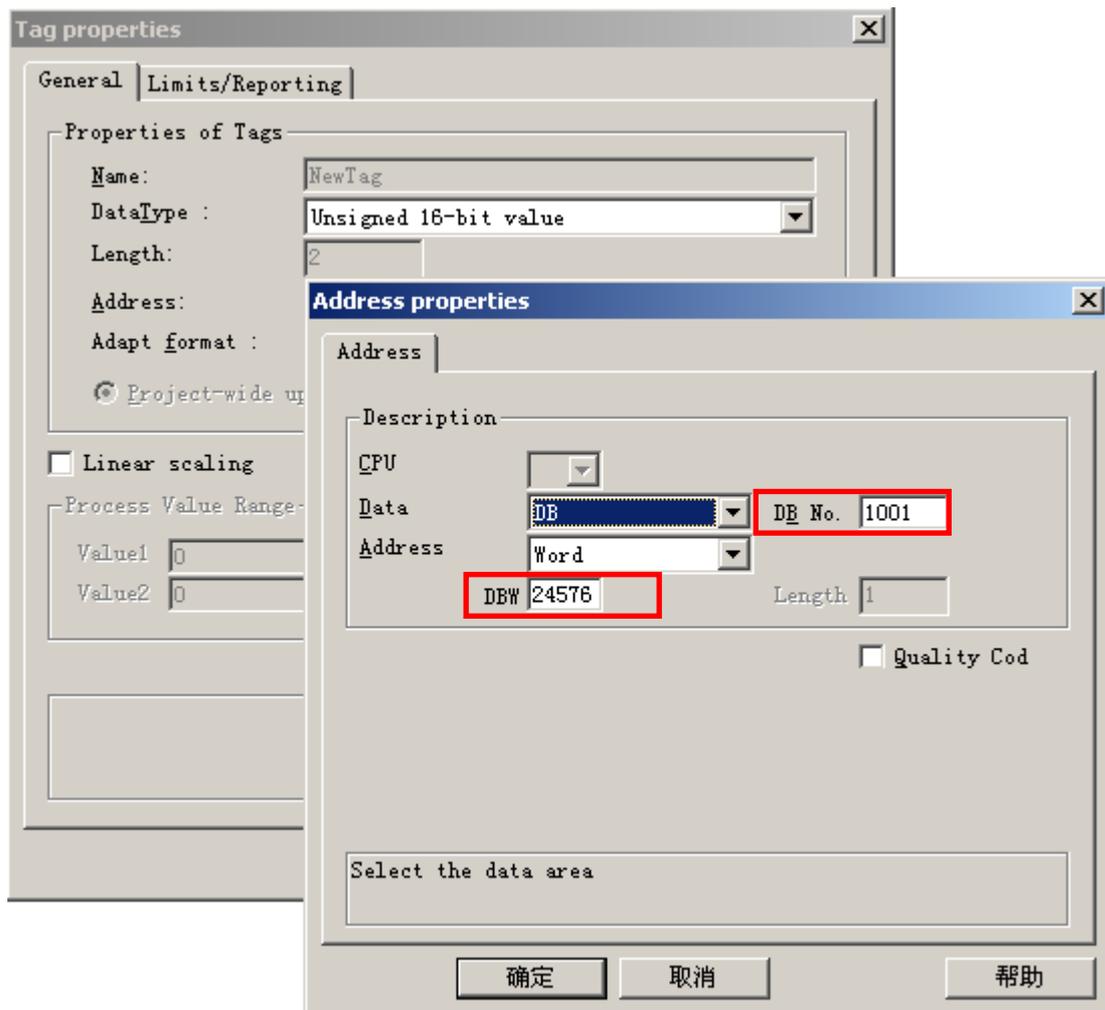
- 打开 WinCCExplore，新建工程并命名为 TDC\_WinCC\_MPI。在 Tag management 中插入 S7 驱动协议包，即 SIMATIC S7 protocol suite。选择 MPI 通道，新建一个连接 NewConnection，设置连接 MPI 通道的属性。设置 Station address 为 2，即 TDC 模板 CP50MO 的 MPI 地址。设置 Slot Number 为 1，即 SER 功能块的 US 参数所指定的 CPU1 的槽号。



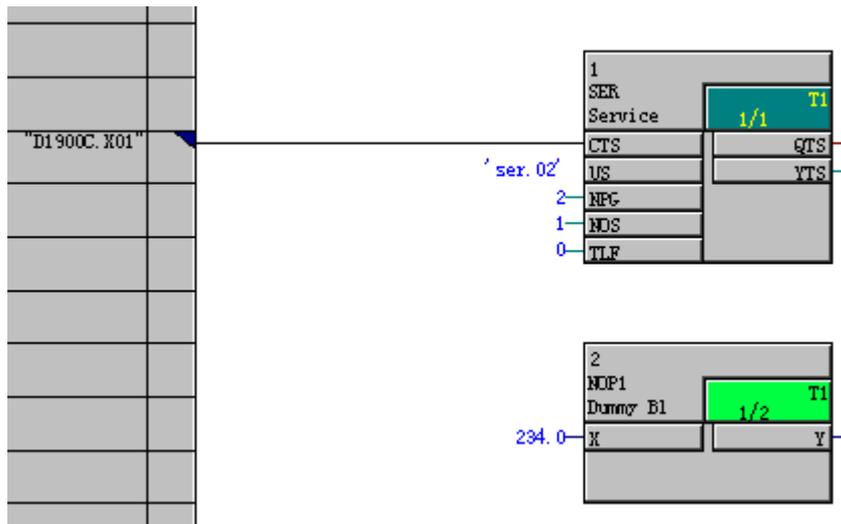
- 选择 MPI 通道，设置系统参数，取消循环管理选项 by PLC。



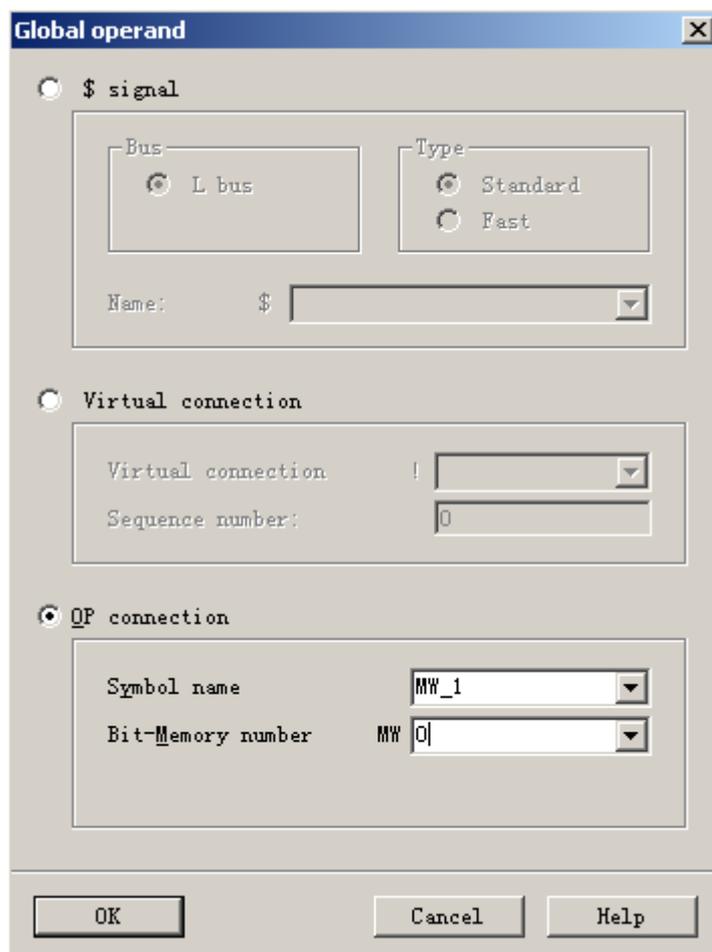
- 在 NewConnection 新建连接中建立新的变量 NewTag，根据编译生成的地址本 A001\_1，设置新建变量的地址。

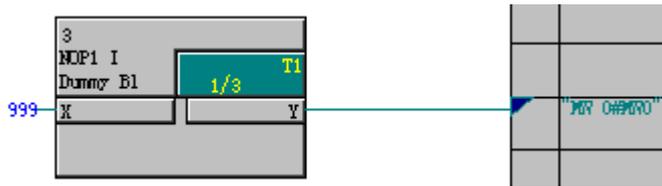


- 使用同样的方法, 在 CPU2 中新建一个名字为 MPI\_WinCC 的 CFC 程序, 添加 SER 功能块和一个 NOP1 功能。注意 NOS 应设置为 1。



- 也可以采用其他方法设置监控变量, 在 CPU2 的 MPI\_WinCC 程序中, 加入 NOP1\_I 的功能块。选中该功能块的输出 Y, 右键选择连接到地址 Interconnection to Address, 弹出全局操作数画面 Global operand, 可以设置为 MW0。



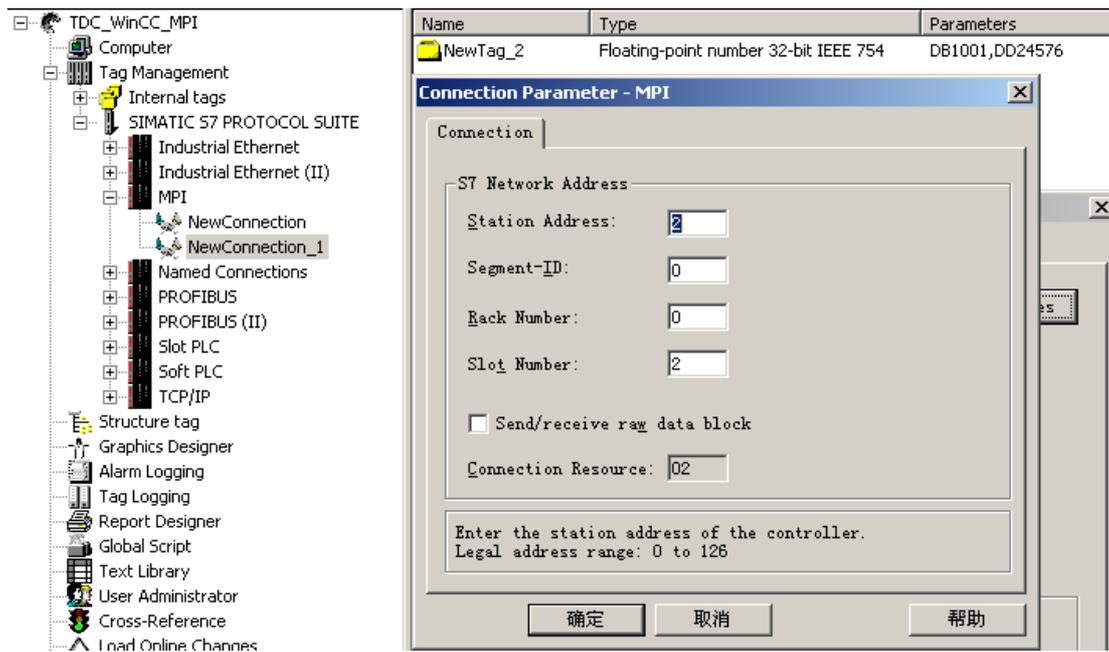


- 使用同样的方法，并对 NOP1 的输出 Y 编译生成成为新的地址本。并下载程序。

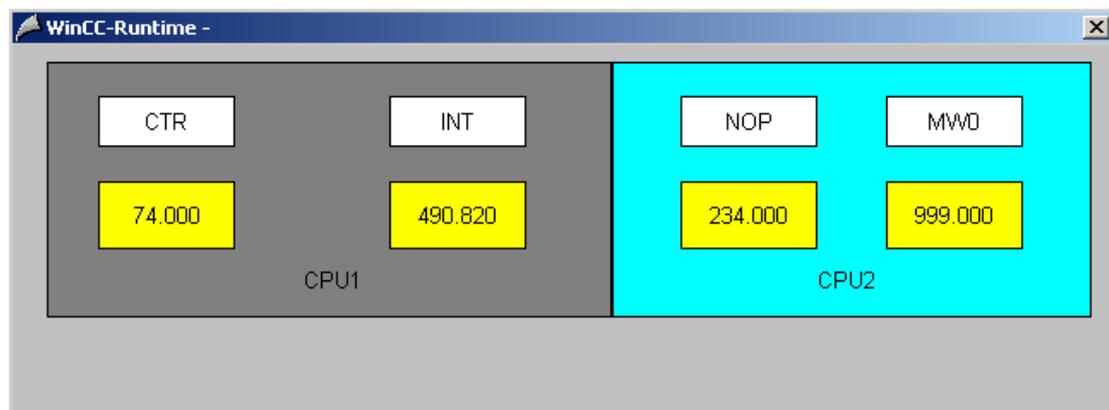
```

A001_2 - 记事本
文件(F) 编辑(E) 格式(O) 帮助(H)
U 5.0.0
31.01.97
A001
D02P02
A001
13.10.2005:17:43
1.0
T1:1.0
T2:4.0
T3:16.0
T4:64.0
T5:256.0
I1:0
I2:0
I3:0
I4:0
I5:0
I6:0
I7:0
I8:0
FP:1
$:0
FP:SERVICE:1:
  FB:2:T1:1
    Y :8091,1001,24576:NF:1.0:::A:::::F
    
```

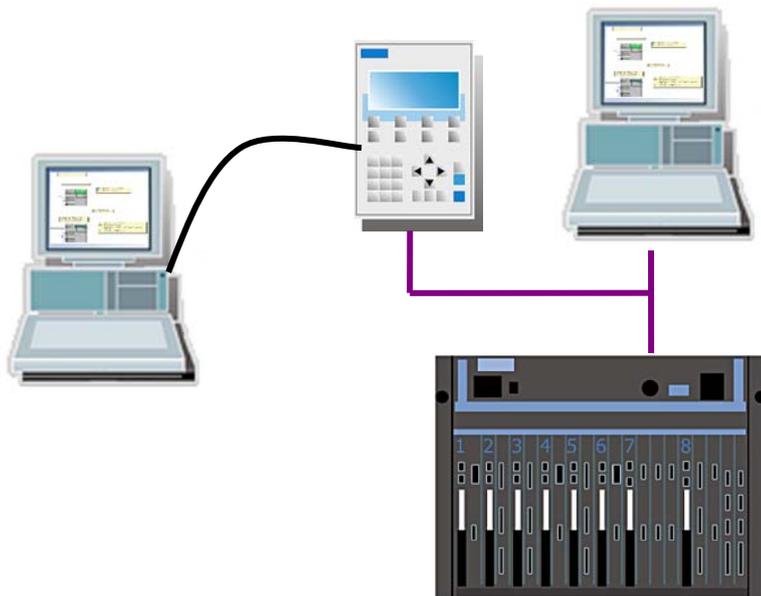
- 使用同样的方法，在 WinCC 中的 MPI 通道新建一个连接，并修改 CPU 的槽号地址为 2，并添加新的变量，用于显示 CPU2 中的监控变量。



- 最后新建一个画面，用于显示连接的 CPU1 和 CPU2 的监控变量。可以在 CPU2 中的程序，在线修改 NOP 功能块的值，察看其变化。



#### 4.4.2 CP50MO 与 OP27 通讯

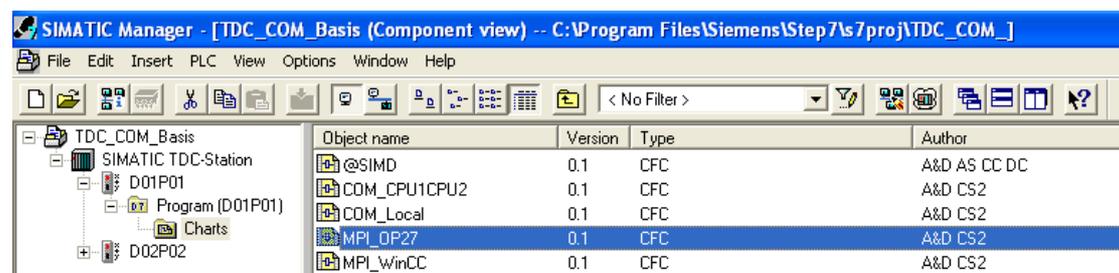


##### 4.4.2.1 在 SIMATIC Manager 中进行硬件组态

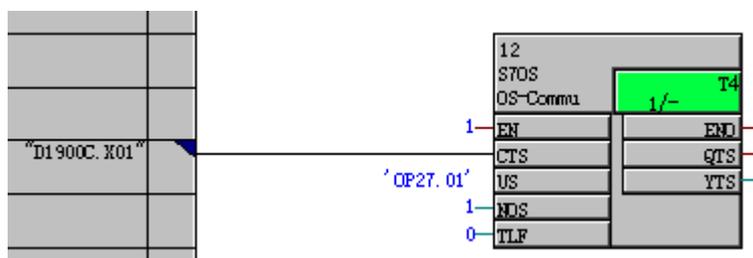
- 硬件组态与 4.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为 MPI 接口。MPI 默认的站地址为 2。
- 计算机 PG 使用标准的 MPI 电缆和 CP5511MPI 网卡，用于下载 CFC 程序和诊断 TDC。
- 另一台 PG 使用串口电缆（6ES5734-2BF00）与 OP27 的串口相连，用于下载组态图形到 OP27 中。

##### 4.4.2.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 MPI\_OP27，并通过双击打开。



- 我仍然使用 CP50MO 的 X1 接口作为 MPI 的通讯接口，在 CPU1 的 CFC 程序 MPI\_WinCC 中，已经编写了关于 @MPI 关联 X1 接口的功能块和 SER 功能块，所以不必再次编写。如果此前没有做该程序，可以参照该 CFC 程序来编写，也可以使用 @MPI 功能块与 CP50MO 的 X2 接口关联。需要注意的是当 MPI 网络上有 2 台显示设备，一台是 WinCC，另一台是 TP/OP，需要把 SER 功能块的 NOS 参数改为 2。当然也可以使用 S7OS 功能块，当使用 S7OS 功能块时，SER 功能块 NOS 参数改为 0，这样不会因显示设备过多而又频繁刷新数据而增加 CPU 的负担。本例中可以使用 S7OS 功能块。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。



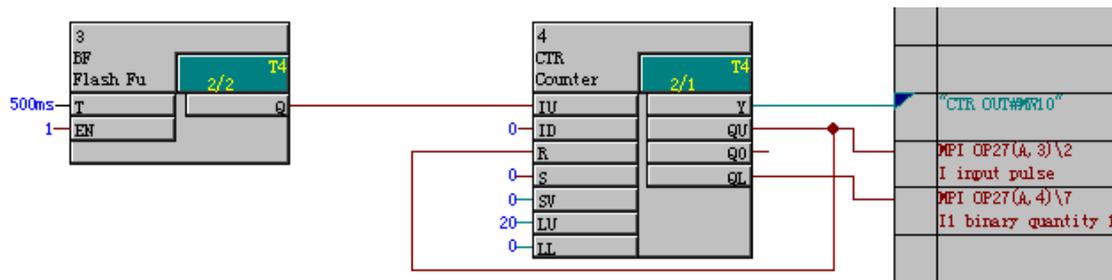
S7OS 功能块的引脚说明：

CTS—连接硬件组态的模板地址为 D1900C.X01，即通过 CP50MO 的 X1 接口作为 MPI 通讯的关联接口。

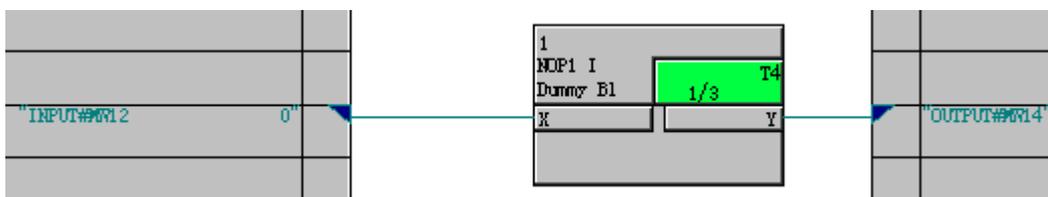
US—任意定义一个不大于 6 个字符长度的通道名，01 表示关联 CPU 的槽号地址。说明通过 SER 功能块，CPU1 来完成 MPI 通讯处理服务。

NOS—指定 OS 的个数，默认为 1，也就是指定监控变量（C and M）的显示设备（例如 OP27）的个数。，即可以使用设定的 MPI 接口 X1 来完成 MPI 网络通讯。

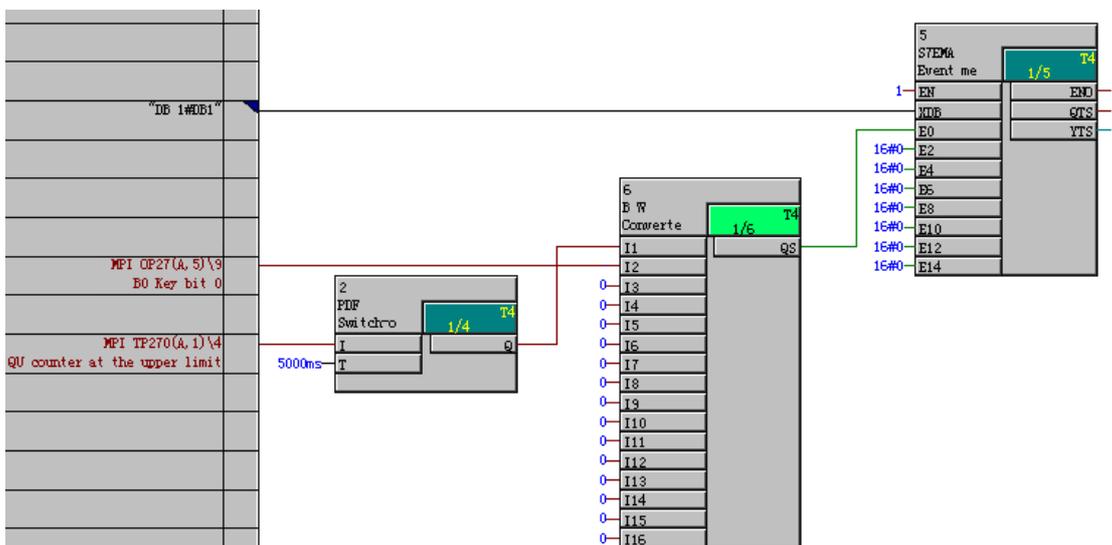
- 定义 1 个功能用于 OP270 触摸屏上的显示。参照 MPI\_WinCC 程序功能解释。定义 CTR 的输出 Y 的 OP connection 全局操作数为 MW10，符号名为 CTR\_OUT。目的是让该变量在 OP27 上显示变化。QU 作为事件消息显示；QL 作为报警消息显示。



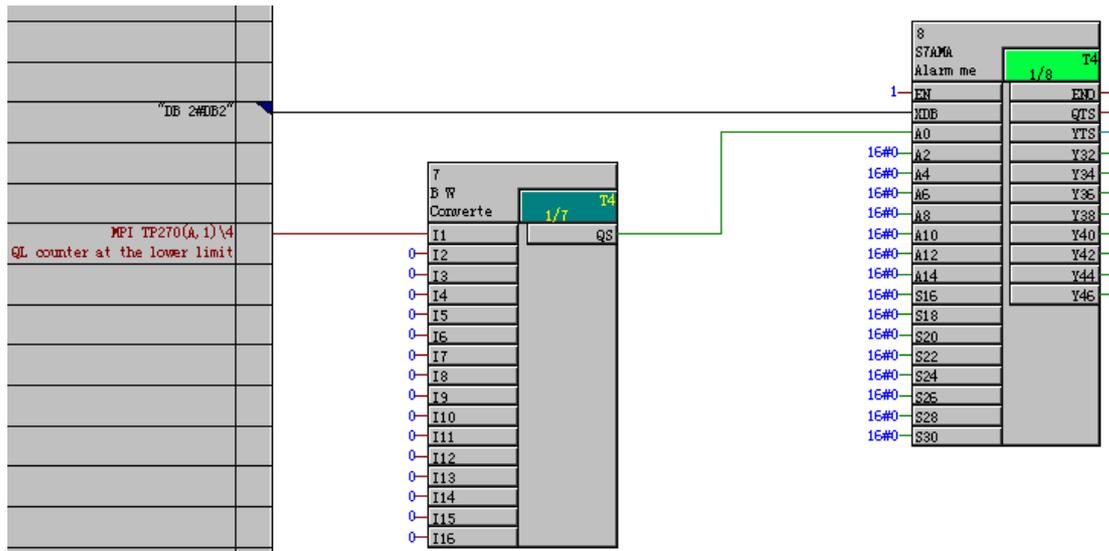
- 定义 1 个功能用于 OP270 触摸屏上的写入数据，再从触摸屏上显示该数据。定义输入的数据变量 X 的 OP connection 的全局操作数地址 MW12，符号名为 INPUT。定义输出变量 Y 的 OP connection 的全局操作数地址 MW14，符号名为 OUTPUT。



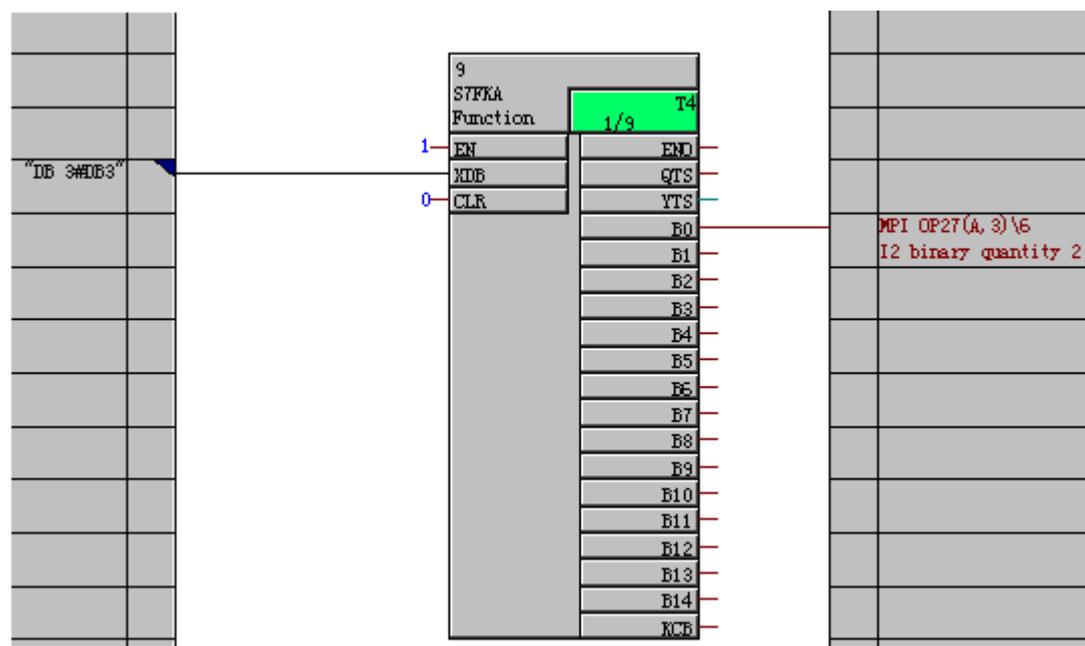
- 定义 1 个功能用于 OP270 触摸屏上组态事件消息。如果计数器 CTR 达到最大值 100 时，作为事件的一个触发，通过功能块 PDF 保持 5s 钟，再通过功能块 B\_W，把 BOOL 值转换为 WORD 值，作为功能块 S7EMA 的事件消息字。通过全局操作数虚拟数据块 DB1 赋予给 S7EMA 功能块输入 XDB。同样使用按下功能键 F1 来激活另一个事件消息。



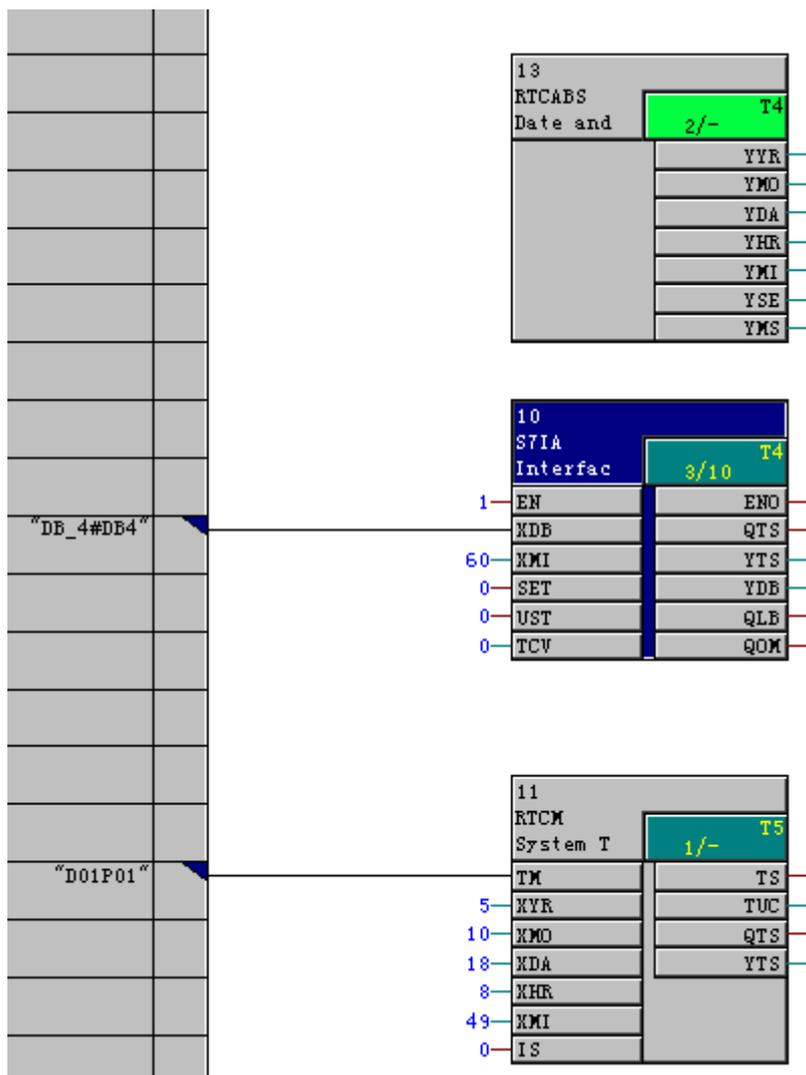
- 定义 1 个功能用于 OP270 触摸屏上组态报警消息。如果计数器 CTR 达到最小值 0 时，作为报警消息的一个触发，再通过功能块 B\_W，把 BOOL 值转换为 WORD 值，作为功能块 S7AMA 的报警消息字。通过全局操作数虚拟数据块 DB2 赋予给 S7AMA 功能块输入 XDB。



- 定义 1 个功能用于 OP270 触摸屏上组态功能键。通过全局操作数虚拟数据块 DB3 赋予给 S7FKA 功能块输入 XDB。



- 定义 OP 的系统时间与 TDC 同步。通过全局操作数虚拟数据块 DB4 赋给 S7IA 功能块输入 XDB。利用 RTCM 功能块来设定 TDC 的系统时间，XYR、XMO、XDA、XHR、XMI 分别对应年、月、日、时、分。输入 IS 信号脉冲由 0 变为 1，可以通过在线的方法，修改该值系统时间设定。通过功能块 RCTABS 可以读出系统的设定时间。



S7IA 功能块的引脚说明:

EN—为 1 时，使能该功能块运行。

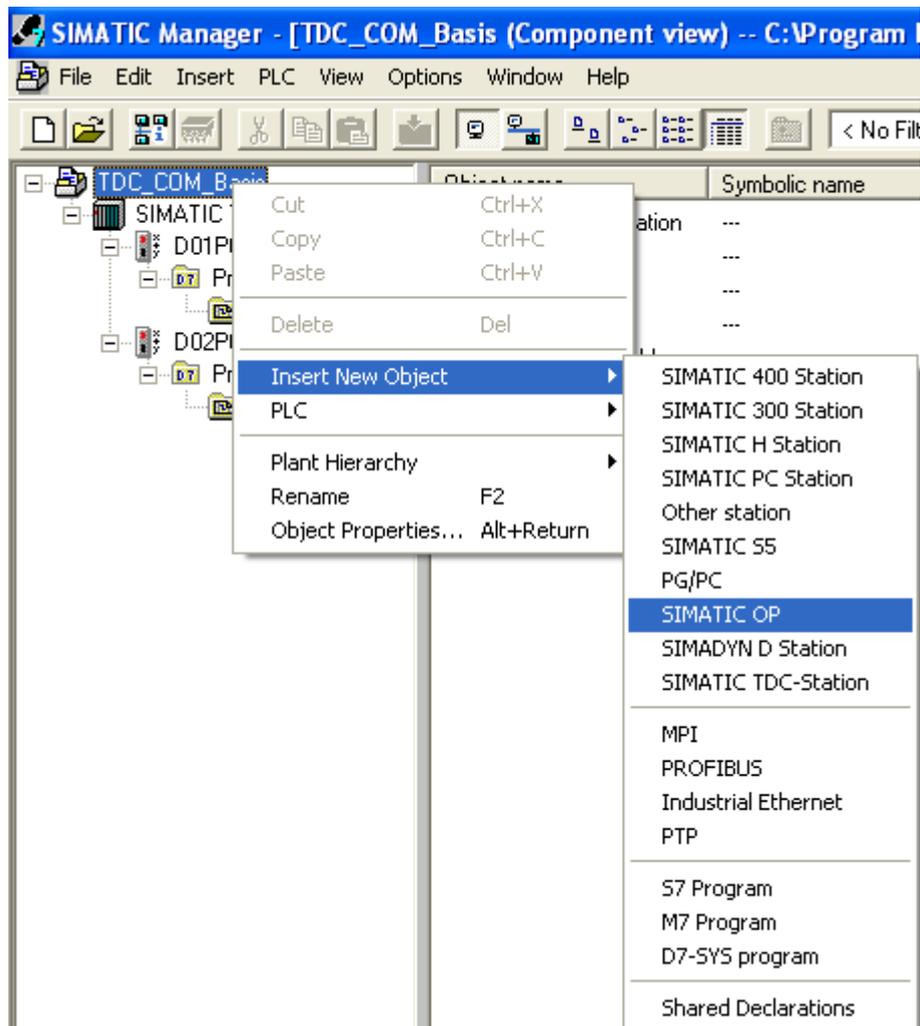
XDB—定义 16 字的用户接口数据区。

XMI—定义 OP27 与 TDC 的时间自动同步的间隔，单位为分钟。

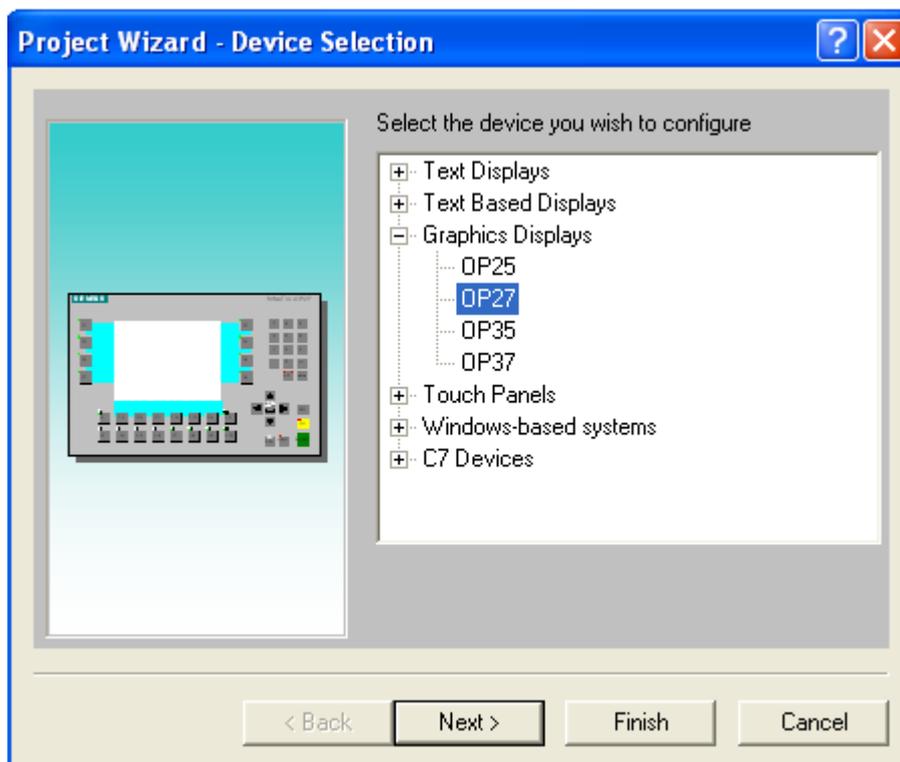
SET—通过正边沿来设定 OP27 的系统时间与 TDC 同步。不要频繁使用该操作。

UST,TCV—用于修改夏，冬令时的时间修正。

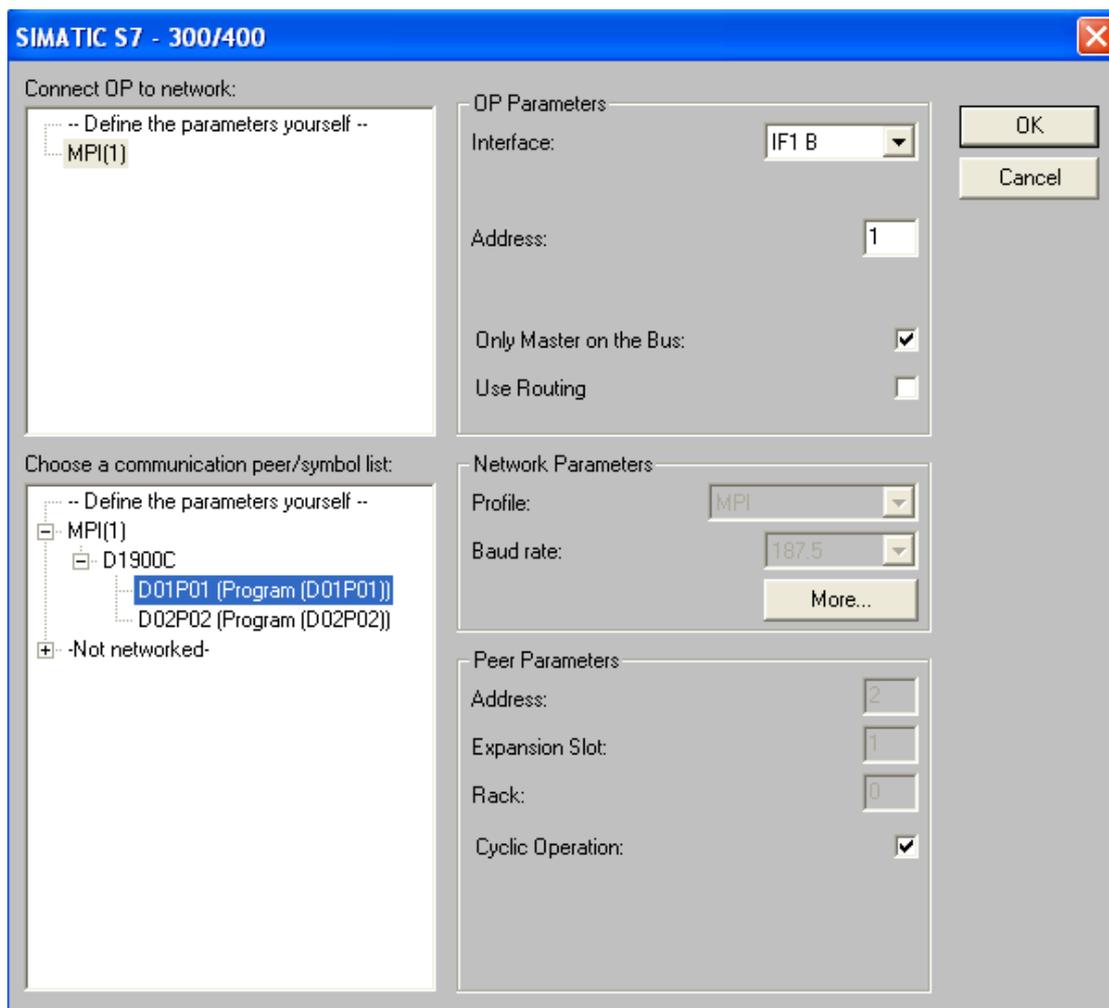
- 对程序进行编译，PG 通过 MPI 总线下载到 TDC 中。
- 在 Step7 的工程中插入一个新的设备 SIMATIC OP，并命名为 OP27。



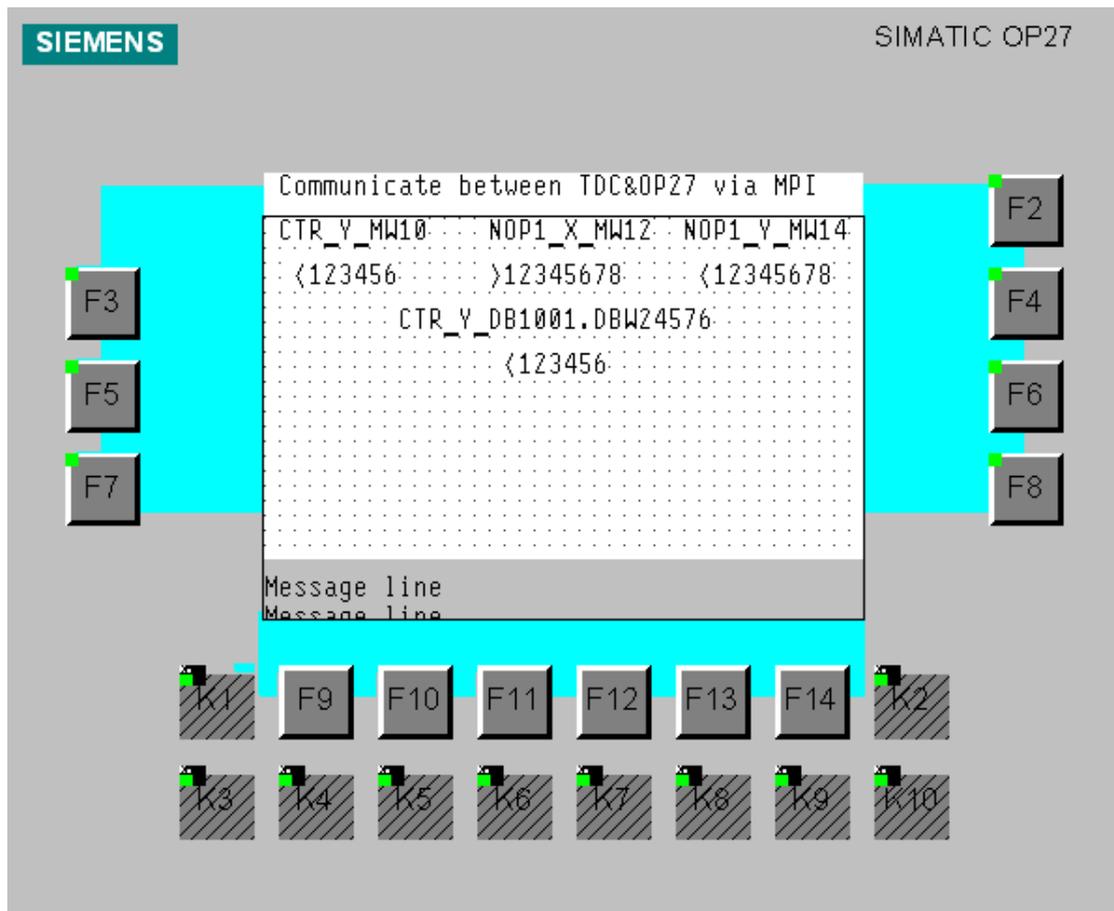
- 双击 OP27 图标，打开 SIMATIC Protocol 画面，选择 OP27 面板。



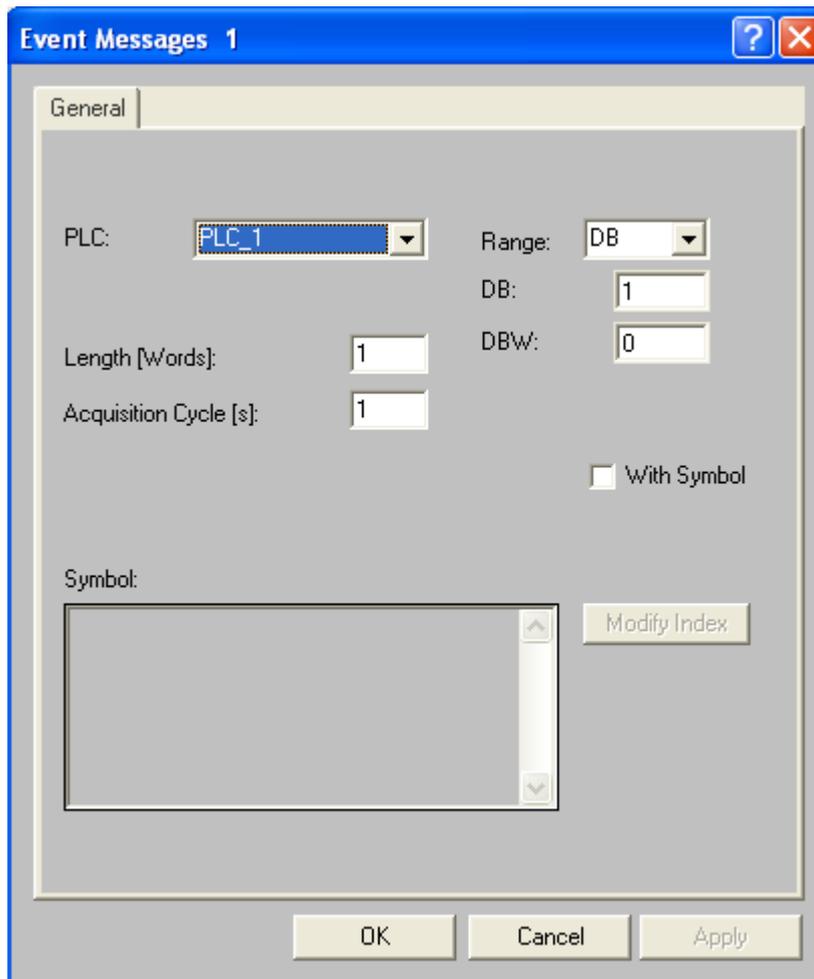
- 点击 Next, 打开 Parameter... 的网络参数配置画面。设置 OP 的 MPI 地址为 1, MPI 网络参数保持不变。Peer Parameters 的 TDC 参数选择 CPU1 作为服务功能的 CPU。这与 S7OS 功能块的输入 US 的设置 ('OP27.01') 相对应。参数 2, 1, 0 为灰色的, 不能更改。2 表示 TDC 的 MPI 网卡 CP50MO 的 MPI 地址为 2, 1 表示 CPU 的槽号, 0 表示 TDC 的机架号。然后点击 Finish, 完成设置。



- 新建一个画面，放入显示和修改的变量。CTR\_Y\_MW10 用于显示功能块 CTR 计数器的输出变量 MW10 的变化。NOP1\_X\_MW12 用于手动输入变量 MW12，NOP1\_Y\_MW14 用于显示输入变量 MW12 的变量值。需要注意的是 CPU1 中的程序 MPI\_WinCC 生成地址本的变量同样可以显示在 OP27 上，例如 CTR 功能块的输出 Y 的地址本中的变量地址是 DB1001.DBW24576。命名为 CTR\_Y\_DB1001.DBW24576 用于 OP27 显示。以上变量均为整型变量。



- 在 Protocol 中的区域指针 Area Pointer 中，新建一个事件消息 Event Message 区域指针，定义该指针的变量为 DB1.DBW0。与 CFC 程序中的功能块 S7EMA 的事件消息字 DB1 对应。在消息组态中，定义 DB1.DBX1.0 的消息文本为“reach the upper limit”。用于显示当计数器的数值达到最大 20 时，显示该事件文本 5 秒钟。

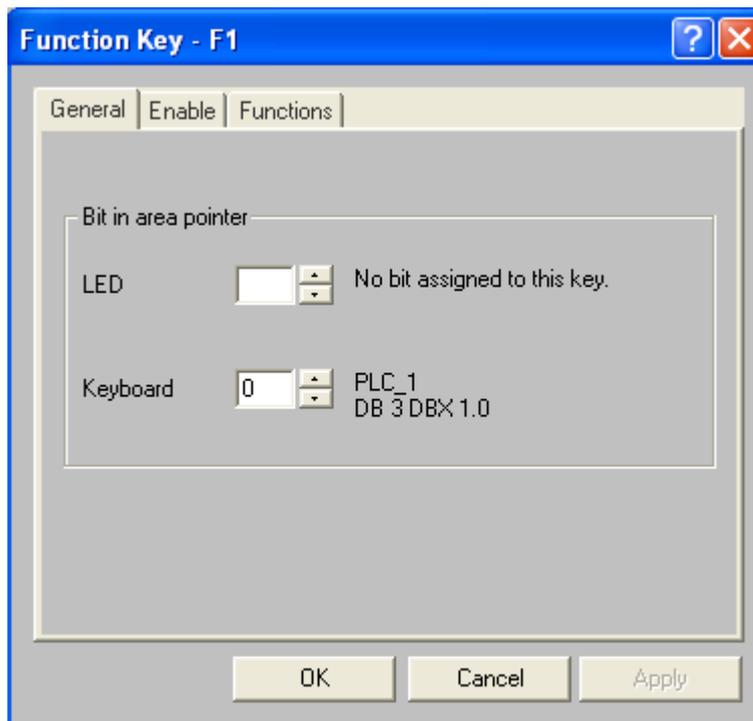


Event Messages - OP 27				
	1	10	20	30
0000				
0001	reach the upper limit			
0002	press F1 button			

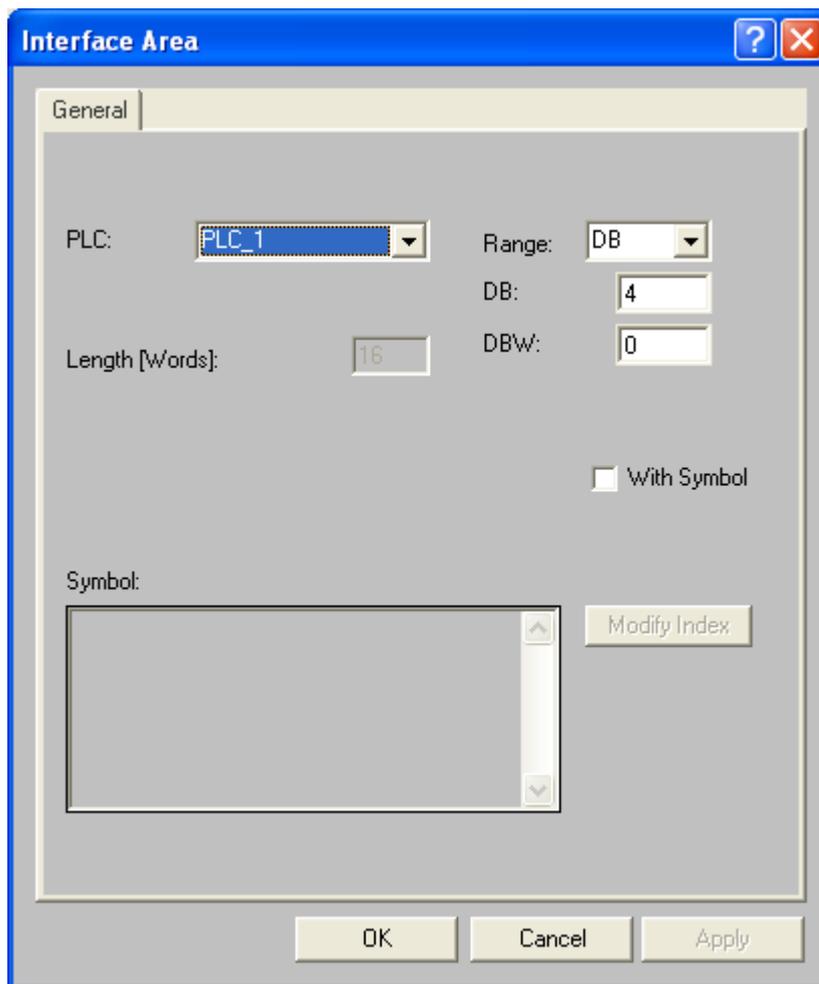
- 在 Protocol 中的区域指针 Area Pointer 中，新建一个报警消息 Alarm Message 区域指针，定义该指针的变量为 DB2.DBW0。与 CFC 程序中的功能块 S7AMA 的事件消息字 DB2 对应。在消息组态中，定义 DB2.DBX1.0 的消息文本为“reach the lower limit”。用于显示当计数器的数值达到 0 时，显示该报警文本。

Alarm Messages - OP 27				
	1	10	20	30
0001	reach the lower limit			
0002				

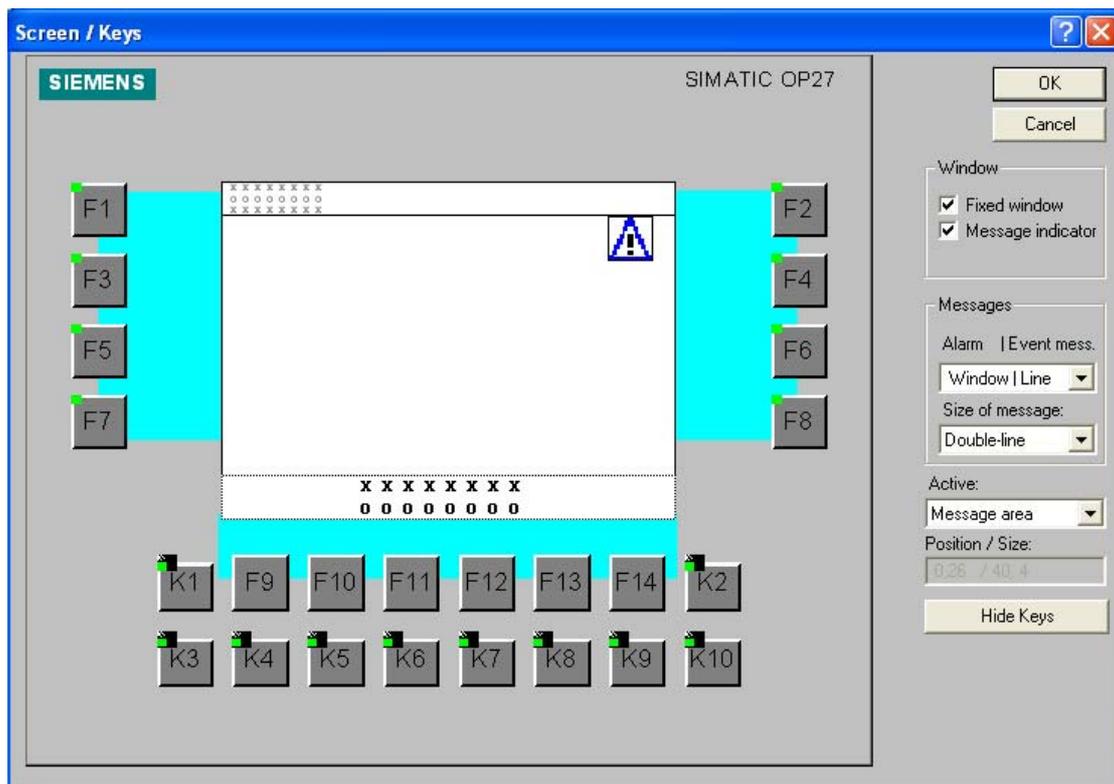
- 在 Protocol 中的区域指针 Area Pointer 中，新建一个功能键分配指针 Function Key Assig.区域指针，定义该指针的变量为 DB3.DBW0。与 CFC 程序中的功能块 S7FKA 的事件消息字 DB3 对应。在系统 System 菜单中，选择 Screen/Keys 菜单，点击 F1 按键，设置其为 DB3.DBX1.0。其它键设置与其相同。点击 F1 键时，CFC 中功能块 S7FKA 的状态。



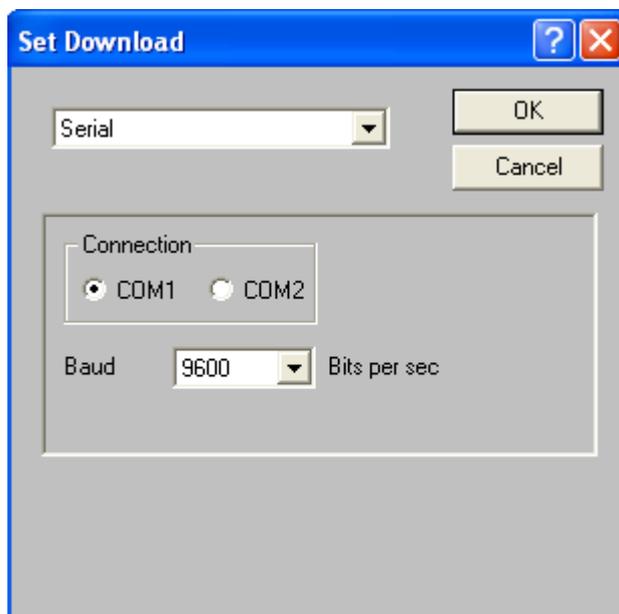
- 在 Protocol 中的区域指针 Area Pointer 中，新建一个接口区域 Interface Area 区域指针，定义该指针的变量为 DB4.DBW0。与 CFC 程序中的功能块 S7IA 的事件消息字 DB4 对应。这样使 OP27 的系统时间与 TDC 的一致，可以在显示报警消息时，显示该时间。



- 定义报警和事件消息为 Window|Line 方式显示，激活区域为 Message area。



- 设置下载属性，波特率为 9600。用 Protocol 通过串口电缆下载 OP27 程序。

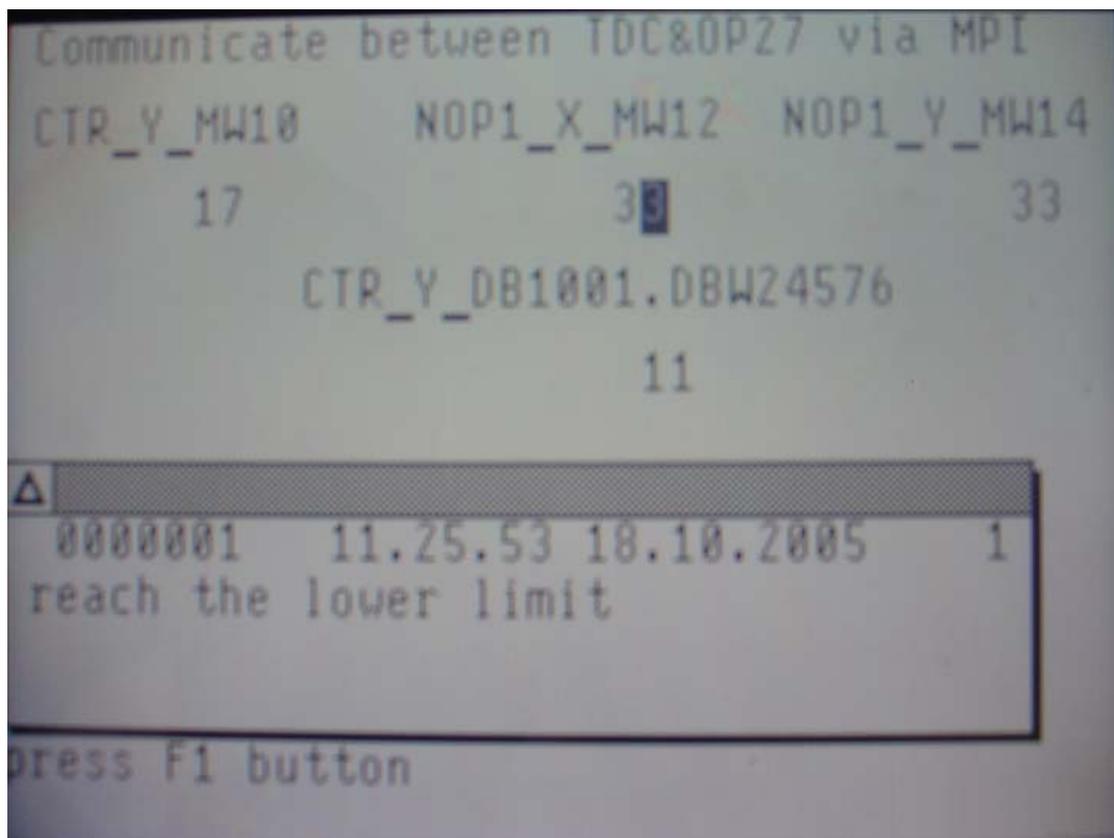


- 显示 OP27 运行结果。

```
Communicate between TDC&OP27 via MPI
CTR_Y_MW10      NOP1_X_MW12  NOP1_Y_MW14
      2              33          33
      CTR_Y_DB1001.DBW24576
              3

△
0000001  11.25.53 18.10.2005  1
reach the lower limit

reach the upper limit
```



#### 4.5 PROFIBUS-DP 通讯

PROFIBUS-DP 通讯是 SIMATIC 产品的标准通讯协议，主站之间的数据传输的方式是令牌方式，主站和从站之间的数据传输方式是主从方式，最高 126 个网络节点。CP50MO 提供 2 个 DP 接口，通讯波特率高达 12M，作为 DP 网卡与其它 SIMATIC CP 网卡相比可以同时作为 DP 主站和 DP 从站，支持 SYNC 和 FREEZE 功能，支持 Shared input 功能(与 S7 PLC 中的 DX 功能类似)，通讯数据量高达 244bytes。

基本需要的硬件包括机架 UR5213, CPU, CP50MO 模板。额外需要的软件 COM PROFIBUS 来组态 PROFIBUS-DP 网络组态。

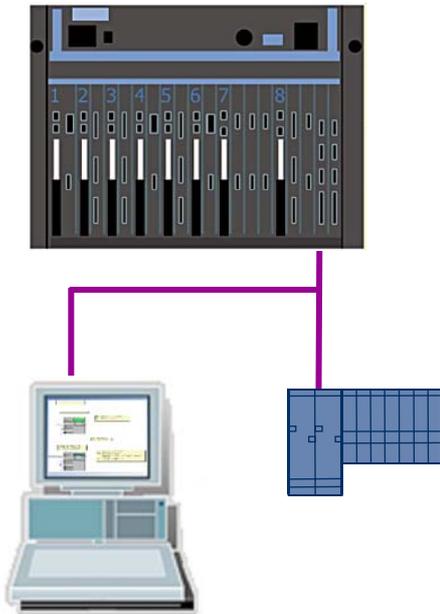
一个 CP50MO 模板，对应一个功能块@PRODP 来组态对应，该功能块用于初始化和监控 PROFIBUS-DP 通讯。使用功能块 CTV 和 CRV 来发送和接收数据。由于不支持与显示设备 (WinCC/OP/TP) 等通讯，所以不需要组态 SER 功能块。

允许的数据传输模式为刷新 (refresh)，对于接收方可以是复合 (multiple) 的数据传输方

式。

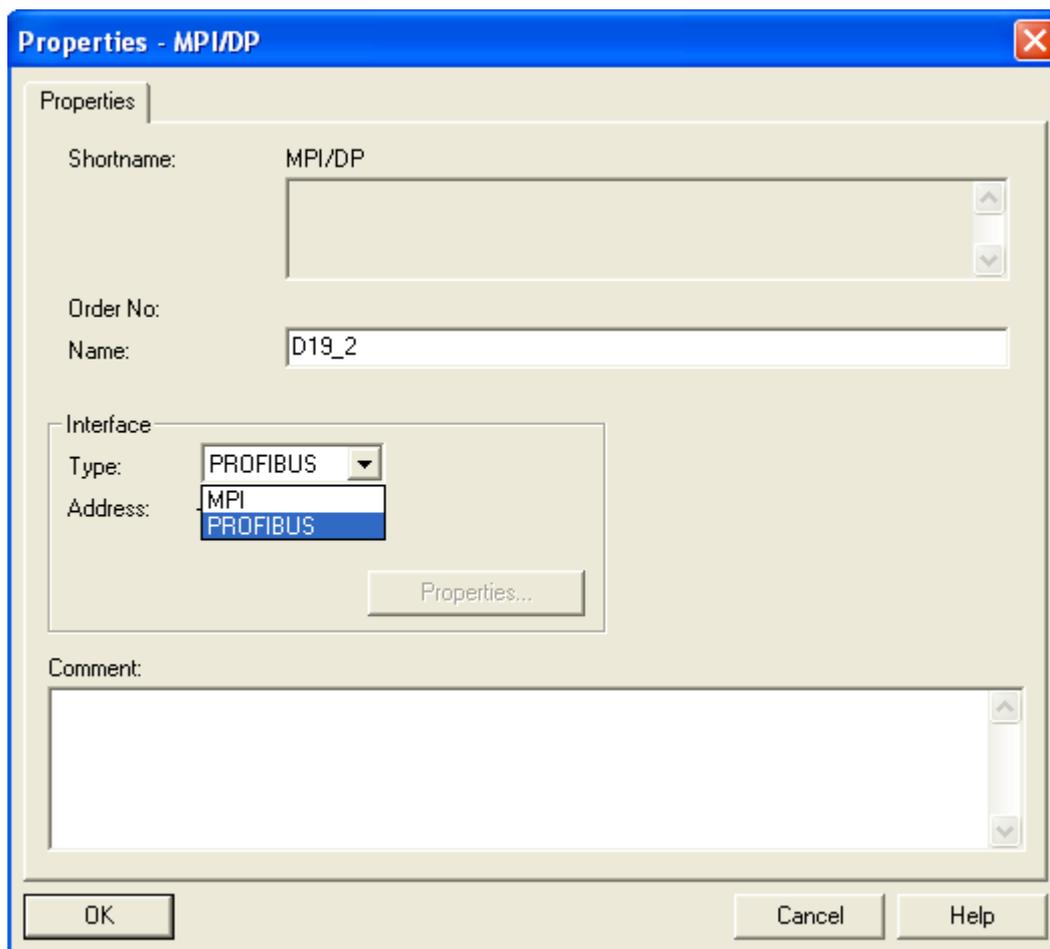
CP50MO 通过 PROFIBUS-DP 协议可以与 SIMATIC S5/S7, 传动, ET200, SIMADYN D / SIMATIC TDC, 以及经过验证第三方设备通讯。

#### 4.5.1 CP50MO 作为主站



##### 4.5.1.1 在 SIMATIC Manager 中进行硬件组态

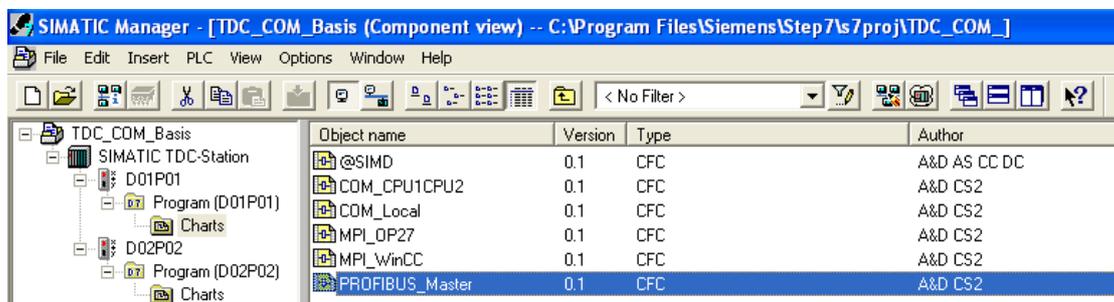
- 硬件组态与 4.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为 MPI 接口。修改该模板的 X2 接口属性为 PROFIBUS。



- 计算机 PG 使用标准的 MPI 电缆和 CP5512MPI 网卡，用于下载 CFC 程序和诊断 TDC。

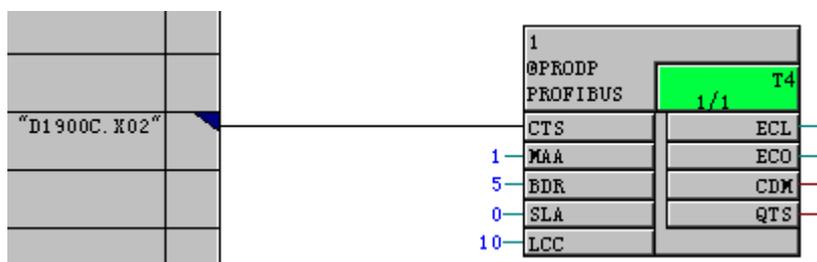
#### 4.5.1.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 PROFIBUS\_Master，并通过双击打开。



- 插入@PRODP 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C.X02。

即利用 CP50MO 的 X2 作为 PROFIBUS 通讯的 DP 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。



@PRODP 功能块的引脚说明：

CTS—连接在硬件组态中的硬件模板地址 D1900C.X02

MAA—定义 CP50MO 的 PROFIBUS-DP 节点站地址，范围从 1~123，默认地址为 1。

BDR—定义 PROFIBUS-DP 的通讯波特率，默认通讯速率为 1.5M。

0 = 9.6 kbaud

1 = 19.2 kbaud

2 = 93.75 kbaud

3 = 187.5 kbaud

4 = 500 kbaud

5 = 1.5 Mbaud

6 = 3 Mbaud

7 = 6 Mbaud

8 = 12 Mbaud

SLA—仅 CP50MO 作为 DP 从站用于初始化，默认为 0:

0: CP50MO 既可以作为 DP 主站又可以作为 DP 从站，此时需要 COM PROFIBUS 软件组态网络信息下载到 TDC 中。

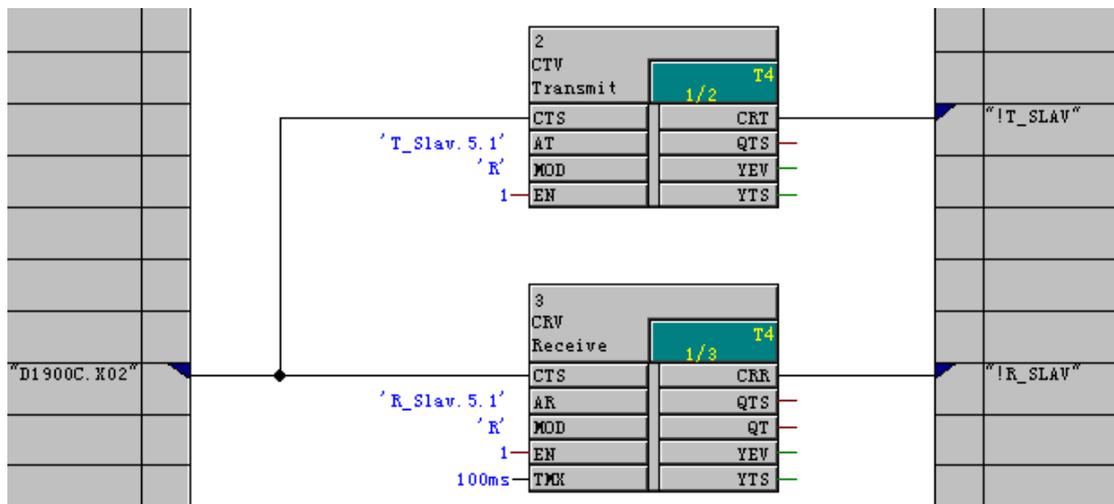
1 or 2: CP50MO 完全作为 DP 从站，不需要 COM PROFIBUS 组态。

1: CP50MO 可以被其它 DP 主站要么读其输出，要么写其输入。

2: CP50MO 可以被其它 DP 主站既可以读其输出，又可以写其输入。

- 分别插入 CTV 和 CRV 功能模块，用于与站地址为 5 的 DP 从站 ET200S 接收和发

送数据。



CTV 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 T\_Slav.5.1。

MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!T\_SLAV

CRV 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

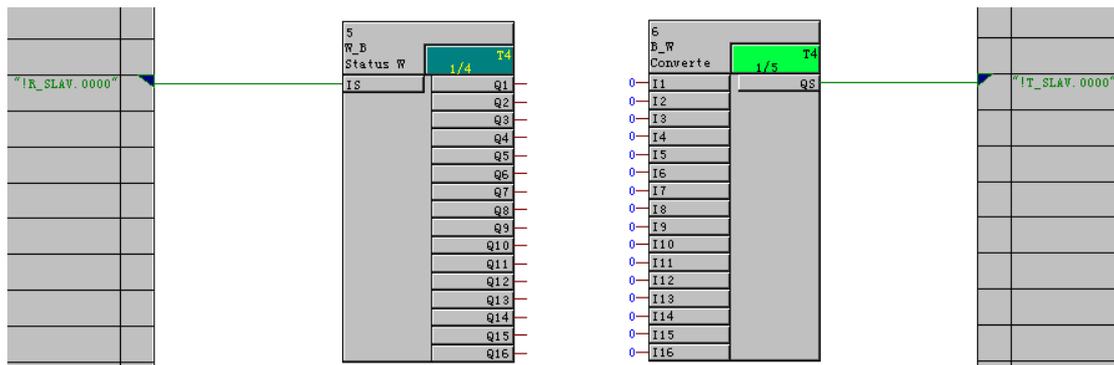
AT—设置通道名称为 R\_Slav.5.1

MOD—定义为 R，刷新 Refresh 模式

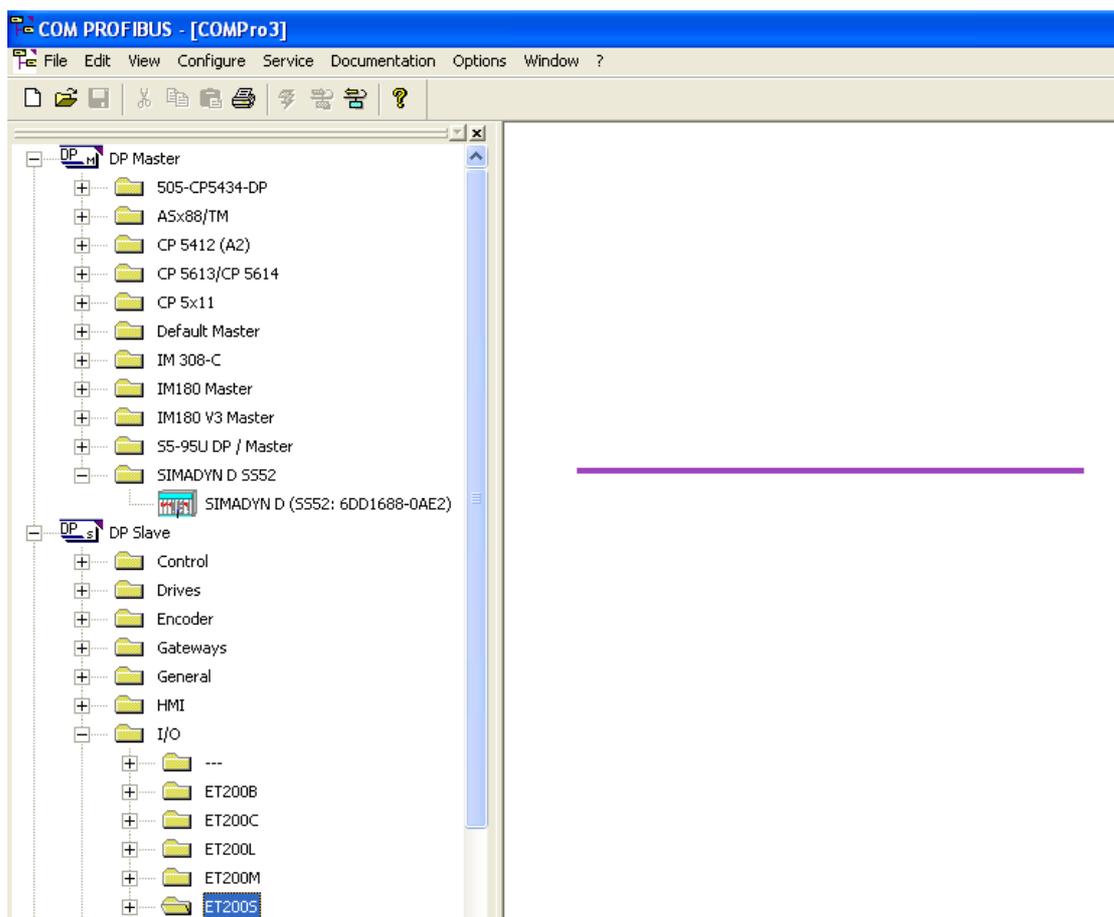
CRT—定义虚拟连接名称为!R\_SLAV

PROFIBUS 通讯设置该参数 AT/AR，由“通道名.地址 1.地址 2”组成。通道名最多有 8 个字符组成，且用同一接口通讯数据的通道名必须不同。地址 1 为从站的 PROFIBUS 地址，范围为 3~123，如果设置为 0，那么该 CP50MO 为从站。地址 2 可以设置 0 或 1。若设置 0 即 Intel 方式，即不必交换高低字节处理可以进行通讯，主要和 TDC 或 SIMADYN D 通讯。设置 1 即 Motorola 方式，即需要交换高低字节处理进行通讯，主要与分布式 I/O，Drives 等通讯。

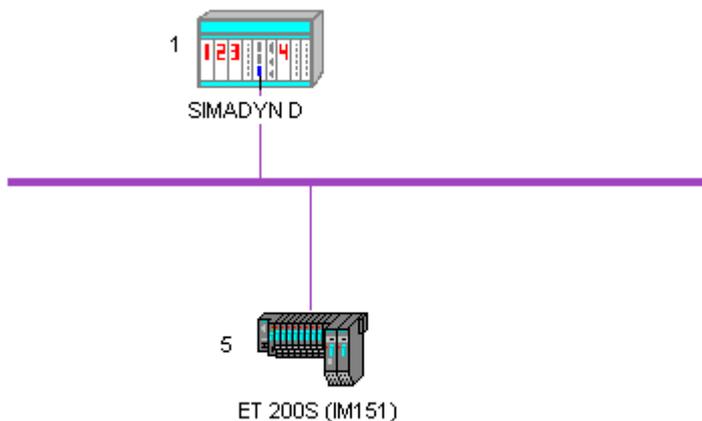
- 插入 B\_W 功能块用于写入从站输出数据和 W\_B 用于读入从站输入数据。



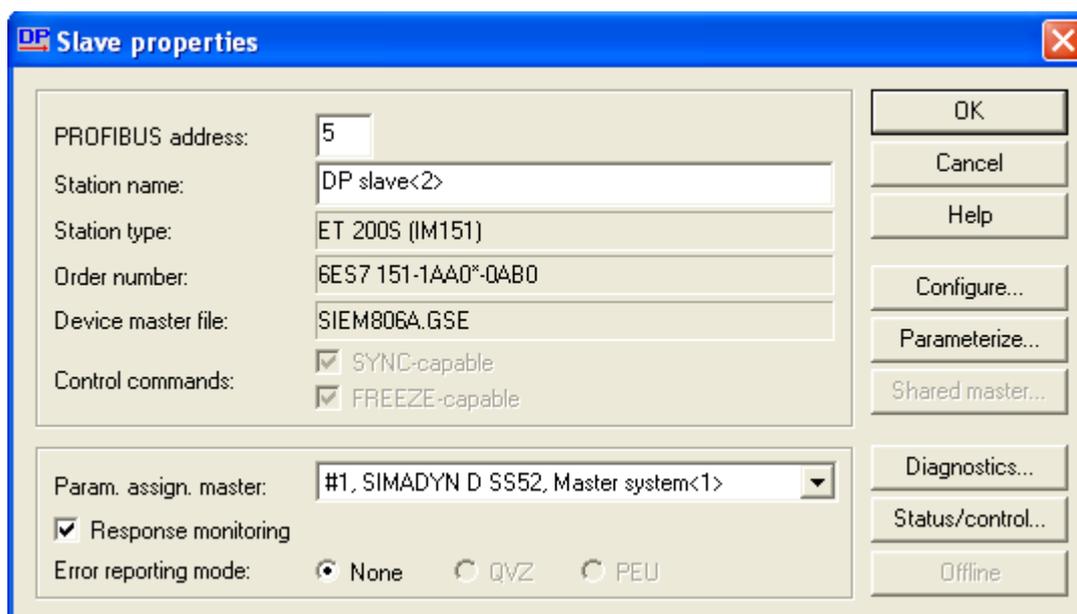
- 编译 CFC 程序，设置 PG/PC 接口为 CP5512(MPI)下载程序和组态的硬件组态信息到 TDC 中。因为@PRODP 的参数 SLA 设为 0，需要 COM PROFIBUS 组态参数。下载后 CPU1 会有故障字“C”闪烁，表示需要把 COM PROFIBUS 软件的组态的网络信息下载到 TDC 中。
- 打开 COM PROFIBUS 软件，左侧栏为 DP 主站和 DP 从站列表。右侧栏为待组态的 DP 网络。



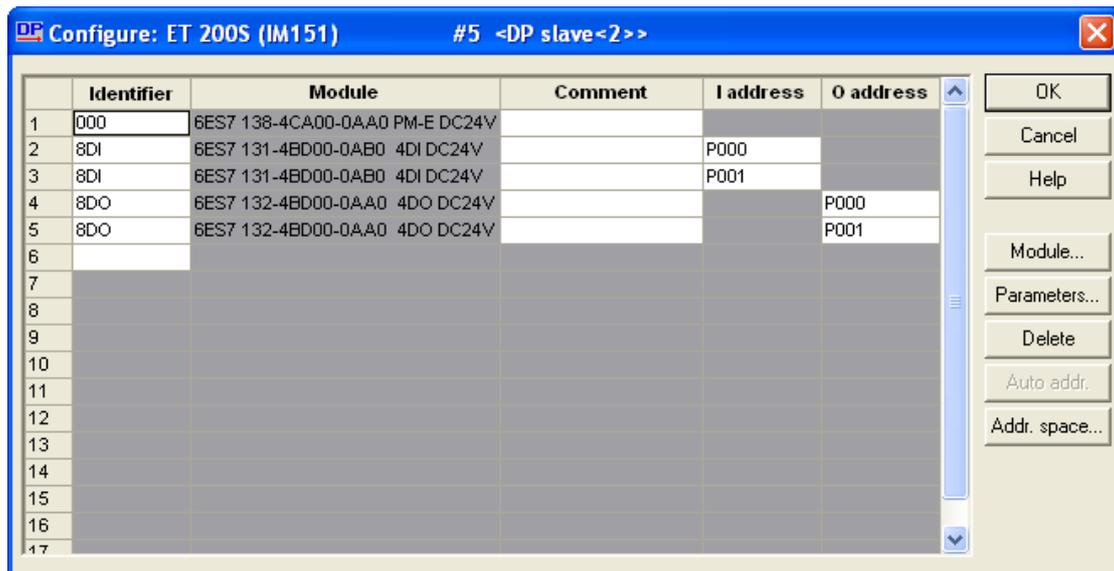
- 从左侧栏 DP Master 内选中 SIMADYN D 到右侧栏，默认的 DP 地址为 1。根据 ET200S 的接口模板 IM 型号从左侧栏 DP Slave 内选择 ET200S(IM151/CPU)到右侧栏。如果没有该模板可以参照附录 A 安装该模板的.gsd 文件。



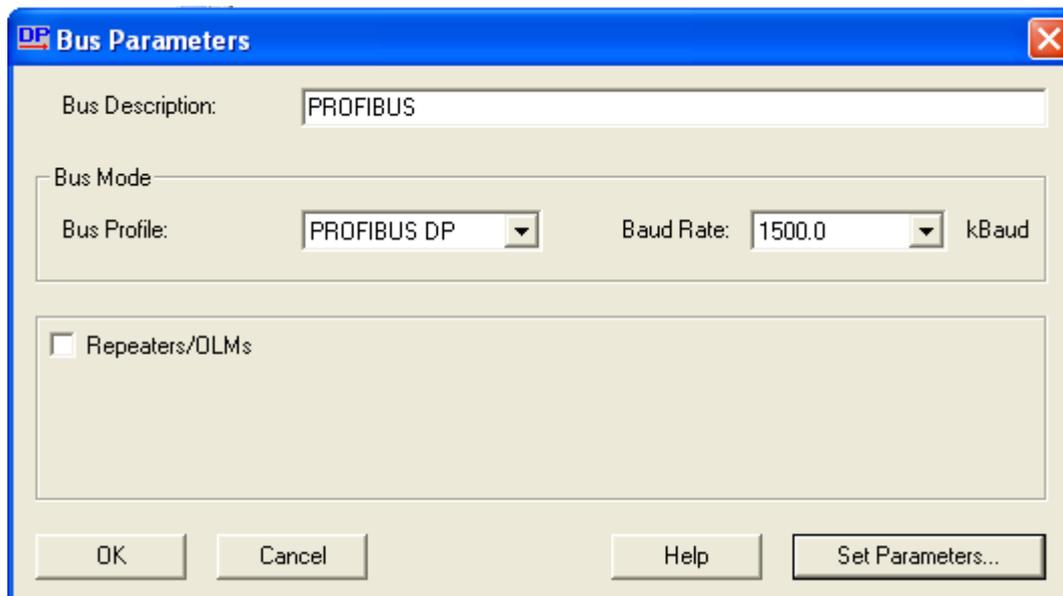
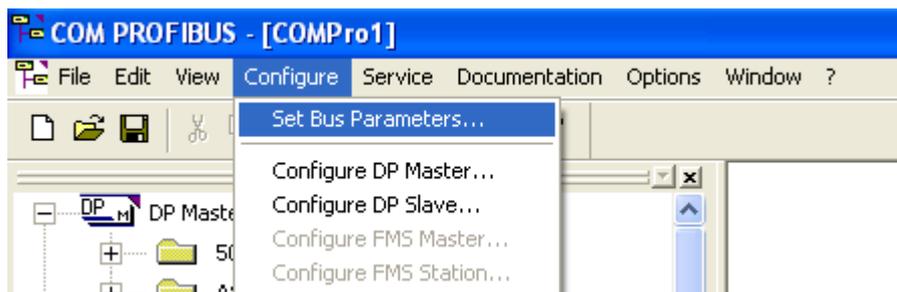
- 双击 ET200S 图标，组态该 DP 从站。首先修改 ET200S 的地址为 5。



- 点击 Configure..., 组态该 ET200S 站的 I/O 模板。点击 Auto addr. 自动分配各个 I/O 模板地址。其它设置为默认，点击 OK 关闭。

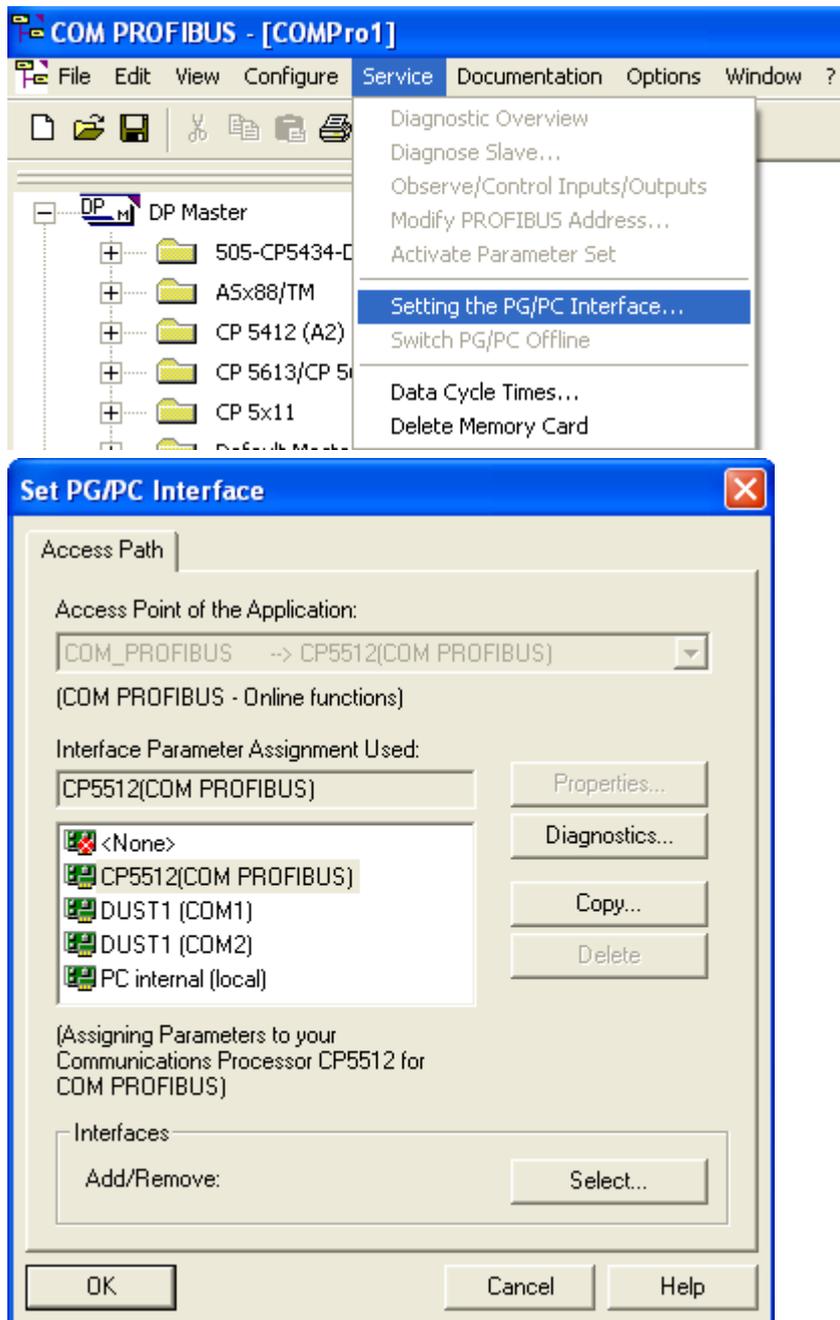


- 在菜单项选择 Configure 中的 Set Bus Parameters..., 通过 Bus Parameters 来设置 PROFIBUS 的网络参数, 本例中保持默认参数。然后保存该文件。

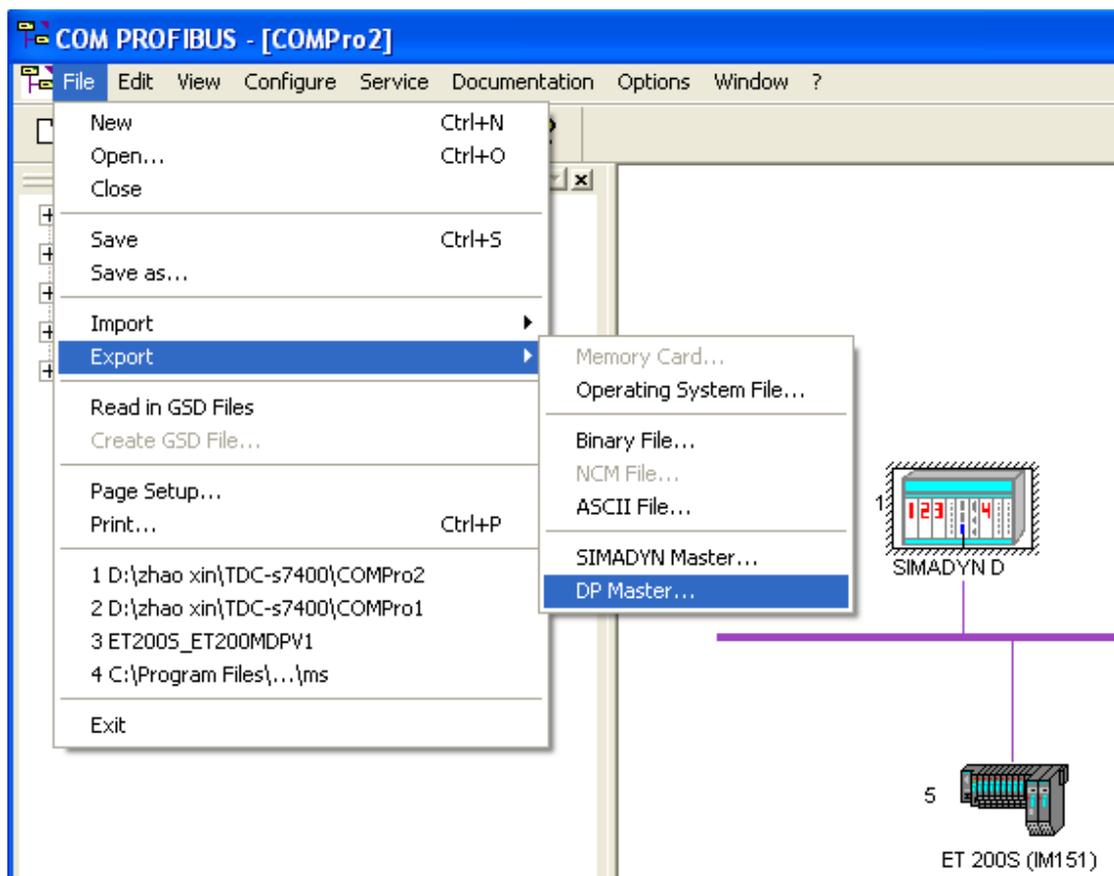


- 在菜单项选择 Service 中的 Setting the PG/PC Interface..., 来设置 COM

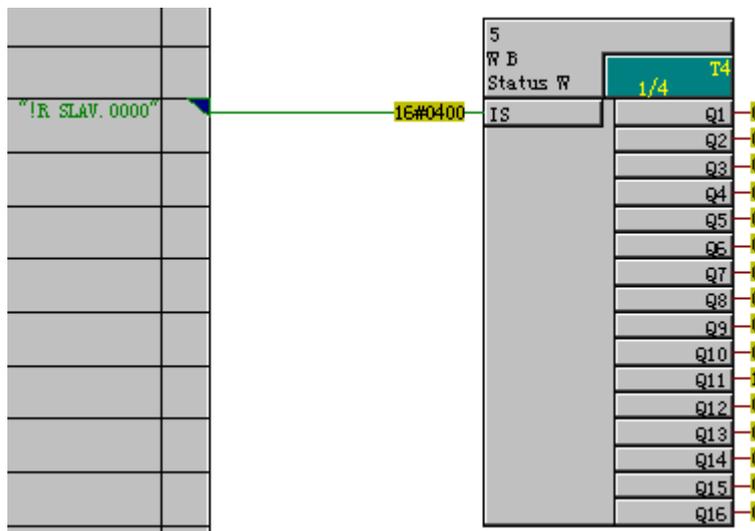
PROFIBUS 的下载参数。

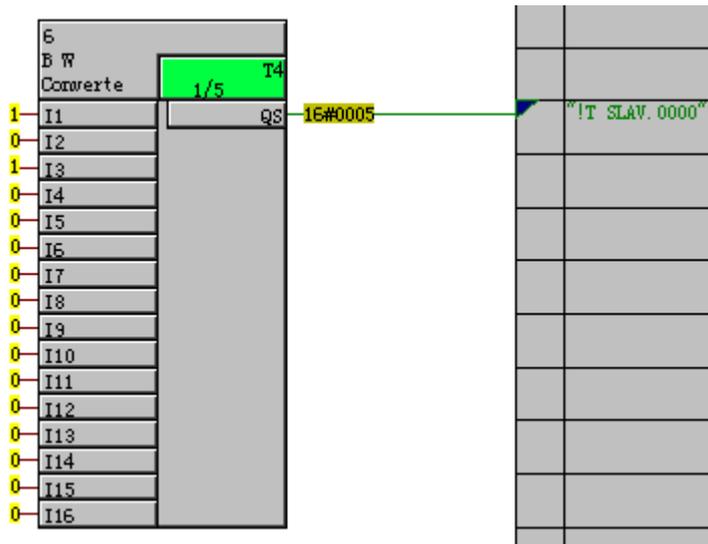


- 首先点击 DP 主站即 SIMADYN D 出现黑色外框。在菜单项选择 File 的 Export, 点击 DP Master 的下载 COM PROFIBUS 的组态参数。需要把下载电缆连接到 CP50MO 的 X2 接口, 因为已经设定该接口为 PROFIBUS 接口。

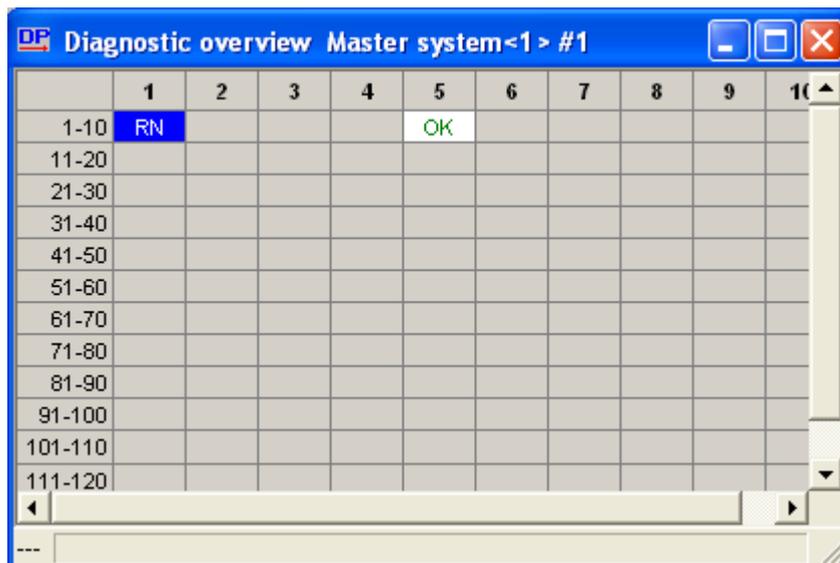


- 下载完成后，在线更改 **B\_W** 功能块的数值，测试输出模板；在线观察 **W\_B** 功能块的显示数值。

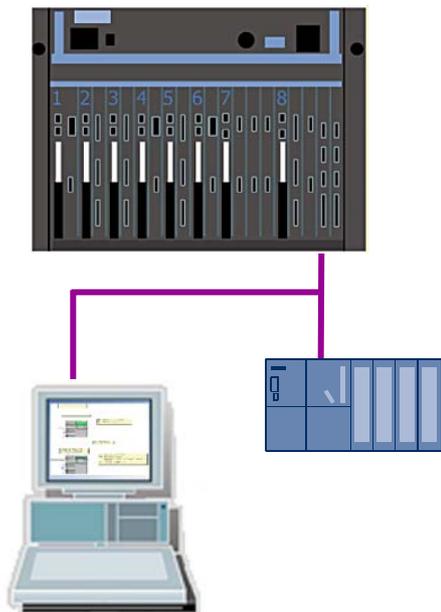




- 下载完成后，可以通过 COM PROFIBUS 软件简单诊断网络状态。点击 DP 主站，然后选择菜单 Service，然后选择 Diagnostic Overview。



#### 4.5.2 CP50MO 作为从站

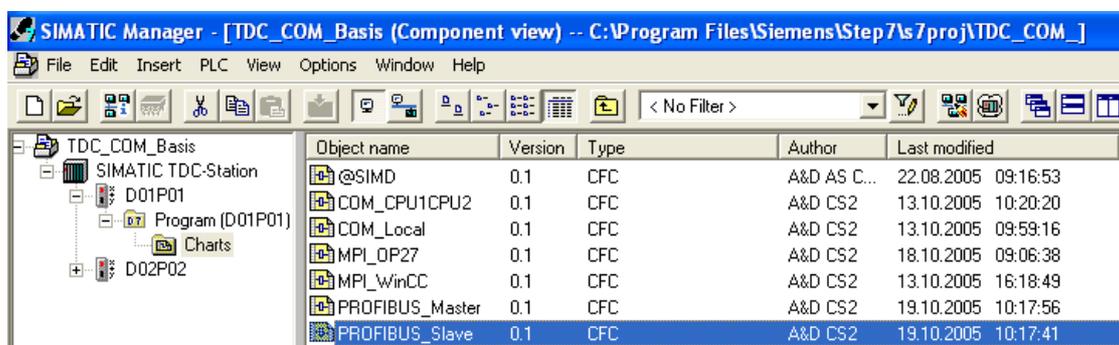


##### 4.5.2.1 在 SIMATIC Manager 中进行硬件组态

- 硬件组态与 4.5.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为 MPI 接口。修改该模板的 X2 接口属性为 PROFIBUS。
- 计算机 PG 使用标准的 MPI 电缆和 CP5512MPI 网卡，用于下载 CFC 程序和诊断 TDC。也可以下载 Step7 程序到 PLC 中。

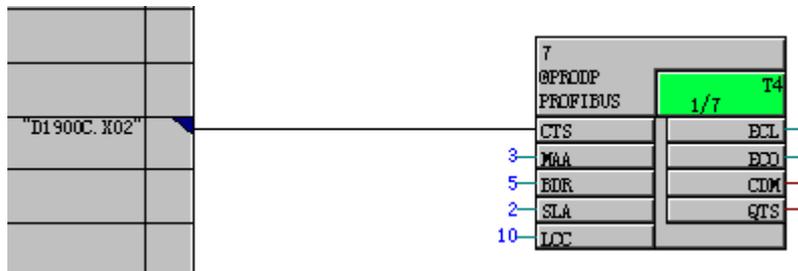
##### 4.5.2.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 PROFIBUS\_Slave。

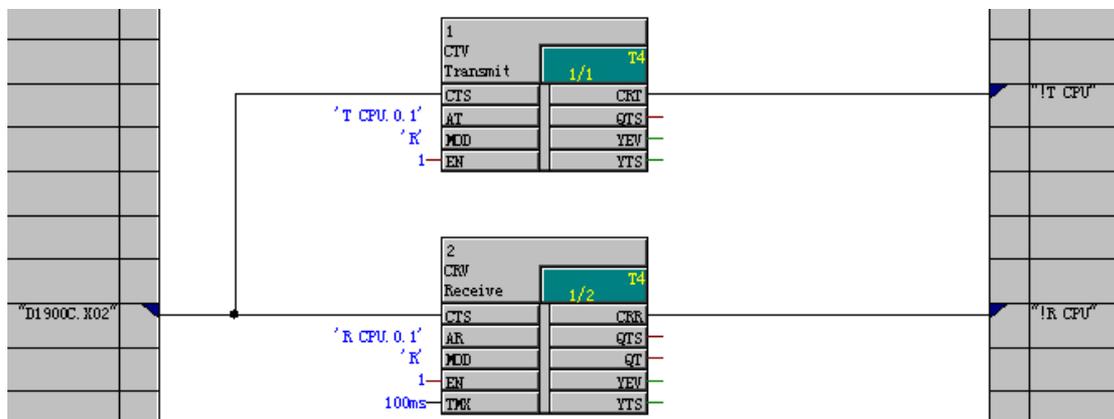


- 插入功能块@PRODP 功能块。设置参数 SLA 为 2, 表示 CP50MO 仅作为 DP 从站，

主站可以对其读写。修改 MAA 参数为 3，表示作为从站 CP50MO 的地址。具体参数信息参照 4.5.1.2。



- 分别插入 CTV 和 CRV 功能模块，用于与站地址为 2 的 DP 主站 CPU314C-2DP 通讯，接收和发送数据。



CTV 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 T\_CPU.0.1。

MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!T\_CPU

CRV 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

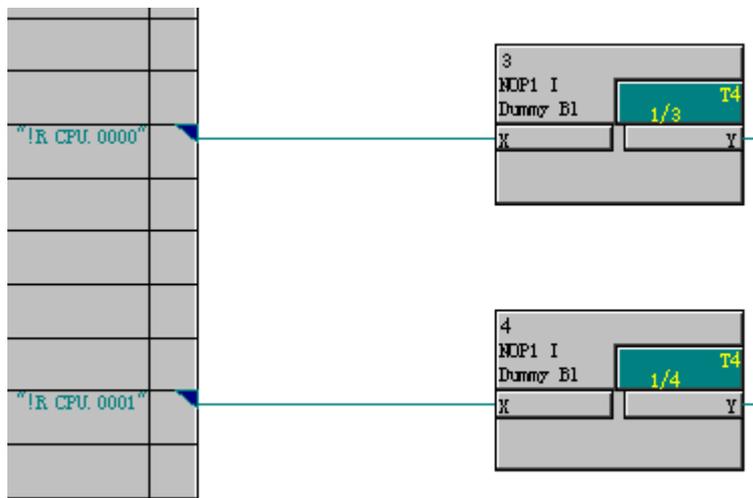
AT—设置通道名称为 R\_CPU.0.1

MOD—定义为 R，刷新 Refresh 模式

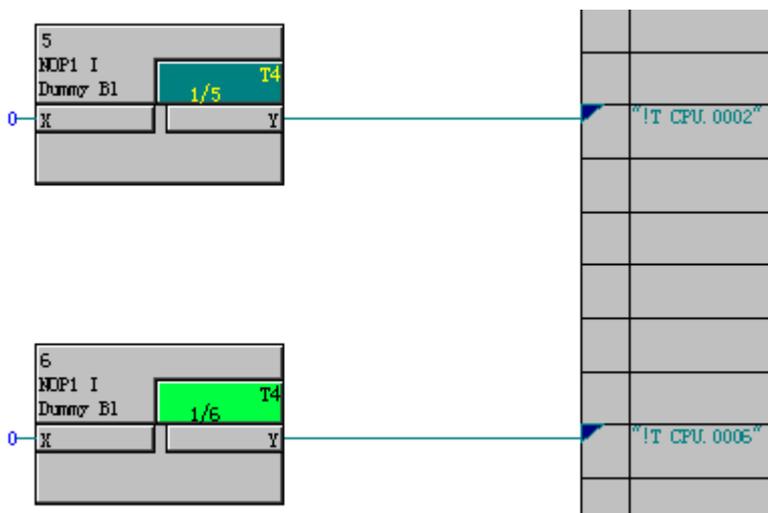
CRT—定义虚拟连接名称为!R\_CPU

PROFIBUS 通讯设置该参数 AT/AR 设置的具体信息参照 4.6.1.2。

- 插入 2 个 NOP1\_I 作为接收数据的功能块。设定功能块 3 变量 X 的虚拟连接名为 R\_CPU, 序号为 0000。设定功能块 4 变量 X 的虚拟连接名为 R\_CPU, 序号为 0001。功能块 3, 4 表示从 CPU 中接收 2 个整型数据, 且根据序号的先后顺序接收 CPU 发出的数据。即 0000 接收第一个整型数据, 0001 接收第二个整型数据。

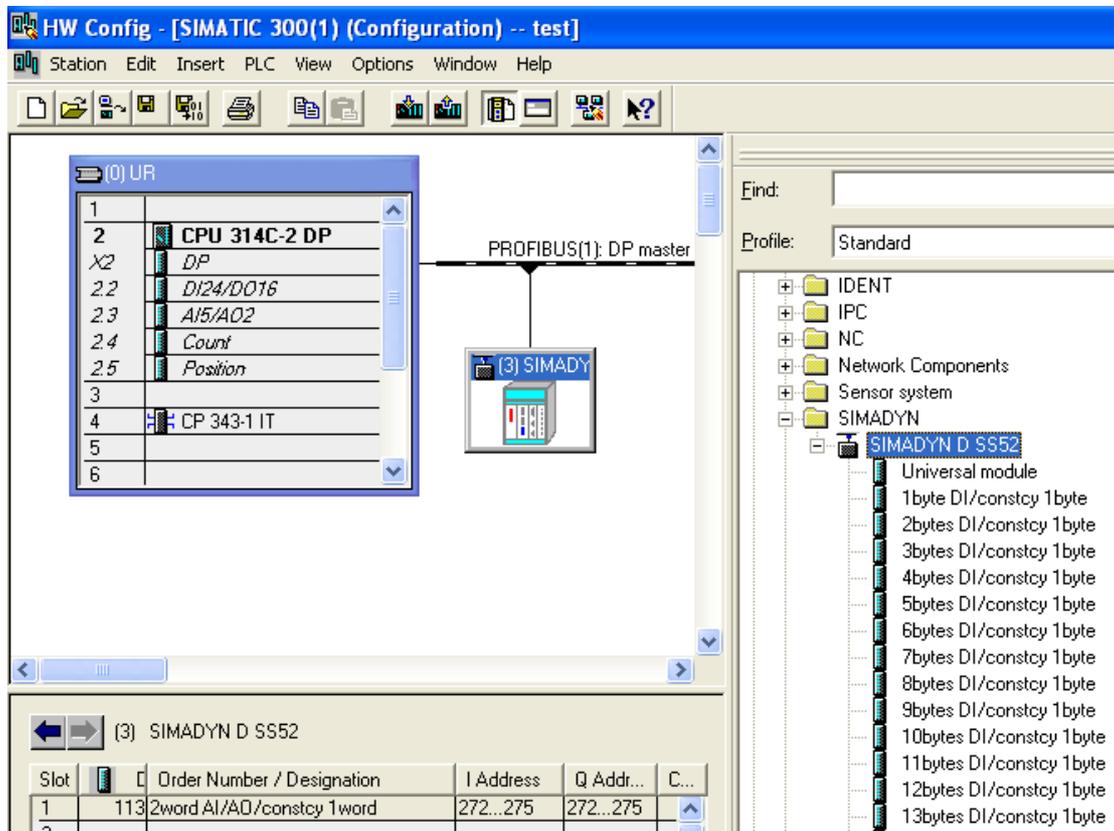


- 插入 2 个 NOP1\_I 作为发送数据的功能块。设定功能块 5 变量 Y 的虚拟连接名为 T\_CPU, 序号为 0002。设定功能块 6 变量 Y 的虚拟连接名为 T\_CPU, 序号为 0006。功能块 5, 6 表示向 CPU 发送 2 个整型数据, 且根据序号的先后顺序接收 CPU 发出的数据。即 0002 发送第一个整型数据, 0006 发送第二个整型数据。

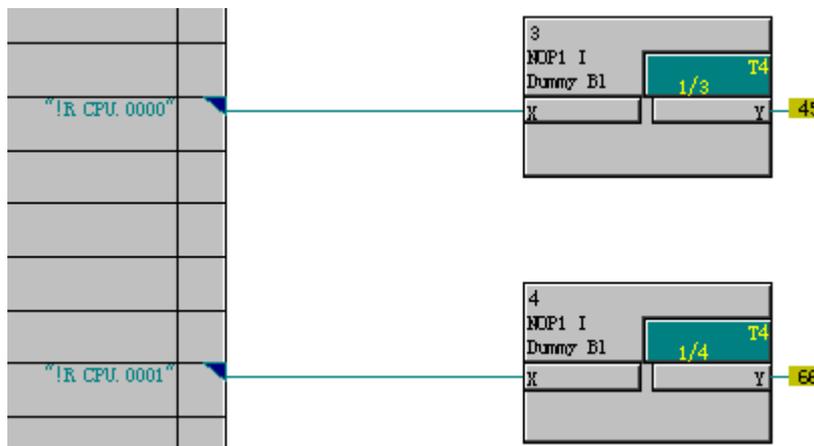


- 编译 CFC 程序, 设置 PG/PC 接口为 CP5512(MPI)下载程序到 TDC 中。需要注意的是需要删除 4.6.1 的程序 PROFIBUS\_Master。

- 在 Step7 中, 新建一个项目, 插入一个 S7-300 站作为 DP 主站, 再插入一个 SIMADYN 作为 DP 从站. 对其进行硬件组态. 设置 CPU314 的 DP 站地址为 2, 设置 SIAMDYN 的 DP 站地址为 3. 组态 2 个字输入/输出作为数据接口区。

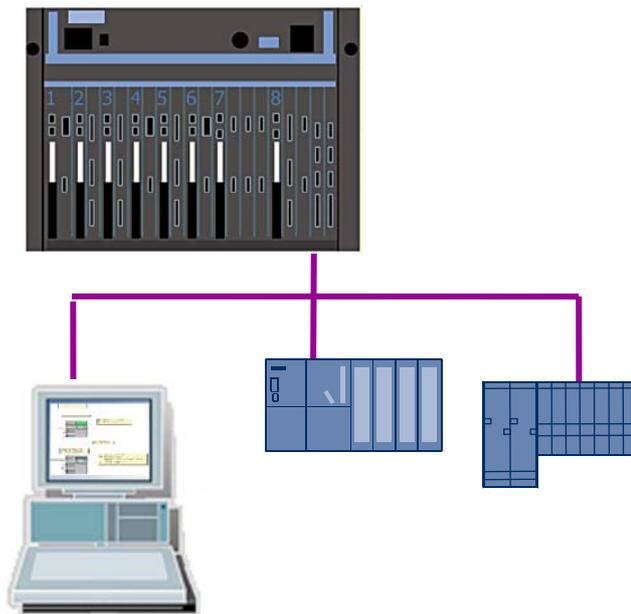


- 下载 Step7 程序, 在 Step7 中强制 PQW272 和 PQW274 分别为 45, 46. 在 CFC 中分别在线修改功能块的输入 X 值为 88, 99, 在 Step7 的 PIW272 和 PIW274 中显示。



VAT_1 -- @test\SIMATIC 300(1)\CPU 314C-2 DP\S7 Program(1) ONLINE					
	Address	Symbol	Display	Status value	Modify value
1	PIW 272		DEC	88	
2	PIW 274		DEC	99	
3	PQW 272		DEC	88	45
4	PQW 274		DEC	88	66
5					

#### 4.5.3 CP50MO 同时作为主站和从站

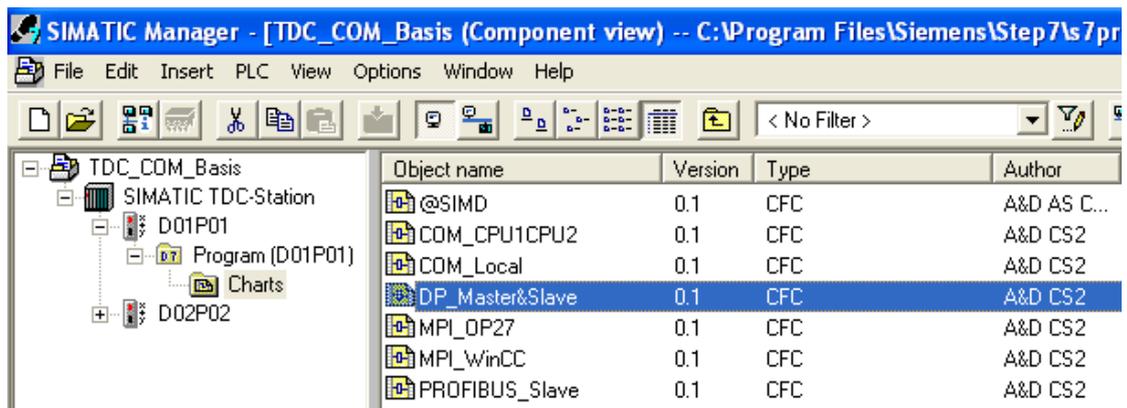


##### 4.5.3.1 在 SIMATIC Manager 中进行硬件组态

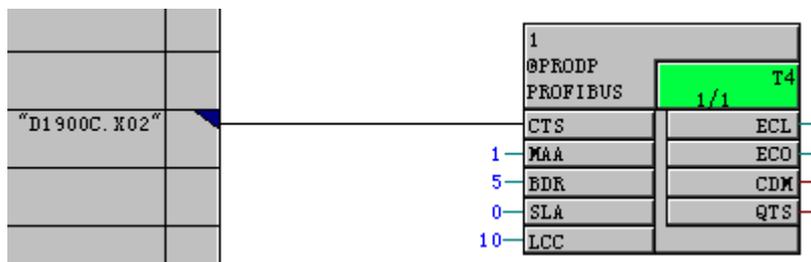
- 硬件组态与 4.5.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为 MPI 接口。修改该模板的 X2 接口属性为 PROFIBUS。
- 计算机 PG 使用标准的 MPI 电缆和 CP5512MPI 网卡, 用于下载 CFC 程序和诊断 TDC。同时也可以下载 Step7 程序, 调试和诊断 PLC。
- 本例中 TDC 是 PLC 的 DP 从站, 同时 TDC 是 ET200S 的 DP 主站。

##### 4.5.3.2 在 SIMATIC Manager 中进行软件组态

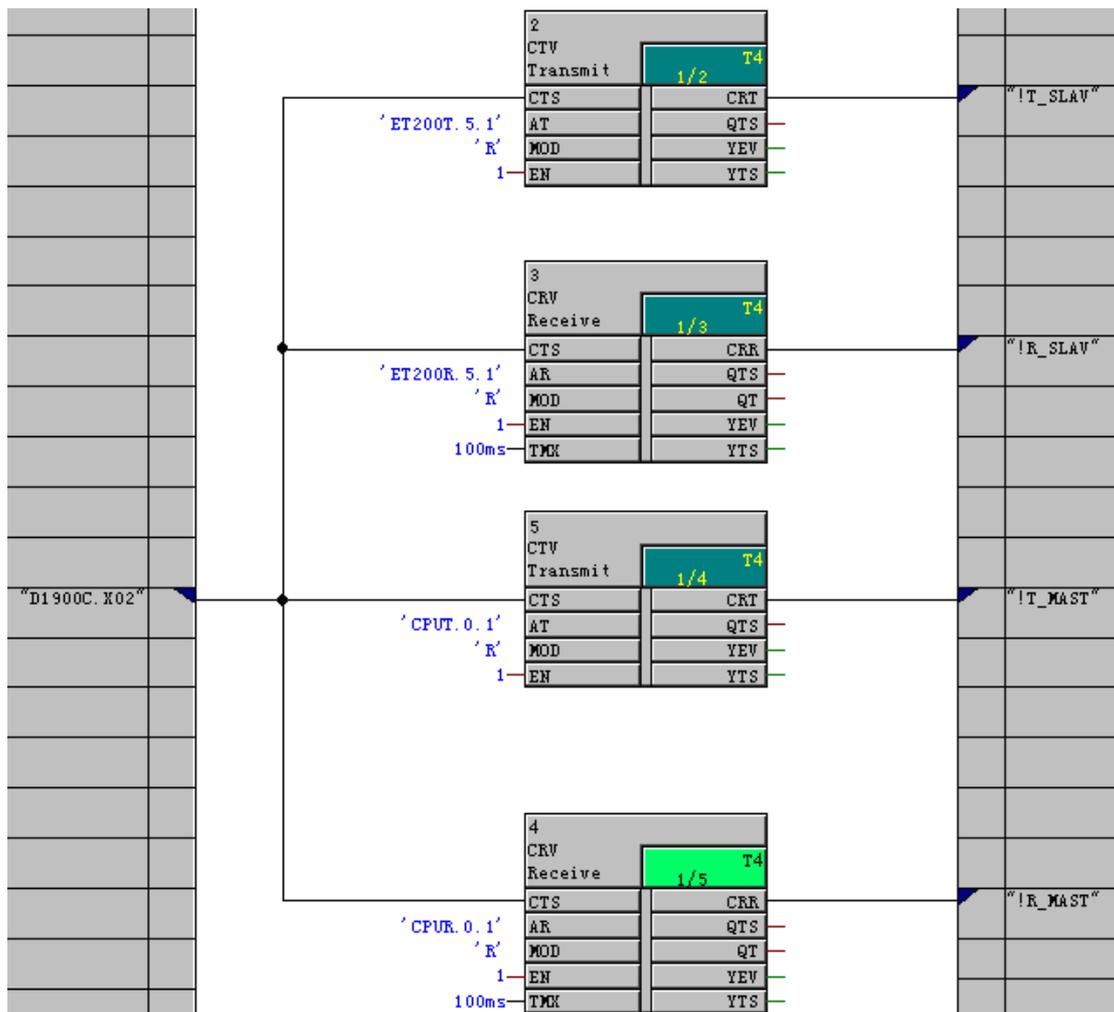
- 在 CPU1:D01P01 中, 在插入新的 CFC 程序, 命名为 DP\_Master&Slave, 并通过双击打开。



- 插入@PRODP 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C.X02。即利用 CP50MO 的 X2 作为 PROFIBUS 通讯的 DP 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。



- 插入 CTV 和 CRV 功能模块，分别用于与站地址为 5 的 DP 从站 ET200S 接收和发送数据。用于与站地址为 2 的 DP 主站接收和发送数据。



CTV 2 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 ET200T.5.1。

MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!T\_SLAV

CRV 3 功能块的引脚说明：

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 ET200R.5.1

MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!R\_SLAV

CTV 5 功能块的引脚说明:

CTS—用于连接通讯的硬件接口, 模板地址 D1900C.X02

AT—设置通道名称为 CPUT.0.1。

MOD—定义为 R, 刷新 Refresh 模式

CRT—定义虚拟连接名称为!T\_MAST

CRV 4 功能块的引脚说明:

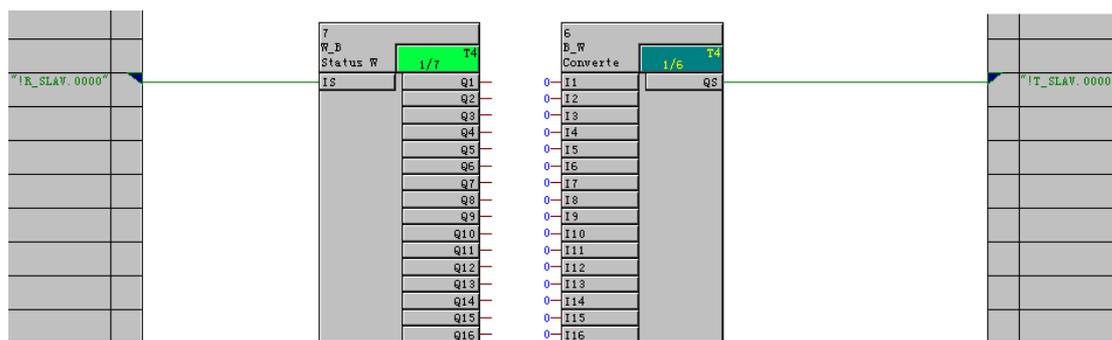
CTS—用于连接通讯的硬件接口, 模板地址 D1900C.X02

AT—设置通道名称为 CPUR.0.1

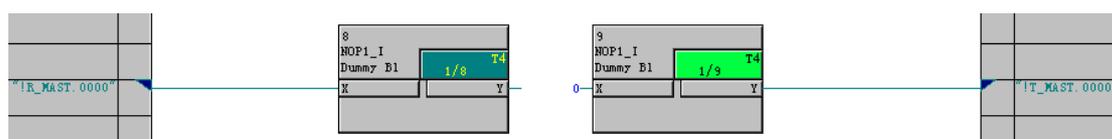
MOD—定义为 R, 刷新 Refresh 模式

CRT—定义虚拟连接名称为!R\_MAST

- 插入 B\_W 功能块用于写入从站输出数据和 W\_B 用于读入 DP 从站输入数据。

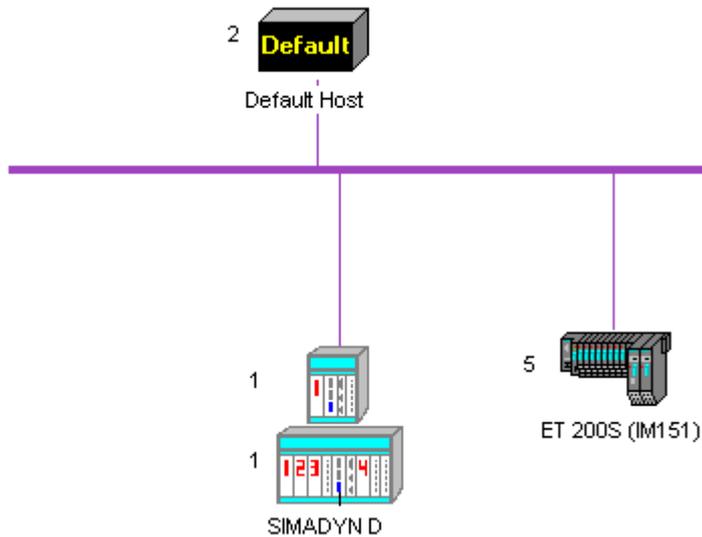


- 插入 NOP1\_I 功能块用于写入 DP 主站输出数据和用于读入 DP 主站输入数据。



- 编译 CFC 程序, 设置 PG/PC 接口为 CP5512(MPI)下载程序和组态的硬件组态信息到 TDC 中。因为@PRODP 的参数 SLA 设为 0, 需要 COM PROFIBUS 组态参数。下载后 CPU1 会有故障字“C”闪烁, 表示需要把 COM PROFIBUS 软件的组态的网络信息下载到 TDC 中。注意需要把程序 PROFIBUS\_Master 和 PROFIBUS\_Slave 程序删除, 因为避免多个@PRODP 冲突。
- 打开 COM PROFIBUS 软件, 从左侧栏 DP Slave 内点击 Drives 选中 SIMADYN D

到右侧栏, 默认的 DP 地址为 1。根据 ET200S 的接口模板 IM 型号从左侧栏 DP Slave 内选择 ET200S(IM151/CPU)到右侧栏。如果没有该模板可以参照附录 A 安装该模板的.gsd 文件。再从左侧栏 DP Master 中选择 Default Host 到右侧栏, 默认 DP 地址为 2。



- 这里组态的 PROFIBUS 网络, Default Host 作为 PLC 为主站。SIMADYN D 为 2 个图标, 小图标是 DP 从站。大图标是 DP 主站, ET200S 是 DP 从站。双击 Default Host 图标。双击 SIMADYN D 的小图标, 参数化该从站到 Default Host 主站中。

**Master parameters**

PROFIBUS address: 2

Station name: Master system<5>

Station type: Default Master

Associated host: Host\_Master system<5>

Address mode: Linear

Defaults

Response monitoring for slaves

Error reporting mode:  None  QVZ  PEU

Buttons: OK, Cancel, Help, Host params., Add. space, Groups, Configuration

**Slave properties**

PROFIBUS address: 1

Station name: DP slave<1>

Station type: SIMADYN D SS52

Order number: 6DD1688-0AE2

Device master file: SIEM8037.GSD

Control commands:  SYNC-capable  FREEZE-capable

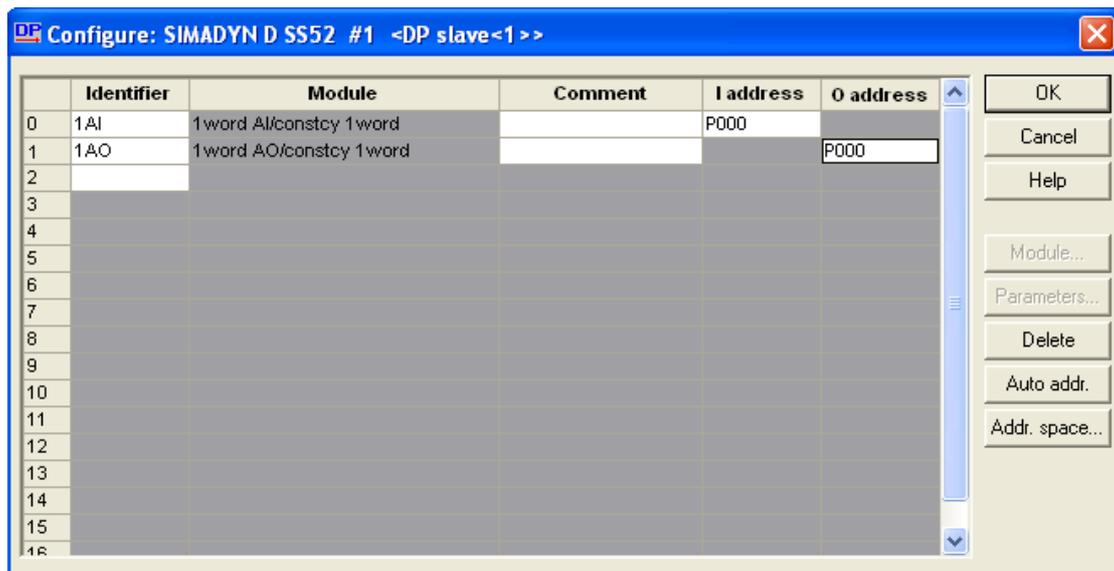
Param. assign. master: #2, Default Master, Master system<5>

Response monitoring

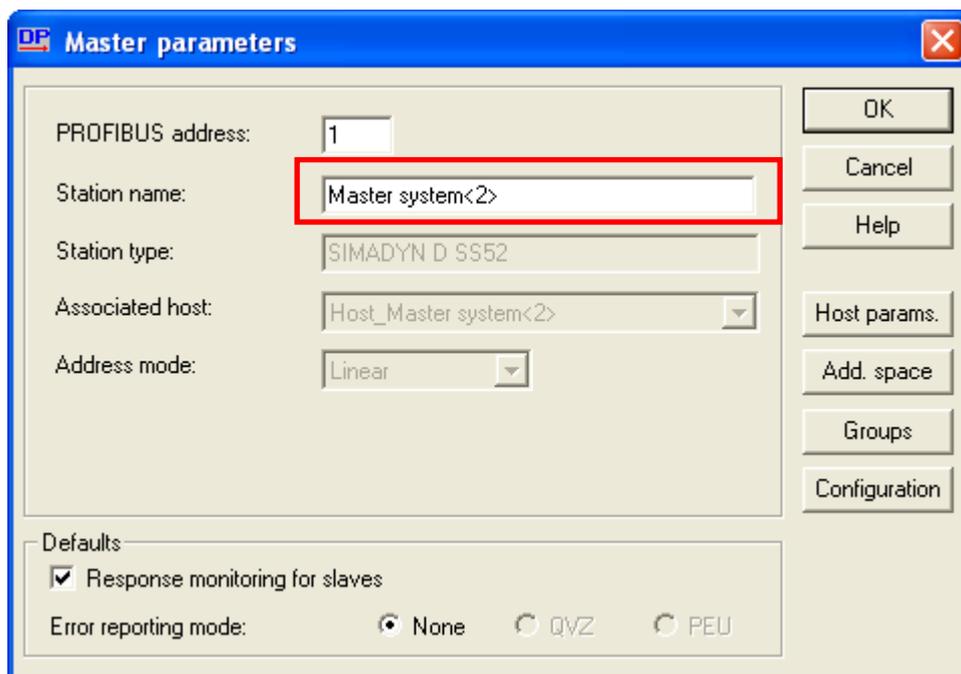
Error reporting mode:  None  QVZ  PEU

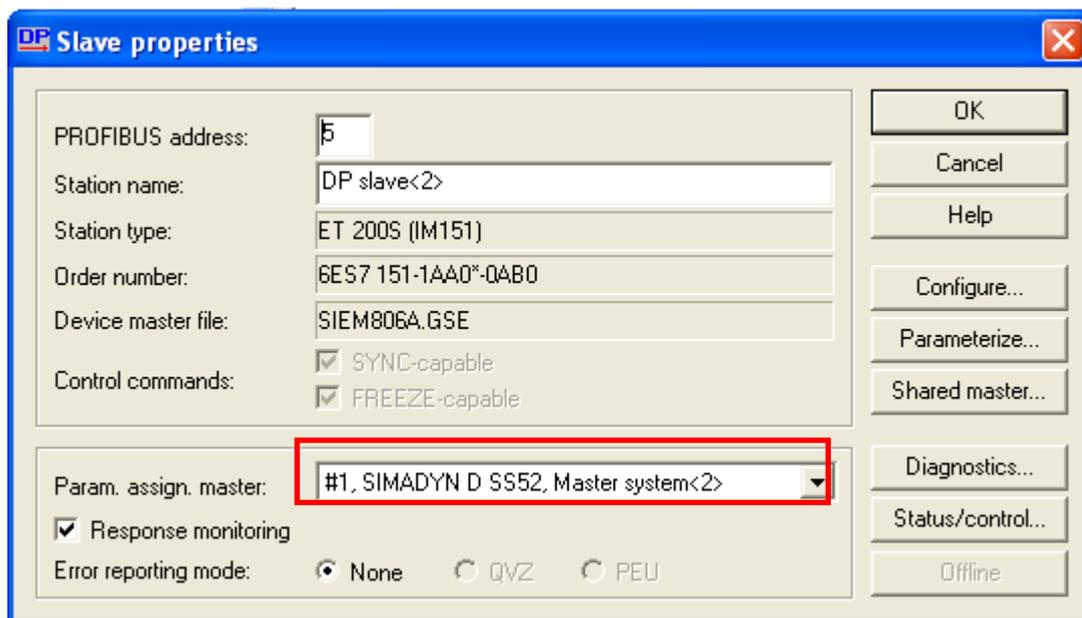
Buttons: OK, Cancel, Help, Configure..., Parameterize..., Shared master..., Diagnostics..., Status/control..., Offline

- 点击 **Configure...**，参照 CFC 中组态的 1 个整型输入/输出组态该 SIMADYN D 站的 I/O 通讯数据区。点击 **Auto addr.** 自动分配各个 I/O 数据区地址。其它设置为默认，点击 **OK** 关闭。

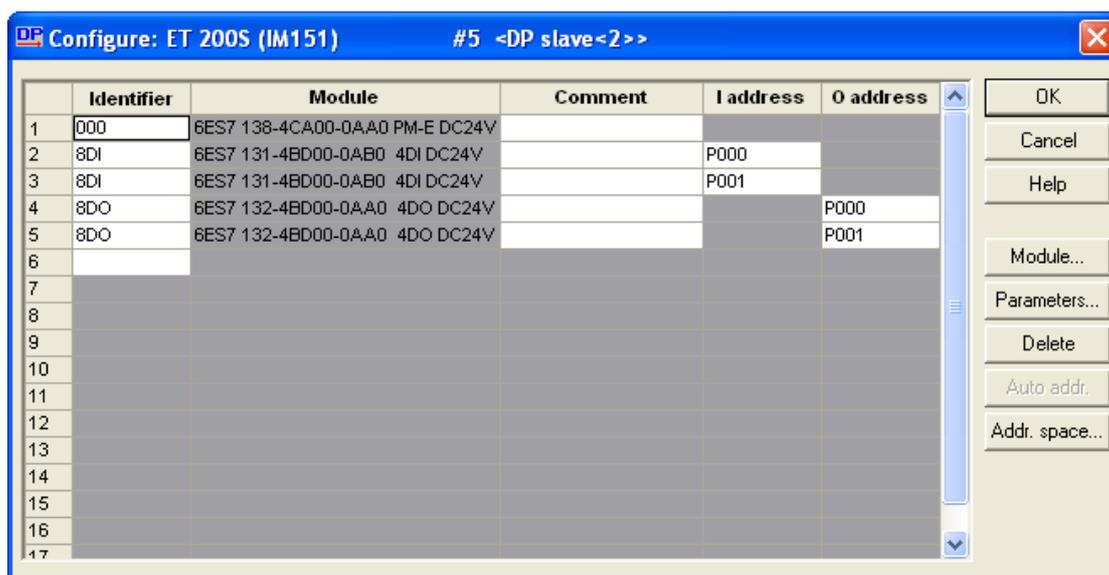


- 双击 SIMADYN D 的大图标。双击 ET200S 参数化该从站到 SIMADYN D 主站中。

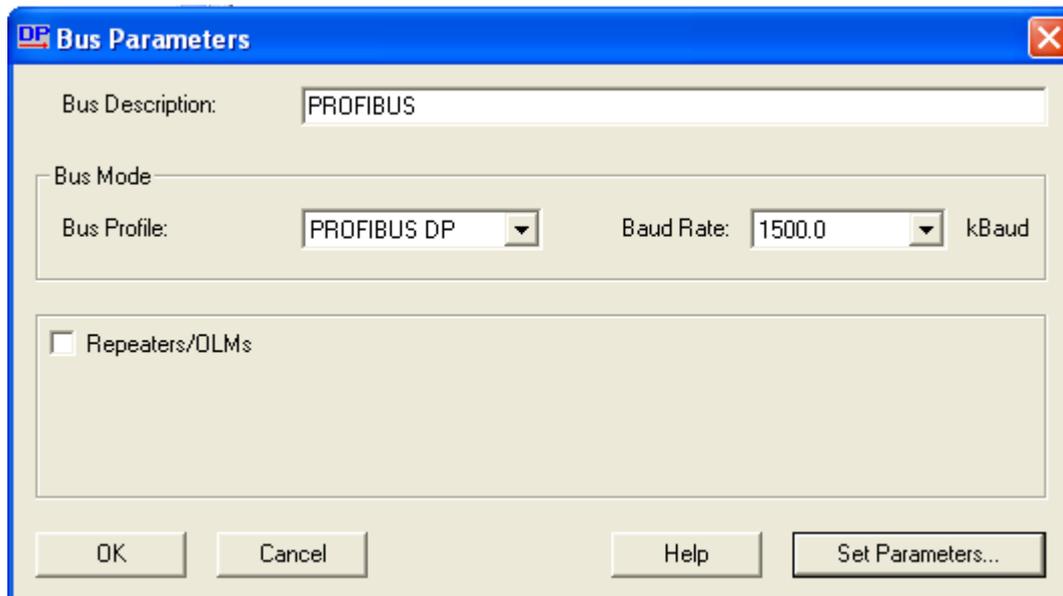




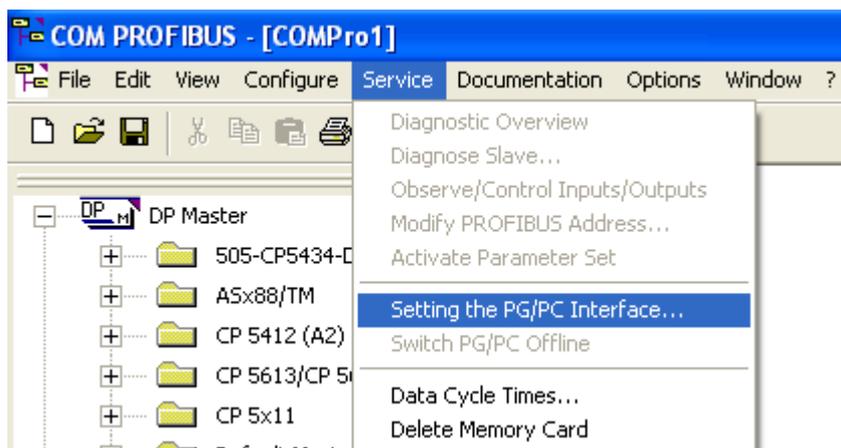
- 点击 Configure..., 组态该 ET200S 站的 I/O 模板。点击 Auto addr. 自动分配各个 I/O 模板地址。其它设置为默认, 点击 OK 关闭。

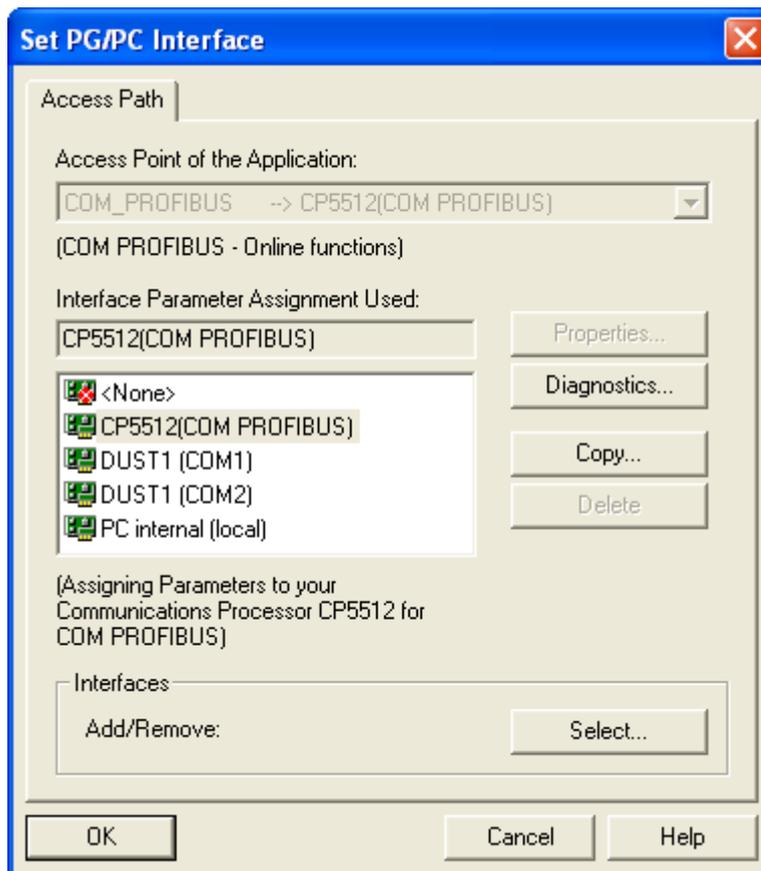


- 在菜单项选择 Configure 中的 Set Bus Parameters..., 通过 Bus Parameters 来设置 PROFIBUS 的网络参数, 本例中保持默认参数。然后保存该文件。



- 在菜单项选择 Service 中的 Setting the PG/PC Interface..., 来设置 COM PROFIBUS 的下载参数。





- 首先点击 DP 主站即 SIMADYN D 出现黑色外框。在菜单项选择 File 的 Export，点击 DP Master 的下载 COM PROFIBUS 的组态参数。需要把下载电缆连接到 CP50MO 的 X2 接口，因为已经设定该接口为 PROFIBUS 接口。具体下载方法请参照 4.5.2。
- 在 Step7 中，新建一个项目，插入一个 S7-300 站作为 DP 主站，再插入一个 SIMADYN 作为 DP 从站。对其进行硬件组态。设置 CPU314 的 DP 站地址为 2，设置 SIAMDYN 的 DP 站地址为 1。组态 1 个字输入/输出作为数据接口区。

The screenshot shows the HW Config software interface for a SIMATIC 300(1) (Configuration) -- test. The rack configuration is as follows:

Slot	Module
1	
2	CPU 314C-2 DP
X2	DP
2.2	DI24/DO16
2.3	AI5/AO2
2.4	Count
2.5	Position
3	
4	CP 343-1 IT
5	
6	

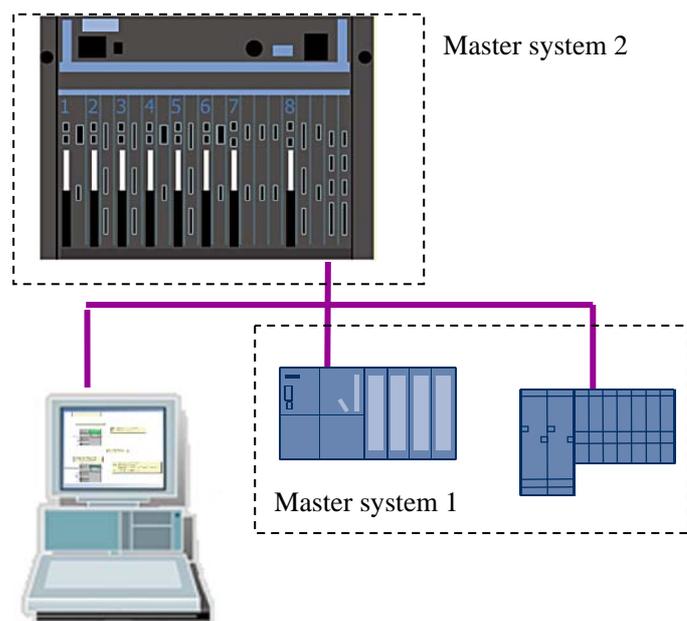
The network diagram shows a PROFIBUS(1): DP master system (1) connected to a SIMADYN D SS52 slave. The slave configuration is detailed in the table below:

Slot	D...	Order Number / Designation	I Address	Q Address	Comment
1	1AI	1word AI/constcy 1word	272...273		
2	1AO	1word AO/constcy 1word		272...273	
3					

- 下载完成后，可以通过 COM PROFIBUS 软件简单诊断网络状态。点击 DP 主站，然后选择菜单 Service，然后选择 Diagnostic Overview。

DP Diagnostic overview Master system<2> #1										
	1	2	3	4	5	6	7	8	9	10
1-10	RN	m			OK					
11-20										
21-30										
31-40										
41-50										
51-60										
61-70										
71-80										
81-90										
91-100										
101-110										
111-120										

#### 4.5.4 Shared Input 方式



在 PROFIBUS 网络中，S7-300PLC 作为 DP 主站，带有 ET200S 从站。同样 TDC 也作为 DP 主站。这样这个网络就可以分为两个主站系统 1 和 2。共享输入 shared input 方式就是主站系统 2 即 TDC 为 DP 主站可以直接读取主站系统 1 中 ET200S 从站的输入模板的状态。

##### 4.5.4.1 在 SIMATIC Manager 中进行硬件组态

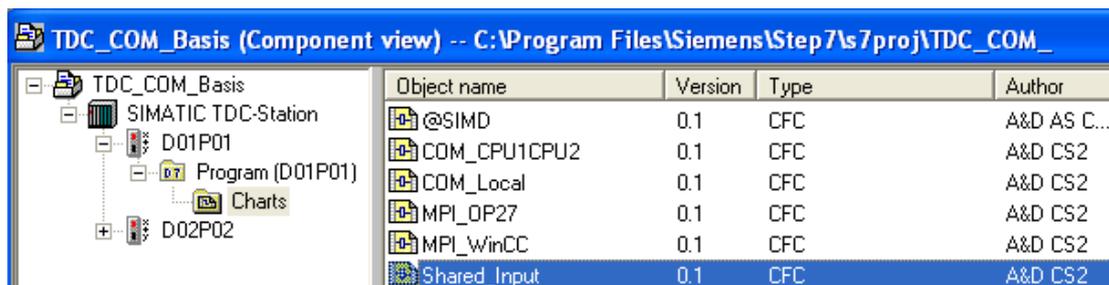
- 硬件组态与 4.5.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为

MPI 接口。修改该模板的 X2 接口属性为 PROFIBUS。

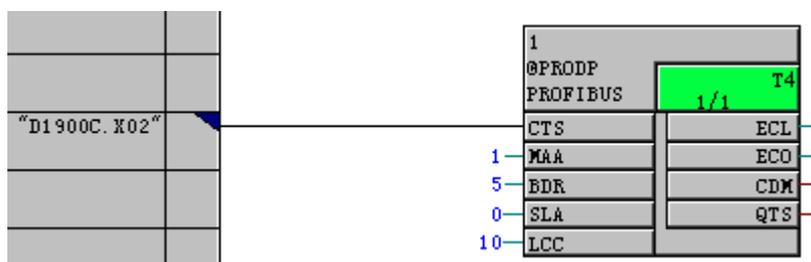
- 计算机 PG 使用标准的 MPI 电缆和 CP5512MPI 网卡，用于下载 CFC 程序和诊断 TDC。同时也可以下载 Step7 程序，调试和诊断 PLC。

#### 4.5.4.2 在 SIMATIC Manager 中进行软件组态

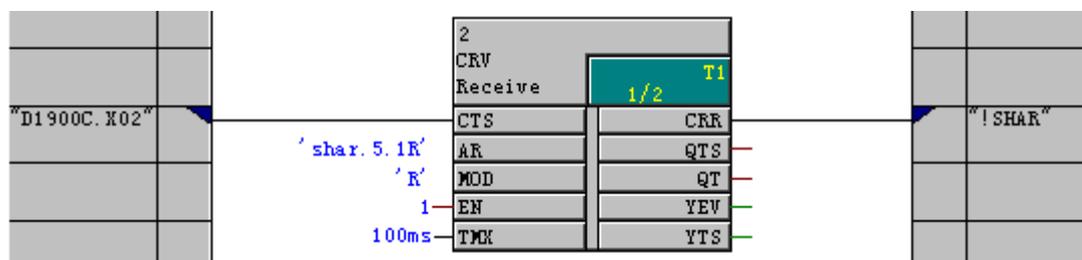
- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 Shared\_Input，并通过双击打开。



- 插入 @PRODP 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C.X02。即利用 CP50MO 的 X2 作为 PROFIBUS 通讯的 DP 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq \text{TA} \leq 256 \text{ ms}$ 。



- 插入 CRV 功能块，用于与另一个主站系统 1 中站地址为 5 的 DP 从站 ET200S 接收数据。



CRV 功能块的引脚说明：

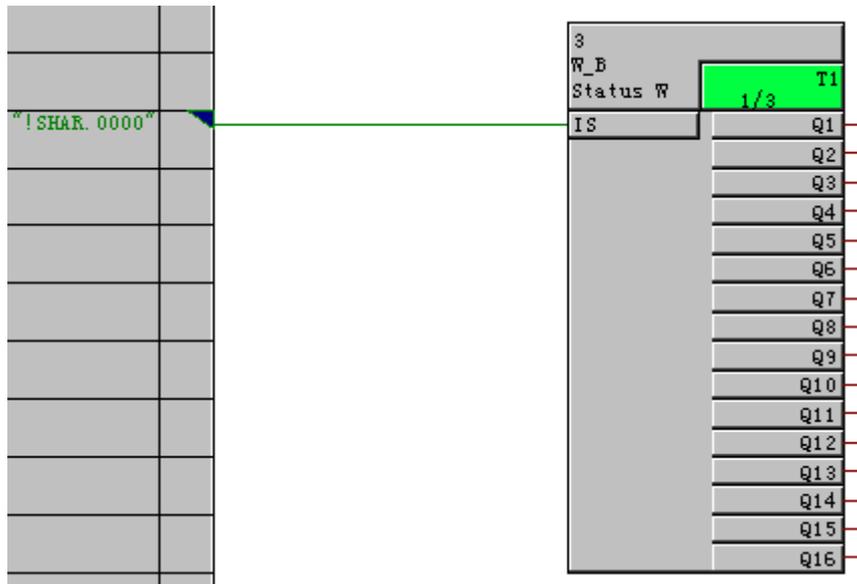
CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 shar.5.1R, R 表示 shared input 功能。

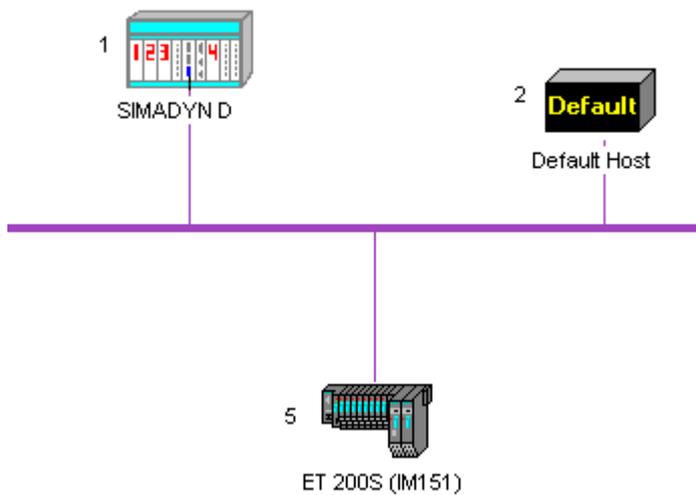
MOD—定义为 R, 刷新 Refresh 模式

CRT—定义虚拟连接名称为!SHAR

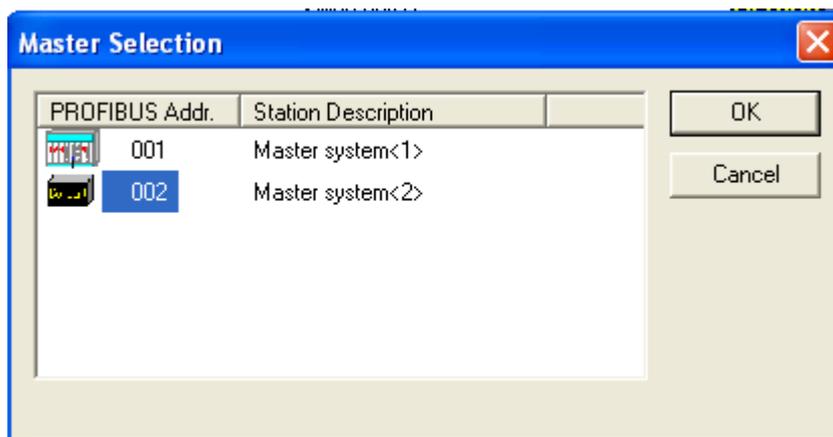
- 插入 W\_B 功能块用于读入 DP 从站 ET200S 输入数据。



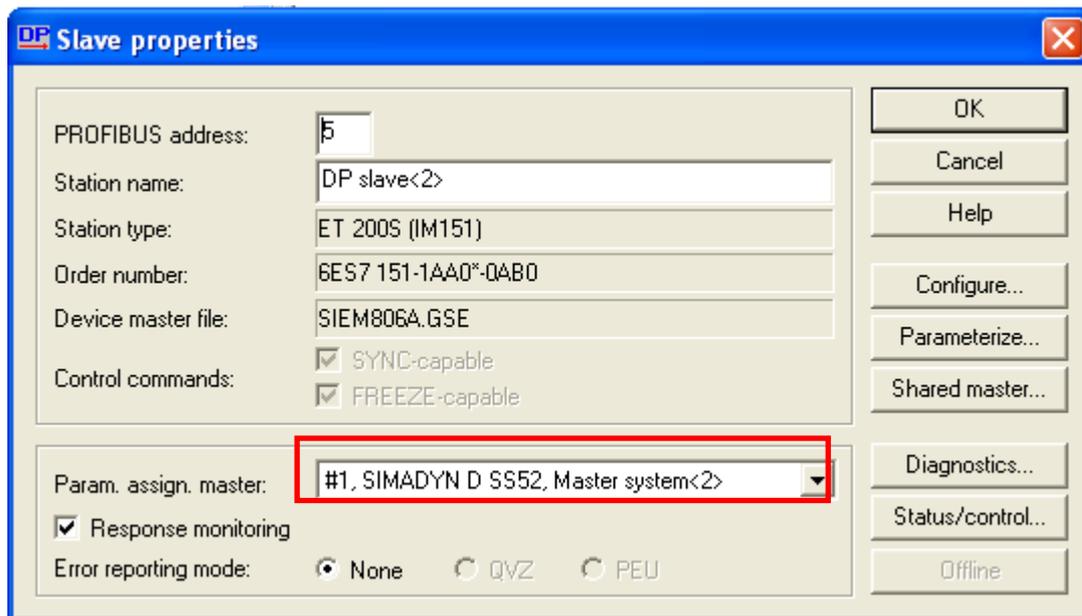
- 编译 CFC 程序, 设置 PG/PC 接口为 CP5512(MPI)下载程序和组态的硬件组态信息到 TDC 中。因为 `@PRODP` 的参数 `SLA` 设为 0, 需要 COM PROFIBUS 组态参数。下载后 CPU1 会有故障字“C”闪烁, 表示需要把 COM PROFIBUS 软件的组态的网络信息下载到 TDC 中。注意需要把程序关于 PROFIBUS 的程序删除, 因为避免多个 `@PRODP` 冲突。
- 打开 COM PROFIBUS 软件, 从左侧栏 DP Master 内选中 SIMADYN D 到右侧栏, 默认的 DP 地址为 1。再选择 Default Host 到右侧栏, 默认 DP 地址为 2。根据 ET200S 的接口模板 IM 型号从左侧栏 DP Slave 内选择 ET200S(IM151/CPU)到右侧栏, 设置 DP 的站地址为 5。



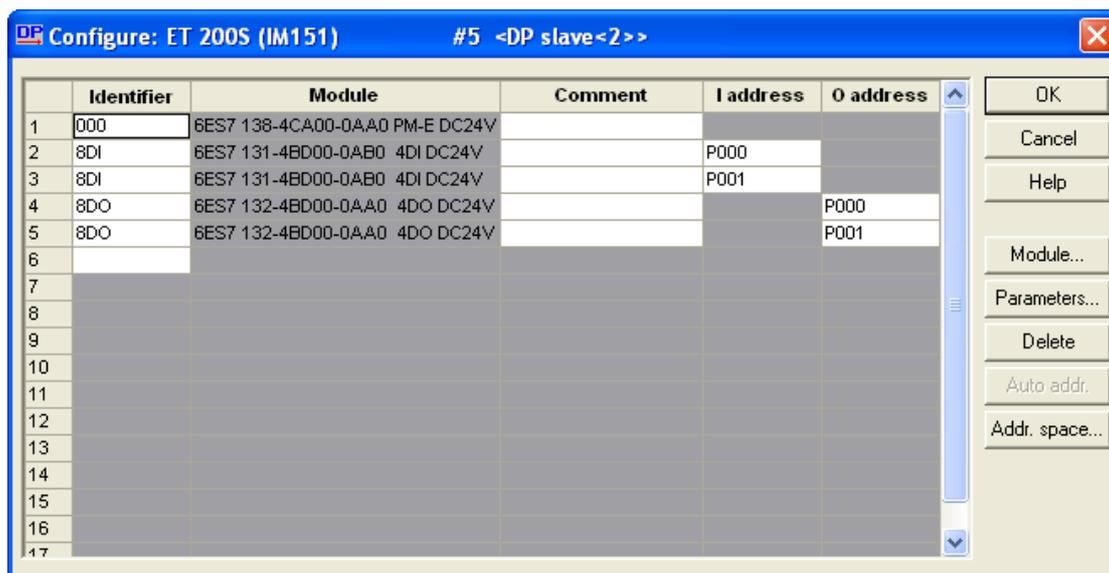
- 拖入 ET200S 时，需要选择参数化 DP 主站。选择 002，即 PLC 是 ET200S 的参数化主站。



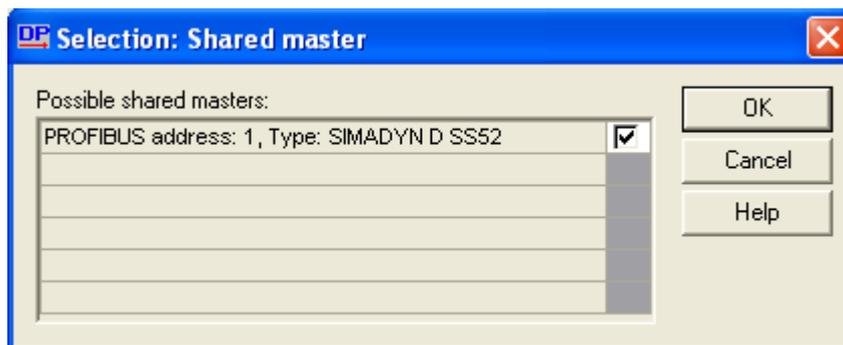
- 这里组态的 PROFIBUS 网络，Default Host 作为 PLC 为主站。ET200S 是该主站的 DP 从站。双击 ET200S 图标，修改其 DP 地址。



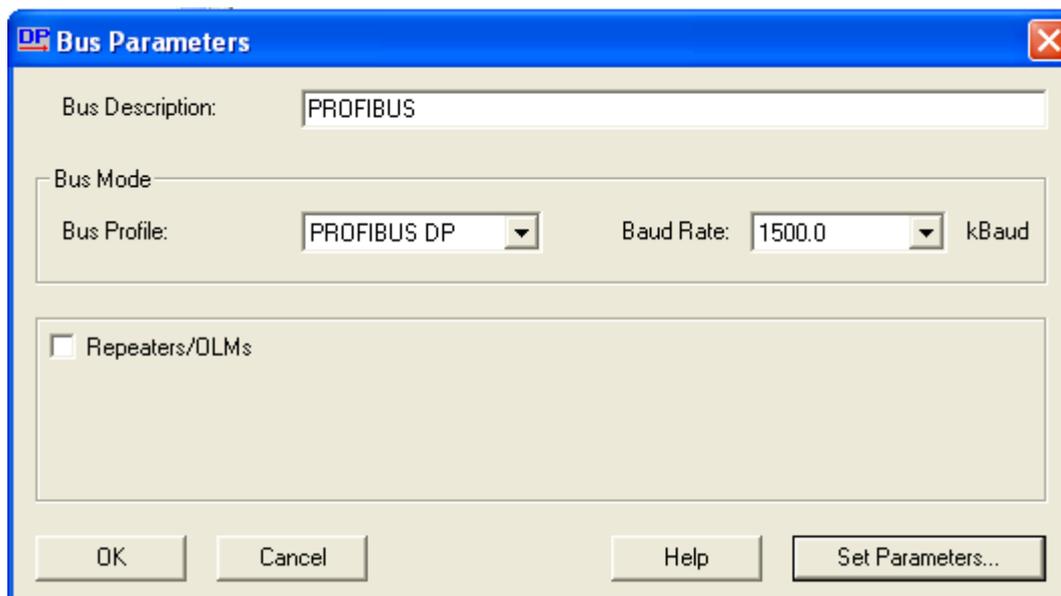
- 点击 Configure..., 组态该 ET200S 站的 I/O 模板。点击 Auto addr. 自动分配各个 I/O 模板地址。其它设置为默认, 点击 OK 关闭。



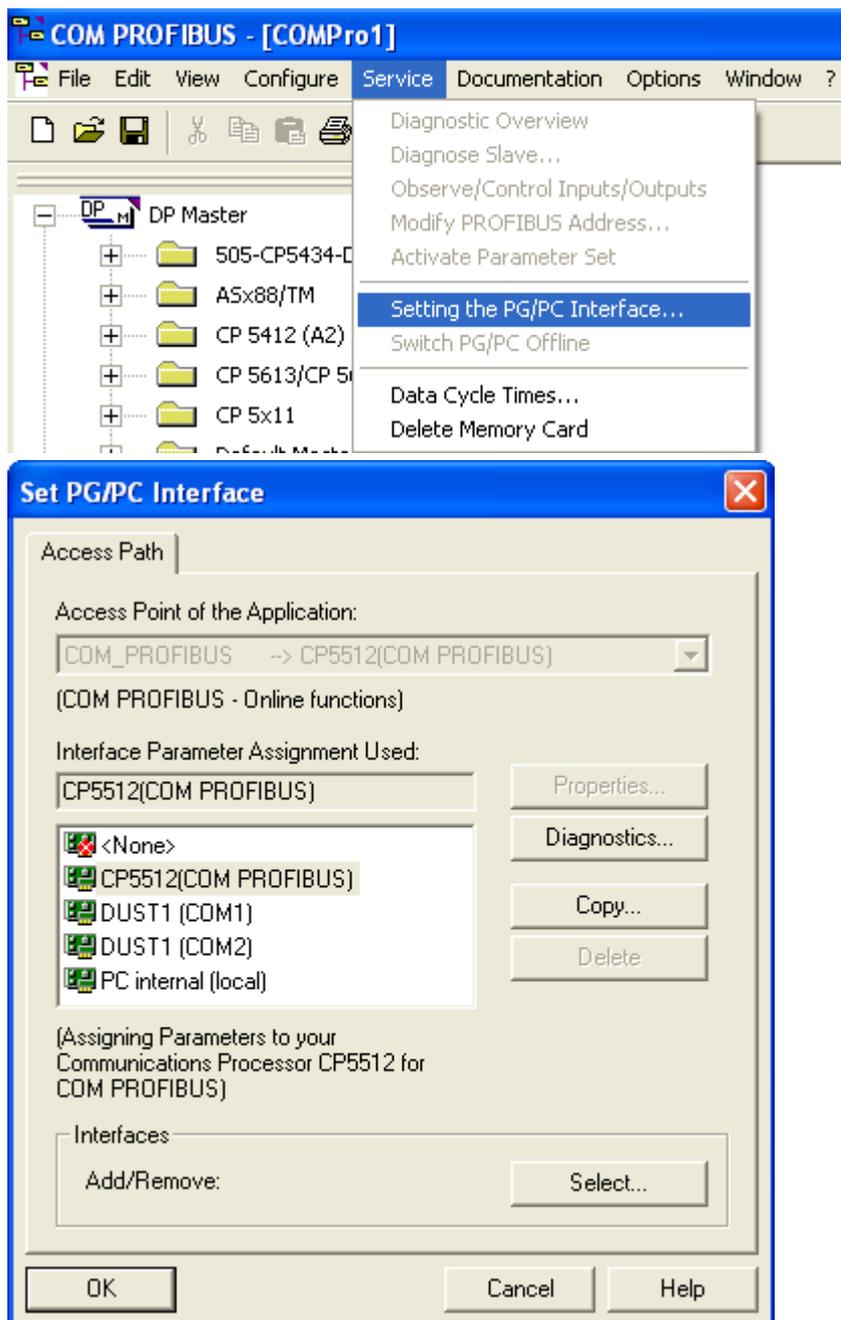
- 在点击 Shared master..., 选中地址为 1 的 DP 主站即 SS52 作为共享该 ET200S 输入的 DP 主站。



- 在菜单项选择 **Configure** 中的 **Set Bus Parameters...**，通过 **Bus Parameters** 来设置 PROFIBUS 的网络参数，本例中保持默认参数。然后保存该文件。



- 在菜单项选择 **Service** 中的 **Setting the PG/PC Interface...**，来设置 COM PROFIBUS 的下载参数。



- 首先点击 DP 主站即 SIMADYN D 出现黑色外框。在菜单项选择 File 的 Export，点击 DP Master 的下载 COM PROFIBUS 的组态参数。需要把下载电缆连接到 CP50MO 的 X2 接口，因为已经设定该接口为 PROFIBUS 接口。具体下载方法请参照 4.5.2。
- 在 Step7 中，新建一个项目，插入一个 S7-300 站作为 DP 主站，再插入一个 ET200S

作为 DP 从站。对其进行硬件组态。设置 CPU314 的 DP 站地址为 2，设置 ET200S 的 DP 站地址为 5。

HW Config - [SIMATIC 300(1) (Configuration) -- test]

Station Edit Insert PLC View Options Window Help

(0) UR

1	
2	<b>CPU 314C-2 DP</b>
X2	DP
2.2	DI24/DO16
2.3	AI5/AO2
2.4	Count
2.5	Position
3	
4	
5	
6	

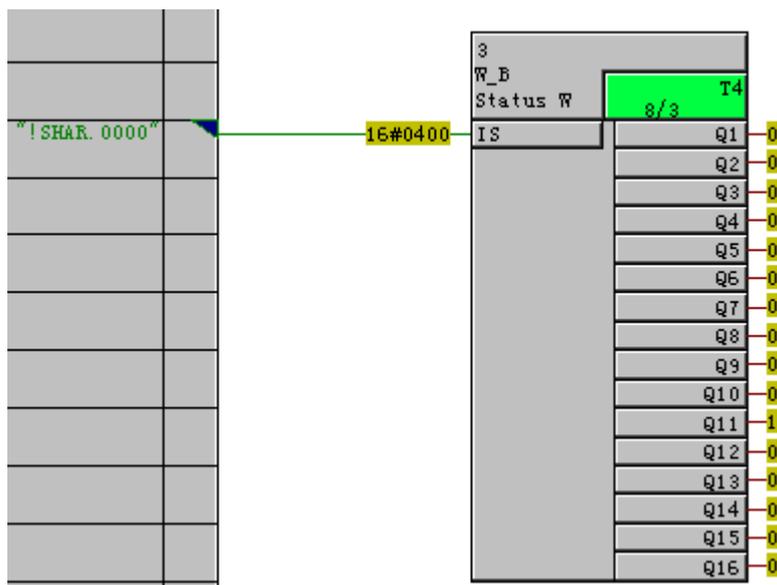
PROFIBUS(1): DP master system (1)

(5) IM151-1

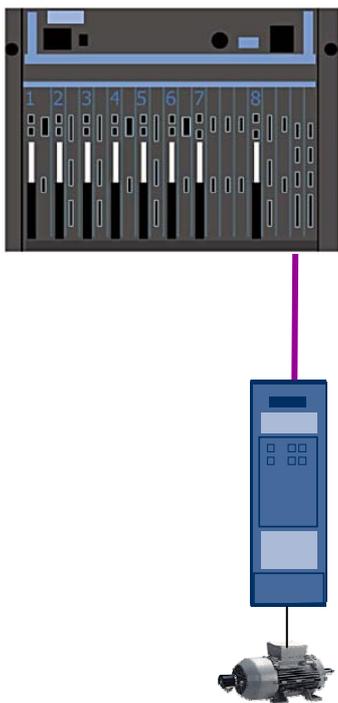
(5) IM151-1 Standard

Slot	Module	Order Number	I Address	Q Address
1	PM-E DC24V	6ES7 138-4CA00-0AA0		
2	4DI DC24V HF	6ES7 131-4BD00-0AB0	0.0...0.3	
3	4DI DC24V HF	6ES7 131-4BD00-0AB0	1.0...1.3	
4	4DO DC24V/0.5A ST	6ES7 132-4BD00-0AA0		0.0...0.3
5	4DO DC24V/0.5A ST	6ES7 132-4BD00-0AA0		1.0...1.3
6				

- 下载完成后，在线观察 CFC 程序，可以读到 ET200S 的输入模板的变化状态。

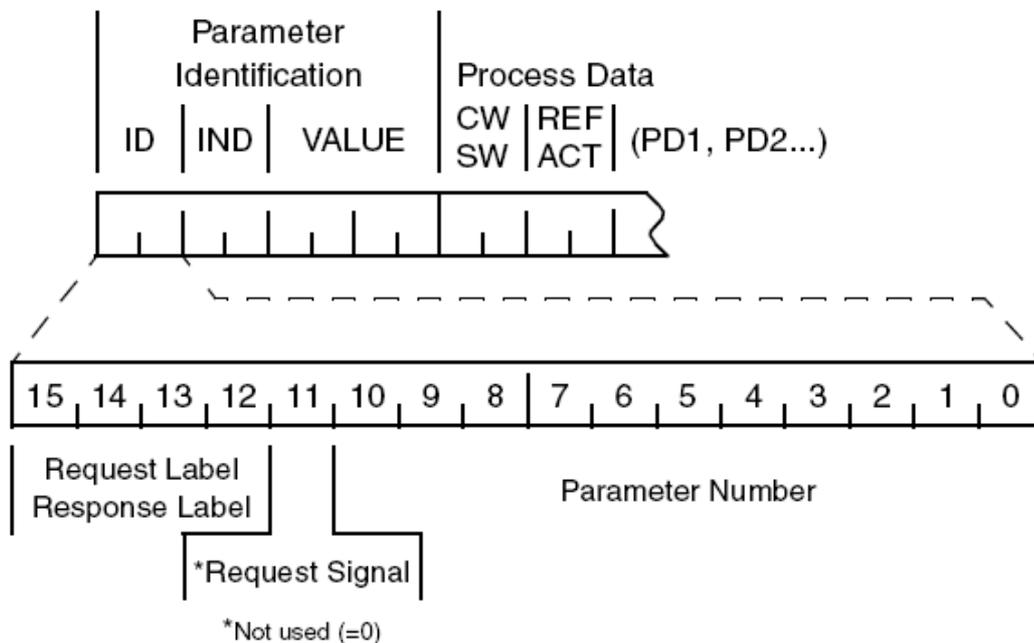


#### 4.5.5 CP50MO 与 MM440 的通讯



TDC 的 CP50MO 作为 DP 主站与传动 MicroMaster440 进行 PROFIBUS-DP 通讯，使用 PPO1 即 4 个 PKW 和 2 个 PZD 来与传动 440 交换数据。其中 PKW 用来读写传动的参数。其中第一个 PZD 作为控制字 CW/状态字 SW，每一位的具体信息，请参照 MM440 手册。第二个 PZD 作为速度给定值 REF/实际速度值 ACT。简要介绍一下数据报文中的 PKW

(Parameter Identification) 的数据区:



ID—分为两个部分，第一个部分是 Request Label，确定了参数在两个通讯设备之间的传输方式，例如参数数值的读写。第二个部分是 Parameter Number，确定了要操作的参数号，例如 P2291。

IND—确定了参数的索引号，例如对于参数 P2000[3]，那么 0, 1, 2 就是索引号。对于 TDC 与 MM440 通讯，不必考虑 PNU 页号。因为 TDC 的功能块提供了绝对值的参数号。

VALUE—确定了要修改参数的数值，包括两个字，分别对应传动中一个字参数 (INT)的数值和两个字参数(DI/Real)的数值。

MM440 的关于 PROFIBUS 的参数设置：

P700=6 COM 设置

P1000=6 COM 设置

P918=5 设置站地址（注意硬件 DIP 开关位置全部 ON 或 OFF）

P971=1 保存设置参数

P2013=127 PKW 的字长可变

4.5.6.1 在 SIMATIC Manager 中进行硬件组态

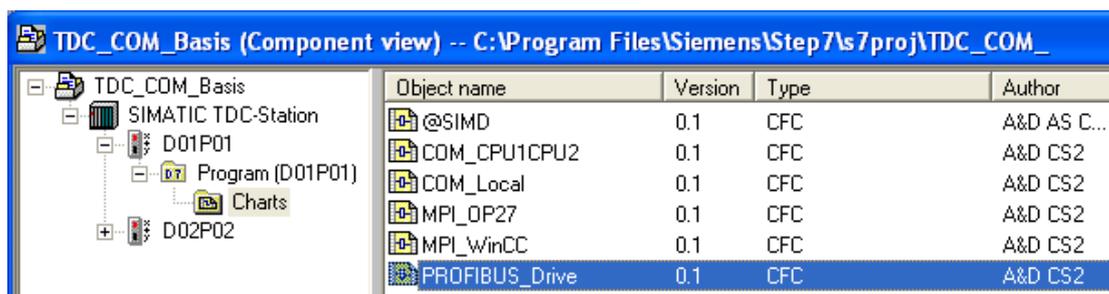
- 硬件组态与 4.5.1 章硬件组态相同。其中对于 CP50MO 模板的 X1, X2 口默认为

MPI 接口。修改该模板的 X2 接口属性为 PROFIBUS。

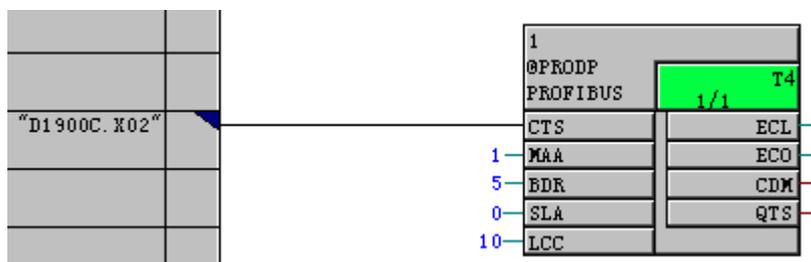
- 计算机 PG 使用标准的 MPI 电缆和 CP5512MPI 网卡，用于下载 CFC 程序和诊断 TDC。

#### 4.5.6.2 在 SIMATIC Manager 中进行软件组态

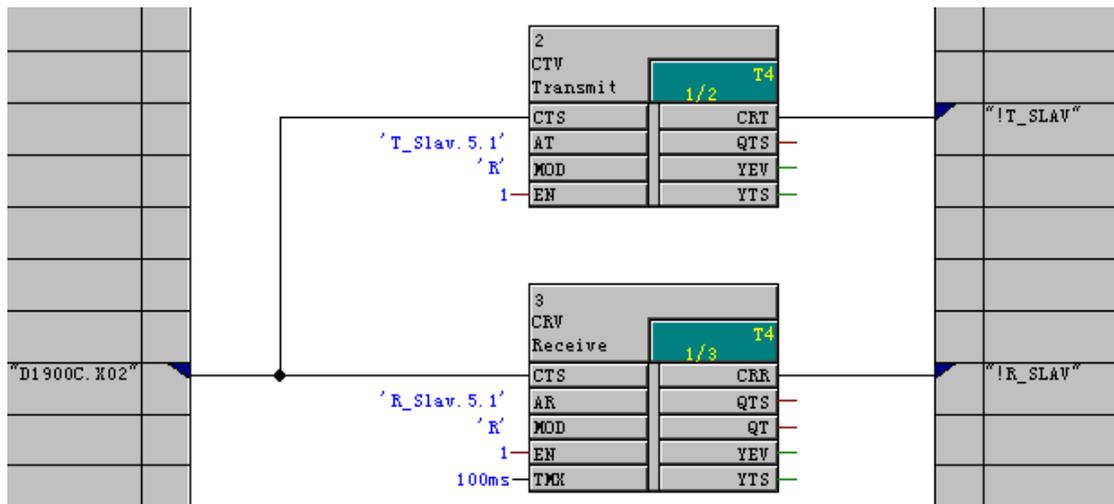
- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 PROFIBUS\_Drive，并通过双击打开。



- 插入@PRODP 功能块到表中，连接在硬件组态中的硬件模板地址 D1900C.X02。即利用 CP50MO 的 X2 作为 PROFIBUS 通讯的 DP 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。



- 分别插入 CTV 和 CRV 功能模块，用于与站地址为 5 的 DP 从站 MM440 接收和发送数据。



CTV 功能块的引脚说明:

CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 T\_Slav.5.1。

MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!T\_SLAV

CRV 功能块的引脚说明:

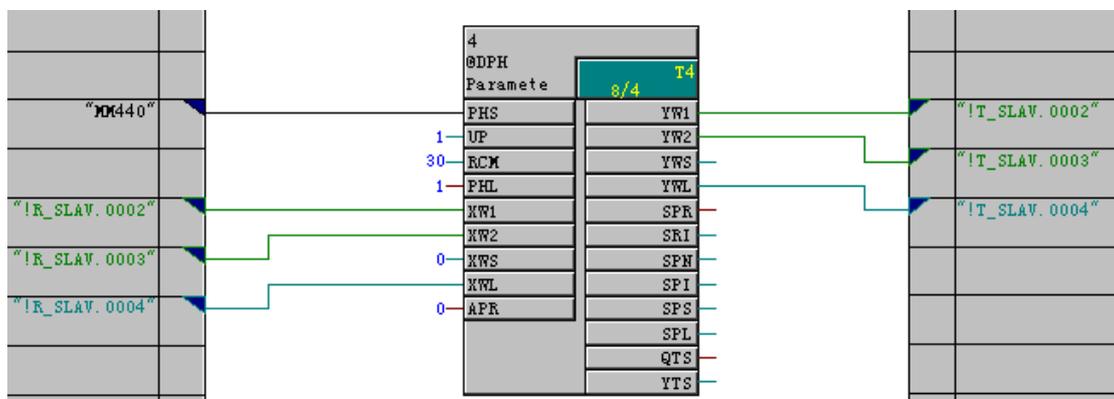
CTS—用于连接通讯的硬件接口，模板地址 D1900C.X02

AT—设置通道名称为 R\_Slav.5.1

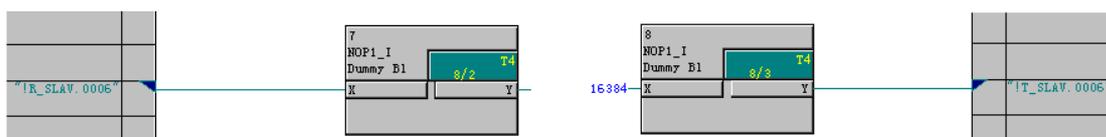
MOD—定义为 R，刷新 Refresh 模式

CRT—定义虚拟连接名称为!R\_SLAV

- 插入@DPH，用于读写传动中的参数数值。

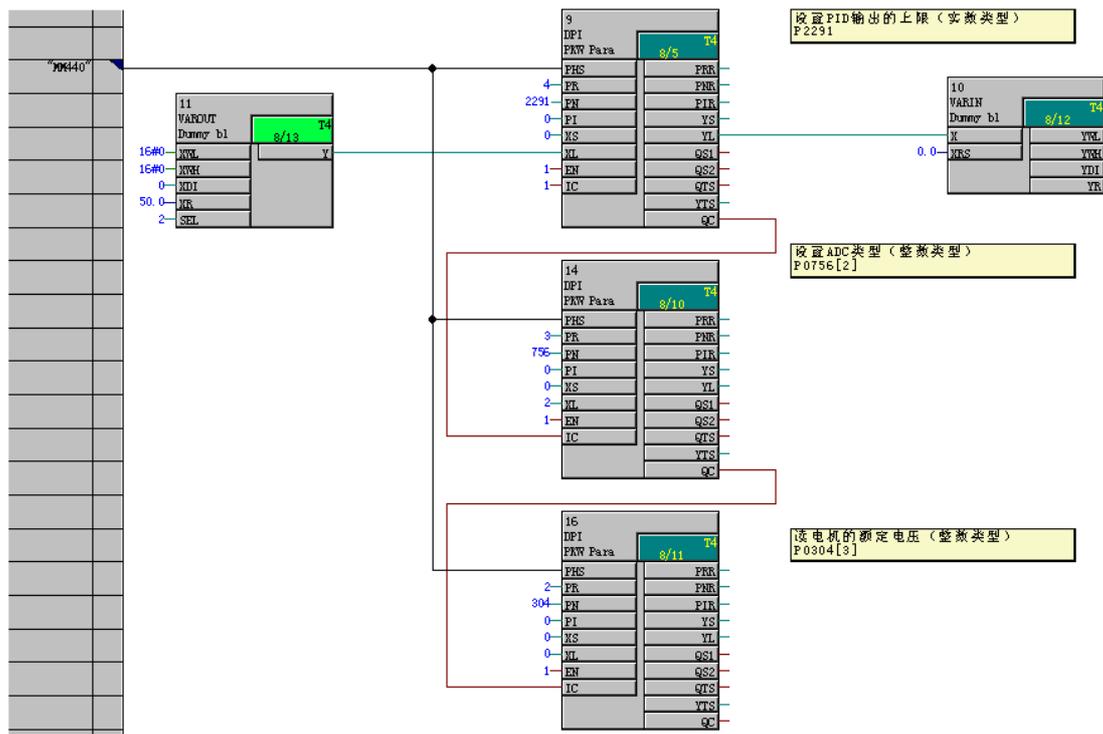






需要注意控制字/状态字以及给定/实际的功能块的虚拟连接号应该按照 PPO 报文中的顺序。

- 插入 DPI 功能块用于修改或读取 MM440 的参数值。



DPI 功能块的说明:

该功能块只能与@DPH 功能块一起使用，DPI 只能用 2byte 或 4byte 的参数值。

PHS—设置全局操作数。与@DPH.PHS的全局操作数设置相同。

PR—就是 PPO 消息类型中 PKW 部分 ID 的 Request Label/Response Label，用于读写参数的操作方式。具体任务号，请察看在线帮助。

PN—就是 PPO 消息类型中 PKW 部分 ID 参数号 Parameter Number。0~39999。

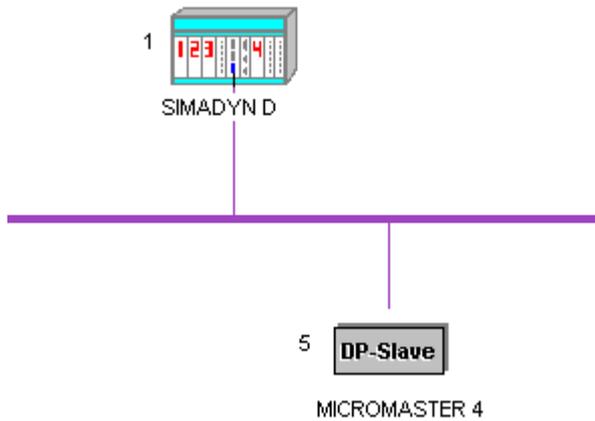
PI—就是 PPO 消息类型中参数识别 Parameter Identification 部分的 IND，用于参数的组的索引。

XS—以 2byte 来修改参数值，VALUE 中的第一个字。

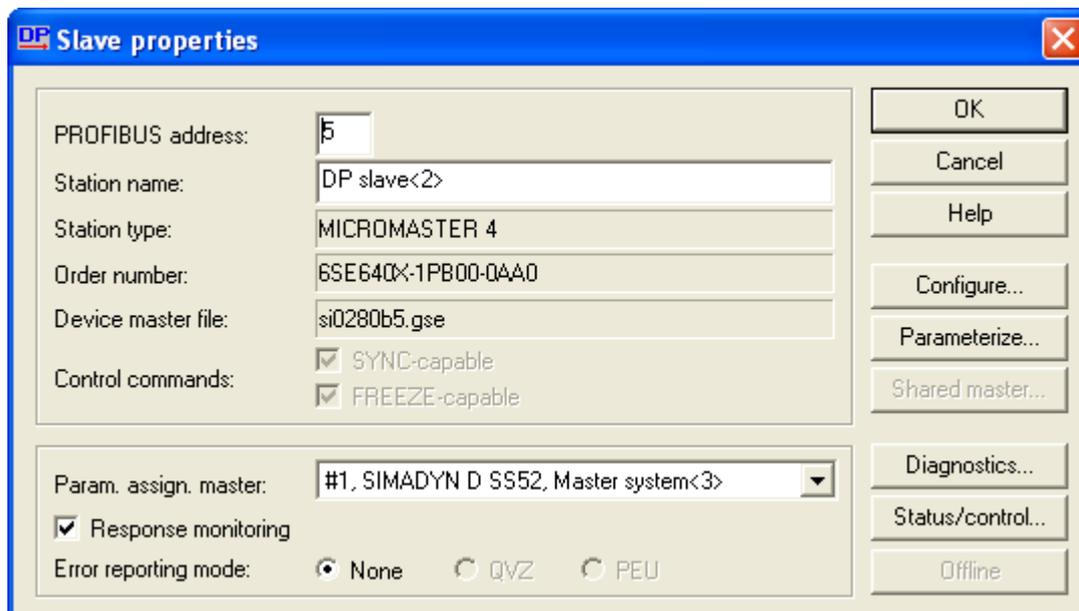
XL—以 4byte 来修改参数值，VALUE。



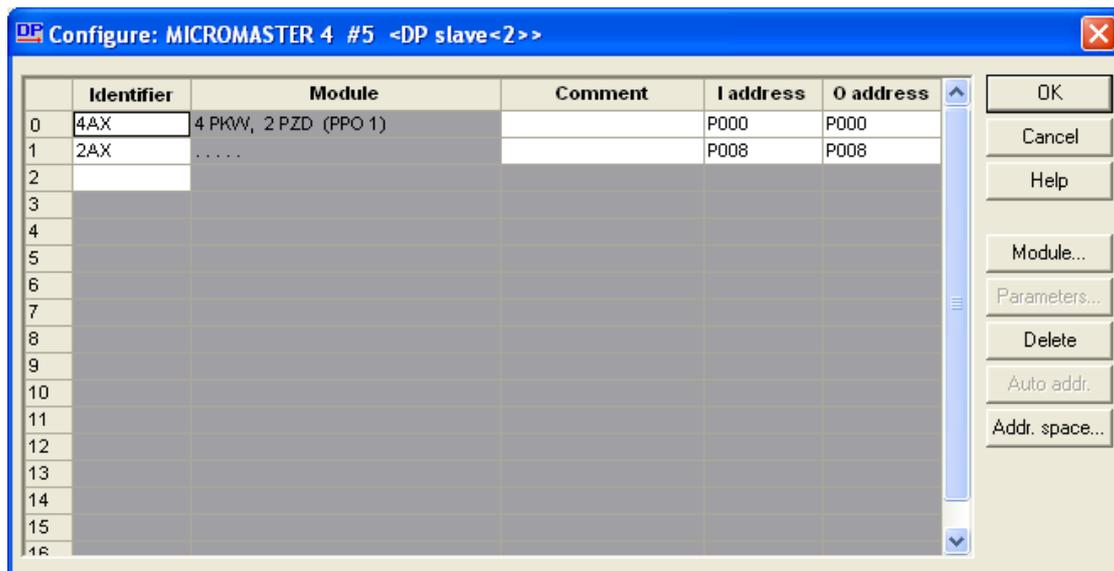
Drives 的 SIMOVERT 中选中 MICROMASTER 4 到右侧栏。



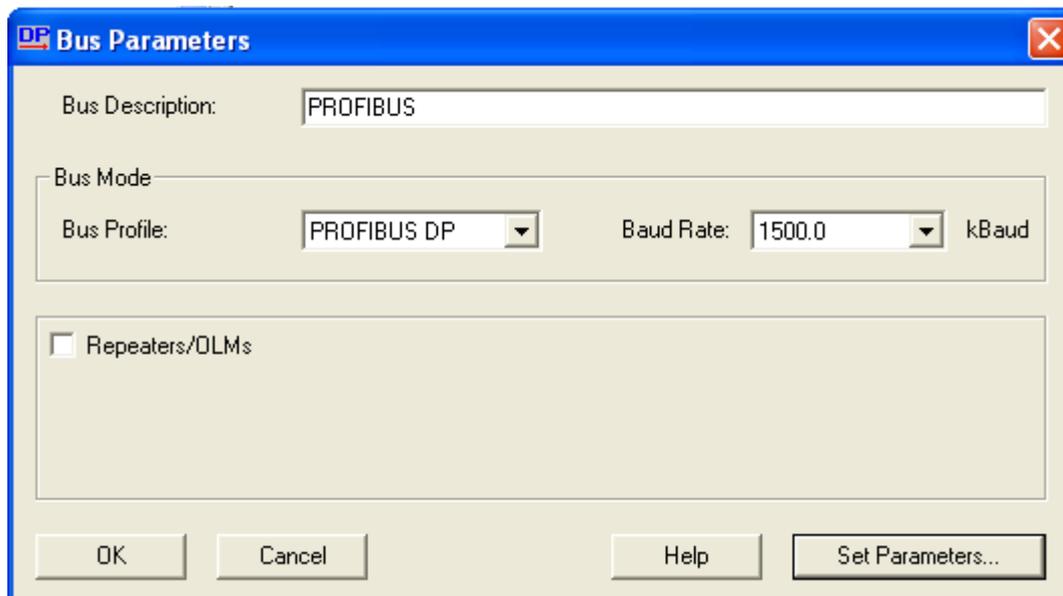
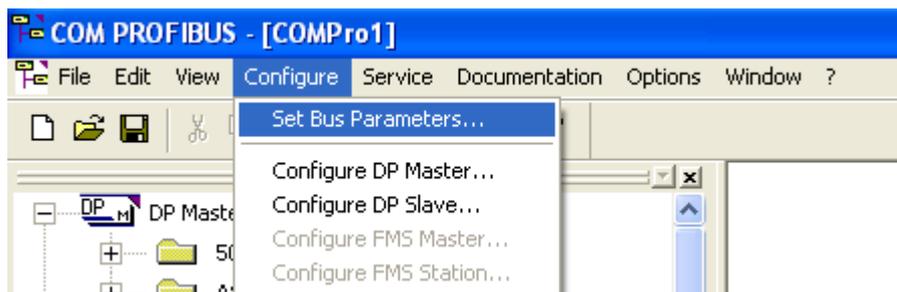
- 双击 MICROMASTER 4 图标，组态该 DP 从站。首先修改 MM440 的地址为 5。



- 点击 Configure..., 组态该 MM440 站为 PPO1 类型通讯。点击 Auto addr. 自动分配各个数据区地址。其它设置为默认，点击 OK 关闭。

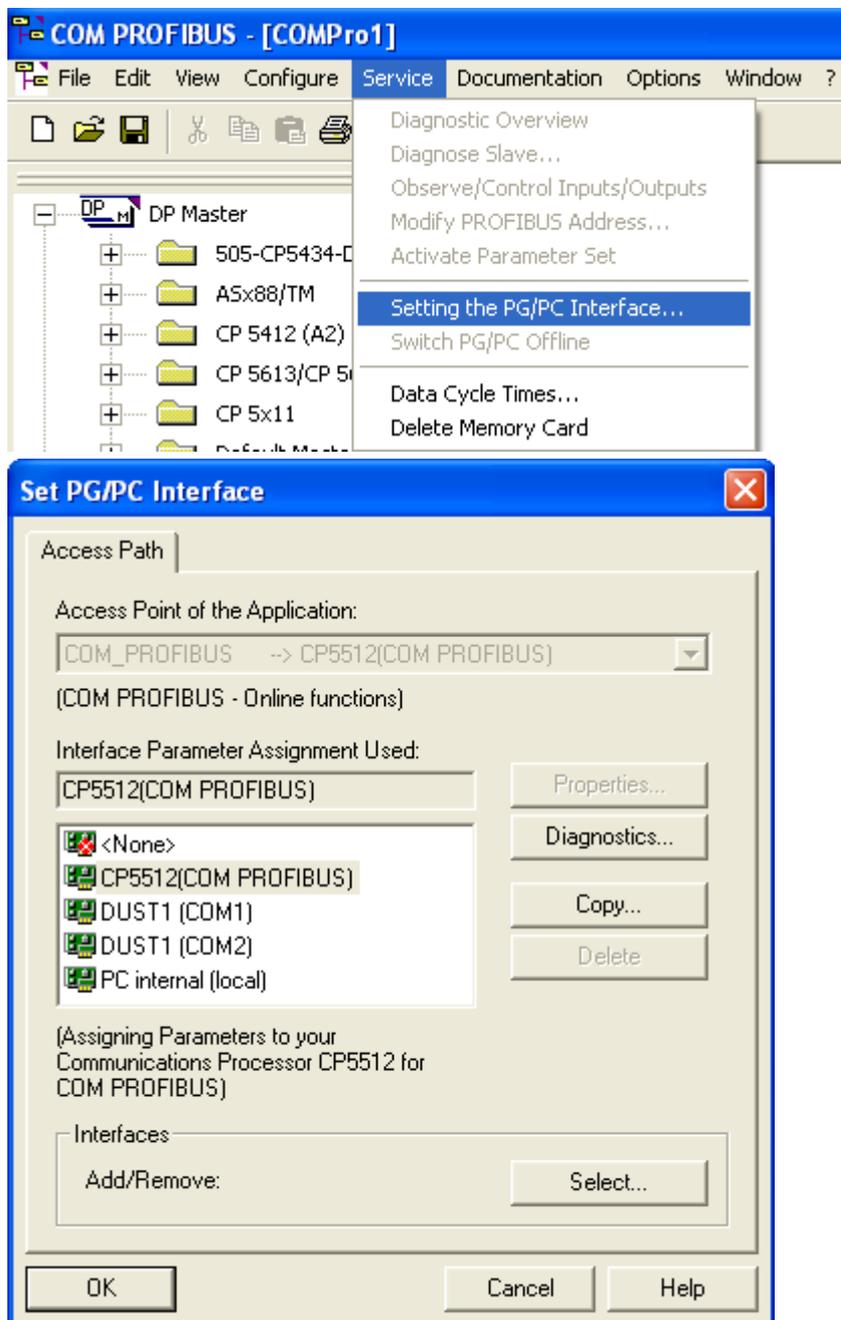


- 在菜单项选择 Configure 中的 Set Bus Parameters..., 通过 Bus Parameters 来设置 PROFIBUS 的网络参数, 本例中保持默认参数。然后保存该文件。



- 在菜单项选择 Service 中的 Setting the PG/PC Interface..., 来设置 COM

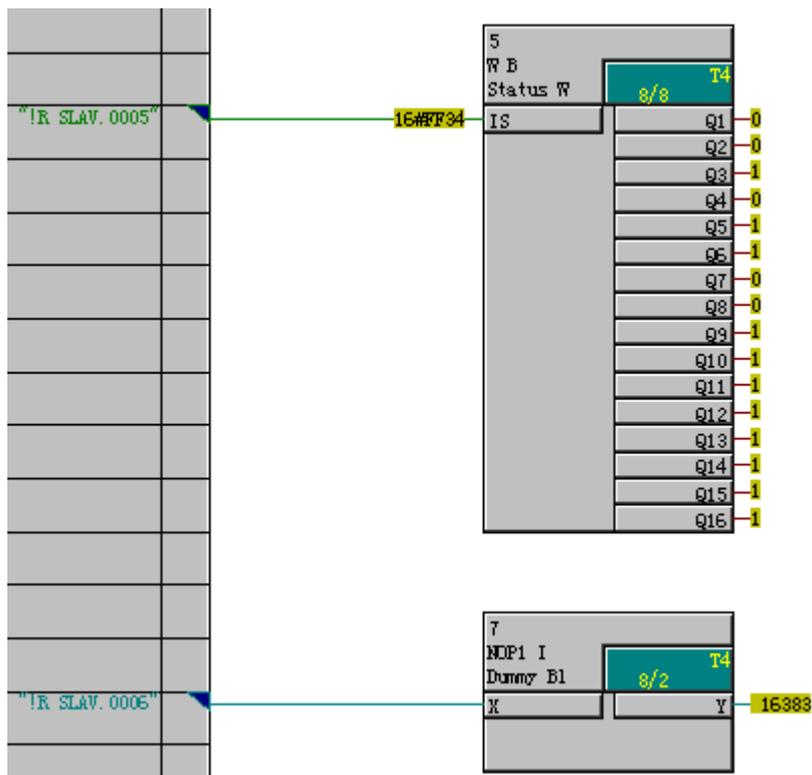
PROFIBUS 的下载参数。并在菜单项选择 File 的 Export，点击 DP Master 的下载 COM PROFIBUS 的组态参数。需要把下载电缆连接到 CP50MO 的 X2 接口，因为已经设定该接口为 PROFIBUS 接口。



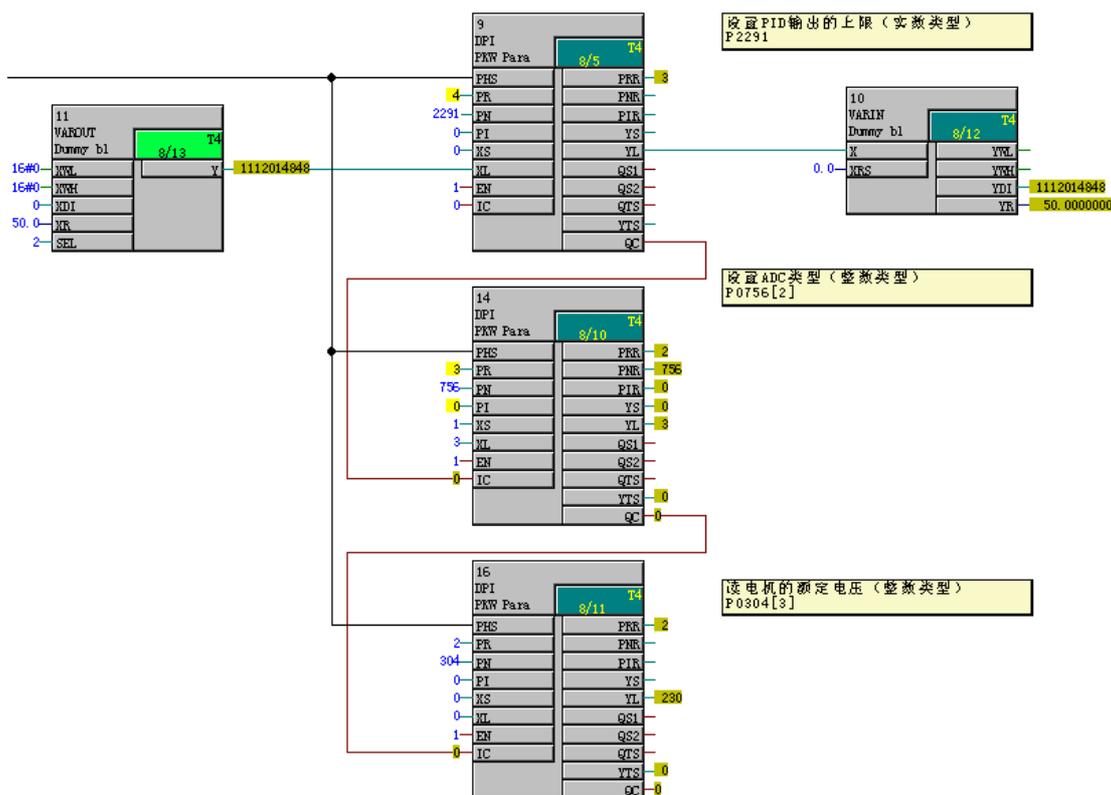
- 下载完成后，MM440 的绿灯常亮，表示通讯正常。在线更改 B\_W 功能块的数值，即修改控制字。从 16#047E 到 16#047F，MM440 启动需要第 0 位为上升沿。然后

输入 8.X 为 16384，可以看到 MM440 从 0Hz 运转到 50Hz。另外也可以通过 W\_B 和 7.Y 察看 MM440 的状态字和实际速度显示值。





- 在线修改参数 P2291，那么该参数的 IND 为 0，所以 PI 设置为 0。PR 设置为 4，即修改 P2291 的 4 个字节的参数字。设置 VAROUT 的 XR 为 50.0。通过 IC 信号的翻转，参数 50.0 设置给 P2291。在线修改参数 P756[0]，那么该参数的 IND 同样也设置 0，如果修改 P756[1]，那么 IND 设置为 1。PR 设置为 3，即修改 P756 的 2 个字节的参数，设置 XL 为 3，通过 IC 信号的翻转，那么 P756 的参数就修改为 3。在线读参数 P0304[0]，结果为 230。



#### 4.6 TCP/IP 通讯

工业以太网是 SIMATIC 产品的支持的网络，TCP/IP，UDP 是工业以太网的标准协议，TDC 可以通过 TCP/IP，UDP 协议与 SIMATIC 产品或其它生产厂家的产品进行通讯。

基本需要的硬件包括机架 UR5213，CPU，CP5100 或 CP51M1 模板。

TDC 中有 2 块工业以太网模板，他们分别是 CP5100 和 CP51M1。其中 CP51M1 为新的工业以太网模板，用于取代 CP5100。他们之间的最大区别在于 CP51M1 可以开辟存储缓冲区用于 CPU 之间的通讯。

一个 CP51M1 只集成一个 10/100 Base TX 的 RJ45 工业以太网接口。该端口支持支持全/半双工，波特率自适应模式。该模板还支持除 TCP/IP 协议外，还支持 UDP，ISO on TCP 以及 S7 协议。

一个 CP5100 模板除集成 10/100 Base T 的 RJ45 以太网接口外，还集成 Cheapernet、MAU 以太网接口。对于 RJ45 的工业以太网接口，支持波特率自适应模式。该模板还支持除 TCP/IP 协议外，只支持 UDP 协议。

任何一个 CP5100 或 CP51M1 对应一个功能块@TCPIP 来组态对应，该功能块用于初始化和监控工业以太网通讯。使用功能块 CTV 和 CRV 来发送和接收数据。

可以组态多个 SER 服务功能块用于关联一个 CPU，通过与通讯接口的耦合，PG/PC 可以通过工业以太网接口下载和调试 CFC（CP51M1 才有此功能）。SER 功能块还初始化和处理 PG 和显示设备（WinCC/OP/TP）的通讯连接。

可以组态一个 S7OS 功能块用于 MPI 服务（一个 CPU 需要一个 S7OS 功能块），通过与工业以太网接口的耦合，用于处理和响应连接的 WinCC 或操作员面板 OP/TP。

SER 功能块可以组态在任意循环时钟周期 T1~T5，但系统处理 SER 功能块的默认采样时间为 32ms。那么意味着该功能块会 32ms 来刷新在显示设备的数据。而 S7OS 的功能块可以组态在任意循环时钟周期 T1~T5，那么可以组态在采样周期长的循环时间中，这样会减少 CPU 的负载。

刷新（refresh）和复合（multiple）两种数据传输模式适用于 UDP 协议。

握手（handshake）和选择（select）两种数据传输模式适用于 TCP 协议。

使用 CP5100 或 CP51M1 可以通过 TCP/IP 协议与 SIMATIC TDC, SIMATIC S5/S7 以及第三方的设备通讯。

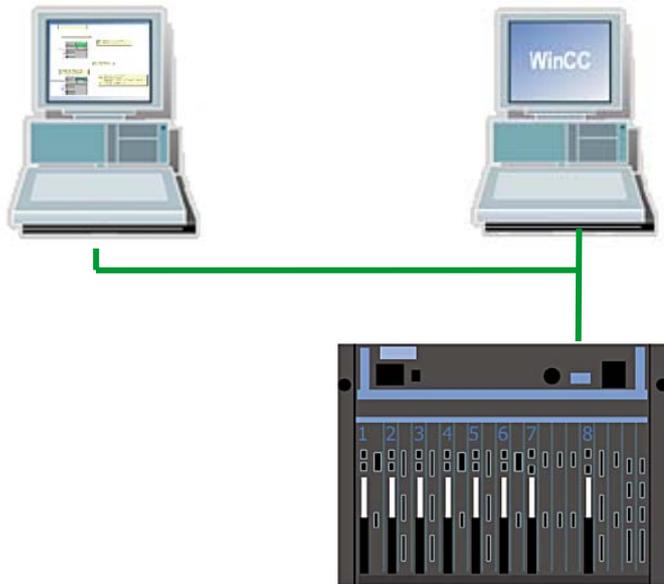
使用 CP5100 或 CP51M1 可以通过 UDP 协议与 SIMATIC TDC, SIMATIC S5/S7, 以及第三方的设备通讯。

使用 CP51M1 可以通过 ISO on TCP 协议与显示设备例如 WinCC 通讯。

使用 PG/PC 可以通过 S7 协议通过 CP51M1 来下载和诊断 TDC。

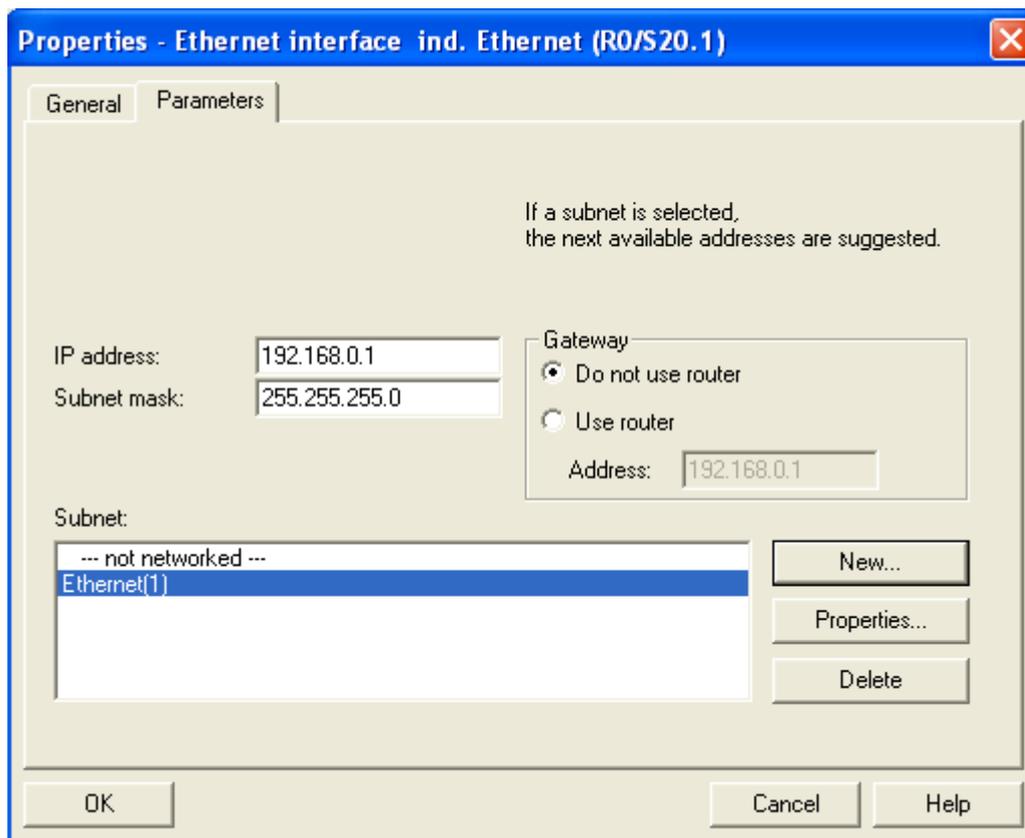
#### 4.6.1 CP51M1 与 WinCC 的标准通讯

CP51M1 可以与 SIMATIC WinCC 通过工业以太网以 ISO-on-TCP 协议进行通讯，不需要增加额外的软件就可以完成与 WinCC 的通讯，但是只能访问过程数据。



#### 4.6.1.1 在 SIMATIC Manager 中进行硬件组态

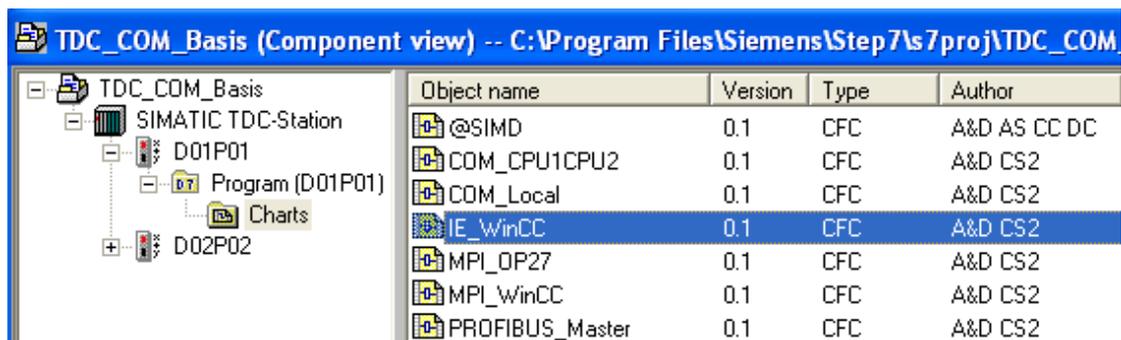
- 硬件组态与 4.1 章硬件组态相同。如果 PC 的网卡端口不是自适应交叉接，那么要使用交叉接的网线与 CP51M1 相连。可以使用带有自适应交叉接的交换机，这样可以使用交叉接或平行接的网线。
- 修改 CP51M1 的硬件组态信息。双击硬件组态 D2000C.X1，新建 Ethernet(1)，并设置 IP 地址 192.168.0.1，子网掩码 255.255.255.0。



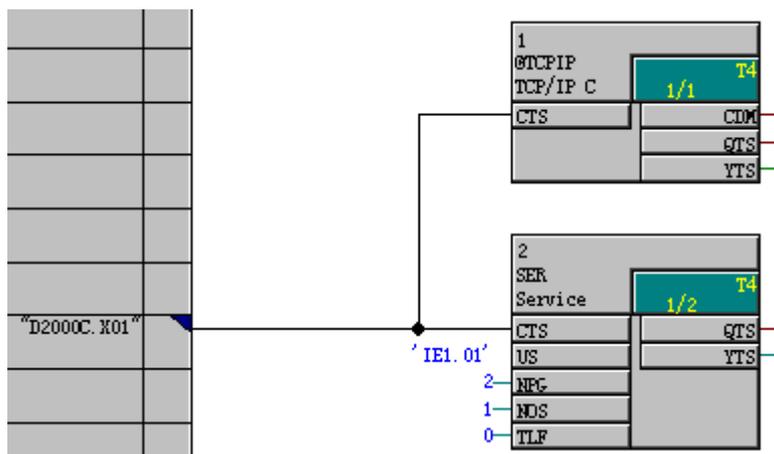
- 计算机 WinCC 操作员站使用标准的以太网电缆和以太网网卡。

#### 4.6.1.2 在 SIMATIC Manager 中进行软件组态

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 IE\_WinCC，并通过双击打开。



- 插入@TCPIP 功能块到表中，连接在硬件组态中的硬件模板地址 D2000C.X01。即利用 CP51M1 的 X1 作为与 WinCC 通讯的 Ethernet 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。插入 SER 功能块，可以任意组态到任何一个时间周期。



SER 功能块的引脚说明：

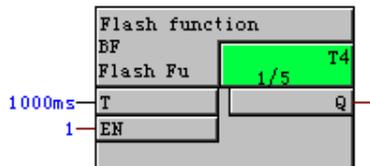
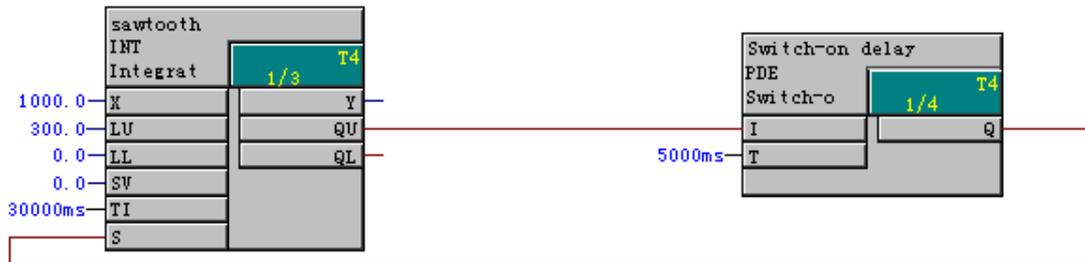
CTS—连接硬件组态的模板地址为 D2000C.X01，即使用 CP51M1 的接口 X1 作为服务端口。

US—任意定义一个不大于 6 个字符长度的通道名，01 表示关联 CPU 的槽号地址。说明通过 SER 功能块，由 CPU1 来完成该接口通讯处理服务。

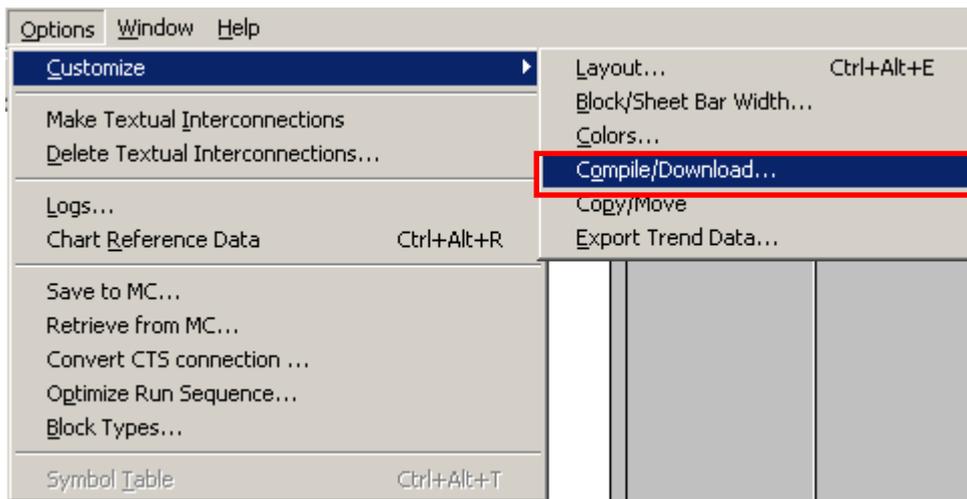
NPG—指定 PG 的个数，默认为 2。也就是可以通过 TDC 的 MPI 接口连接的 PG 的个数。即可以使用设定的 Ethernet 接口 X1 来下载和诊断 TDC 的服务。

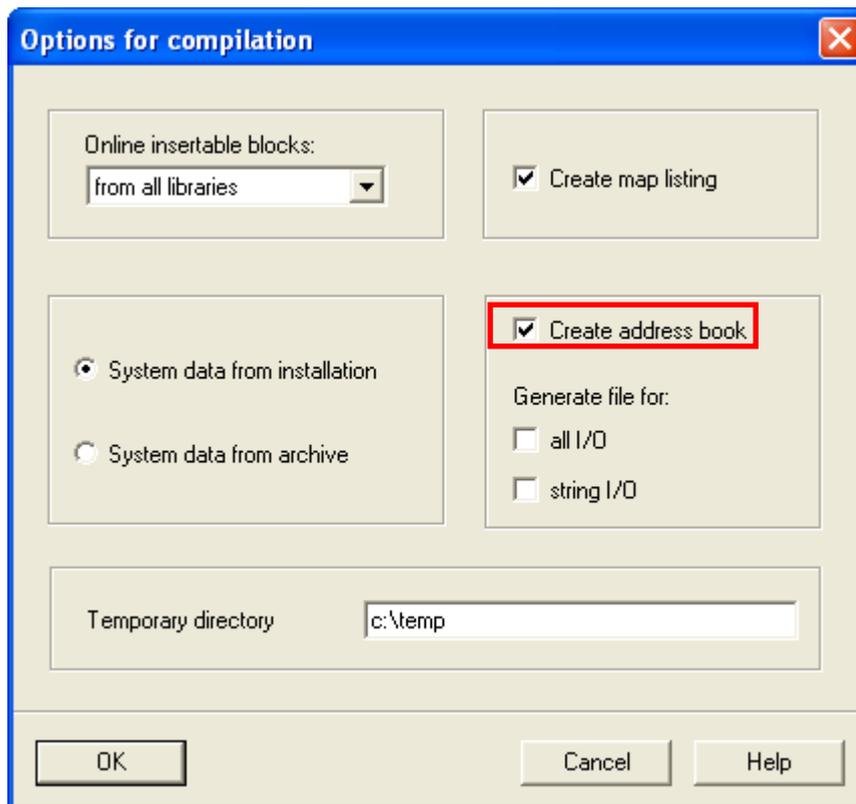
NOS—指定 OS 的个数，默认为 0。本例中为 1，也就是指定监控变量（C and M）的显示设备（例如 WinCC）的个数。即可以使用设定的 Ethernet 接口 X1 来实现工业以太网通讯。

- 在 CFC 中定义两个功能，第一个功能通过 INT 命名为 sawtooth 功能块进行积分运算当达到高限 300 时通过 PDE 命名为 Switch-on delay 功能块延迟 5 秒钟复位该积分功能块，然后重新积分。插入 BF 功能块命名为 Flash function 实现 BF.Q 的信号每隔 1 秒翻转一次。

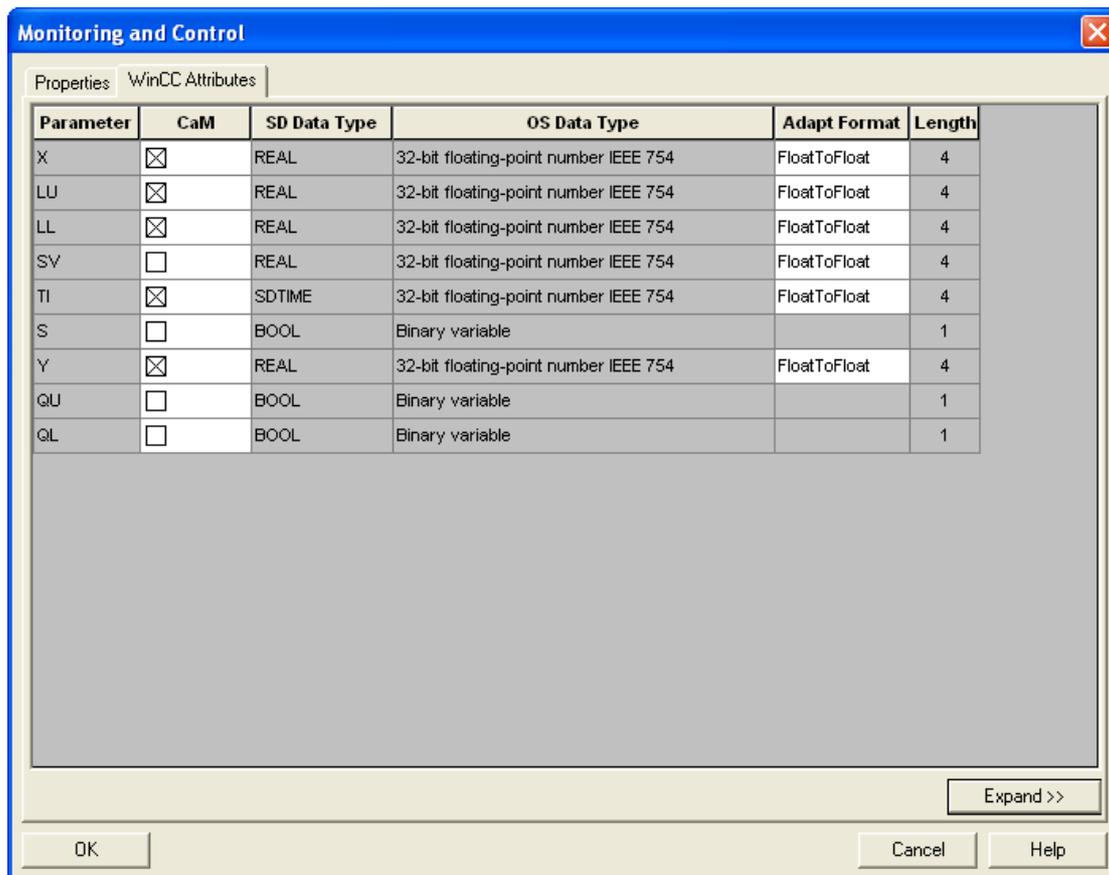


- 在 CFC 的菜单中，选择 Options 中的 Customize，选择 Compile/Download...，选中修改地址本选项 Create address book。

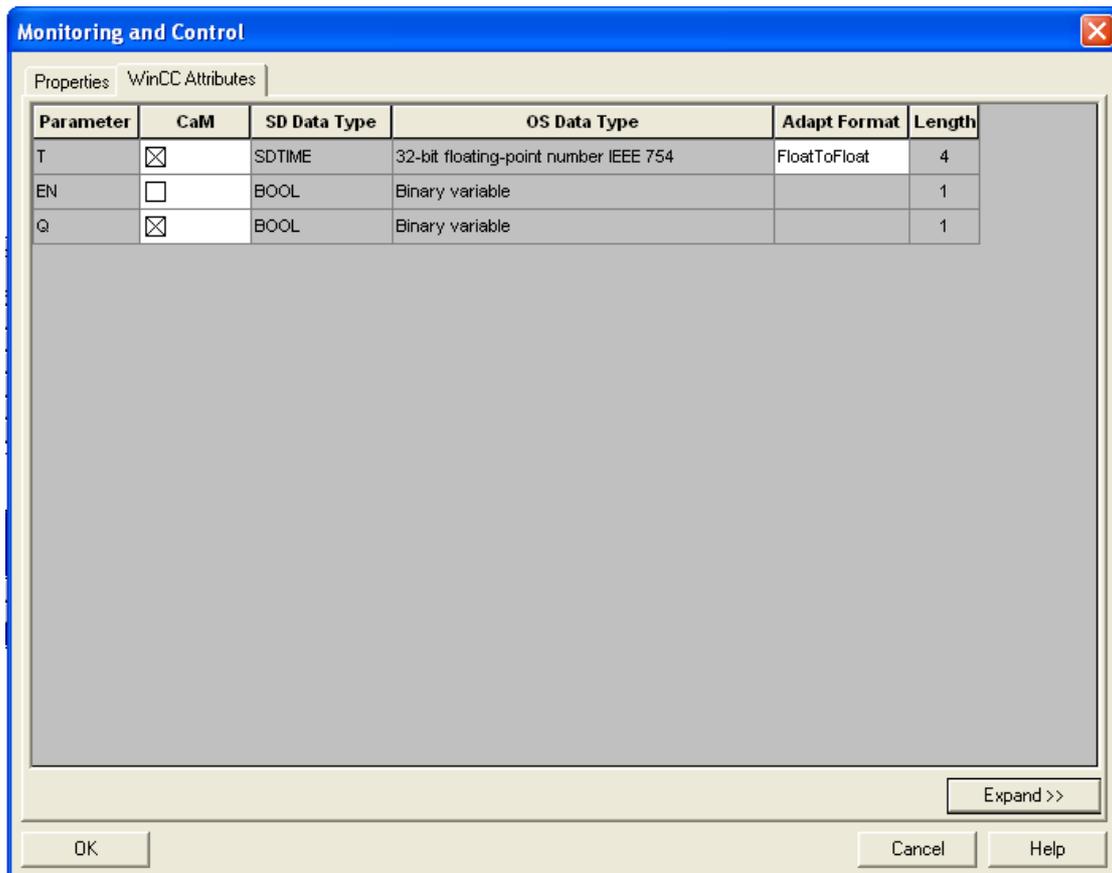




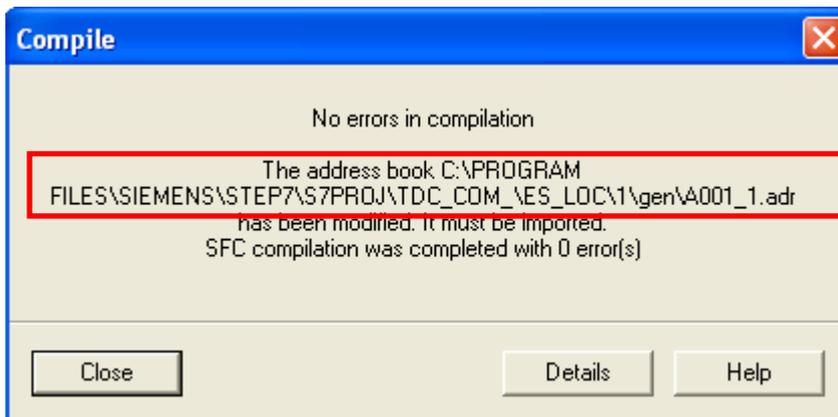
- 双击 INT 功能块, 打开 INT 功能块的属性。选择 OCM possible。点击 Operator C and M..., 打开 Monitoring and Control 属性。注意该功能块的映像区数据块为 1006。可以选中 complete block structure 选项, 选中该选项, 就是对该功能块的所有输入和输出进行监控。在打开的 Monitoring and Control 属性中, 切换到 WinCC Attributes 页面, 选择要监控的变量。并且使用同样的方法选中 BF 功能块的输出 Y 作为监控变量。



- 对于 BF 功能块，使用相同的方法设置。注意该功能块的映像区数据块为 1007。



- 设置完成后，对 CFC 进行编译，出现生成的地址本的路径。

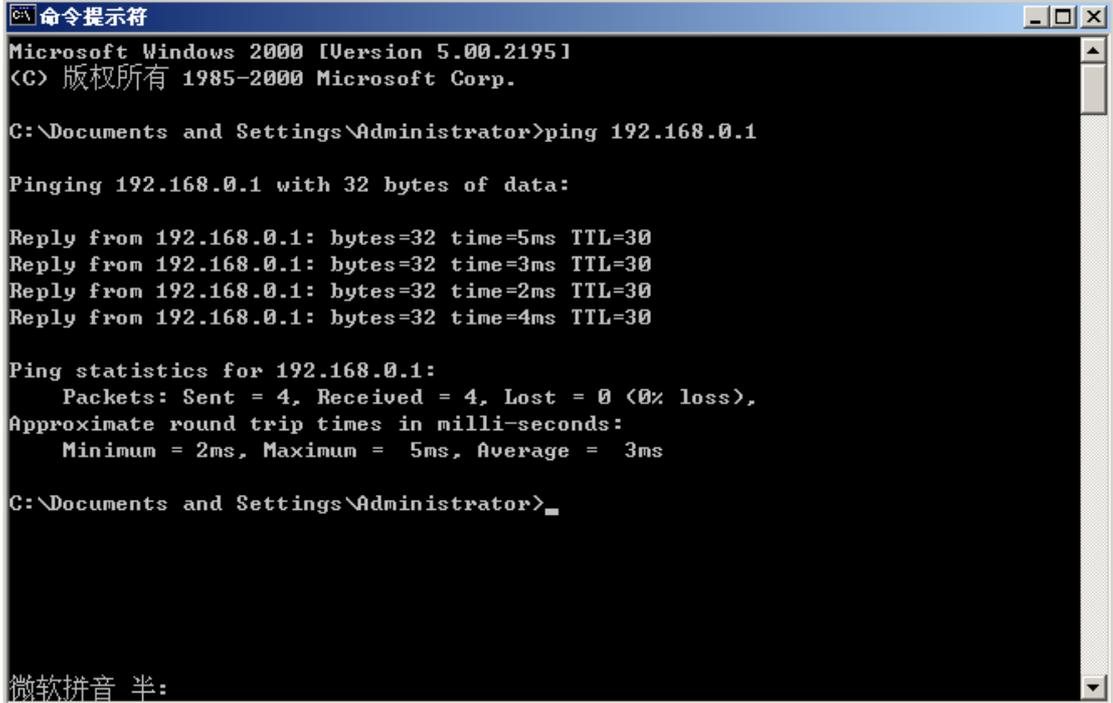


- 根据该路径，通过 WordPad 打开该文件。上面记录着 CPU1 即 D01P01 的周期任务（T1~T5）和中断任务(I1~I8)的设置信息。最重要的是提供了所要监控功能块 sawtooth 和 flash function 的变量和所分别对应的数据功能块的代号 1006、1007，以及对应该变量的偏移地址。

```

V 5.0.0
31.01.97
A001
DO1P01
A001
26.10.2005:09:44
1.0
T1:1.0
T2:4.0
T3:16.0
T4:64.0
T5:256.0
I1:0
I2:0
I3:0
I4:0
I5:0
I6:0
I7:0
I8:0
FP:3
$:0
FP:IE_WinCC:2:
  FB:Flash function:T4:2
    T :8091,1007,0:TF:::E:::F
    Q :8091,1007,24576:B1:::A:::N
  FB:sawtooth:T4:5
    X :8091,1006,0:NF:1.0::E:::F
    LU :8091,1006,4:NF:1.0::E:::F
    LL :8091,1006,8:NF:1.0::E:::F
    TI :8091,1006,16:TF:::E:::F
    Y :8091,1006,24576:NF:1.0::A:::F
  
```

- 设置 PG/PC 接口为 CP5512(MPI)下载硬件组态信息和 CFC 程序。通过 Ping 命令来查看通讯是否连通。



```
命令提示符
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:

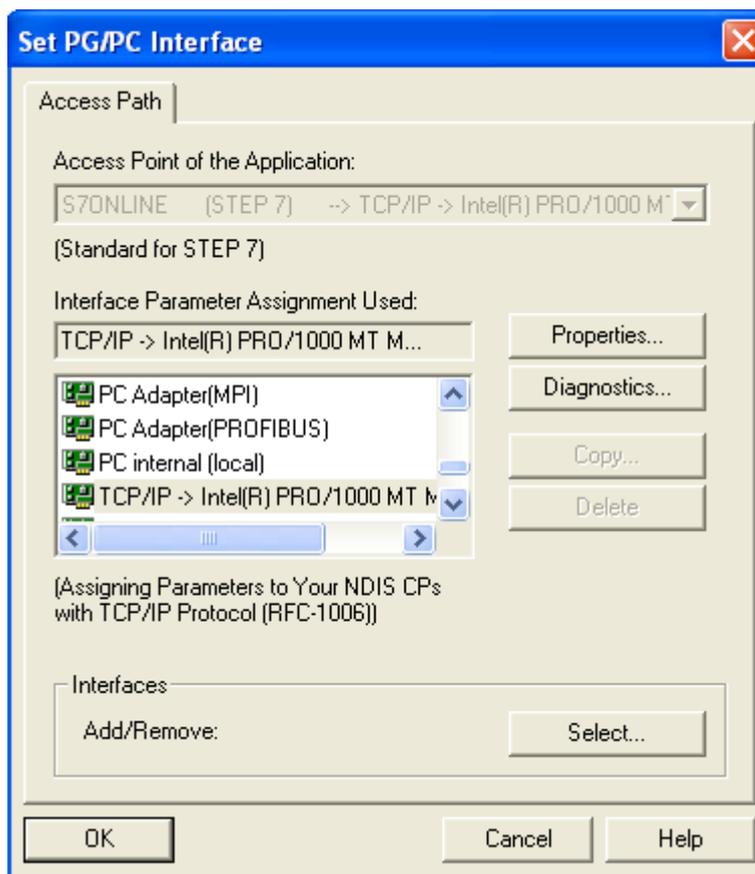
Reply from 192.168.0.1: bytes=32 time=5ms TTL=30
Reply from 192.168.0.1: bytes=32 time=3ms TTL=30
Reply from 192.168.0.1: bytes=32 time=2ms TTL=30
Reply from 192.168.0.1: bytes=32 time=4ms TTL=30

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 5ms, Average = 3ms

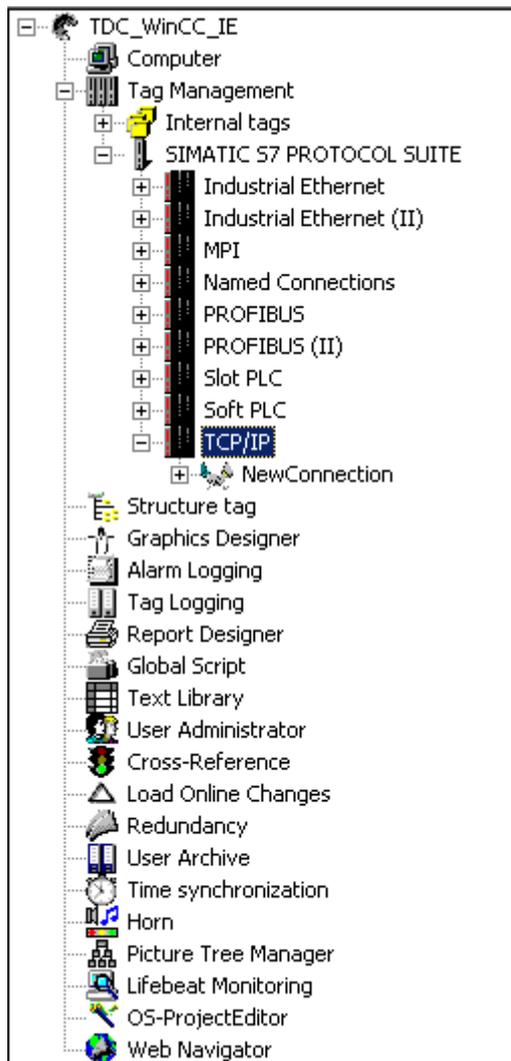
C:\Documents and Settings\Administrator>
```

微软拼音 半:

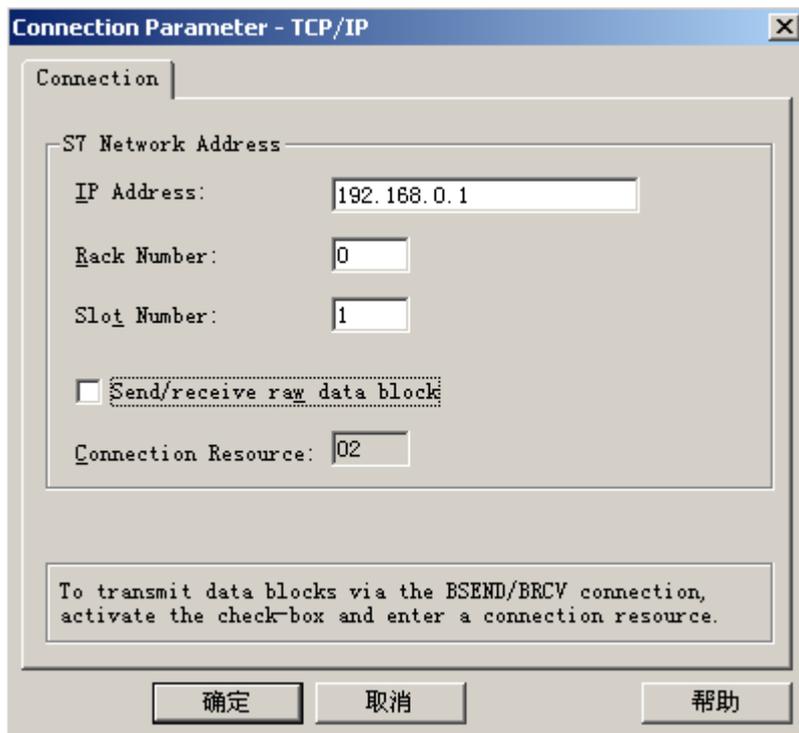
- 这时可以通过设置 PG/PC 接口为 TCP/IP->Intel(R) PRO/1000 MT M...表示可以使用本机网卡下载和诊断 TDC。



- 打开 WinCCExplore，新建工程并命名为 TDC\_WinCC\_IE。在 Tag management 中插入 S7 驱动协议包，即 SIMATIC S7 protocol suite。选择 TCP/IP 通道来建立一个连接。注：该 SIMATIC S7 PROTOCOL SUITE 的 TCP/IP 通道使用的是 ISO-on-TCP 协议。



- 新建一个连接 NewConnection，设置连接 TCP/IP 通道的属性。设置 IP Address 为 192.168.0.1，即 TDC 模板 CP51M1 的工业以太网的地址。设置 Slot Number 为 1，即 SER 功能块的 US 参数所指定的 CPU1 的槽号。



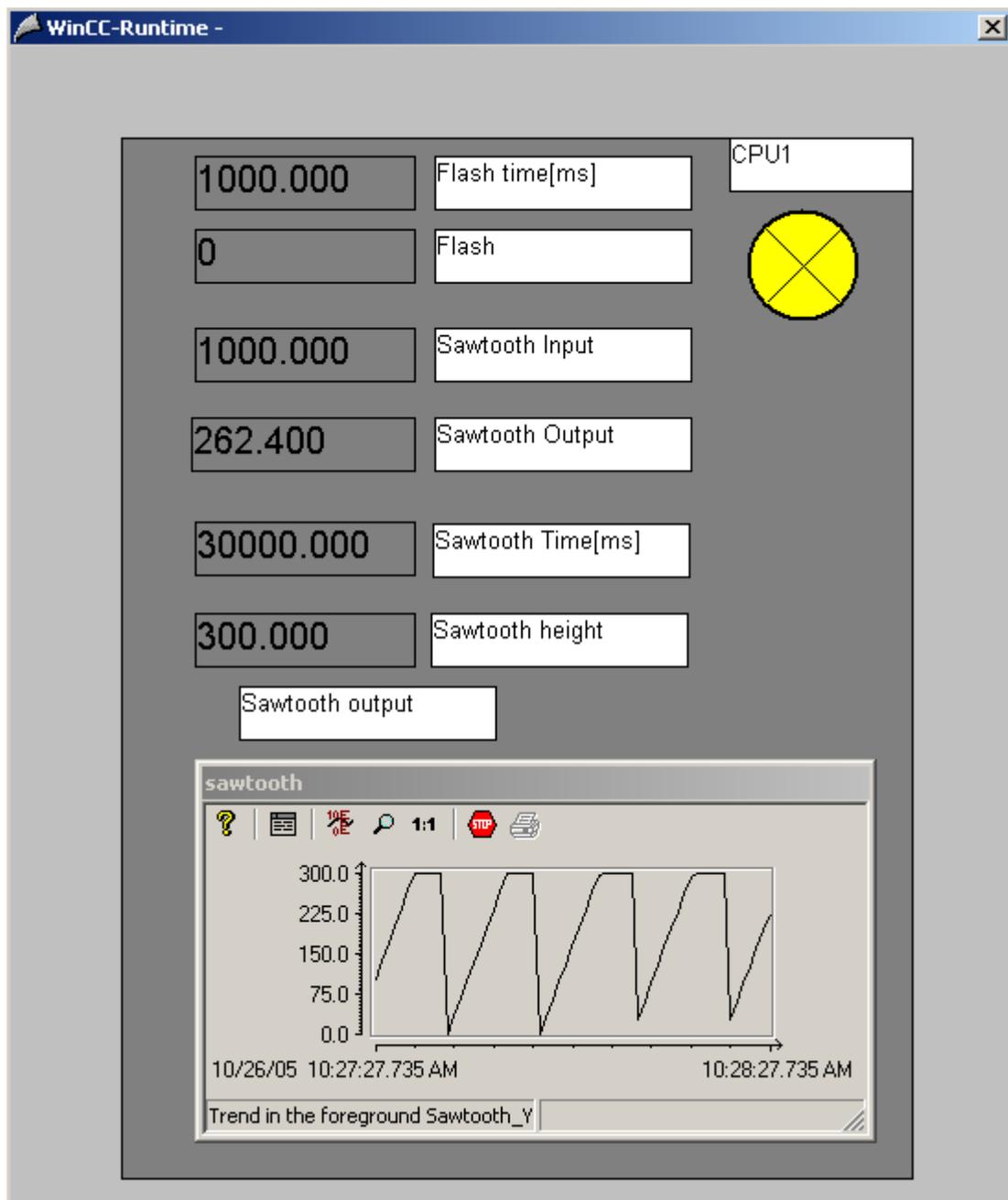
- 选择 TCP/IP 通道，设置系统参数，在 SIMATIC S7 栏中取消循环管理选项 by PLC。



- 在 NewConnection 新建连接中建立新的变量 NewTag，根据编译生成的地址本 A001\_1，设置新建变量的地址。

Name	Type	Parameters
Flash_T	Floating-point number 32-bit IEEE 754	DB1007,DD0
Flash_Q	Binary Tag	DB1007,D24576.0
Sawtooth_X	Floating-point number 32-bit IEEE 754	DB1006,DD0
Sawtooth_LU	Floating-point number 32-bit IEEE 754	DB1006,DD4
Sawtooth_LL	Floating-point number 32-bit IEEE 754	DB1006,DD8
Sawtooth_TI	Floating-point number 32-bit IEEE 754	DB1006,DD16
Sawtooth_Y	Floating-point number 32-bit IEEE 754	DB1006,DD24576

- 最后新建一个画面，用于显示连接的 CPU1 的监控变量。可以使用类似的方法显示连接 CPU2 中的监控变量。

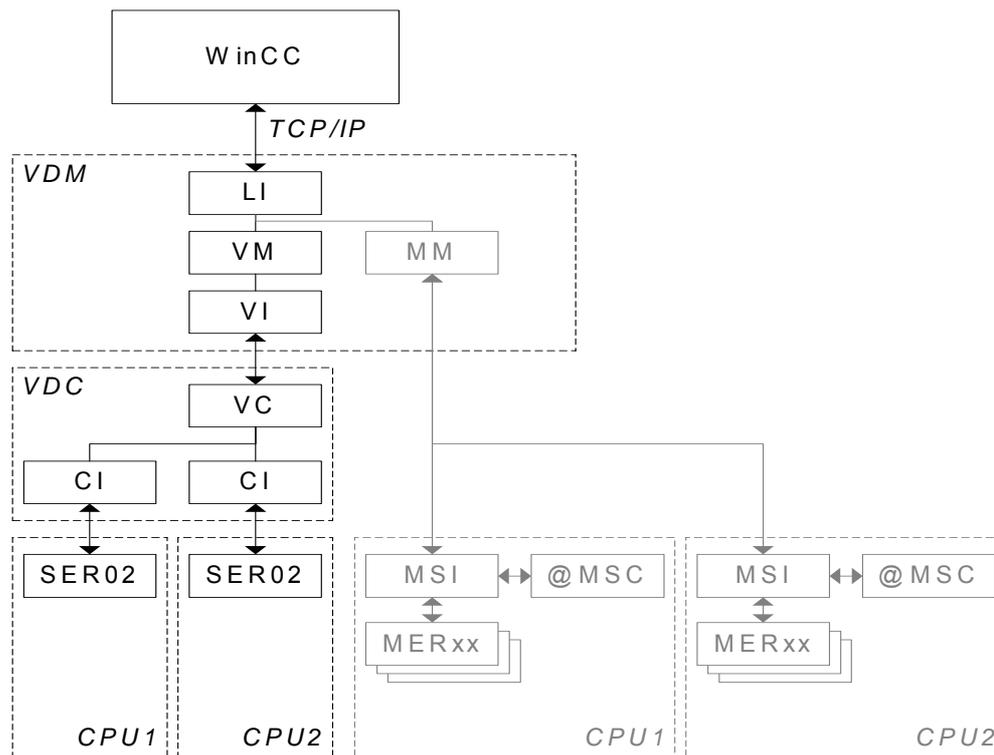


#### 4.6.2 CP51M1 与 WinCC 的 PMC 通讯

CP51M1 可以与 SIMATIC WinCC 通过工业以太网以 TDC PMC TCP 通道进行通讯，需要给 WinCC 增加额外的 DLL 软件，不但可以访问过程数据和还可以访问过程消息。但其组态有些繁琐，需要安装 SIMATIC TDC PMC TCP/IP 驱动在 WinCC 操作员站上，并且一定要安装相应的授权。另外还要安装准备工具软件 PROBI 和地址列表导入工具软件 ADRIMP(安

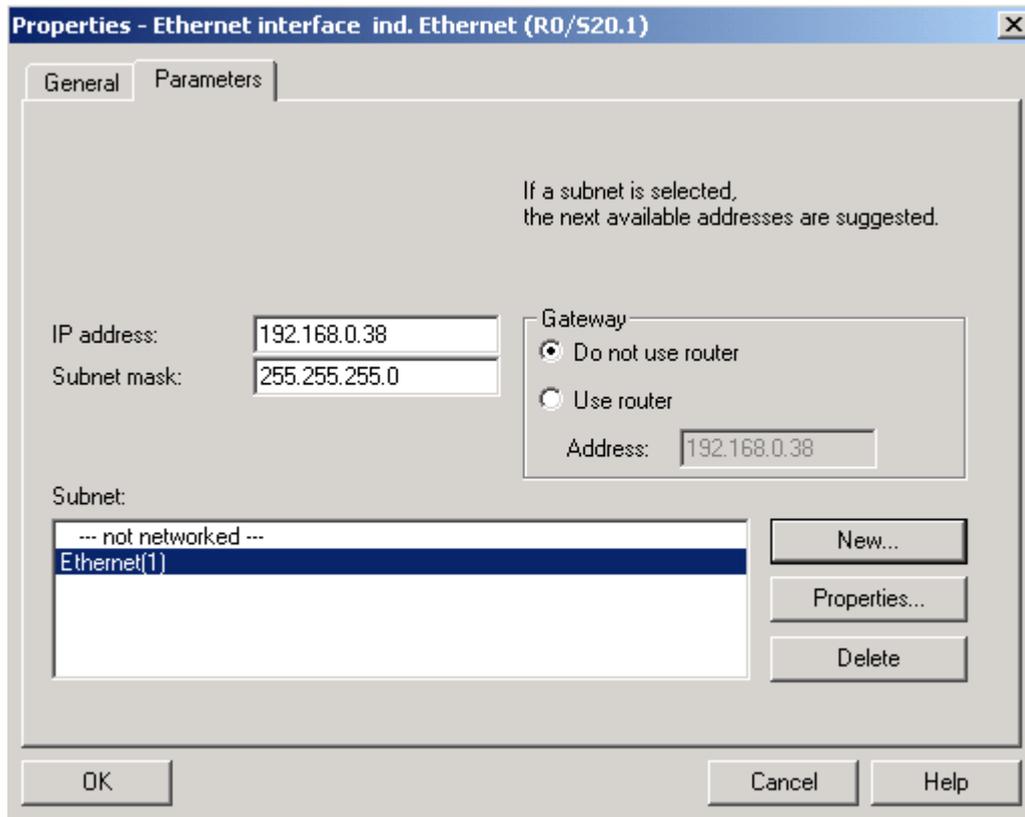
装 TDC PMC TCP/IP 的程序目录中)。TDC 的 CP5100 模板只能够通过 TDC PMC TCP/IP 驱动来实现 TCP/IP 的通讯。组态方法与 CP51M1 的 PMC 方式类似。

下图为 PMC 在 TDC 编程的功能框图，VDM 即 Visualization data manager，表示可视化数据管理。VDC 即 Viasulization data concentrator，表示可视化数据集中。该功能框图表示通过 PMC 通道可以处理过程数据（实体部分）和消息数据（虚体部分）。



#### 4.6.2.1 在 SIMATIC Manager 中进行硬件组态

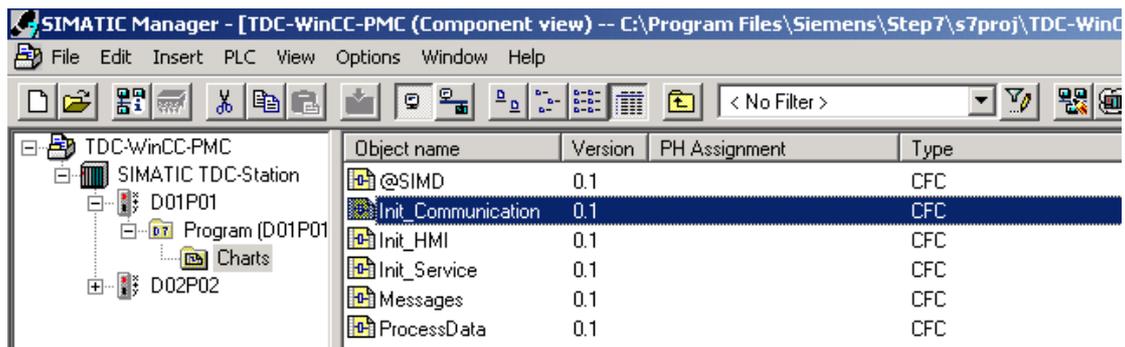
- 硬件组态与 4.1 章硬件组态相同。如果 PC 的网卡端口不是自适应交叉接，那么要使用交叉接的网线与 CP51M1 相连。可以使用带有自适应交叉接的交换机，这样可以使交叉接或平行接的网线。
- 修改 CP51M1 的硬件组态信息。双击硬件组态 D2000C.X1，新建 Ethernet(1)，并设置 IP 地址 192.168.0.38，子网掩码 255.255.255.0。



- 计算机 WinCC 操作员站使用标准的以太网电缆和以太网网卡。

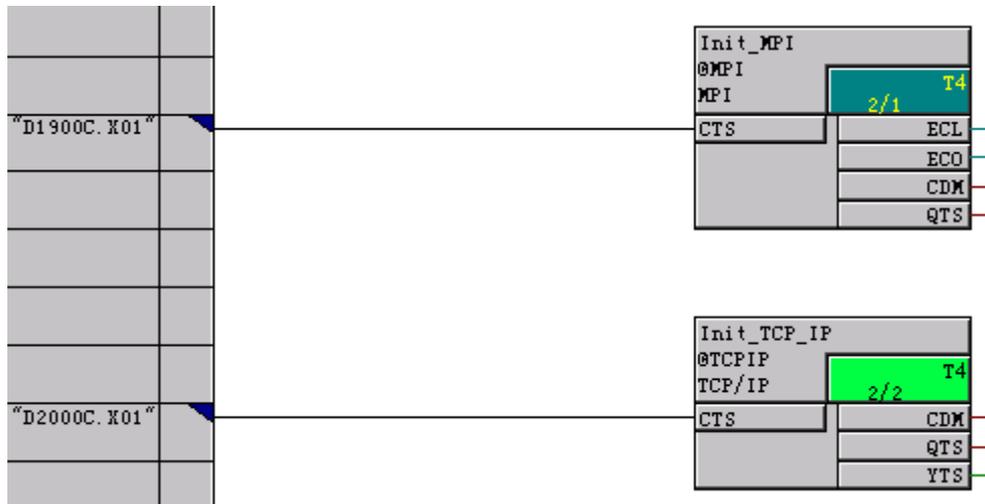
#### 4.6.1.2 在 SIMATIC Manager 中进行软件组态

- 新建一个项目，命名为 TDC-WinCC-PMC。在 CPU1 即 D01P01 的 Charts 表中插入新的 CFC 程序，命名为 Init-Communication。用于初始化通讯。双击打开。



- 插入@TCPIP 功能块到表中，连接在硬件组态中的硬件模板地址 D2000C.X01。即利用 CP51M1 的 X1 作为与 WinCC 通讯的 Ethernet 接口。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32\text{ ms} \leq TA \leq 256\text{ ms}$ 。插入@MPI 功能

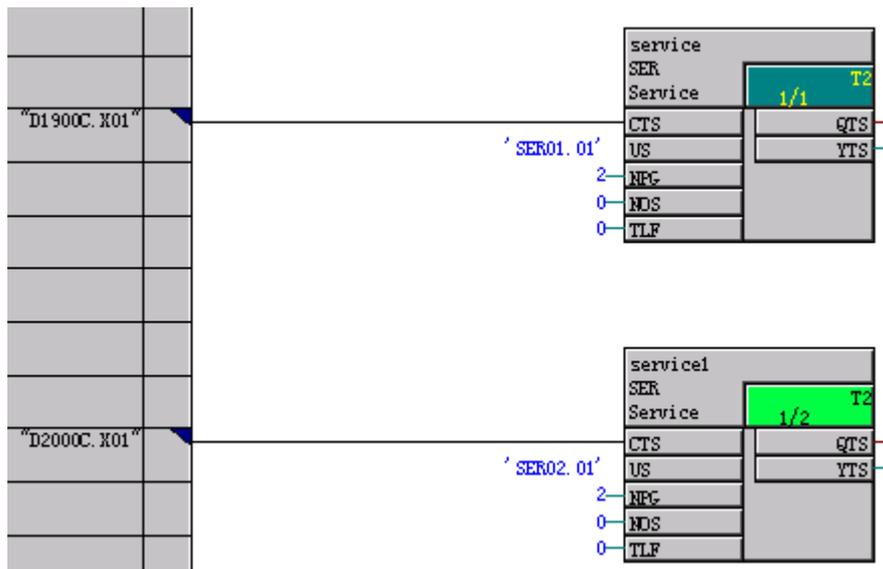
块到该表中，连接在硬件组态中的硬件模板地址 D1900C.X01。即利用该 MPI 接口用于通讯任务。并把该功能块组态到 T4 的采样时间。因为该功能块要求组态到  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$ 。



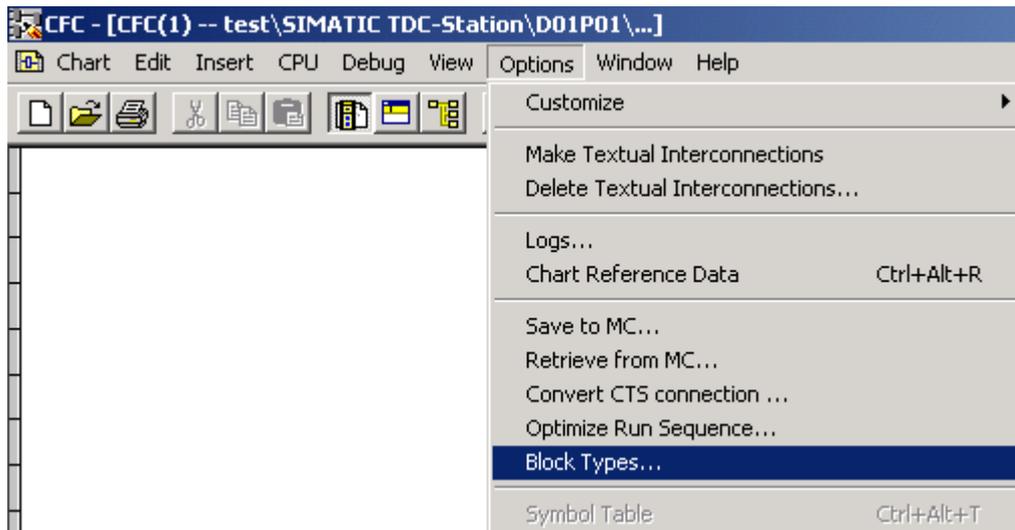
- 在该程序中一定要组态@GLOB 功能块，用于初始化和监控关联的通讯缓冲区。在 CP50MO 中建立通讯缓冲区，这样 WinCC 与 TDC 通过该数据缓冲区交换过程数据和消息数据。

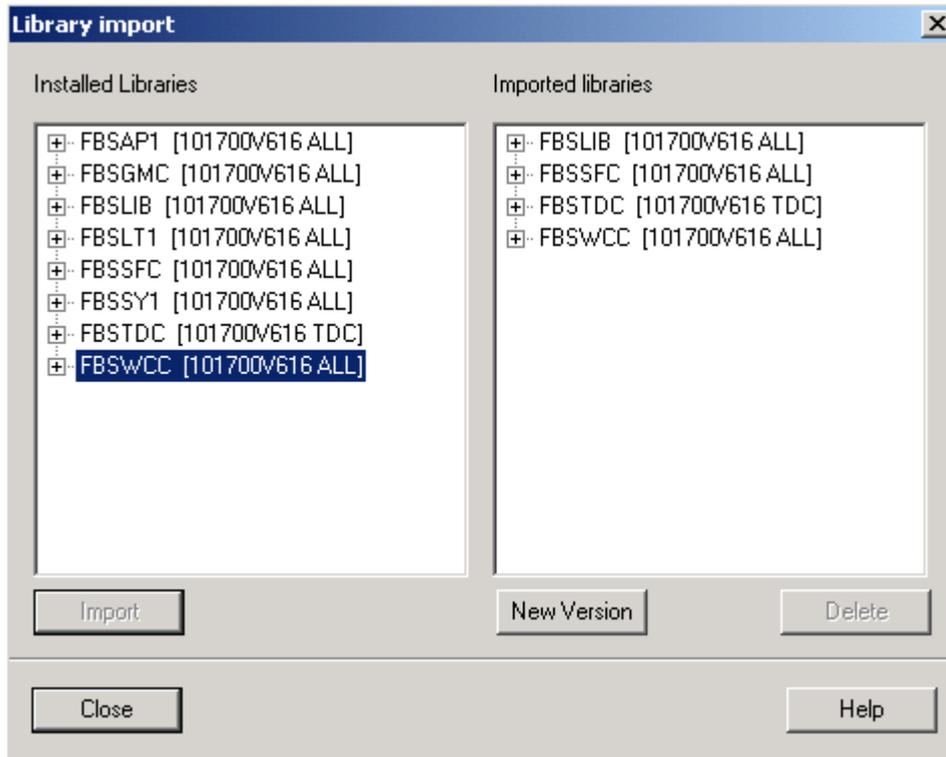


- 在 CPU1 即 D01P01 的 Charts 表中再插入新的 CFC 程序，命名为 Init-Service。用于初始化服务。双击打开。服务功能块关联在 MPI 接口或 Ethernet 接口，就可以使用该接口进行下载程序和诊断 TDC。

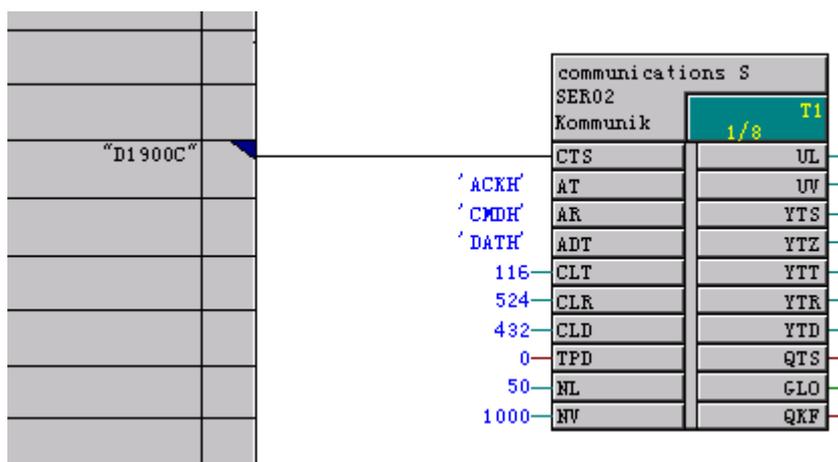


- 首先在 CFC 的菜单 Options 选择 Block types..., 弹出 Library import 画面, 需要导入所需要的库。点击左侧 FBSWCC 库, 点击 Import, 这样该库就导入到右侧栏, 此时可以使用该库进行编写关于 PMC 程序。





- 在 CPU1 即 D01P01 的 Charts 表中再插入新的 CFC 程序，命名为 Init-HMI。用于初始化人机界面即 WinCC。在该 CFC 中按照 VDC 和 VDM 功能框图编写程序。先介绍过程数据功能框图。插入 SER02 功能块。连接在硬件组态中的硬件模板地址 D1900C。即利用 CP50MO 所建立的通讯缓冲区来实现数据交换。为减少对 TDC 的 CPU 的运算负载可以把该功能块组态到  $16 \text{ ms} \leq TA \leq 512 \text{ ms}$ 。



SER02 功能块的引脚说明：

该功能块可以读、写由 CP50MO 模板的数据缓冲区通道的数据。

AT—连接到 CI 功能块的发送通道名。需要与 CI 功能块的接收通道名一致。

AR—连接到 CI 功能块的接收通道名。需要与 CI 功能块的发送通道名一致。

ADT—连接到 CI 功能块的数据通道名。需要与 CI 功能块的数据通道名一致。

CLT—发送通道的数据长度，单位为 Byte。对于连接到 CI 功能块那么必须写入 116。

CLR—接收通道的数据长度，单位为 Byte。对于连接到 CI 功能块那么必须写入 524。

CLD—数据通道的数据长度，单位为 Byte。对于连接到 CI 功能块那么必须写入 432。

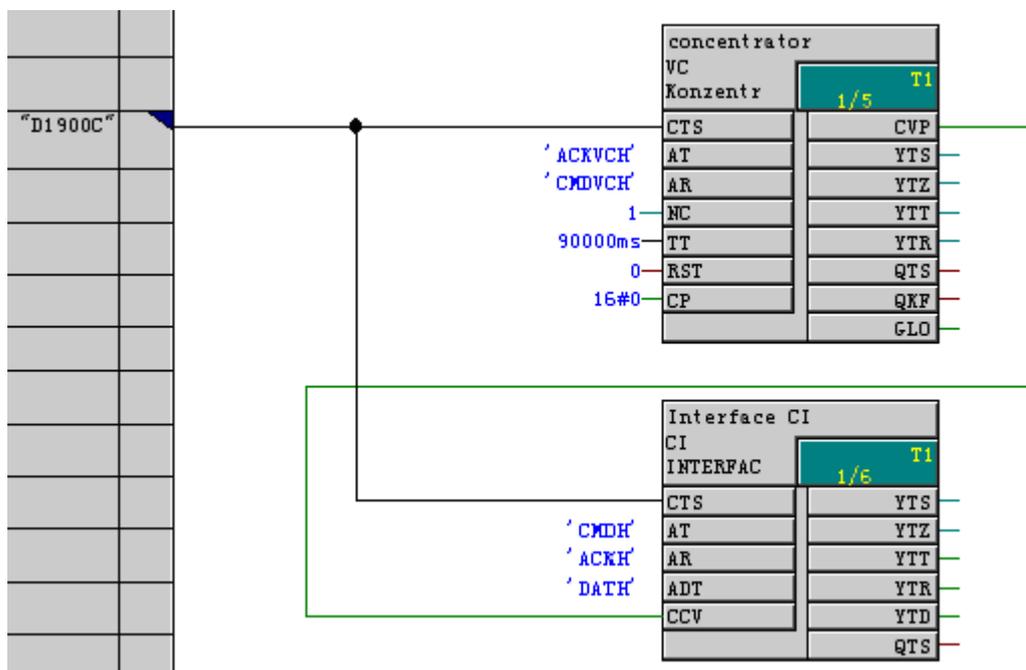
TPD—定义数据交换的传输模式，‘0’为握手模式，‘1’为刷新模式。对于连接到 CI 功能块那么必须是握手模式 0。

NL—定义可以有多少个数据列表可以同时导入。一个数据列表占 20bytes。

NV—定义可以有多少个数值可以被数据列表得到。一个数值占 12bytes。

- 建立 VDC 功能框图。插入 CI 功能块。该功能块是 VC 和 SER02 的接口功能块。

插入 VC 功能块，该功能块是可视化集中器。用于与 VI 功能块连接。



CI 功能块的引脚说明：

AT—连接到 SER02 功能块的发送通道名。需要与 SER02 功能块的接收通道名一致。

AR—连接到 SER02 功能块的接收通道名。需要与 SER02 功能块的发送通道名一致。

ADT—连接到 SER02 功能块的数据通道名。需要与 SER02 功能块的数据通道名一致。

CCV—用于连接到 VC 功能块的输出引脚 CVP。

VC 功能块的引脚说明：

该功能块在 VDM 和 SER02 功能块间针对每一个单独的 CPU 建立一个连接。从 VI 中接收到数据时，会拆分出来给下一级的 CI 功能块，再传给 SER02 功能块，即把数据传送给 CPU1 或 CPU2。CPU1 和 CPU2 发送给 WinCC 的数据会通过 VC 功能块组合集中，在传送给 VI 功能块。该功能块的采样时间，应该尽可能的短，要低于在 WinCC 中组态的变量的采样时间。

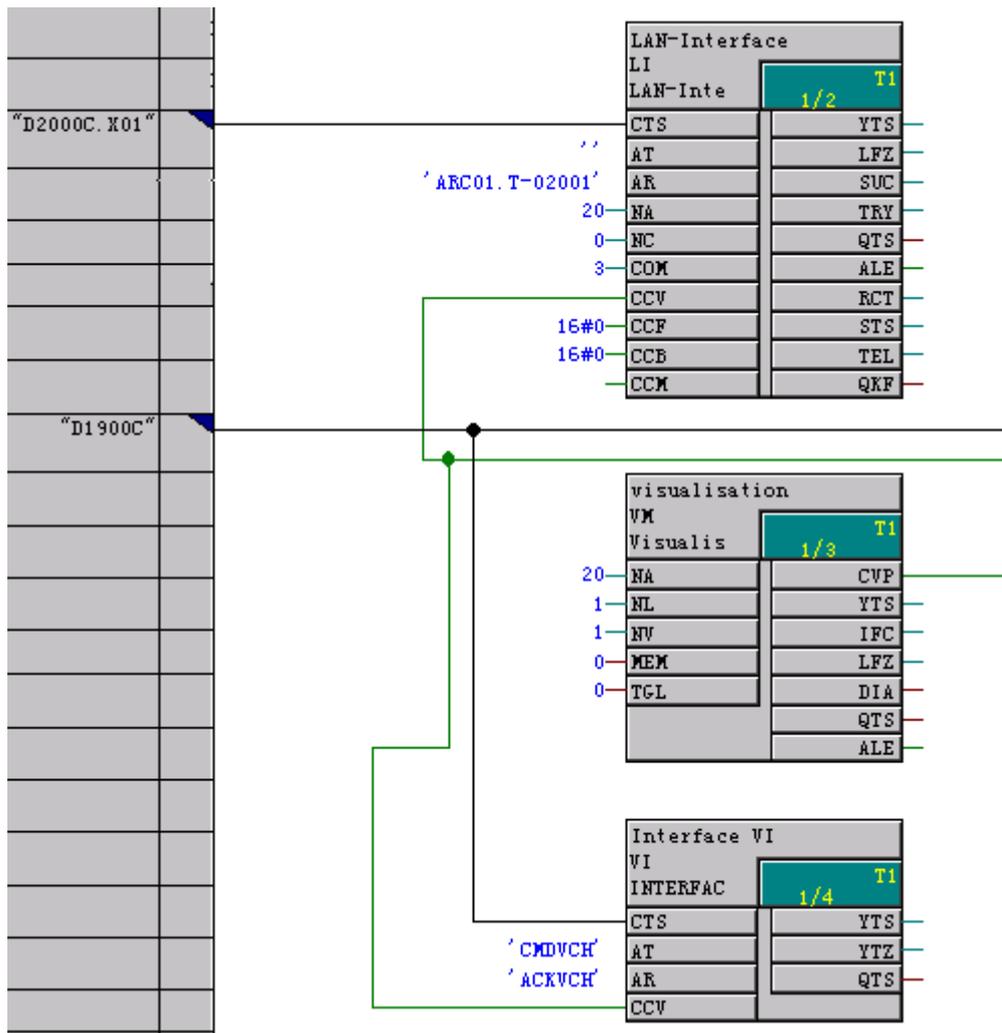
AT—连接到 VI 功能块的发送通道名。需要与 VI 功能块的接收通道名一致。

AR—连接到 VI 功能块的接收通道名。需要与 VI 功能块的发送通道名一致。

NC—表示 VC 功能块所连接 CI 功能块的数量。本例中为 1。

CVP—用于连接到 CI 功能块的输入引脚 CCV。

- 建立 VDM 功能框图。插入 VI 功能块。该功能块是 VC 和 VM 的接口功能块。插入 VM 功能块，该功能块是用于管理 WinCC 过程任务。插入 LI 功能块，该功能块用于建立 WinCC 和 VDM 功能框图的通讯接口。



VI 功能块的引脚说明:

VI 功能块的采样时间需要组态小于或等于 VM 功能块的采样时间。并且应该尽可能的短，要低于在 WinCC 中组态的变量的采样时间。

AT—连接到 VC 功能块的发送通道名。需要与 VC 功能块的接收通道名一致。

AR—连接到 VC 功能块的接收通道名。需要与 VC 功能块的发送通道名一致。

CCV—用于连接到 VM 功能块的输出引脚 CVP。

VM 功能块的引脚说明:

该功能块的采样时间需要组态到大于或等于所有 LI 功能块和 VI 功能块。允许最大的采样时间为 1024ms。当运行时，VM 功能块允许与其 CVP 相连的 LI 和 VI 功能块登陆，即 LI 和 VI 通过 VM 功能块实现数据的双向通讯。一个 VM 可以与多个 LI 功能块相连，表示数据可

以传送给多个 WinCC 系统。或者是多个 VM 可以与一个 LI 功能块相连，表示一个 WinCC 系统可以显示多个 CPU 的数据。

NA—定义同时可以被处理的最大任务数，该参数定义至少要与全部的 LI 功能块 NA 所定义的任务数的和同样大。

NL—连接到 LI 功能块的数量。

NV—连接到 VI 功能块的数量。

CVP—用于连接到 VI 功能块的输入引脚 CCV 和 LI 的输入引脚 CCV。

LI 功能块的引脚说明：

LI 功能块的采样时间需要组态小于或等于所有 VM, FM, BM, MM 功能块的采样时间。并且应该尽可能的短，要低于在 WinCC 中组态的变量的采样时间。

CTS—与所关联的 WinCC 通讯的以太网接口，即 CP51M1 的 X01 接口。

AT—连接到 WinCC 的发送通道名。

AR—连接到 WinCC 的接收通道名。本质上与发送通道名一致。

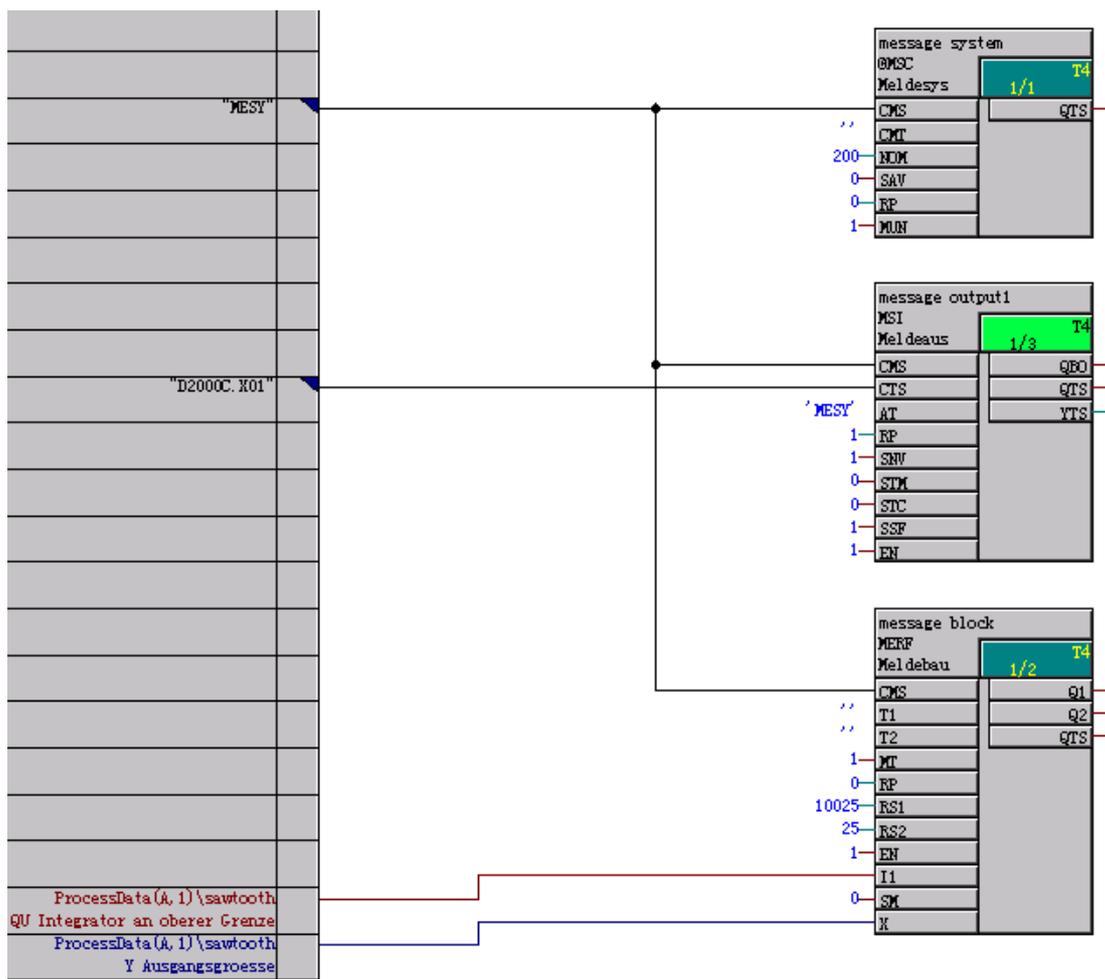
NA—定义同时可以被处理的最大任务数，最多 41 个。该参数尽可能设的大些，这样可以保证在 WinCC 中所组态的显示任务和 MM 的任务能够并行运行和处理。

NC—定义一个 WinCC-ID，用于多个 LI 功能块与一个 VM 连接的逻辑地址区分。

COM—定义所使用的通讯协议。

CCV/CCF/CCB/CCM—用于连接到相应功能块 VM/FM/BM/MM 的输出引脚 CVP。其中只有 VM 与 MM 与 TDC 相关，那么对于 CCF 和 CCB 必须设置为 0。

- 在 CPU1 即 D01P01 的 Charts 表中再插入新的 CFC 程序，命名为 Messages。用于初始化消息系统。介绍消息数据功能框图。插入 @MSC 功能块。该功能块来初始化和监控消息系统。把该功能块组态到  $32\text{ms} \leq TA$ 。插入 MSI 功能块，该功能块用于输出在消息系统中的消息。插入 MERF 功能块，该功能块用于往消息系统中输入消息。



### @MSC 功能块的引脚说明:

对于每一个 CPU 只能组态一个 @MSC 功能块。且该 @MSC 功能块监测和管理该 CPU 中的消息。一个消息包含一个文本，两段号码（前缀和后缀），一个类型，以及可以选择的检测值。一个文本就是指消息文本。两段号码（前缀和后缀）用于区分消息的高低优先级。类型包括 S/C/F/W，分别为系统错误消息/通讯错误消息/一般错误消息/一般报警消息。

CMS—定义一个消息系统名称。最多 6 个字符。

CMT—用于发生通讯错误时产生的消息文本。如果不设置该参数，那么有通讯错误发生时，只会出现消息号和符号‘C’。

NOM—定义消息系统的大小，消息数最多为 32767，最小为 15 个。

RP—设为 0，表示消息产生是系统错误消息或通讯错误消息，后缀自动产生。

MUN—使能该消息系统。

MSI 功能块的引脚说明:

MSI 可以组态多次在同一个 CPU 中, 根据其接口 CTS 的不同, 消息可以输出给不同的接口。经过每一个循环周期, 输出一个消息。

CMS—用于连接由@MSC 所建立的消息系统名称。

CTS—用于经过该接口发送消息。

AT—初始化地址名, 用于在关联的该接口上建立消息发送通道。本例中与消息系统的名称相同。

RP—如果消息系统满, 会产生溢出消息, 该消息的前缀会是该参数。本例为 1。

SNV—1 表示输出消息号, 否则, 0 不输出。

STM—1 表示输出消息文本, 否则, 0 不输出。

STC—0 表示消息文本按照实际的文本长度输出。否则按照固定的文本长度输出。

SSF—1 表示消息输出按照标准模式输出, 否则按照 16 进制输出。

EN—1 使能消息输出。

MERF 功能块的引脚说明:

MERF 可以在消息系统中输入两个消息。使用该功能块时, 一定需要@MSC 和 MSI 功能块。通过该功能块可以定义一个消息的一个文本, 两段号码 (前缀和后缀), 一个类型, 以及可以选择的检测值。

CMS—用于连接由@MSC 所建立的消息系统名称。

T1—初始化激活的消息文本, 最多 60 个字符。

T2—初始化非激活的消息文本, 最多 60 个字符。

MT—1 表示输入的是错误信息, 否则是报警信息。

RP—消息前缀。

RS1—激活消息的后缀。

RS2—非激活消息的后缀。

EN—1 功能块使能。

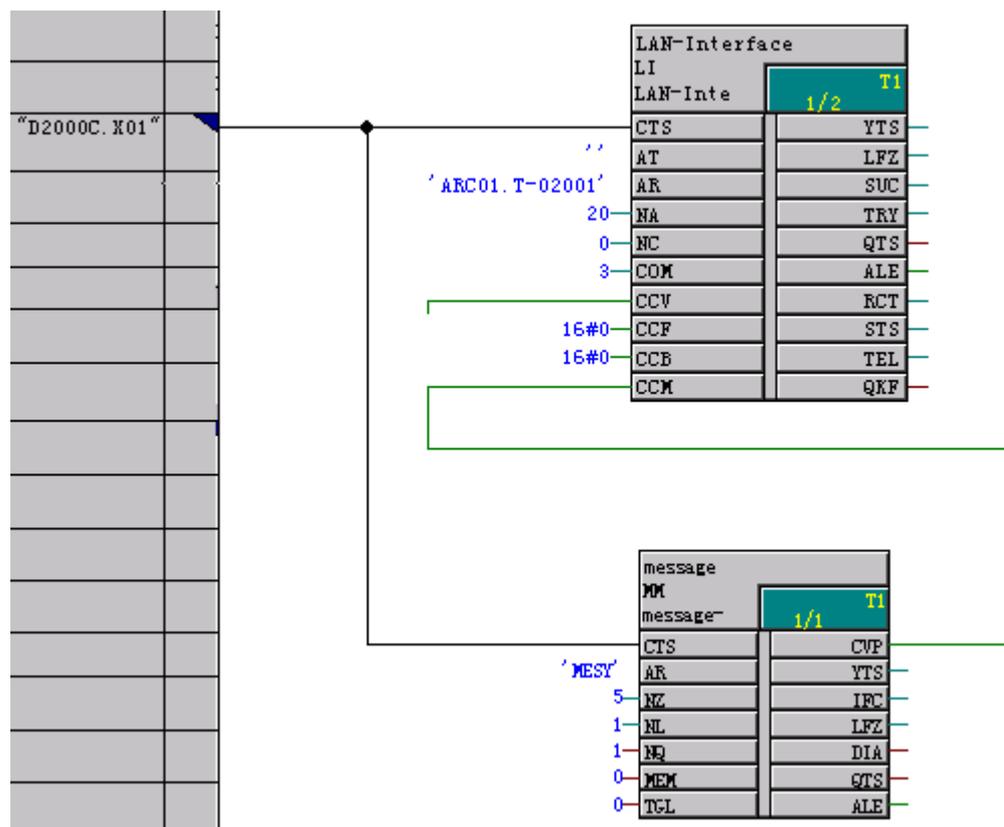
I1—用于消息触发。由 0 变化到 1 表示激活的消息文本 T1 加上激活消息的后缀 RS1。由 1 变化到 0 表示非激活的消息文本 T2 加上非激活消息的后缀 RS2。本例中该参数与积分器的

输出是否达到上限 QU 相连。

SM—1 表示禁止消息被重新输出。

X—检测的值。本例中该参数与积分器的输出 Y 相连。

- 打开 Init\_HMI 程序，在 VDM 的功能框图中加入 MM 功能块。该功能块是处理 TDC 消息的一个管理功能块。



MM 功能块的引脚说明：

MM 功能块的组态采样周期必须大于或等于所有与之相连功能块 LI 的采样周期。

CTS—用于初始化组态关联的模块名称，在这个数据接口上建立接收通道。

AR—所建立的接收通道名，与 MSI 功能块的 AT 相对应。

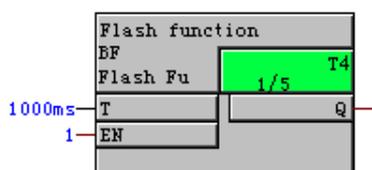
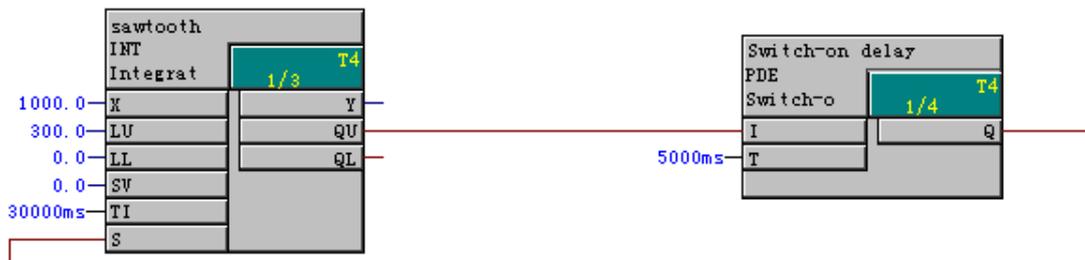
NZ—初始化 MM 功能块的循环数，用于给 WinCC 发送两个连续的消息的间隔。

NL—表示连接 LI 功能块的个数。

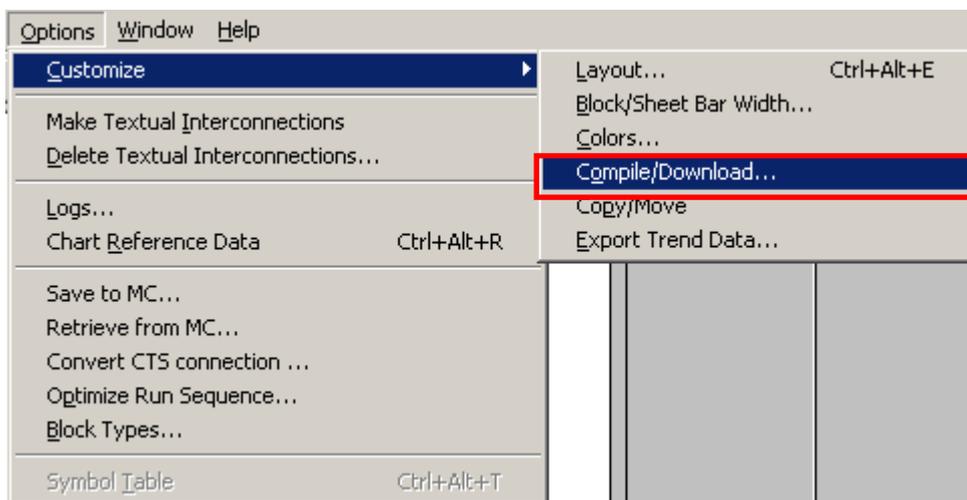
NQ—初始化对于到来和离开的消息的应答，0 表示由应答，1 表示没有应答。

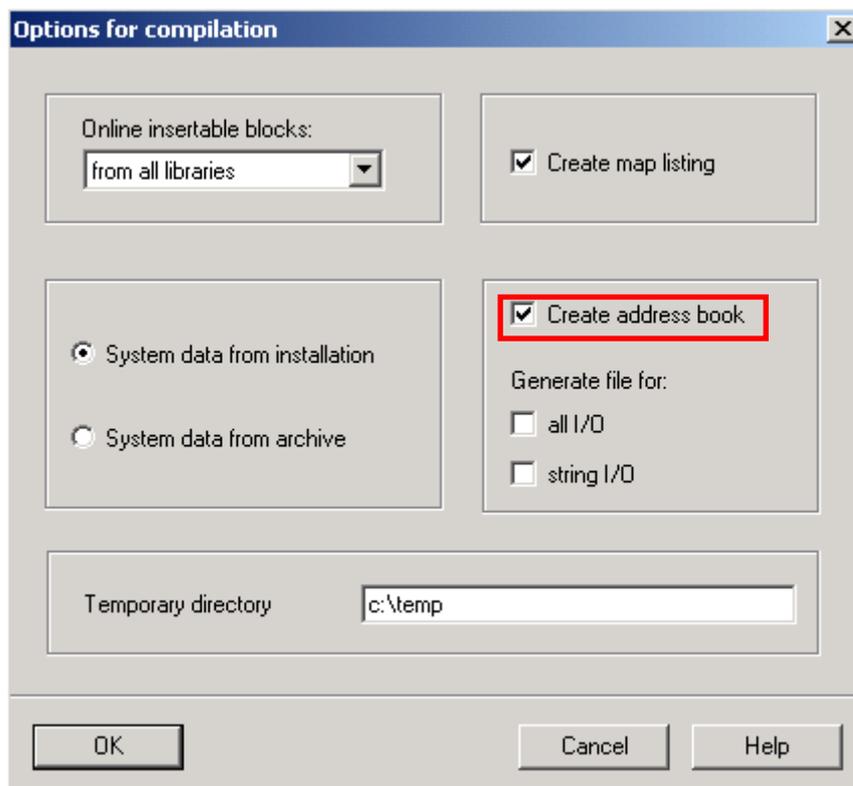
CVP—与 LI 功能块的 CCM 相连。

- 在 CPU1 即 D01P01 的 Charts 表中再插入新的 CFC 程序，命名为 ProcessData。  
用于初始化过程数据。在 CFC 中定义两个功能，第一个功能通过 INT 命名为 sawtooth 功能块进行积分运算当达到高限 300 时通过 PDE 命名为 Switch-on delay 功能块延迟 5 秒钟复位该积分功能块，然后重新积分。插入 BF 功能块命名为 Flash function 实现 BF.Q 的信号每隔 1 秒翻转一次。

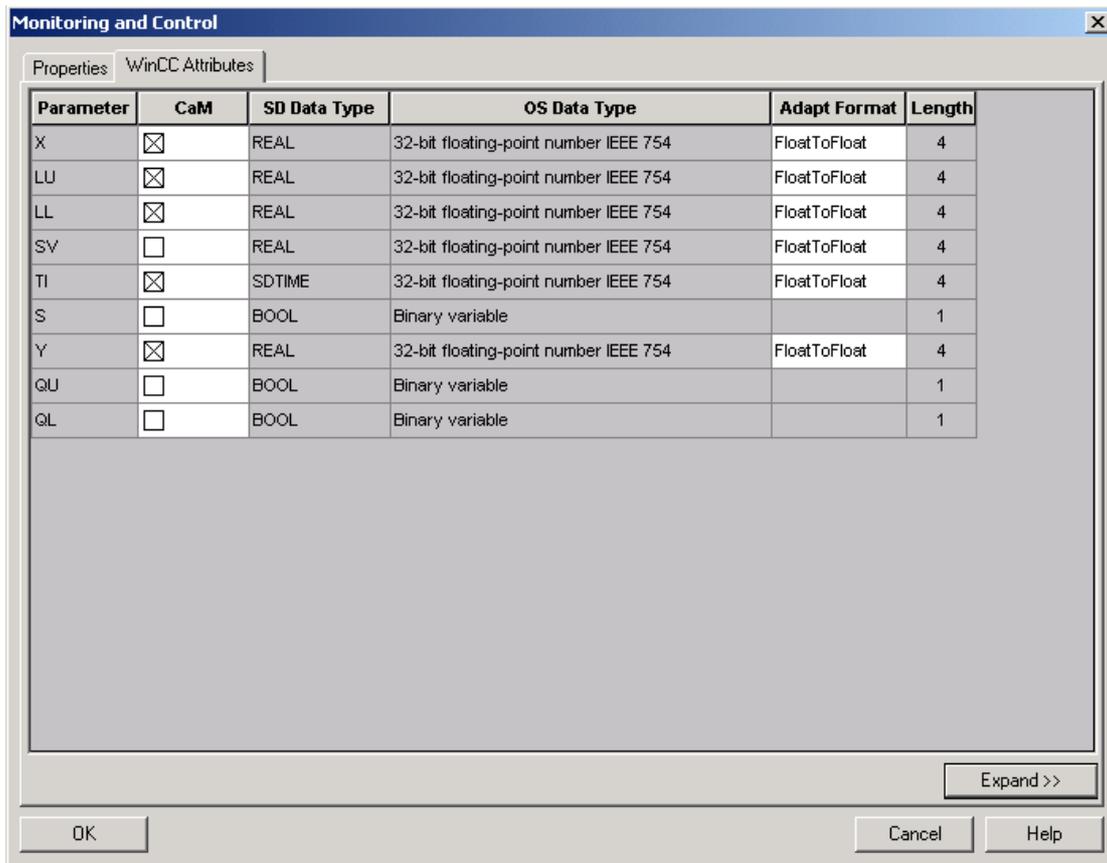


- 在 CFC 的菜单中，选择 Options 中的 Customize，选择 Compile/Download...，选中修改地址本选项 Create address book。

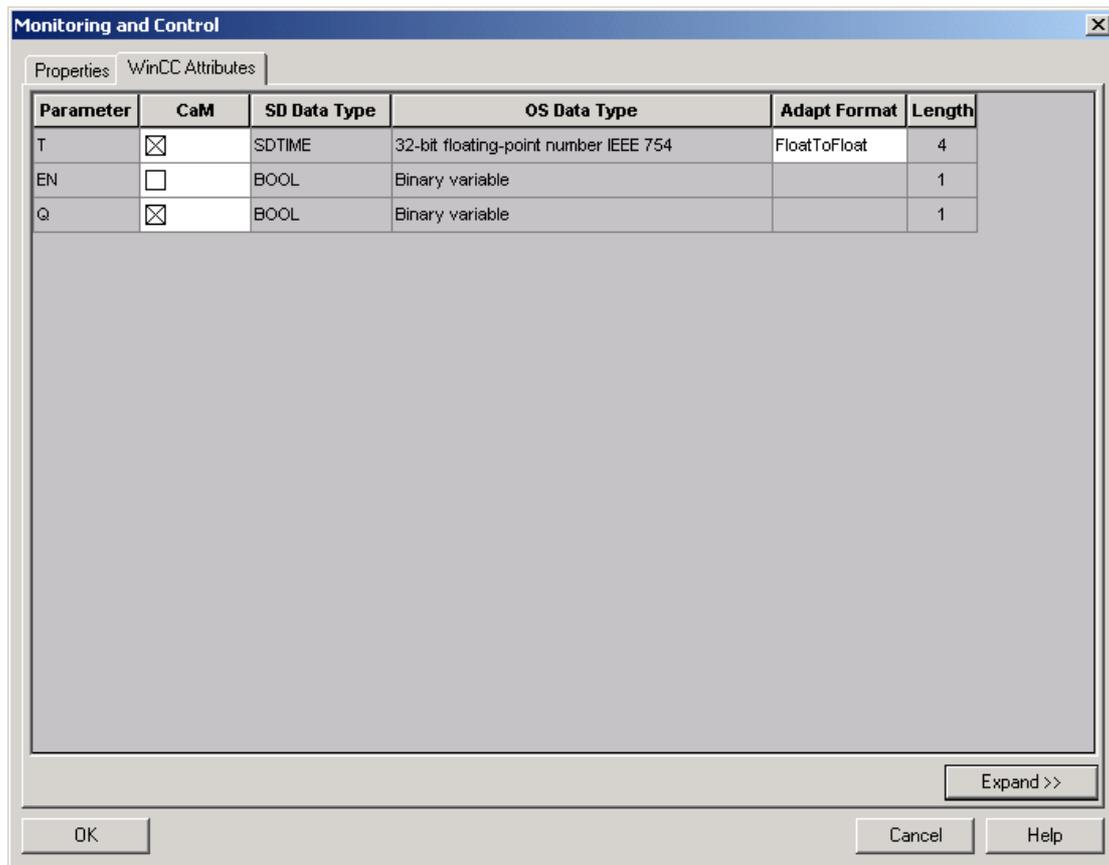




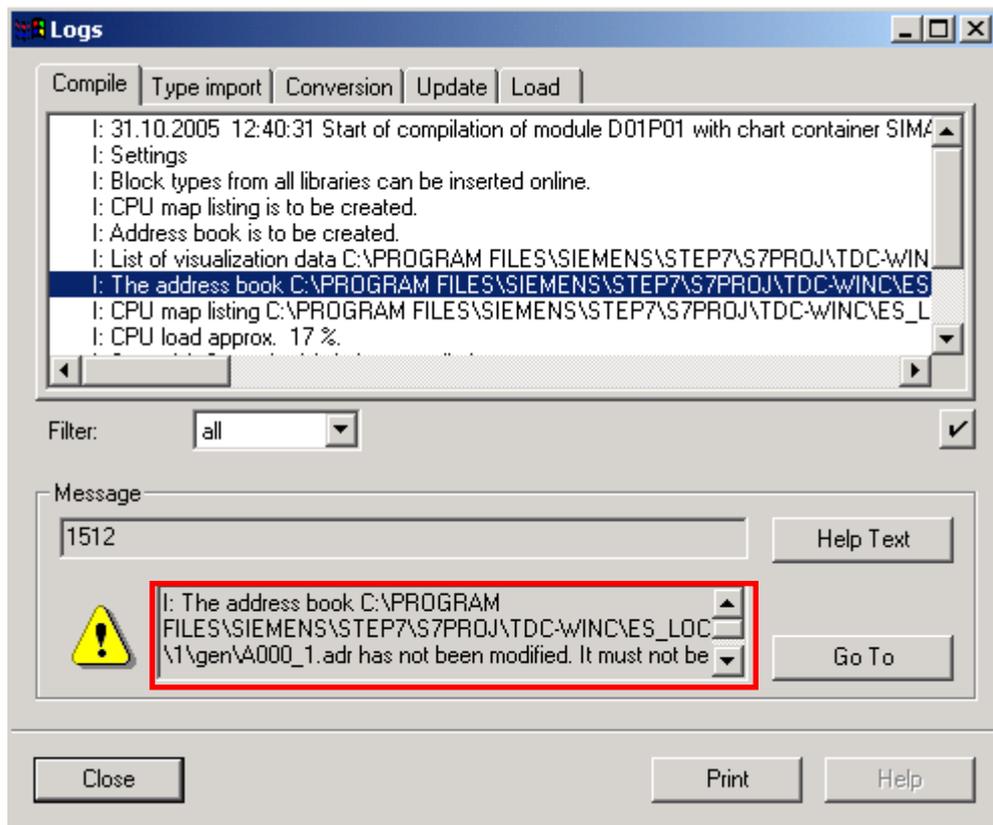
- 双击 INT 功能块, 打开 INT 功能块的属性。选择 OCM possible。点击 Operator C and M..., 打开 Monitoring and Control 属性。注意该功能块的映像区数据块为 1001。可以选中 complete block structure 选项, 选中该选项, 就是对该功能块的所有输入和输出进行监控。在打开的 Monitoring and Control 属性中, 切换到 WinCC Attributes 页面, 选择要监控的变量。并且使用同样的方法选中 BF 功能块的输出 Y 作为监控变量。



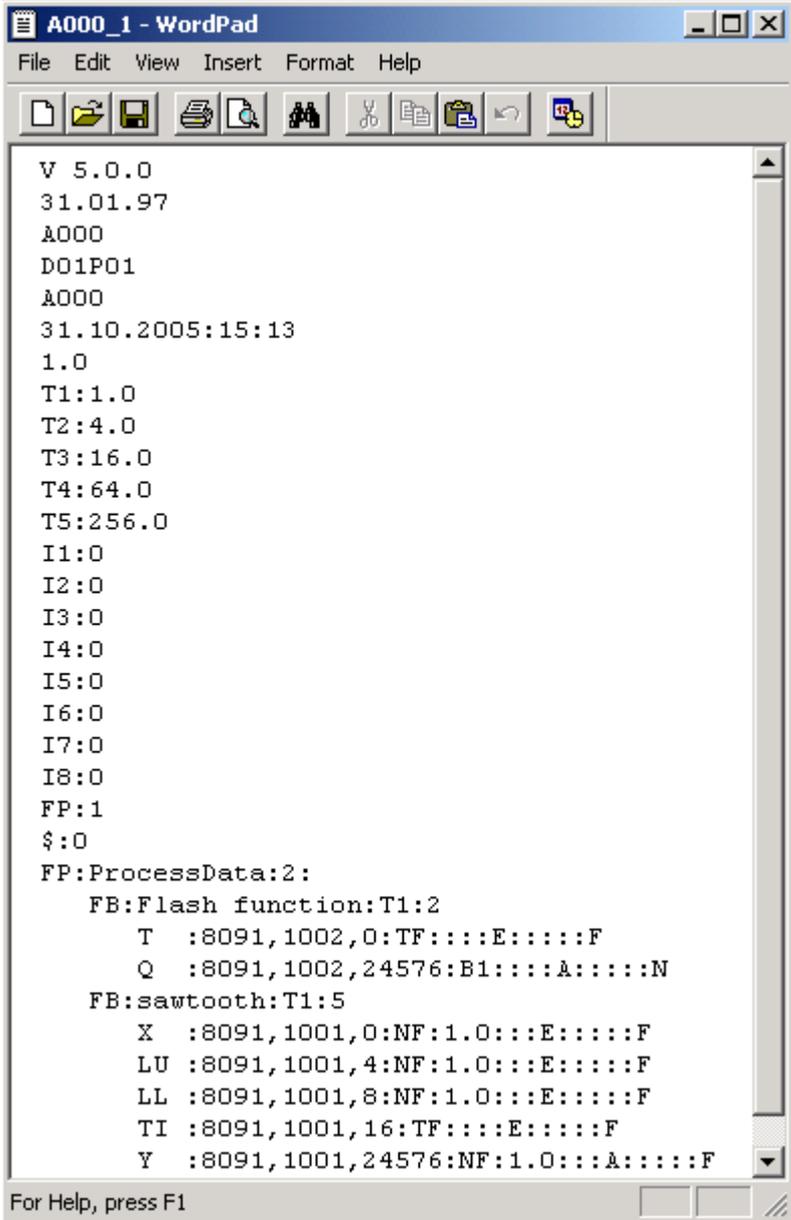
- 对于 BF 功能块，使用相同的方法设置。注意该功能块的映像区数据块为 1002。



- 设置完成后，对 CFC 进行编译，创建地址本，这个地址本包含了所有创建映像 DB 的信息。下图是编译后自动弹出的编译结果，其中包含地址本的路径。如果经过多次编译，系统不会出现直接带有地址本的路径的信息，可以通过点击 **Details** 来获取地址本的路径信息。



- 根据该路径，通过 WordPad 打开该文件。上面记录着 CPU1 即 D01P01 的周期任务（T1~T5）和中断任务(I1~I8)的设置信息。最重要的是提供了所要监控功能块 sawtooth 和 Flash function 的变量和所分别对应的数据功能块的代号 1001、1002，以及对应该变量的偏移地址。



```
V 5.0.0
31.01.97
A000
DO1P01
A000
31.10.2005:15:13
1.0
T1:1.0
T2:4.0
T3:16.0
T4:64.0
T5:256.0
I1:0
I2:0
I3:0
I4:0
I5:0
I6:0
I7:0
I8:0
FP:1
$:0
FP:ProcessData:2:
  FB:Flash function:T1:2
    T :8091,1002,0:TF::::E::::F
    Q :8091,1002,24576:B1::::A::::N
  FB:sawtooth:T1:5
    X :8091,1001,0:NF:1.0::E::::F
    LU:8091,1001,4:NF:1.0::E::::F
    LL:8091,1001,8:NF:1.0::E::::F
    TI:8091,1001,16:TF::::E::::F
    Y :8091,1001,24576:NF:1.0::A::::F
```

- 设置 PG/PC 接口为 CP5512(MPI)下载硬件组态信息和 CFC 程序。通过工业以太网的

网线，连接 TDC 和 WinCC 操作员站，在 WinCC 操作员站上使用 Ping 命令，察看以太网是否连通。

```

C:\>ping 192.168.0.38

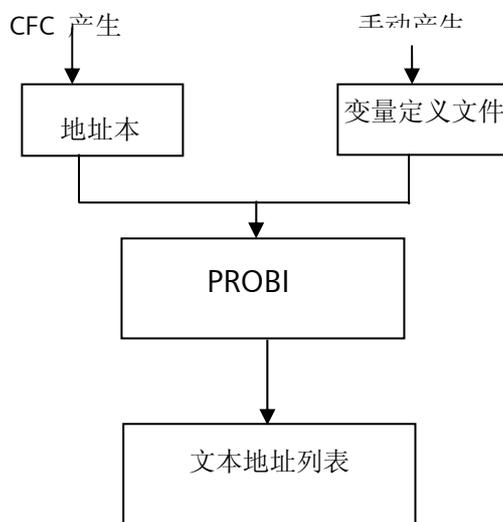
Pinging 192.168.0.38 with 32 bytes of data:

Reply from 192.168.0.38: bytes=32 time<10ms TTL=30

Ping statistics for 192.168.0.38:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
    
```

- 使用 PROBI 软件来实现 WinCC 的地址列表导入。参考下图的地址列表导入方法。



地址本由 CFC 编译自动产生的，变量定义文件由手动编写完成。为了增加软件的透明度，应该使地址本和变量定义文件处于相同的路径中。变量定义文件是一个文本文件，必须严格按照定义的要求来手动完成。

变量定义文件的结构：

[VDM:<LI 的通道连接名>]

[PN:<编译生成的地址本名>]

<WinCC 过程变量名>,<Chart>.<功能块名>.<引脚名>,<I>

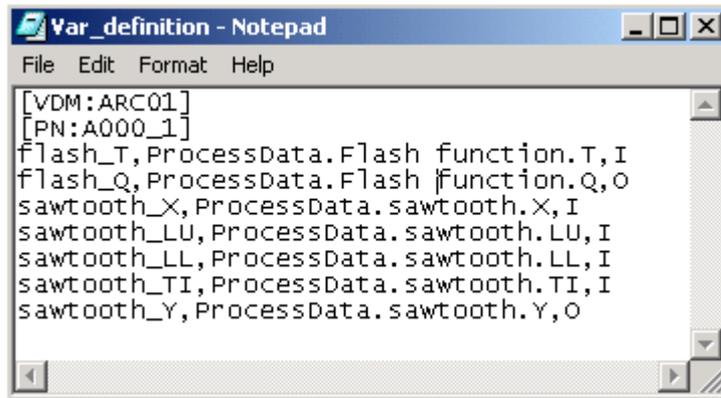
<WinCC 过程变量名>,<Chart>.<功能块名>.<引脚名>,<O>

其中，I 表示输入引脚；O 表示输出引脚。<>符号不定在变量定义文件中。且 WinCC 变

量名可以自由定义。

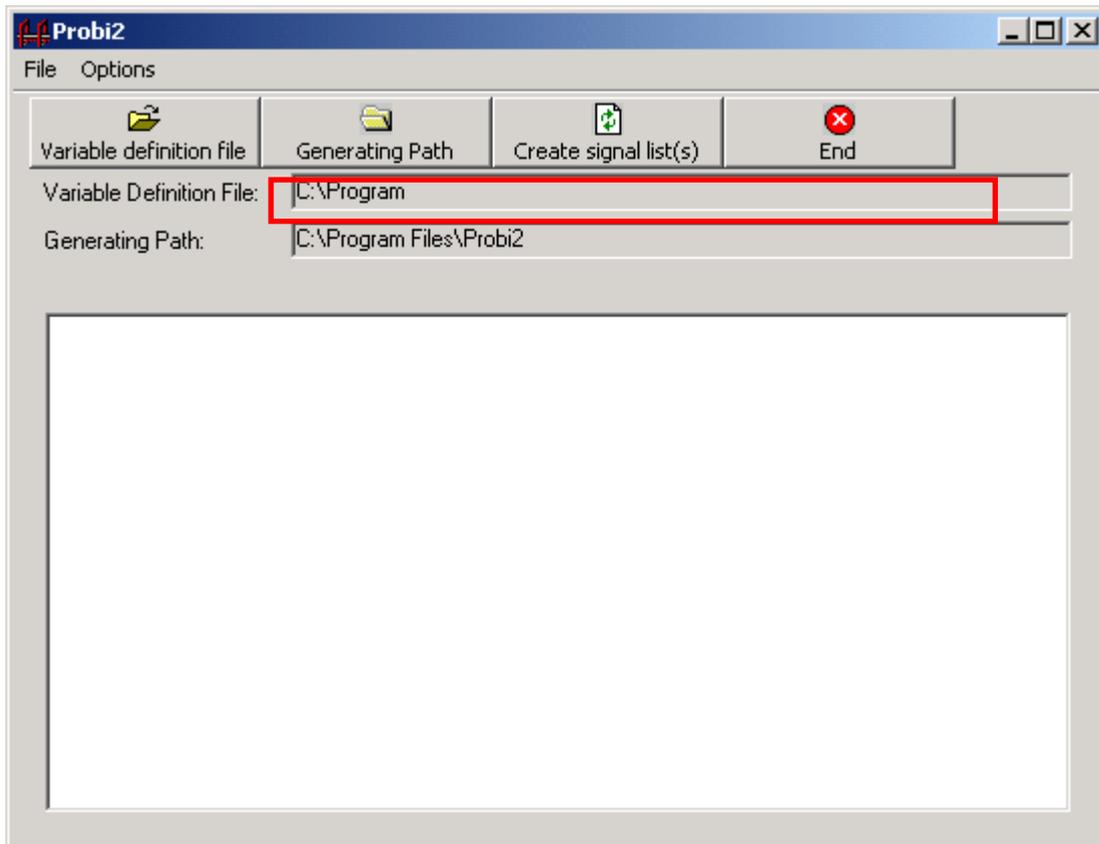
在 C:\Program Files\Siemens\Step7\S7Proj\TDC-WinCIES\_LOC\1\Gen 的目录中，新建一个文本文件，命名为“Var\_definition”。并且打开该文件，按照上述文件结构手动输入该文件。

注意一定要严格按照该文件结构编写。

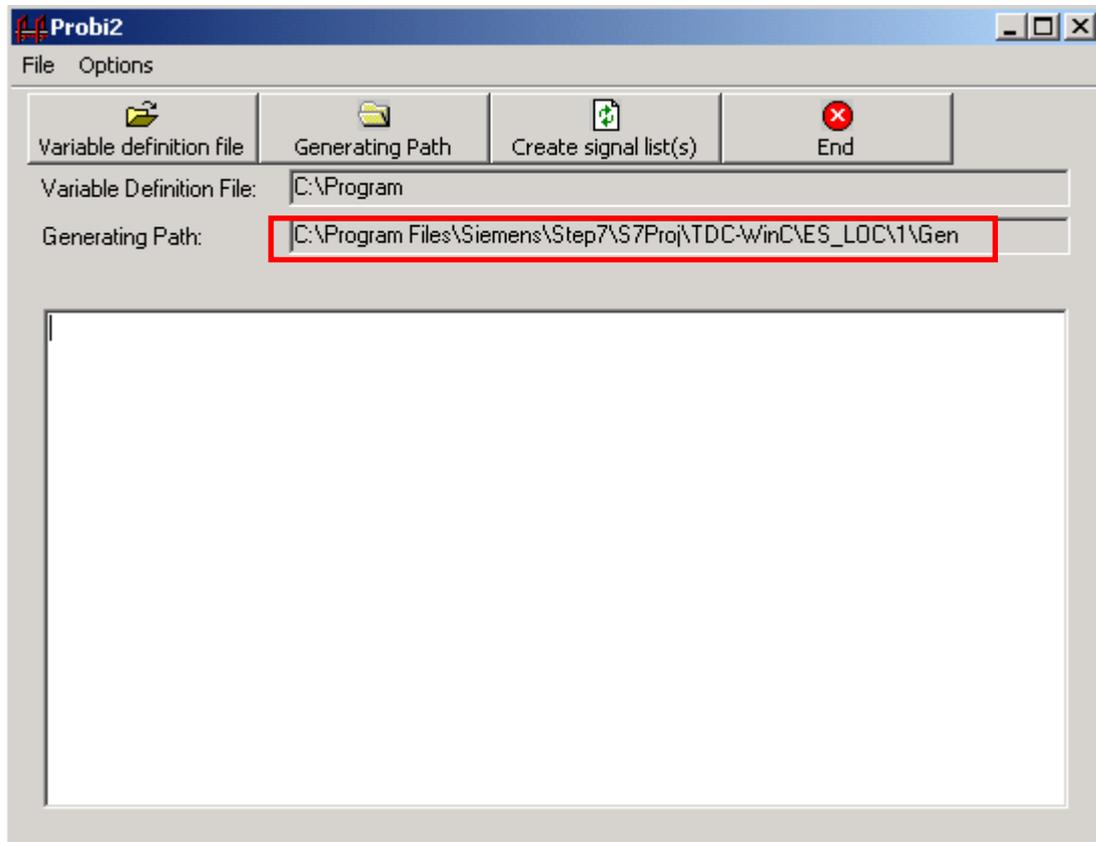


```
[VDM:ARC01]
[PN:A000_1]
flash_T, ProcessData.Flash function.T,I
flash_Q, ProcessData.Flash function.Q,O
sawtooth_X, ProcessData.sawtooth.X,I
sawtooth_LU, ProcessData.sawtooth.LU,I
sawtooth_LL, ProcessData.sawtooth.LL,I
sawtooth_TI, ProcessData.sawtooth.TI,I
sawtooth_Y, ProcessData.sawtooth.Y,O
```

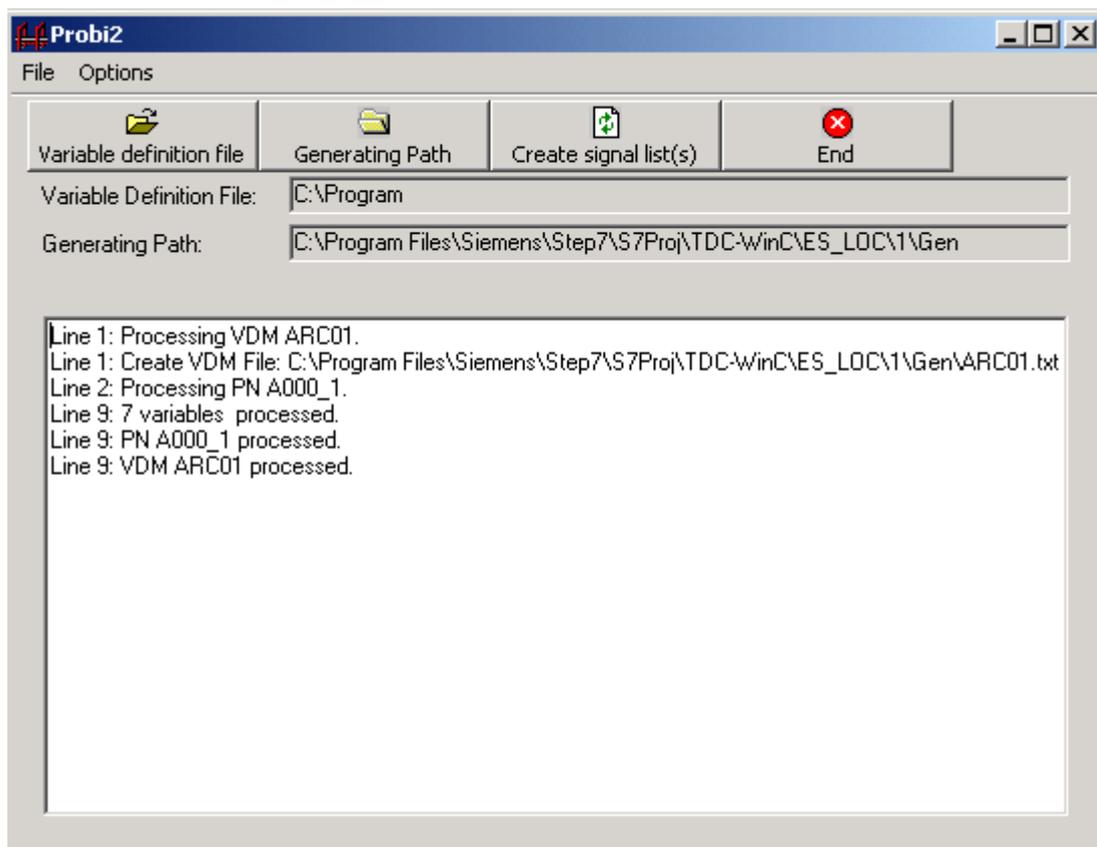
- 打开 Probi 软件，搜索刚定义的变量定义文件 Var\_definition。由于路径很长，只显示 C:\Program。



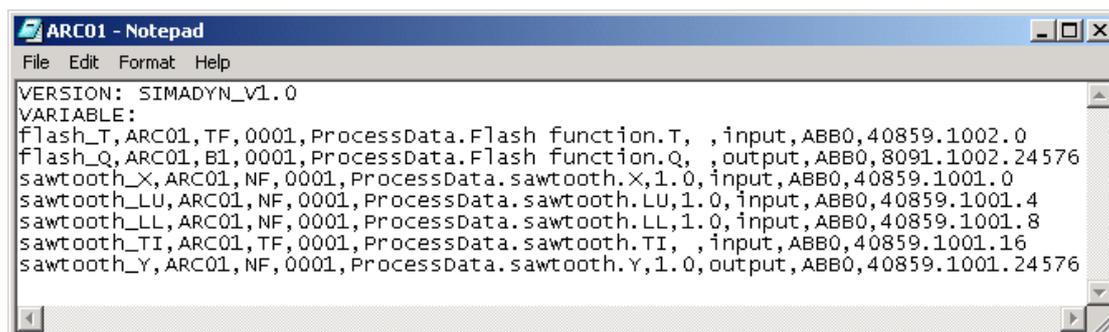
- 点击 Probi 软件中的 **Generating Path**，来定义生成的路径，与变量定义文件的路径相同。



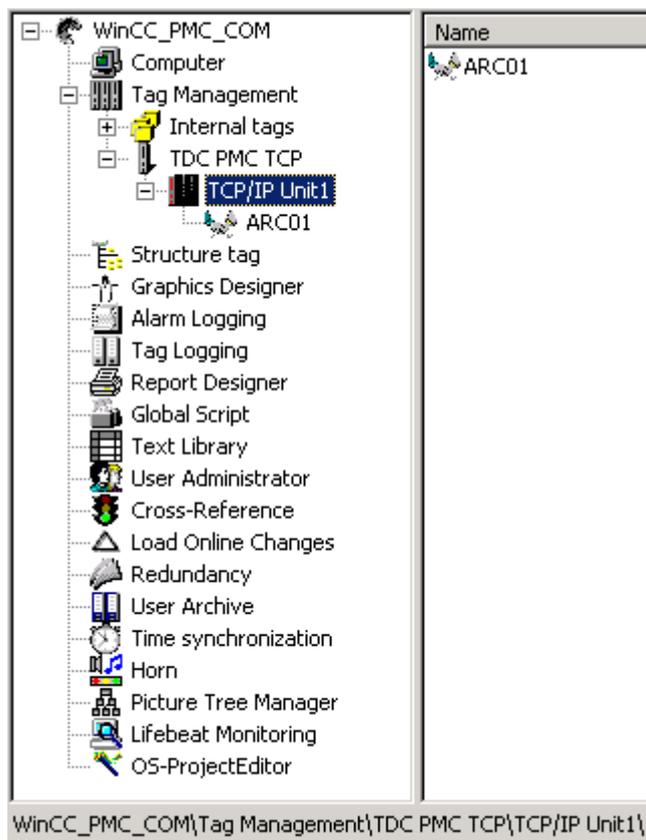
- 点击 Probi 软件中的 **Create signal list(s)**，来生成文件地址列表为 ARC01.txt。并且显示生成列表文件的处理信息。



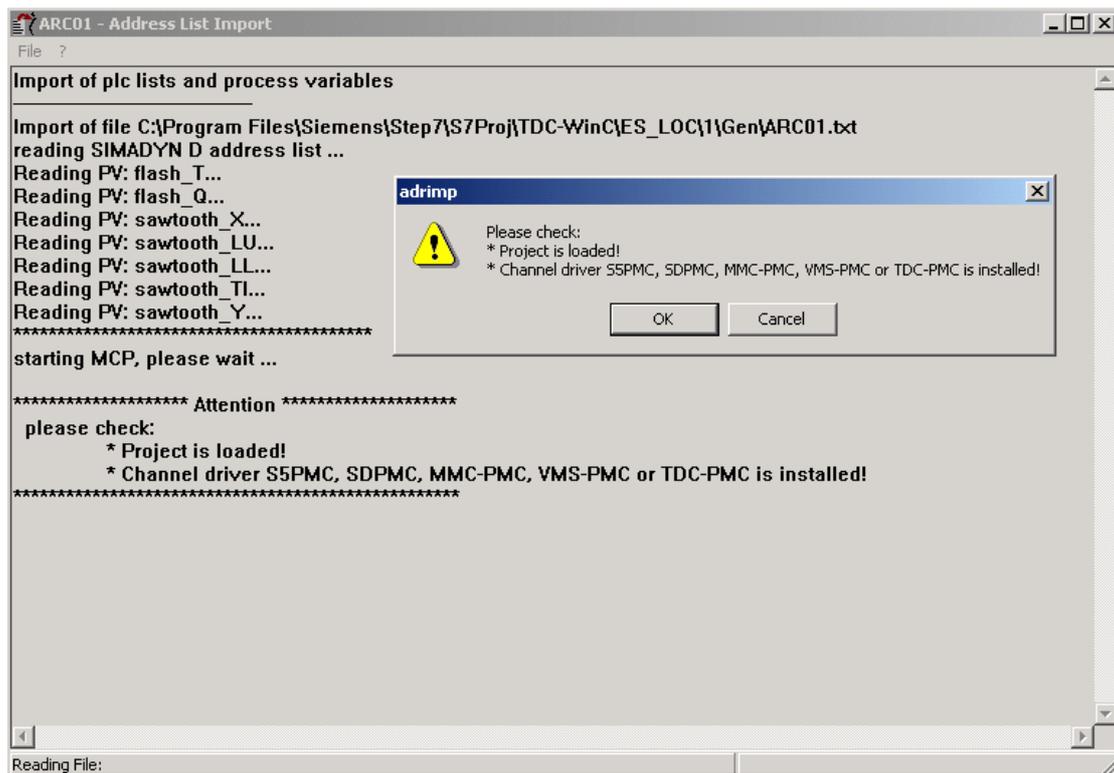
- 退出Probi软件。在C:\Program Files\Siemens\Step7\S7Proj\TDC-WinC\ES\_LOC\1\Gen\中找到ARC01.txt文件，打开察看结果。



- 打开 WinCCExplore，新建工程并命名为 WinCC\_PMC\_COM。在 Tag management 中插入 PMC 驱动协议包，即 TDC PMC TCP。新建一个连接命名为 ARC01，需要注意的是该连接名与地址列表文件名一致。



- 打开 ADRIMP 软件，选择“File”菜单项，选择“Open”菜单项，根据路径找到通过 PROBI 软件生成的地址文件列表 ARC01.txt，点击 Open。出现导入画面，此时提示需要在打开的 WinCC 中安装 PMC 驱动通道。

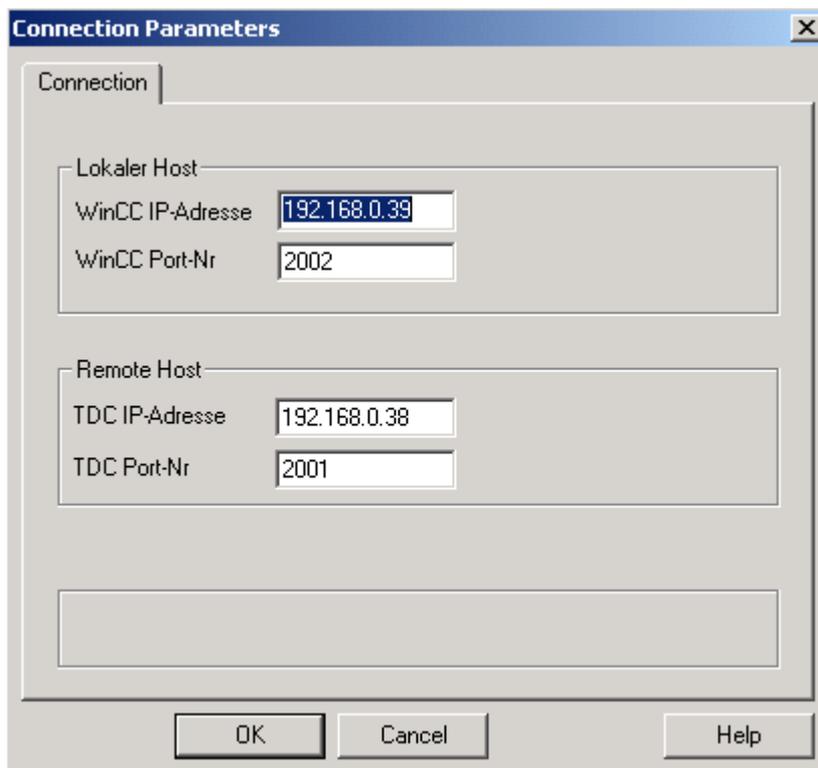


- 点击 OK，显示变量导入状态和结果。该状态和结果信息全部存储在 ARC01.LOG 文件中。

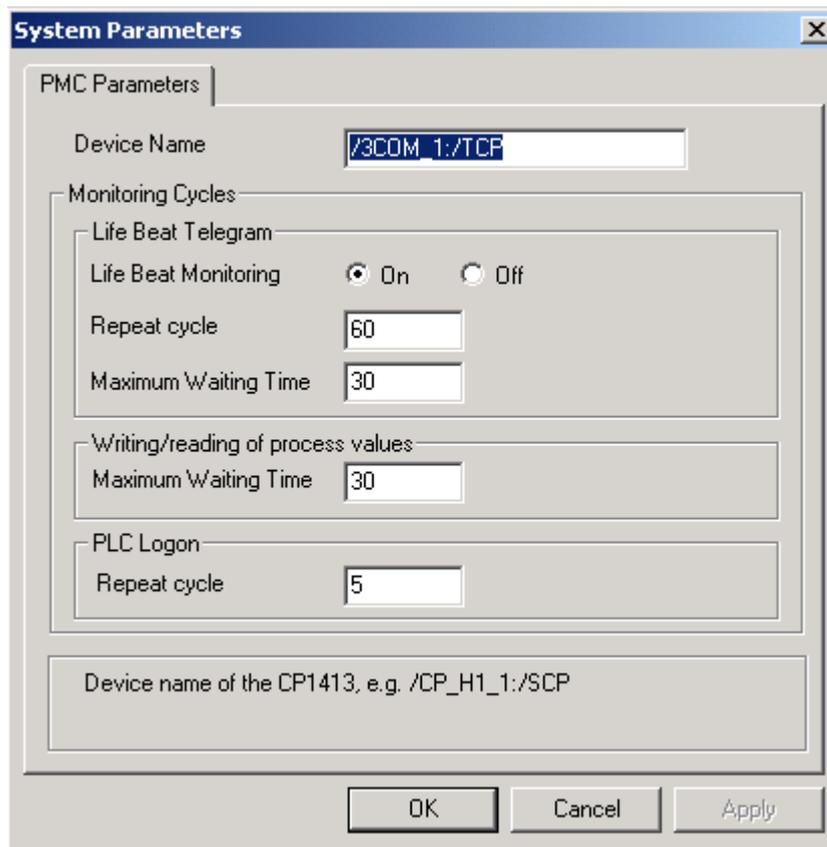
- 点击 ARC01 连接通道，察看生成的变量，这里包含着变量的逻辑名称信息和路径信息。

Name	Type	Parameters
flash_T	Floating-point number 32-bit IEEE 754	TF ProcessData.Flash 1 1 40859 1002 0
flash_Q	Binary Tag	B1 ProcessData.Flash 1 1 8091 1002 24576
sawtooth_X	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 0
sawtooth_LU	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 4
sawtooth_LL	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 8
sawtooth_TI	Floating-point number 32-bit IEEE 754	TF ProcessData.sawtoo 1 1 40859 1001 16
sawtooth_Y	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 24576

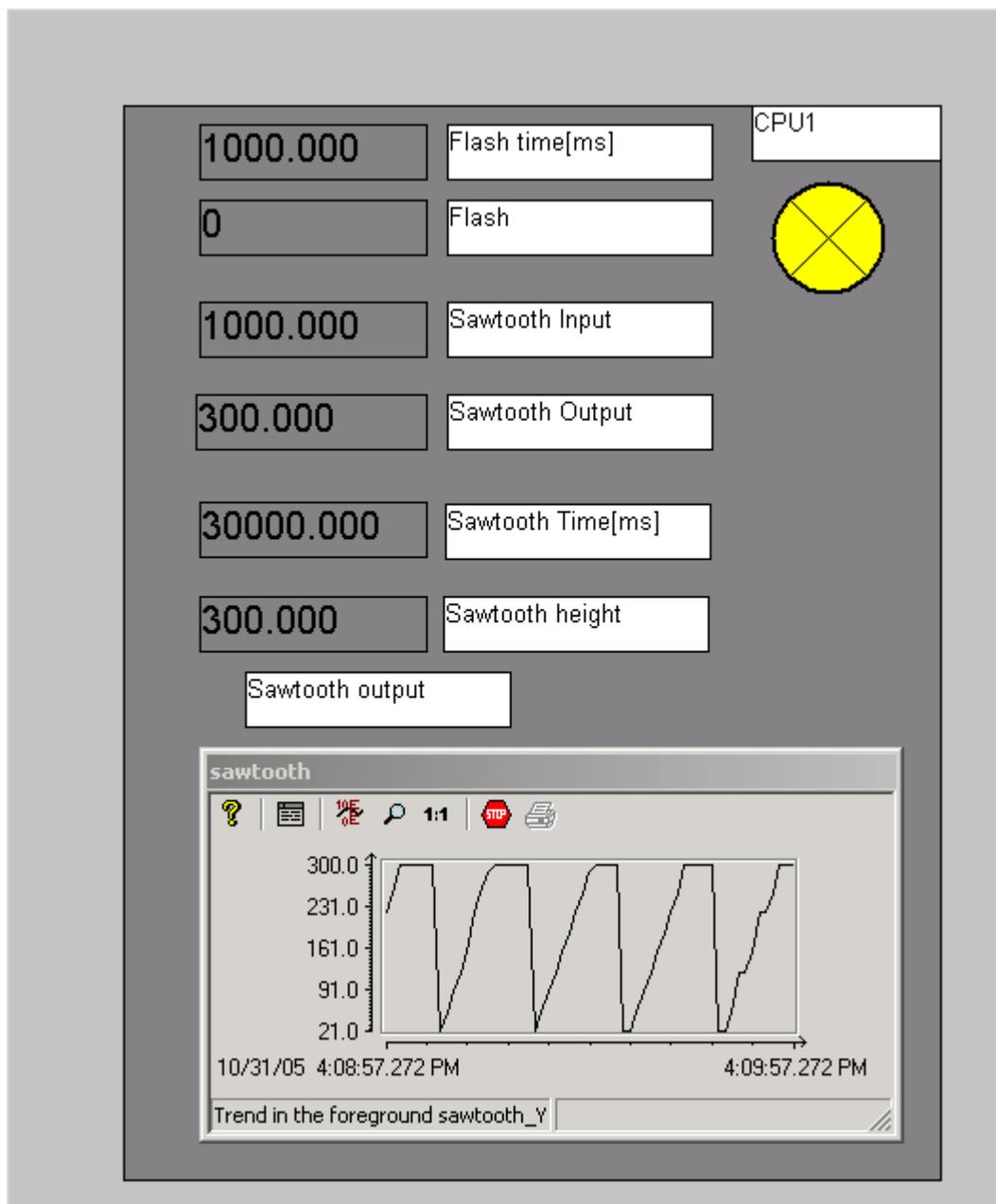
- 选择 ARC01 通道，设置该通道属性，WinCC 的 IP 地址为该 WinCC 操作员站的操作地址。WinCC 端口号任意设定。因为 CP51M1 作为服务器。TDC IP 地址根据 TDC 硬件组态的 IP 设置。TDC 的端口号按照 CFC 程序中的 LI 的 AR 设置。



- 察看 TCP/IP Unite1 的系统参数。其中/3COM\_1:/TCP 是通讯的逻辑设备名。该页面设置按照默认状态即可。



- 定义一个过程画面，激活 WinCC 程序，察看过程变量的运行结果。



- 对于消息变量，在 PROBI 软件生成的地址文件列表 ARC01.txt 中手动添加。

添加消息变量的文件结构：

<消息的变量名>,< LI 的通道连接名>,<MNR,MNR<消息号>,zustand,EKL0,HRM0

其中， <>符号不定以在地址文件列表中。且消息变量名可以自定义。

```

VERSION: SIMADYN_V1.0
VARIABLE:
flash_T, ARC01, TF, 0001, ProcessData.Flash function.T, , input, ABB0,40859.1002.0
flash_Q, ARC01, B1, 0001, ProcessData.Flash function.Q, , output, ABB0,8091.1002.24576
sawtooth_X, ARC01, NF, 0001, ProcessData.sawtooth.X,1.0, input, ABB0,40859.1001.0
sawtooth_LU, ARC01, NF, 0001, ProcessData.sawtooth.LU,1.0, input, ABB0,40859.1001.4
sawtooth_LL, ARC01, NF, 0001, ProcessData.sawtooth.LL,1.0, input, ABB0,40859.1001.8
sawtooth_TI, ARC01, TF, 0001, ProcessData.sawtooth.TI, , input, ABB0,40859.1001.16
sawtooth_Y, ARC01, NF, 0001, ProcessData.sawtooth.Y,1.0, output, ABB0,40859.1001.24576
Message1, ARC01, MNR, MNR25, zustand, EKLO, HRMO
Message2, ARC01, MNR, MNR26, zustand, EKLO, HRMO
    
```

- 通过 ADRIMP 再次对地址列表文件导入到 WinCC 中。察看 ADRIMP 中的结果。

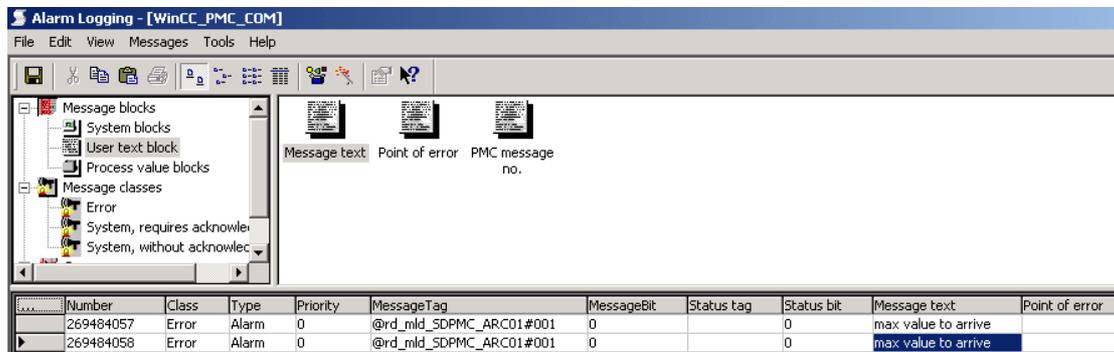
```

***** Attention *****
please check:
  * Project is loaded!
  * Channel driver S5PMC, SDPMC, MMC-PMC, VMS-PMC or TDC-PMC is installed!
*****
*** open project ...
*** creating connection ...
*** creating messages ...
*** creating PVs ...
  creating flash_T ...
  creating flash_Q ...
  creating sawtooth_X ...
  creating sawtooth_LU ...
  creating sawtooth_LL ...
  creating sawtooth_TI ...
  creating sawtooth_Y ...
  creating Message1 ...
  creating @rd_mld_SDPMC_ARC01#001
  open message system, please wait ...
  creating message 269484057
PLS-Options changed! Try it again...
  creating message 269484057
  creating Message2 ...
  creating message 269484058
***** Closing System ... *****
***** Ready *****
*** logfile is C:\Program Files\Siemens\Step7\S7Proj\TDC-WinCC\ES_LOCC\1\Gen\ARC01.LOG ***
    
```

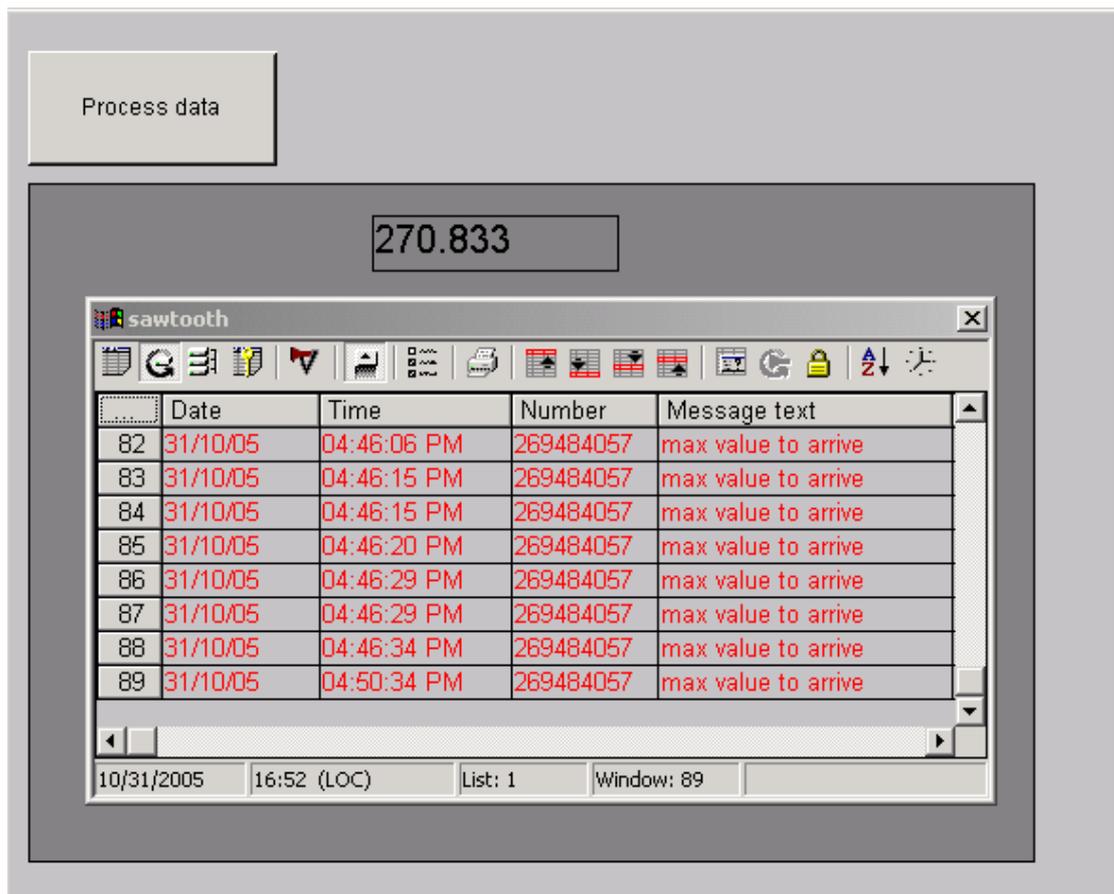
- 察看 WinCC 中变量列表中的结果，可以看到@rd\_mld\_SDPMC\_AEC01#001 的原始数据类型导入到变量表中。

Name	Type	Parameters
flash_T	Floating-point number 32-bit IEEE 754	TF ProcessData.Flash 1 1 40859 1002 0
flash_Q	Binary Tag	B1 ProcessData.Flash 1 1 8091 1002 24576
sawtooth_X	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 0
sawtooth_LU	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 4
sawtooth_LL	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 8
sawtooth_TI	Floating-point number 32-bit IEEE 754	TF ProcessData.sawtoo 1 1 40859 1001 16
sawtooth_Y	Floating-point number 32-bit IEEE 754	NF ProcessData.sawtoo 1 1 40859 1001 24576
@rd_mld_SDPMC_ARC01#001	Raw Data Type	RAW_EVENT

- 打开 WinCC\_PMC\_COM 项目中的 Alarm Logging, 可以直接定义消息文本在 Alarm Logging 中。

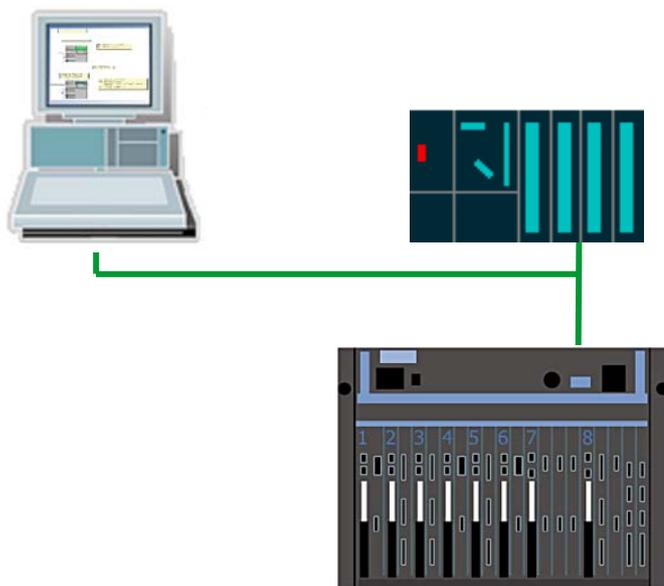


- 定义一个消息画面, 用于显示消息报警的结果。在 WinCC\_PMC\_COM 项目中的启动属性中需要选中 Alarm Logging Runtime。需要注意的是该消息出现的时间是 TDC 的系统时间, 而不是 WinCC 操作员站的 PC 系统时间。



#### 4.6.3 CP51M1 与 CP343-1 通讯 (TCP/UDP)

CP51M1 可以与 SIMATIC S5/S7 通过工业以太网以 TCP 协议进行通讯，需要指定服务器与客户机以及端口号。那么主动建立连接的一方称为客户机。TCP 协议面向确定性连接，也就是能够寻址通讯方才可以发送数据。该连接一旦建立，那么数据就能保证到达通讯方，如果发生错误，数据会自动重新发送几次。UDP 协议是非面向确定性连接的。也就是在暂时不能够寻址通讯方也可以发送数据。但不能保证数据正确达到，一旦接收到数据，由于数据安全校验机制，可以保证接收到的数据是正确的。



##### 4.6.3.1 在 SIMATIC Manager 中进行硬件组态

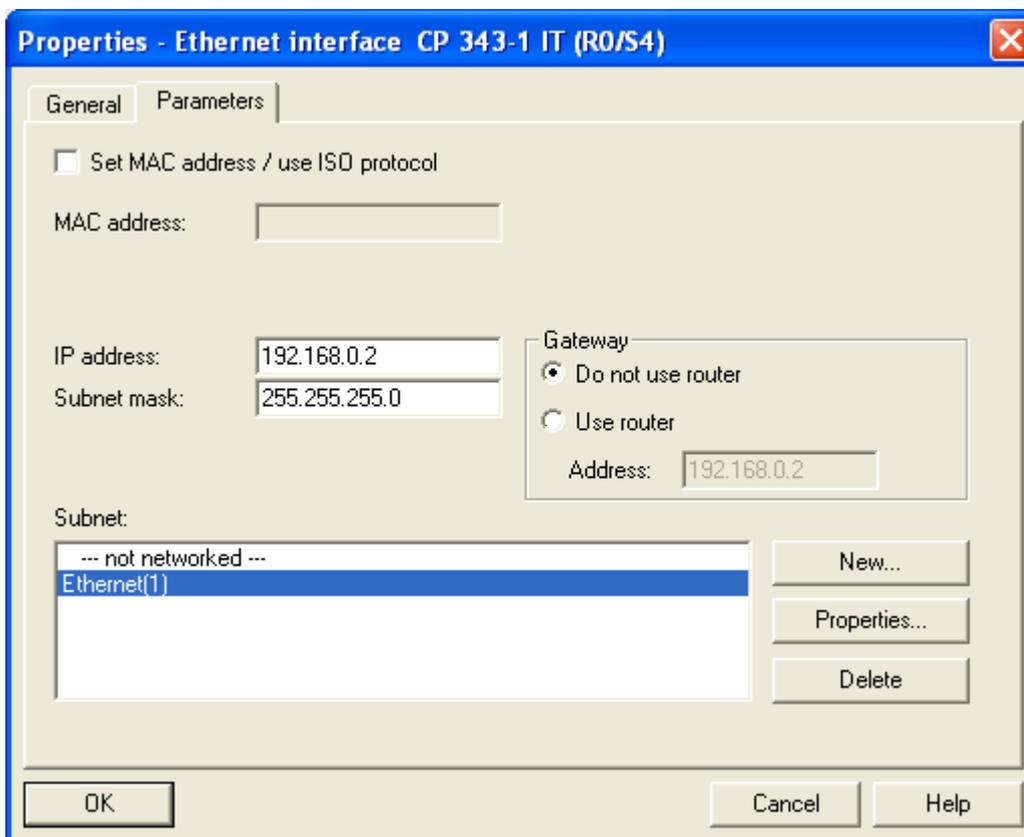
- 硬件组态与 4.6.1 章硬件组态相同。设置 IP 地址 192.168.0.1，子网掩码 255.255.255.0。
- 计算机 PG 使用标准的以太网电缆和以太网网卡。

##### 4.6.3.2 在 SIMATIC Manager 中进行软件组态

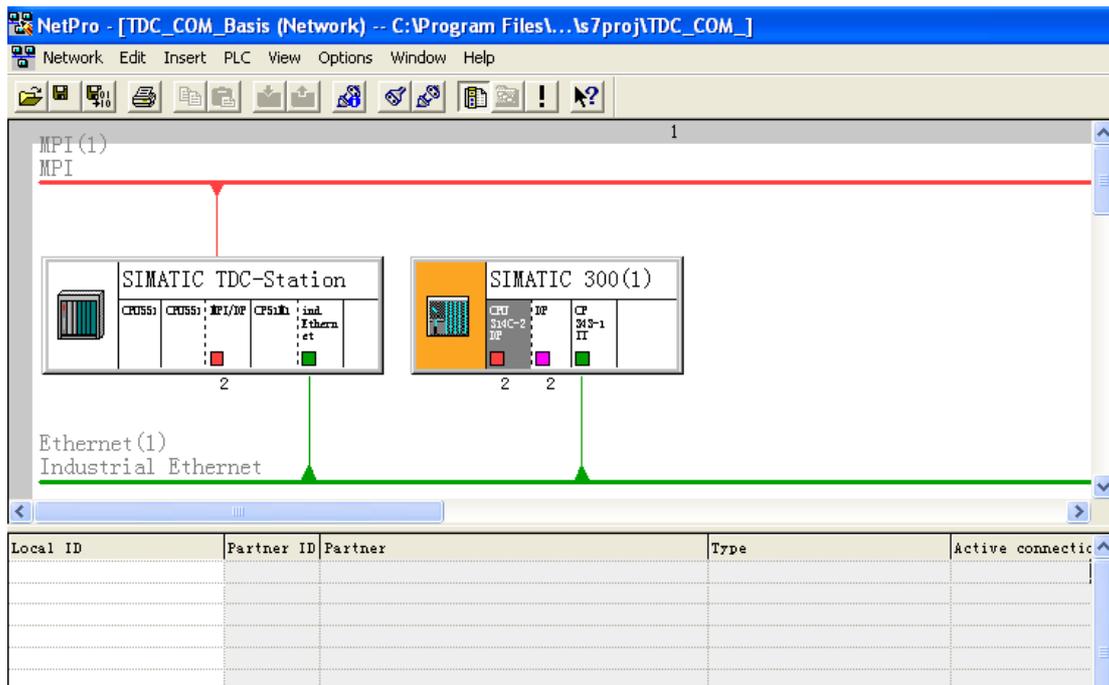
- 在项目 TDC\_COM\_Basis 中，插入一个 SIMATIC 300 Station，对其进行硬件组态。该 CP343-1 支持 TCP 通讯。

Slot	Module	Order number	Fir...	M...	I address	Q addr...
1						
2	<b>CPU 314C-2 DP</b>	<b>6ES7 314-6CF00-0AB</b>	<b>V1.0</b>	<b>2</b>		
X2	<i>DP</i>				1023*	
2.2	<i>DI24/DO16</i>				124...126	124...125
2.3	<i>AI5/AO2</i>				752...761	752...755
2.4	<i>Count</i>				768...783	768...783
2.5	<i>Position</i>				784...799	784...799
3						
4	CP 343-1 IT	6GK7 343-1GX11-0XE0	V2.0	3	256...271	256...271
5						

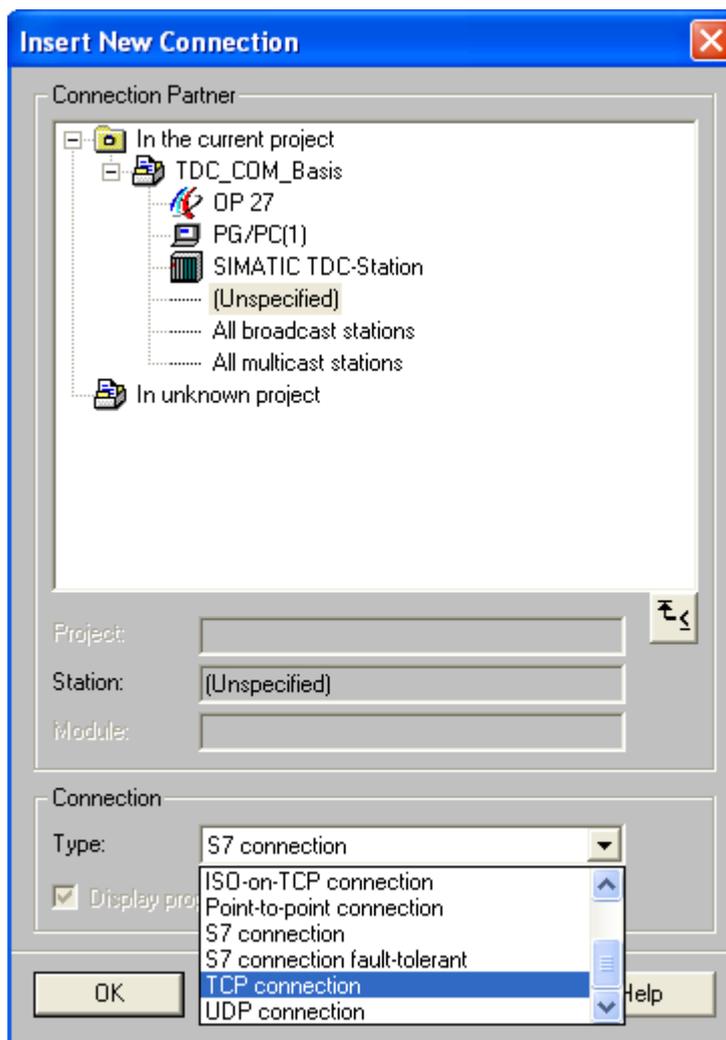
- 双击 CP343-1，对其进行以太网参数设置组态。IP 地址为 192.168.0.2，子网掩码为 255.255.255.0。选择 Ethernet(1)与 CP51M1 处于同一子网中。



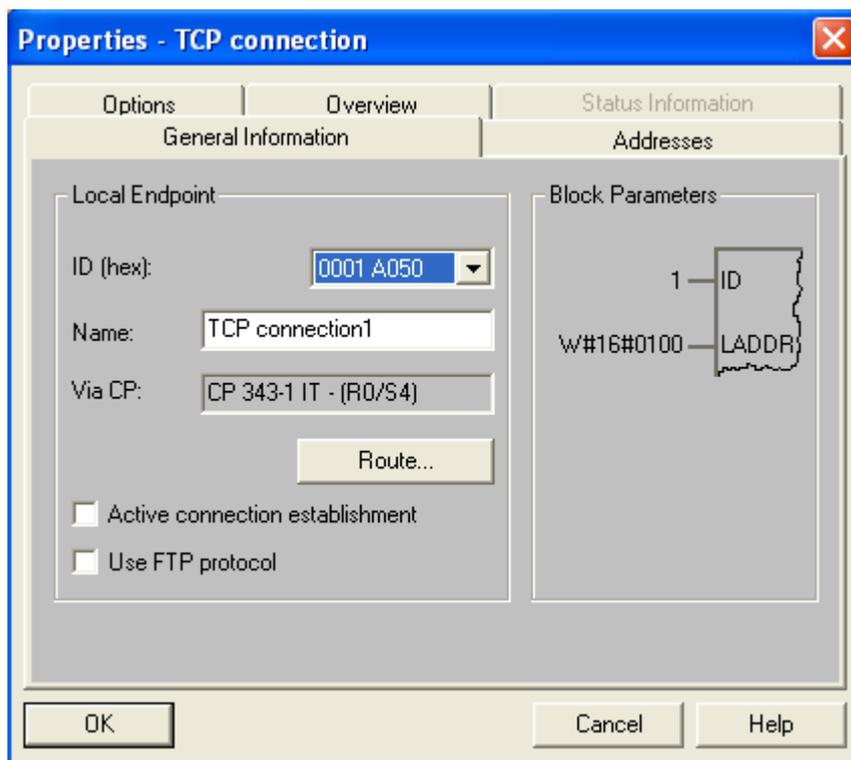
- 打开 NetPro，对 CP343-1 和 CP51M1 组态网络参数。



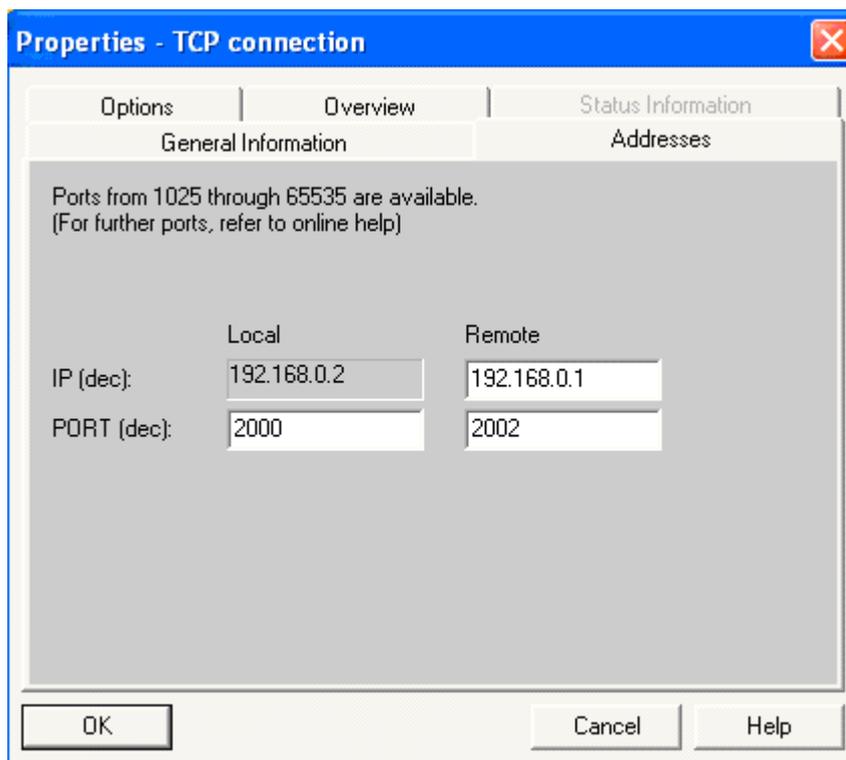
- 点击 SIMATIC 300(1)的 CPU314C-2 DP，出现设置连接的画面栏，双击该画面栏设置新建连接参数，选择（Unspecified）站，设置 TCP connection。注意本节中的 UDP 例子，同样选择的 Unspecified 站。



- 在这个画面中，有一个选项“Active connection establishment”，即是否要建立主动连接，本例中不选择该选项，表示 CP343-1 作为服务器。



- 之后在 **Addresses** 栏中，设置对方的即 CP51M1 的 IP 地址和端口号。IP 地址为 192.168.0.1 与 CP51M1 的硬件组态相对应。端口号设置为 2002。其他设置保持默认。



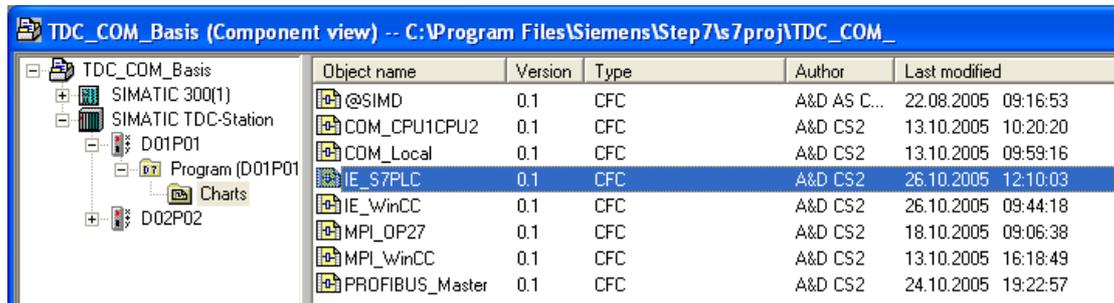
- 点击 NetPro 工具栏的 ，对网络组态进行保存和编译。
- 在 300 站的程序表中，打开 OB1 编写 SEND/RCV 程序。使用 AG\_SEND 和 AG\_RECV 功能块来与 CP51M1 进行数据交换。本例中设置 4 个字节进行数据交换。

```
CALL "AG_SEND"
ACT :=M0.0
ID :=1
LADDR :=W#16#100
SEND :=P#M 10.0 BYTE 4
LEN :=4
DONE :=M0.1
ERROR :=M0.2
STATUS:=MW2
```

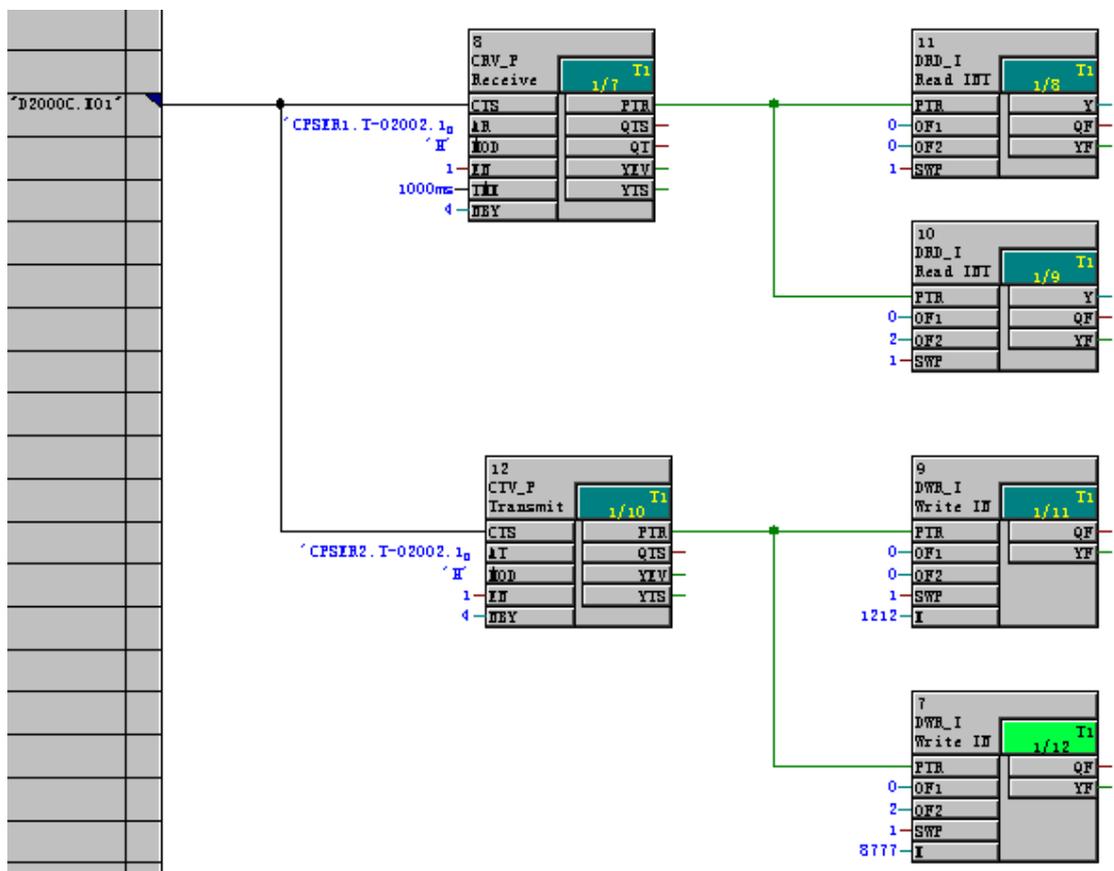
```
CALL "AG_RECV"
ID :=1
LADDR :=W#16#100
RCV :=P#M 20.0 BYTE 4
NDR :=M0.3
ERROR :=M0.4
STATUS:=MW4
LEN :=MW6
```

- 对 300 站的程序和硬件组态以及网络组态参数下载。

- 在 CPU1:D01P01 中，在插入新的 CFC 程序，命名为 IE\_S7PLC，并通过双击打开。



- 在 IE\_WinCC 中已经组态了 @TCPIP 功能块，所以不必再次组态该功能块。插入 CTV\_P 和 CRV\_P 功能块以及 DRD\_I 和 DWD\_I 功能块实现与 CP343-1 数据交换。组态到 T1 时间周期，这样会建立与 CP343-1 的快速连接。



CRV\_P 功能块的说明：

该功能块是以指针的方式来接收数据报文。即对方法送过来的数据放在 CPU 的数据缓冲区中，可以由相应的指针数据块来读取该数据。

CTS—设置全局操作数，定义该模块名称为 D2000C.X01。表示 TDC 通过该接口来接收数据。

AR—定义地址信息。地址信息格式是“通道名称.协议类型-CP51M1 的端口号.对方的 IP 地址-对方的端口号”。第一段由通道名称.协议类型-CP51M1 的端口号组成；第二段由对方的 IP 地址-对方的端口号组成。根据通讯方式是否是服务器来组态该地址信息。

CP51M1 (CTV/CRV)		通讯方
有第二段地址信息	主动建立连接 → ← 接收数据 发送数据 →	服务器
无第二段地址信息	← 主动建立连接 接收数据 → ← 发送数据	客户机

由于已经在 NetPro 中建立 CP343-1 为服务器，那么 AR 的参数中就包含第二段地址信息。即 CP343.T-02002.192168000002-02000。CP343 是通道名，可以随意设置。T 表示 TCP 协议，若是用 U 表示 UDP 协议。02002 表示 CP51M1 的端口号，在 2002 前一定要加 0。192168000002 表示 CP343-1 的 IP 地址，注意中间没有.符号。02000 表示 CP343-1 的端口号，在 2000 前加 0。

MOD—定义通讯接收模式为握手方式 H。表示只要数据没有被接收，那么就不会发送新的数据。

EN—为该功能块的使能。

TMX—监控时间，即要在监控时间内接收到新数据。默认为 1000ms。

NBY—定义接收数据缓冲器的大小。单位是 byte。需要与接收的数据缓冲区大小一致。

S7-300 的程序中定义数据缓冲区为 4 个 byte。所以这里定义为 4。

PTR—数据缓冲区的指针首地址，用于与其他指针的读功能块一起使用。

DRD\_I 功能块的说明：

PTR—数据缓冲区的指针首地址，用于与其他指针的通讯功能块一起使用。

OF1/OF2—数据缓冲区的偏移地址。单位 byte。OF1+OF2 为总的偏移地址。

SWP—交换通讯字的高低字节。对于指针通讯方式使用 PROFIBUS 或 S7 通讯数据。要设置为 1。本例中要设置为 1。如果 SWP≠1，那么在 S7-300 中发送数据区设置一个数据数值为 1(Dec)=0001(Hex)，那么在 TDC 中会收到数值为 256(Dec)=0100(Hex)。这是由于高低 byte 没有交换造成的错误。

Y—表示接收到的数据。

CTV\_P 功能块的说明：

参照 CRV\_P 功能块的说明。

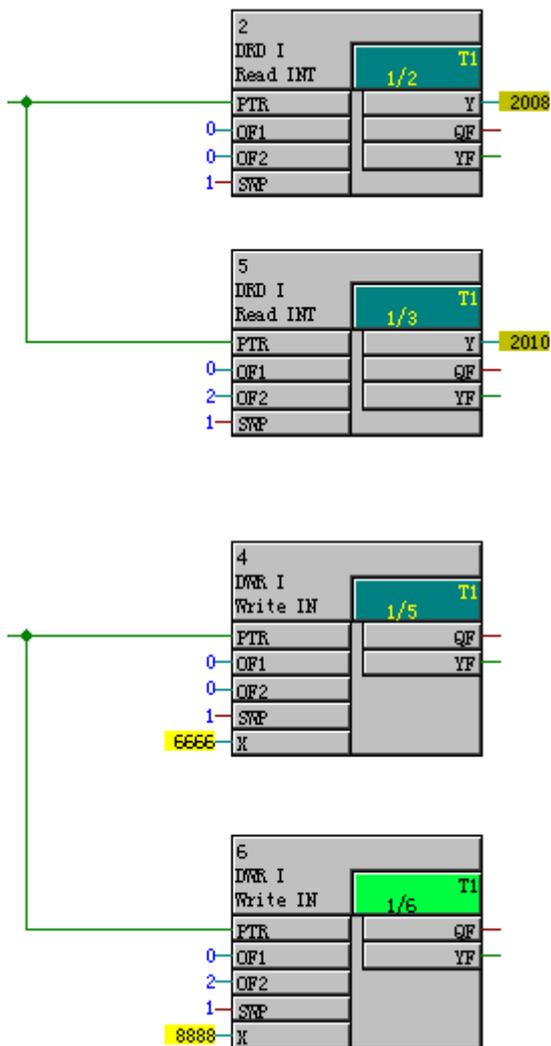
DWR\_I 功能块的说明：

参照 DRD\_I 功能块的说明。

Y—表示发送的数据。

- 编译 CFC 程序，通过工业以太网下载和诊断 TDC。在 300 站中新建一个变量表，在线修改通讯数据 MW10，MW12 和读取 MW20，MW22。观察 CFC 和变量表中的结果。

VAT 1 -- @TDC_COM_Basis\SIMATIC 300(1)\CPU 314C-2 DP\S7 Program(1) ONLINE						
	Address	Symbol	Display	Status value	Modify value	
1	MW 10		DEC	2008	2008	
2	MW 12		DEC	2010	2010	
3	MW 20		DEC	6666		
4	MW 22		DEC	8888		
5						



- 用类似的方法可以在 S7-300 站的 NetPro 中组态一个连接, 由 S7-300 作为客户机, 那么 TDC 作为服务器。这样, CP51M1 与 CP343-1 同时可以即作为服务器也可以作为客户机。此时需要双方的端口号要改变, 即 CP51M1 服务器的端口号为 2008, 客户机的端口号为 2002。CP343-1 服务器的端口号为 2000, 客户机的端口号为 2001。

Local		Remote	
IP (dec):	192.168.0.2	192.168.0.1	
PORT (dec):	2001	2008	

Local ID	Partner ID	Partner	Type	Active connection partner	Subnet
0002 A050		TCP connection5	TCP connection	Yes	Ethernet(1) [IE]
0001 A050		TCP connection1	TCP connection	No	Ethernet(1) [IE]

Local		Remote	
IP (dec):	192.168.0.2	192.168.0.1	
PORT (dec):	2000	2002	

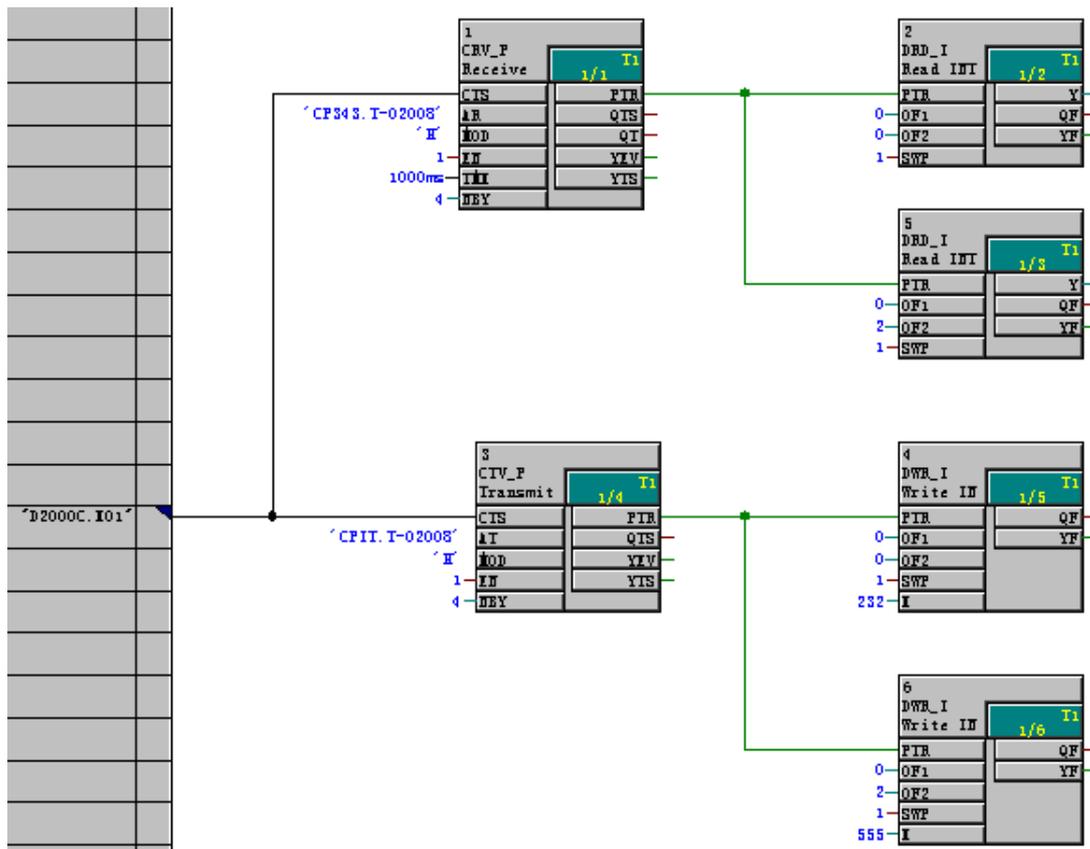
- 同样在 300 站的 OB1 中编写程序，用于与服务器 CP51M1 作为通讯。同样交换 4 个 Bytes。

```

CALL "AG_SEND"
ACT :=M1.0
ID :=2
LADDR :=W#16#100
SEND :=P#M 30.0 BYTE 4
LEN :=4
DONE :=M1.1
ERROR :=M1.2
STATUS:=MW8
CALL "AG_RECV"
ID :=2
LADDR :=W#16#100
RECV :=P#M 40.0 BYTE 4
NDR :=M1.3
ERROR :=M1.4
STATUS:=MW70
LEN :=MW72

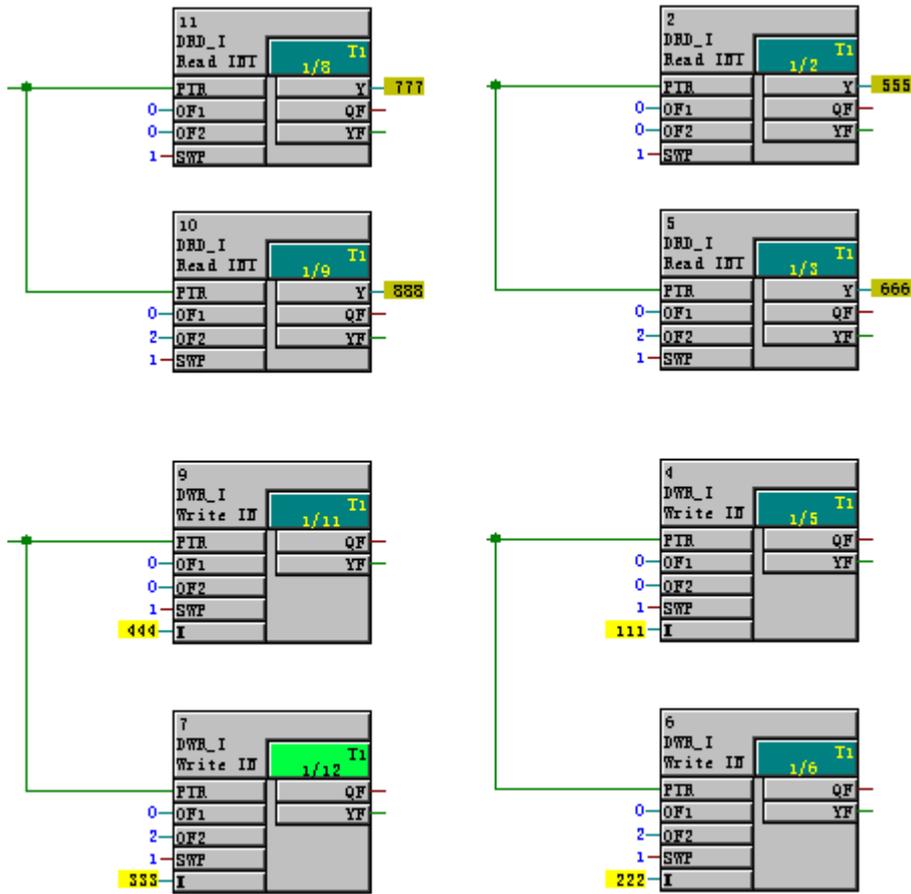
```

- 同样在 TDC 站的 CFC 中编写程序，用于与客户机 CP343-1 作为通讯。同样交换 4 个 Bytes。

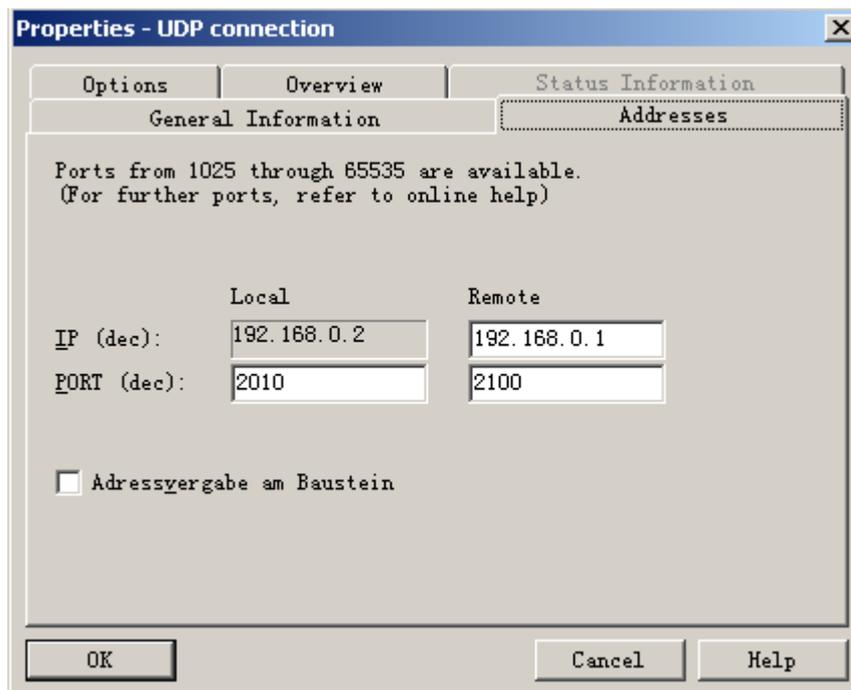


- 编译 CFC 程序，通过工业以太网下载和诊断 TDC。打开 300 站的变量表，在线修改通讯数据 MW10, MW12, MW30, MW32 和读取 MW20, MW22, MW40, MW42。观察 CFC 和变量表中的结果。

VAT 1 -- @TDC_COM_Basis\SIMATIC 300(1)\CPU 314C-2 DP\S7 Program(1) ONLINE					
	Address	Symbol	Display Comment	Status value	Modify value
1	MW 10		DEC	777	777
2	MW 12		DEC	888	888
3	MW 20		DEC	444	
4	MW 22		DEC	333	
5	MW 30		DEC	555	555
6	MW 32		DEC	666	666
7	MW 40		DEC	111	
8	MW 42		DEC	222	
9					



- 用类似的方法在 S7-300 站的 NetPro 中组态一个 UDP 连接，输入 CP511M1 的 IP 地址和端口号。



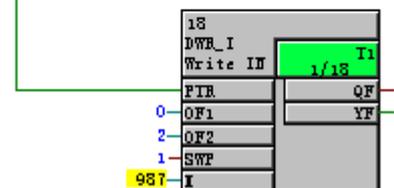
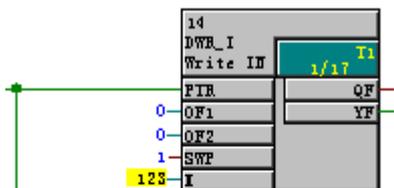
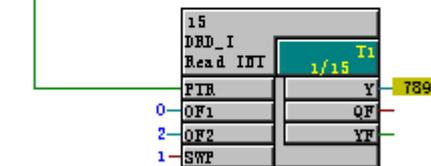
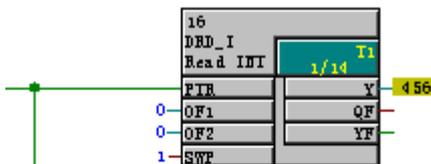
- 同样在 300 站的 OB1 中编写程序，用于与服务器 CP51M1 作为通讯。同样交换 4 个 Bytes。

```
CALL "AG_SEND"
ACT :=M0.5
ID :=3
LADDR :=W#16#100
SEND :=P#M 50.0 BYTE 4
LEN :=4
DONE :=M0.6
ERROR :=M0.7
STATUS:=MW74
CALL "AG_RECV"
ID :=3
LADDR :=W#16#100
RECV :=P#M 60.0 BYTE 4
NDR :=M1.5
ERROR :=M1.6
STATUS:=MW76
LEN :=MW78
```

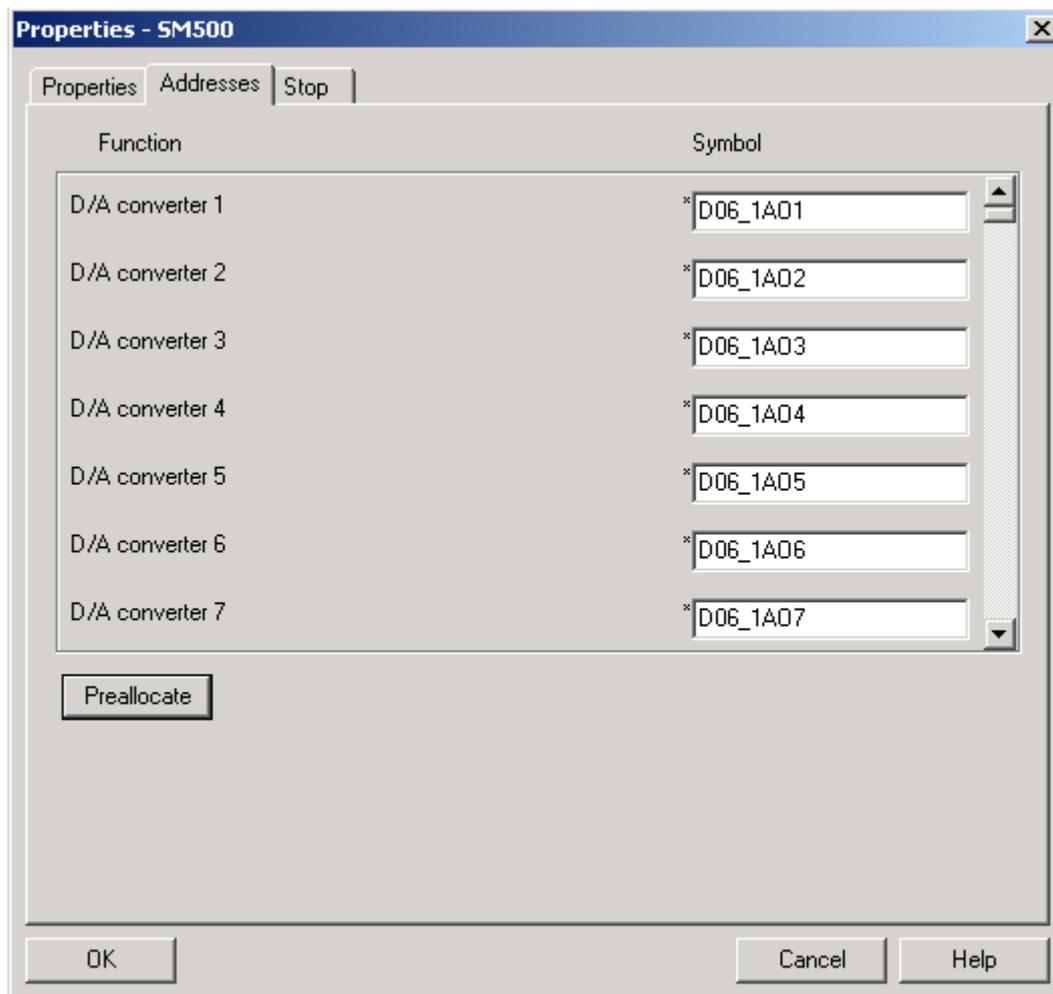
- 同样在 TDC 站的 CFC 中编写程序，用于与 CP343-1 作为 UDP 通讯。同样交换 4 个 Bytes。CRV\_PAR='CP1.U-02100'，CTV\_PAT='CP2.U-02100'。U 表示 UDP 协议。



VAT 1 -- @TDC_COM Basis\SIMATIC 300(1)\CPU 314C-2 DP\S7 Program(1) ONLINE						
	Address	Symbol	Displ	Status value	Modify value	
1	MW	10		DEC	777	777
2	MW	12		DEC	888	888
3	MW	20		DEC	444	
4	MW	22		DEC	333	
5	MW	30		DEC	555	555
6	MW	32		DEC	666	666
7	MW	40		DEC	111	
8	MW	42		DEC	222	
9	MW	50		DEC	456	456
10	MW	52		DEC	789	789
11	MW	60		DEC	123	
12	MW	62		DEC	987	
13						



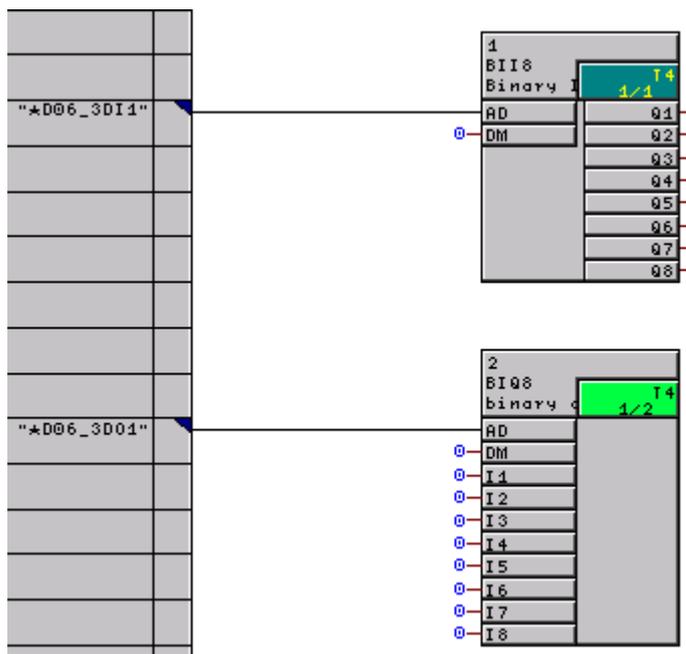




- 计算机 PG 使用标准的 MPI 电缆用于下载和调试程序。

### 5.1.2 在 SIMATIC Manager 中进行软件组态

- 在 Charts 中，插入一个新的 CFC 程序，命名为 DI\_DO。在该程序中对数字量进行读写操作。从 I/O FB 库中，选择 BI18 和 BIQ8 组态到 DI\_DO 程序中。



#### BI18 功能块引脚说明:

用于读入 8 位二进制值。

AD—连接硬件 DI 的地址。该地址是硬件组态分配的符号地址。可以读入 8DI，那么选择其中一组读取。

DM—1，表示直接模式，即数字量的读入操作在该采样时间内执行该功能块时发生。0，表示系统模式，其在该采样时间的开始就执行该读入该数字量的操作。

Q1~Q8—是对应读入的数字量，其中 Q1 为最低位，Q8 为最高位。其它类推。

#### BIQ8 功能块引脚说明:

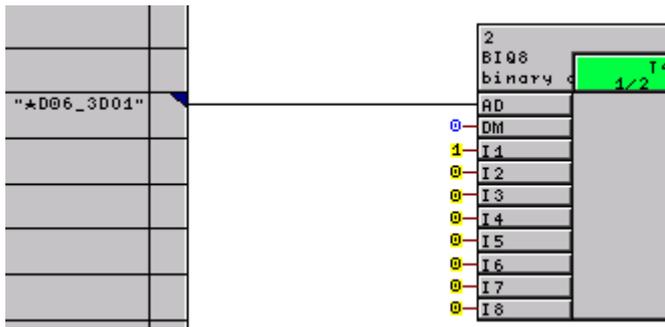
用于输出 8 位二进制值。

AD—连接硬件 DO 的地址。该地址是硬件组态分配的符号地址。可以输出 8DO，那么选择其中一组。

DM—1，表示直接模式，即数字量的输出操作在该采样时间内执行该功能块时发生。0，表示系统模式，其在该采样时间的开始就执行该数字量的操作。

I1~I8—是对应的输出数字量，其中 Q1 为最低位，Q8 为最高位。其它类推。

- 对 CFC 程序进行编译，下载到 TDC 中。DO 输出接 PLC 的 DI 端子来测试。察看结果。



**Monitor/Modify - DI24/DO16 - (R0/S2)**

Online via assigned CPU services

Path: TDC-WinCC-PMC\SIMATIC 300(1)\CPU 314C-2 DP

	Address	Symbol	Displa	Status value	Modify value
1	I 0.0		BIN	2#0	
2	I 0.1		BIN	2#1	
3	I 0.2		BIN	2#0	
4	I 0.3		BIN	2#0	
5	I 0.4		BIN	2#0	
6	I 0.5		BIN	2#0	
7	I 0.6		BIN	2#0	
8	I 0.7		BIN	2#0	
9	I 1.0		BIN	2#0	
10	I 1.1		BIN	2#0	
11	I 1.2		BIN	2#0	

Row Not Effective      Update Force Symbol with F5

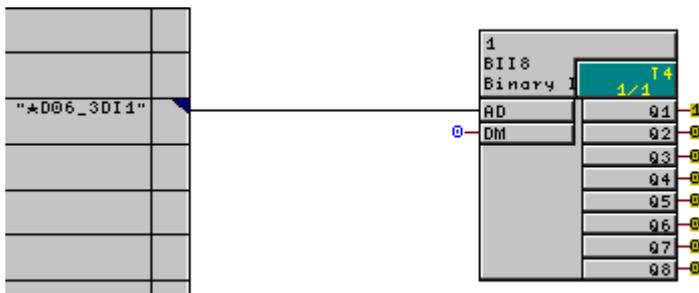
Run conditionally:  Monitor       Modify      Trigger...

Run immediately:  Status Value       Enable Periph. Outputs       I/O Display

**RUNNING**

Close      Help

- DI 输出接 PLC 的 DO 端子来测试。察看结果。



**Monitor/Modify - DI24/DO16 - (R0/S2)**

Online via assigned CPU services

Path: TDC-WinCC-PMC\SIMATIC 300(1)\CPU 314C-2 DP

	Address	Symbol	Displa	Status value	Modify value
22	I 2.5		BIN	2#0	
23	I 2.6		BIN	2#0	
24	I 2.7		BIN	2#0	
25	Q 0.0		BIN	2#0	
26	Q 0.1		BIN	2#0	
27	Q 0.2		BIN	2#1	2#1
28	Q 0.3		BIN	2#0	
29	Q 0.4		BIN	2#0	
30	Q 0.5		BIN	2#0	
31	Q 0.6		BIN	2#0	
32	Q 0.7		BIN	2#0	

Row Not Effective      Update Force Symbol with F5

Run conditionally:  Monitor,  Modify,  Trigger...

Run immediately:  Status Value,  Modify Value

Enable Periph. Outputs,  I/O Display

**RUNNING**

Close      Help

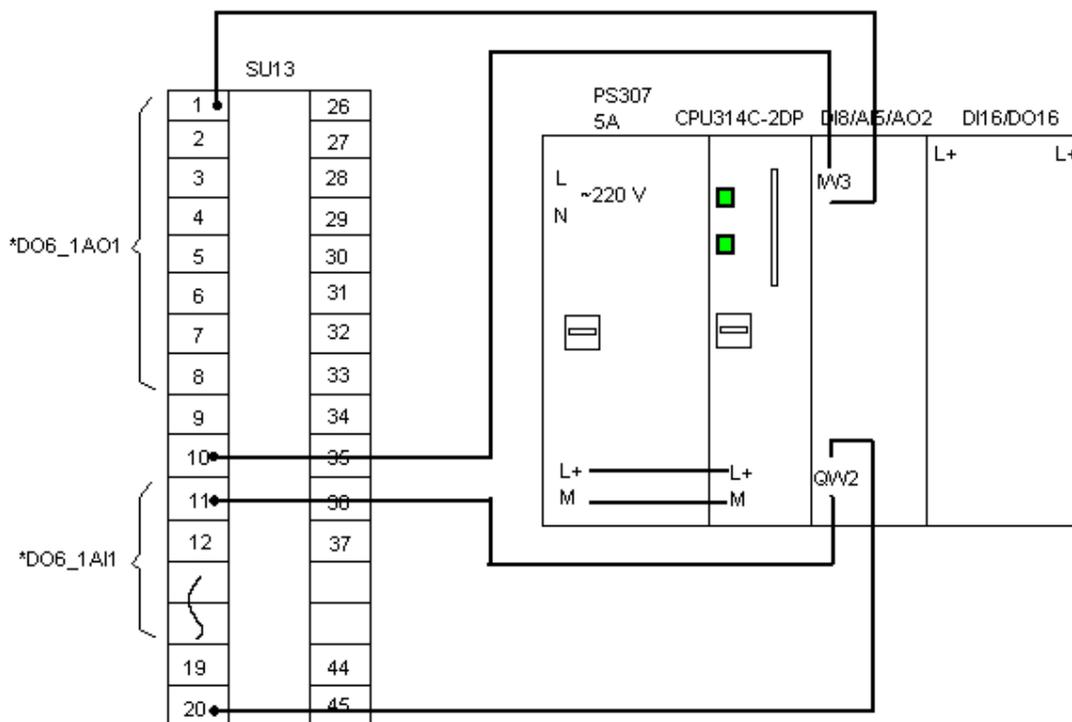
## 5.2 模拟量的读入/输出

SM500 的模拟量输入信号为-10V~+10V，精度 12 位，没有电气隔离。模拟量输出信号为-10V~+10V，精度 12 位，没有电气隔离。具体的详细技术数据请参照本书中的技术数据一章。

### 5.2.1. 在 SIMATIC Manager 中进行硬件组态

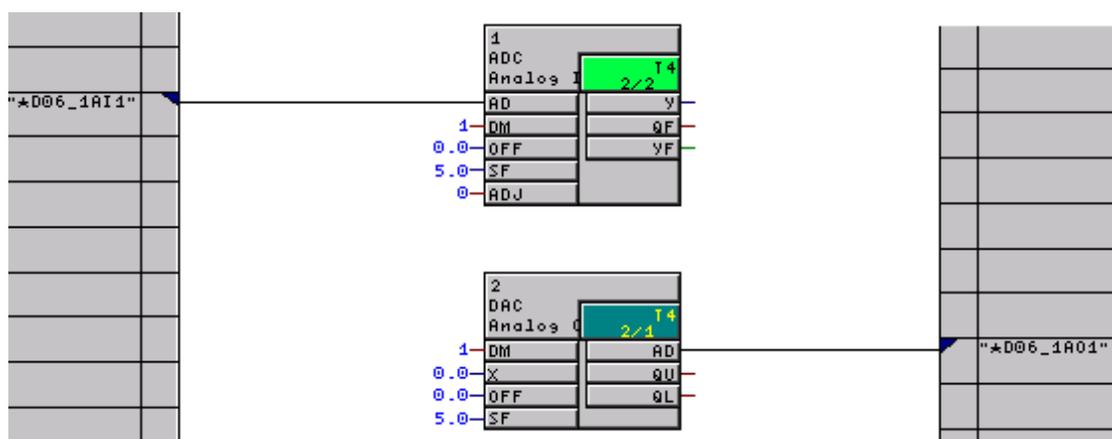
- 硬件组态与 5.1.1 章硬件组态相同。
- 通过 SC63 连接 SM500 的 X1 端子和 SU13。通过 PLC 的 AI 和 AO 模板来做测试。

电路图如下：



### 5.2.2 在 SIMATIC Manager 中进行软件组态

- 在 Charts 中，插入一个新的 CFC 程序，命名为 AI\_AO。在该程序中对模拟量进行读写操作。从 I/O FB 库中，选择 ADC 和 DAC 组态到 AI\_AO 程序中。



ADC 功能块引脚说明：

模拟量输入功能块。转换公式为

$$Y = \frac{U}{5V} \times SF - OFF$$

AD—连接硬件 AI 的地址。该地址是硬件组态分配的符号地址，选择其中一组读取。

DM—1，表示直接模式，即模拟量的输入操作在该采样时间内执行该功能块时发生。0，表示系统模式，其在该采样时间的开始就执行该模拟量的操作。

OFF—偏移补偿。

SF—标么系数。

ADJ—0 表示不要调整；>1 表示在当前的采样周期调整；=1 在 65536 个采样周期后调整。

DAC 功能块引脚说明：

模拟量输出功能块。转换公式为

$$V_{AD} = \frac{X + OFF}{SF} \times 5V$$

表达式 (X+OFF) /SF 要求限制在-2.0~+2.0。>=2.0，输出为 10V。<=-2.0，输出为-10V。

AD—连接硬件 AO 的地址。该地址是硬件组态分配的符号地址，选择其中一组。

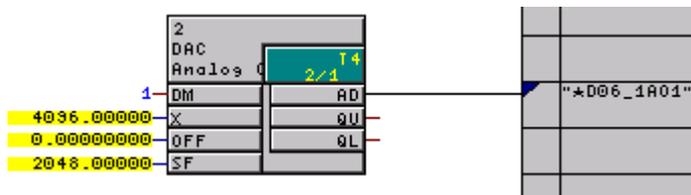
DM—1，表示直接模式，即模拟量的输出操作在该采样时间内执行该功能块时发生。0，表示系统模式，其在该采样时间的开始就执行该模拟量的操作。

X—是对应的输入数字量。

OFF—偏移补偿。

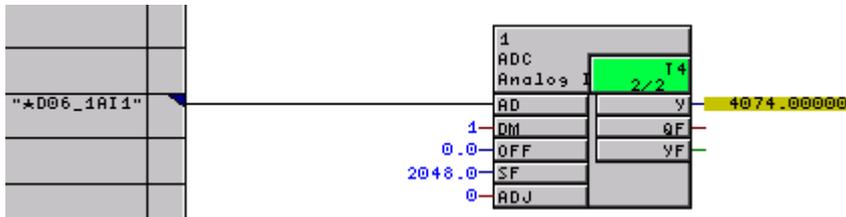
SF—标么系数。

- 对 CFC 程序进行编译，下载到 TDC 中。DO 输出接 PLC 的 DI 端子来测试。PLC 的 DO 接 TDC 的 DI 端子。按照输出精度是 12 位的。定义 SF=2048，那么 X=-4096~+4096 即可满足精度。察看结果。



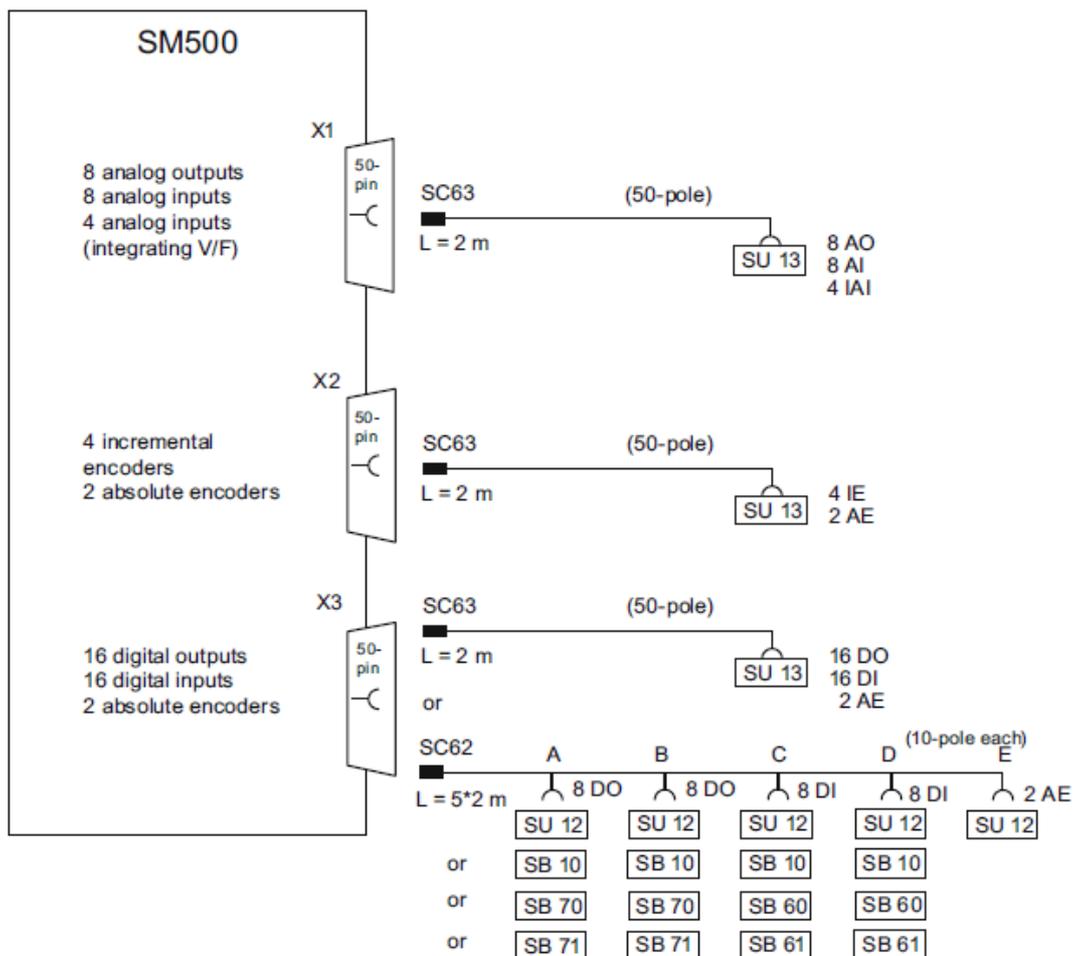
VAT\_1 -- @S7-300 AIAO\SIMATIC 300(1)\CPU 314C-2 DP\S7 Program(1) ONLINE

	Address	Symbol	Displa	Status value	Modify value
1	IW 3		DEC	16	
2	QW 2		DEC	27648	27648
3					

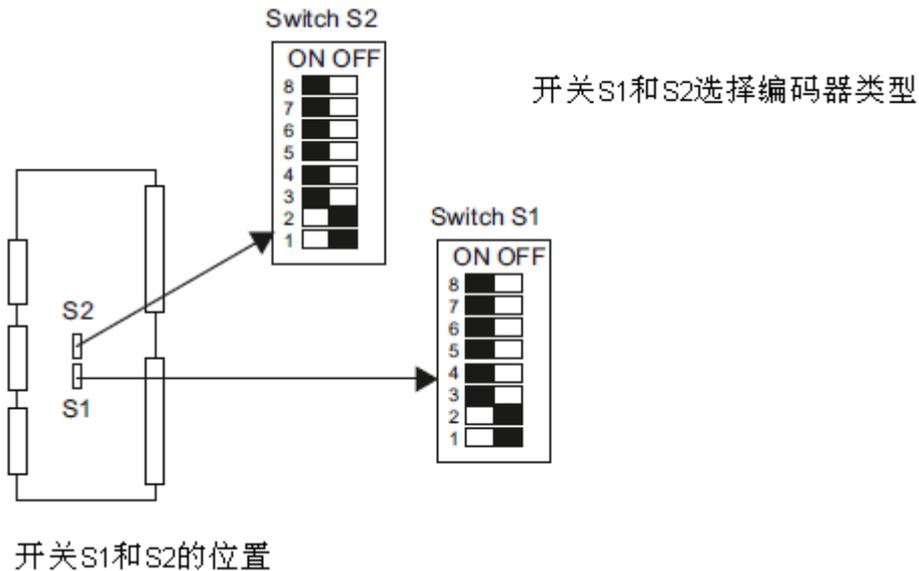


### 5.3 检测增量型编码器的输入

增量型编码器连接如下图



根据编码器输出信号 15V 和 5V 的不同，需要在 SM500 模板上调节 DIP 开关 S1 和 S2。如下图所示



编码器类型设置如下表所示

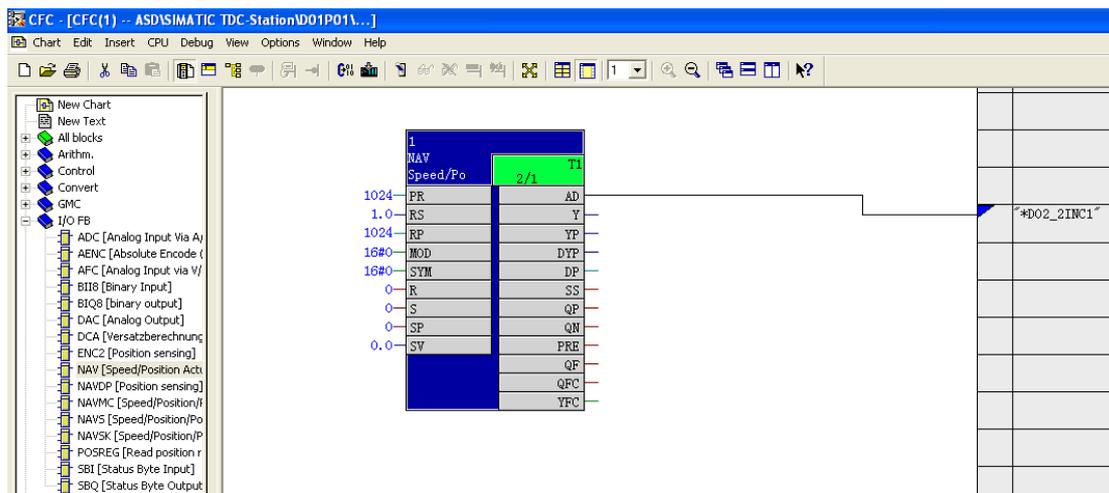
Channel	Track	Switches	15 V encoder	5 V encoder
Encoder 1	A/VW	S1, 1	OFF	ON
	B/RW	S1, 2		
	N/-	S1, 3		
Encoder 2	A/VW	S1, 4	OFF	ON
	B/RW	S1, 5		
	N/-	S1, 6		
Encoder 3	A/VW	S2, 1	OFF	ON
	B/RW	S2, 2		
	N/-	S2, 3		
Encoder 4	A/VW	S2, 4	OFF	ON
	B/RW	S2, 5		
	N/-	S2, 6		

SM500 上的 X2 端子通过 SC63 与 SU13 接口模板相连。

X2 接口集成了 4 个增量型编码器通道，用于检测增量型编码器的输出。接线图如下表所示。

X2	Designation	SU13	X2	Designation	SU13
1	Incremental encoder 1, track A+	1	26	Incremental encoder 2, track A+	26
2	Incremental encoder 1, track A-	2	27	Incremental encoder 2, track A-	27
3	Incremental encoder 1, track B+	3	28	Incremental encoder 2, track B+	28
4	Incremental encoder 1, track B-	4	29	Incremental encoder 2, track B-	29
5	Incremental encoder 1, track N+	5	30	Incremental encoder 2, track N+	30
6	Incremental encoder 1, track N-	6	31	Incremental encoder 2, track N-	31
7	Incremental encoder 3, track A+	7	32	Incremental encoder 3, track B+	32
8	Incremental encoder 3, track A-	8	33	Incremental encoder 3, track B-	33
9	Incremental encoder 3, track N+	9	34	Incremental encoder 3, track N-	34
10	Ground, encoder	10	35	Ground, encoder	35
11	Incremental encoder 4, track A+	11	36	Alarm input 1	36
12	Incremental encoder 4, track A-	12	37	Alarm input 2	37
13	Incremental encoder 4, track B+	13	38	Alarm input 3	38
14	Incremental encoder 4, track B-	14	39	Alarm input 4	39
15	Incremental encoder 4, track N+	15	40	Alarm acknowledgment output 1	40
16	Incremental encoder 4, track N-	16	41	Alarm acknowledgment output 2	41

在 CFC 中，使用功能块 NAV 读取增量型编码器的脉冲数，进行速度和位置的操作。从 I/O FB 库中，选择 NAV 组态到程序中，并且把该功能块组态到 T1 中，功能块的采样时间要求  $TA \leq 20ms$ 。



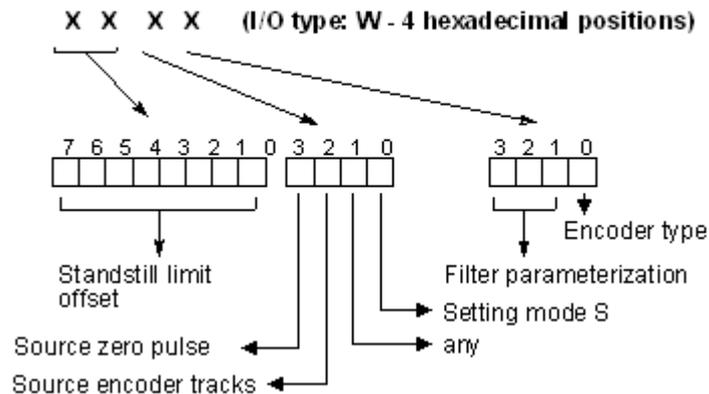
NAV 功能块参数说明如下表，该功能块可以通过读取增量型编码器的脉冲来计算速度和位置。

引脚	说明
AD	连接硬件组态脉冲编码器的地址。该地址是硬件组态分配的符号地址，选择其中一组读取，*D02_2INC1，2 代表 SM500 的 X2 端子，1 代表第一路编码器信号。

PR	编码器的每圈脉冲数
RS	机械的额定转速，单位 rpm。 $Y = (\text{forward\_pulses} - \text{reverse\_pulses}) / \text{time} \times \text{NF}$ 也就是单位时间内计算的脉冲数乘以标准化系数 NF。 NF 是按照参数 PR 和 RS 来计算的，当机械转速达到额定转速 RS 时，Y 输出为 1。
RP	设定编码器的 4 倍频脉冲数，该参数与 MOD 参数有关。
MOD	由 4 个 16 进制数组成，具体格式如下图。
SYM	设置同步类型
R	针对位置值 YP 复位 0
S	针对位置值 YP 置位 SV
SP	SP=1 即设定同步位置
SV	位置设定值
Y	输出实际速度。
YP	输出实际位置。实际位置计算公式： $YP = \sum(\text{forward\_pulses} - \text{reverse\_pulses}) / \text{RP}$ 也就是检查脉冲数的累加除以 RP。RP 为编码器的每圈脉冲数，当根据 MOD 选择 0 对应的码盘是应该乘以 4 倍频。
PRE	检查是否检测到脉冲
YFC	故障代码

MOD 各位含义如下

### MOD format:



**Standstill limit offset (静止极限偏差):** 如果在采样时间 TA 没有接收到任何脉冲, 此时输出参数 PRE=0, Y=0, 那么检测时间会由 TA 延长到静止极限偏差值 (由参数 MOD 设定), 该值由 FD 到 7F, 对应的十进制数为-3~127, 127ms 是最高限。

**Source zero pulse/Source encoder tracks/any:** 这些设置与 TDC 无关。

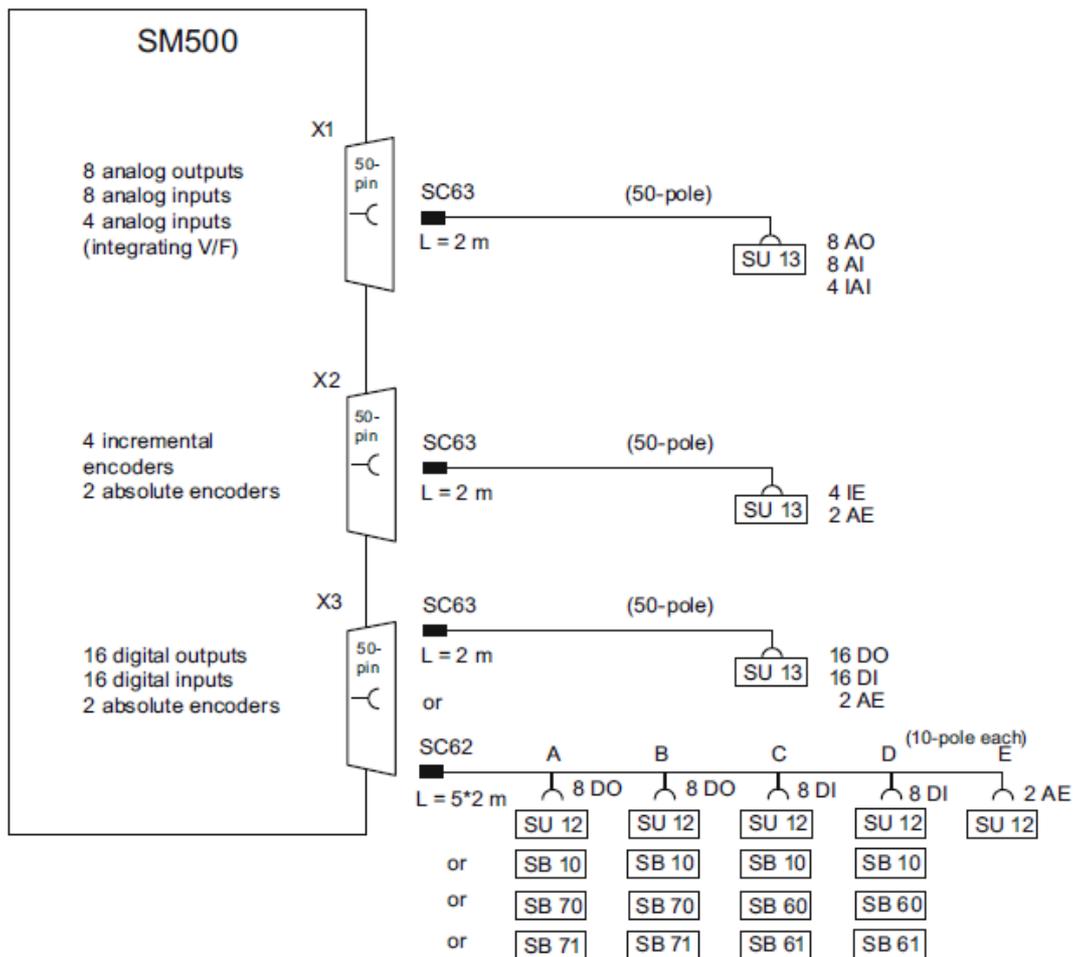
**Setting Mode SP:** 0 表示设置 SV 的值给 YP, 1 表示设置 YP-SV 给 YP。

**Filter parameterization:** 设置脉冲的滤波时间常数, 详细值可以参考功能块手册。

**Encoder type:** 设置编码器类型。0 表示编码器有 A, B 两个通道, 且相位差 90 度, 有 Z 或者没有 Z 信号。计数该种类型的编码器采样 4 倍频。1 表示该编码器只有一个通道, 且没有 Z 信号。使用 1 倍频方式计数。

#### 5.4 检测绝对值型编码器的输入

绝对值编码器的连接如下图



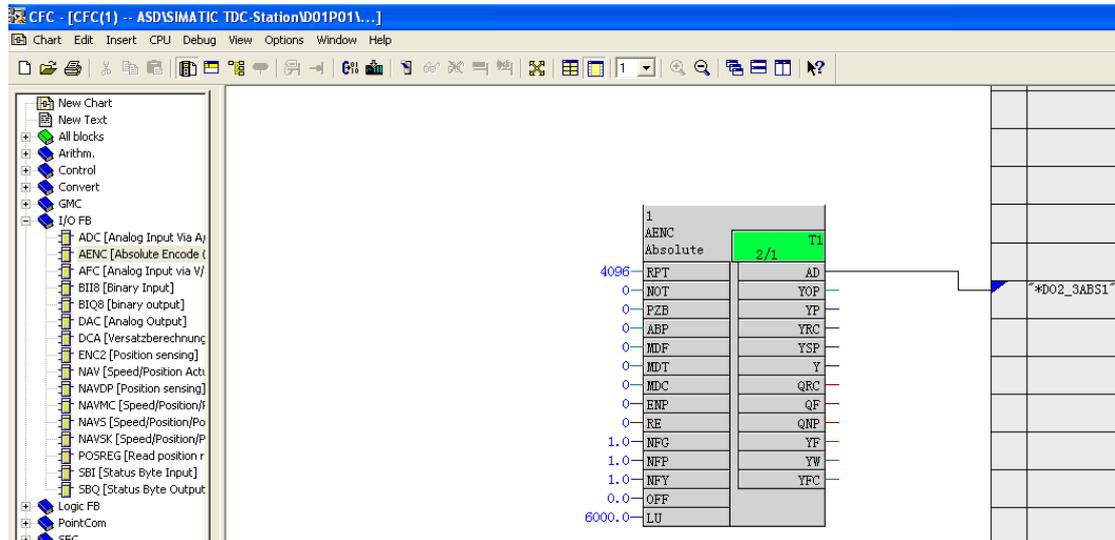
SM500 可以读取 SSI 或 EnDat 的绝对值编码器，最大脉冲频率 2MHz。X2 接口集成有 2 个绝对值编码器的通道，用于检测绝对值编码器的输出。接线图如下表所示。

X2	Designation	SU13	X2	Designation	SU13
21	Absolute encoder 3, data D+	21	46	Absolute encoder 4, data D+	46
22	Absolute encoder 3, data D-	22	47	Absolute encoder 4, data D-	47
23	Absolute encoder 3, clock C+	23	48	Absolute encoder 4, clock C+	48
24	Absolute encoder 3, clock C-	24	49	Absolute encoder 4, clock C-	49
25	Ground, encoder SSI	25	50	Ground, encoder SSI	50

X3	Designation	SU13	X3	Designation	SU13
21	Absolute encoder 1, data D+	21	46	Absolute encoder 2, data D+	46
22	Absolute encoder 1, data D-	22	47	Absolute encoder 2, data D-	47
23	Absolute encoder 1, clock C+	23	48	Absolute encoder 2, clock C+	48
24	Absolute encoder 1, clock C-	24	49	Absolute encoder 2, clock C-	49
25	Ground, encoder SSI	25	50	Ground, encoder SSI	50

从 I/O FB 库中，选择 AENC 组态到程序中，并且把该功能块组态到 T1 中，功能块的采样

时间要求  $TA \leq 10ms$ 。



**ANEC 功能块说明:**

该功能块可以通过读取绝对值编码器的脉冲来计算位置和速度。

AD 连接硬件组态脉冲编码器的地址。该地址是硬件组态分配的符号地址，选择其中一组读取，即 \*D02\_3ABS1

RPT/NOT/PZB/ABP 用于初始化绝对值编码器，如下图所示:

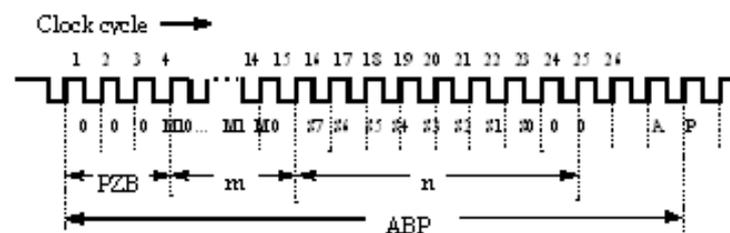
SSI 多圈编码器，带报警位 A 和奇偶校验位 P

NOT = 2048; m = 11;

RPT =256; n=8;

ABP = 25; 报警位在第 25 位;

PZB =3; 开始位置时，不相关的位数为 3;



Mi: 多圈编码器的二进制圈数编码。

Si: 码盘的二进制角度编码。

A: 报警位。

P: 奇偶校验位

ANEC 功能块的引脚说明如下表

PZB	仅用于 SSI 编码器，传递的开始位置时其不相关的位数
NOT	多圈绝对值编码器的圈数，为 2 的 m 次幂
RPT	绝对值编码器每圈指定的步数，也就是每圈的精度，为 2 的 n 次幂
ABP	用于指定报警位的位置号
MDF	用于设定编码器的数据传输频率，0 表示 100KHz
MDT	指定编码器的类型，0 表示旋转型的 SSI 编码器
MDC	连接绝对值编码器的数据编码，1 表示 SSI 输出的格雷码
ENP	使能绝对值编码器奇偶校验位，即第 26 为 P
RE	复位故障信息，上升沿有效
NFG	编码器与驱动轴之间的齿轮箱的变比
NFP	用于标么化位置输出值，YP, YRC, YSP。NFP 是轴旋转的位置值，计算公式为 $YSP = (\text{measured value} / \text{RPT}) * \text{NFG} * \text{NFP} - \text{OFF}$
NFY	输出实际速度标么化系数
OFF	实际位置偏移量。该参数与位置输出的参数可以标么化同一种单位
LU	编码器的限制最大操作速度
YOP	编码器的最初位置
YP	绝对值编码器的实际位置角度
YRC	圈数计数值。在默认 NFP=1 的标么中，YRC 表示实际码盘转动的圈数。如果 NFP>1，那么 YRC 就是 NFP 的倍数
YSP	YP 与 YRC 的和
Y	根据编码器的旋转位置计算的速度值

QF	错误信息。用于指示当前的速度，位置是否有效。如果 QF=1 那么 YP, YRC, YSP 及 Y 无效
QRC	圈数计数器溢出
QNP	指示接收到正确的有效的新位置
YF	如果 SSI 编码器不设置报警位，那么 YF=0，否则输出为 FFFFh
YW	对于 SSI 编码器无效
YFC	表示 AENC 的错误位，具体错误代码可以查看手册。

## 附录

### A. 在 COM PROFIBUS 中安装新的.gsd 文件

若没有 ET200M 的 IM153-2 模板 6ES7153-2BA00-0XB0 的.gsd 文件，那么就在网上下载该.gsd 文件，分别集成到 Step7 和 COM PROFIBUS 中。

集成到 Step7 中的方法:

1. 打开没有项目的硬件组态画面。
2. 在菜单"**Options > Install New GSD**".

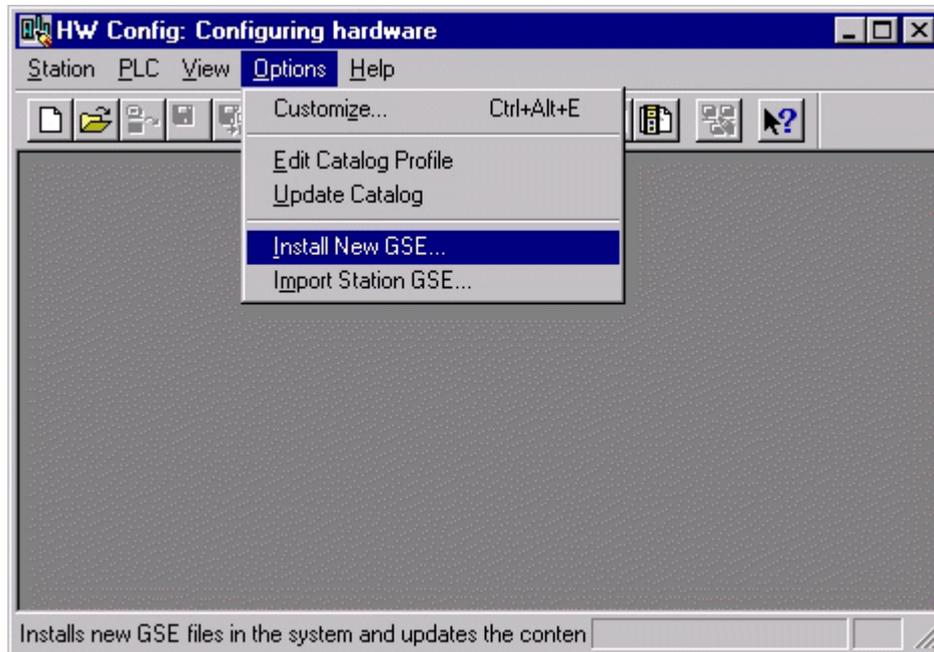


Figure 1: HW config

3. 在出现的对话框中，选中要安装的.gsd 文件，之后选择"Open".



Figure 2: GSD installation

4. 最后选择菜单中"Options > Update Catalog".

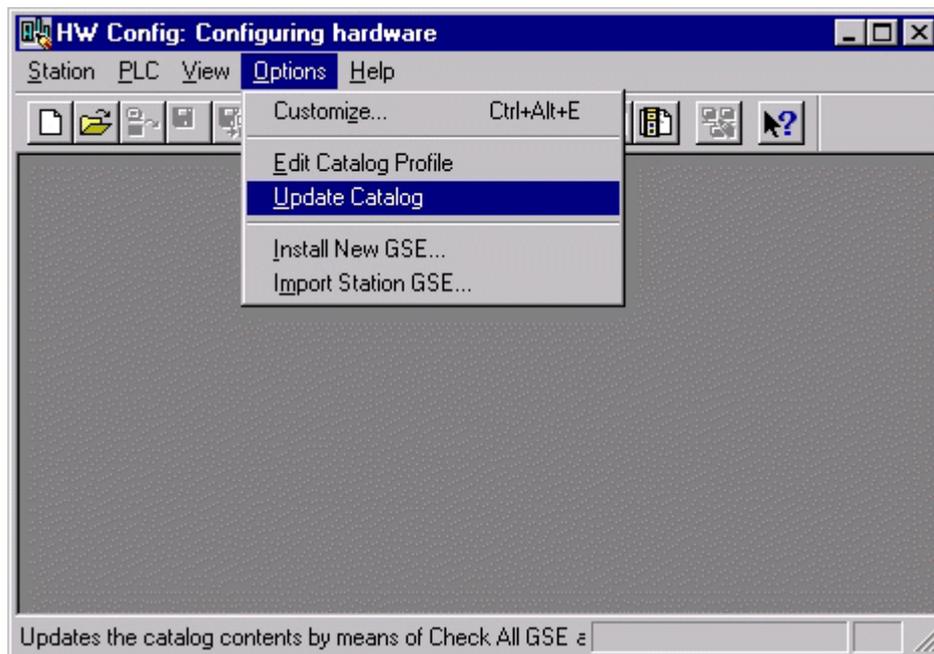


Figure 3: Updating the catalog

5. 那么您就可以在 Hardware catalog 中发现新添加的硬件"PROFIBUS DP > More DP devices"

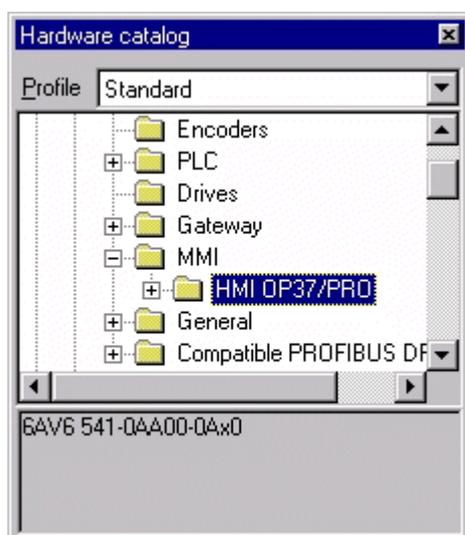


Figure 4: Hardware catalog

6. 现在这个从站就集成到您的 Step7 中。.

集成到 COM PROFIBUS 中的方法:

1. Copy .gsd 文件到文件夹"<(drive)>:\Siemens\CPBV51\gsd"中.
2. 启动 COM PROFIBUS 并选择"Read in GSD files".

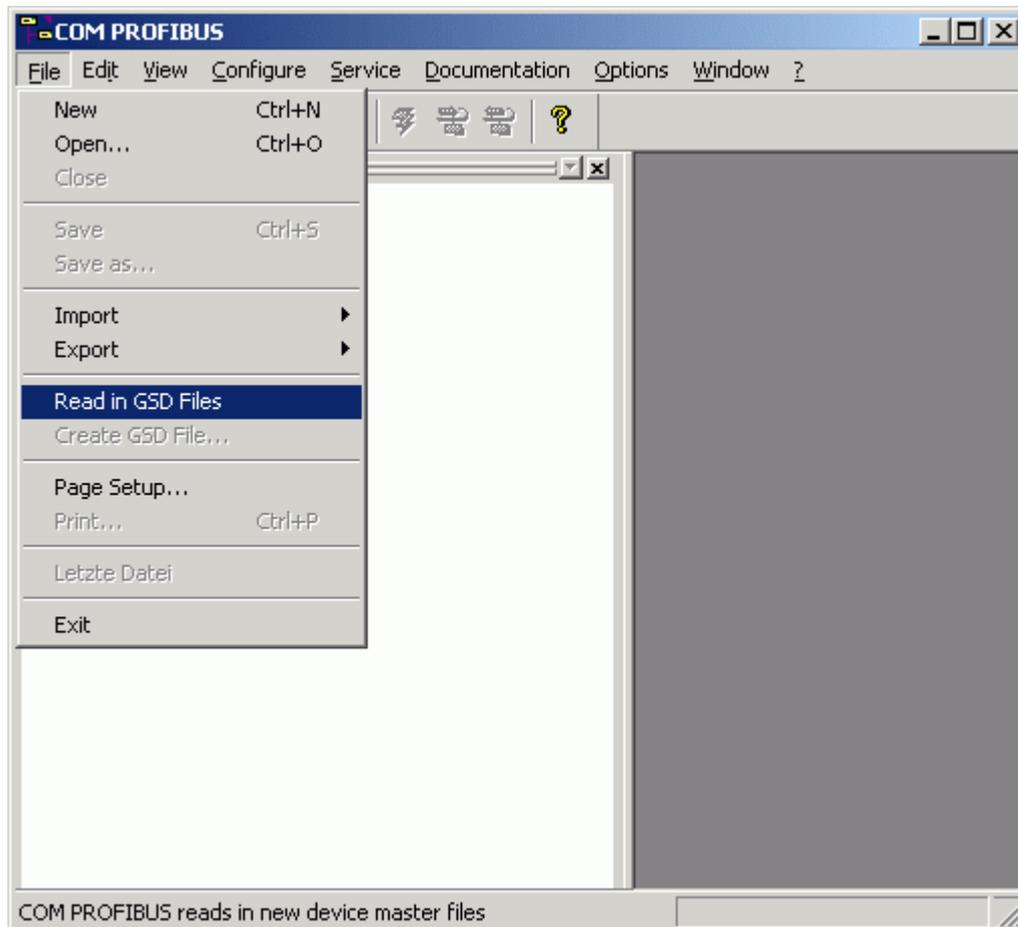


Figure 5: Reading in GSD files in COM PROFIBUS

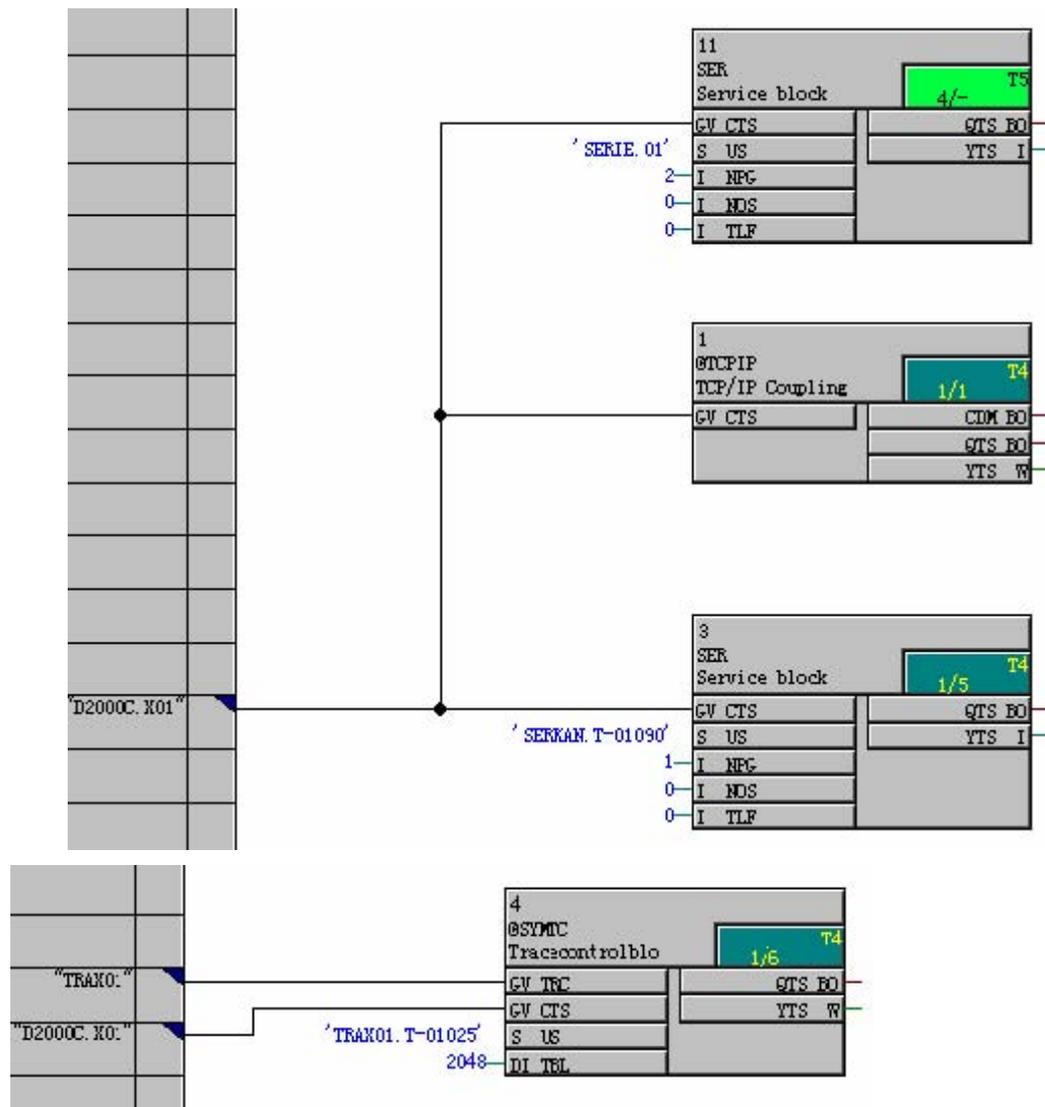
3. 新的 ET200 的子站就集成到 COM PROFIBUS 中。

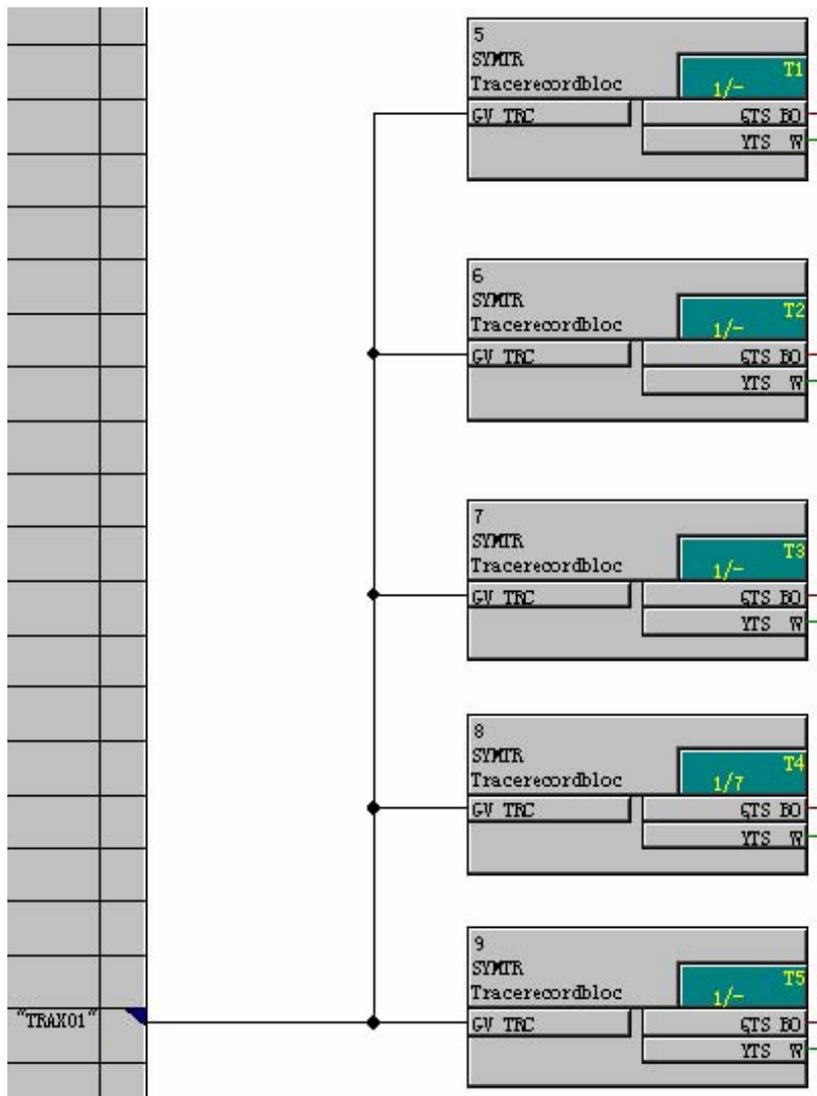
#### B.使用 Symtrace-D7 可视化过程数据和逻辑分析

Symtrace-D7 是一种用于调试和维护TDC 系统的支持系统软件。Symtrace D7 V3.4 版本还可以使用TCP/IP 来读出TDC 中组态的运行值。在V3.3 以前版本中PC 只能使用串口使用DUST1 协议来来与TDC 通讯。该软件支持两种跟踪方式，一种是在线跟踪；另一种是离线跟踪。在线跟踪用于粗略的显示数据趋势，无限变量可以显示。最小的采样周期为

32ms。离线跟踪用于实时数据显示，最多可以显示22个变量。最短的采样时间为0.1ms。可以确定激活条件。

- 1.在CFC 中导入FBSSY1 功能块库，方法参照4.6.2 种的方法。
- 2.使用CP51M1 通过TCP/IP 方式实现TDC 数据跟踪。组态SER 功能块用于提供Symtrace 通讯服务。令US 为'SERKAN.T-01090'，其中SERKAN 是通道名。T 表示
- 3.在CFC 组态@SYMTC 功能块，设置该数据跟踪任务名称为TRAX01。CTS 连接CP51M1 的以太网接口即硬件组态中D2000C.X01。令US 为'TRAX01.T-01025'，其中该通道号与跟踪任务名称相同即TRAX01。其端口号为1025。2048 为设定的数据跟踪缓冲区的大小，单位为双整数。一个CPU 中只能组态一个@SYMTC 功能块。且只能组态大于32ms 的采样任务周期中。
4. 组态SYMTR 功能块5 个，分别组态到各个时间周期T1-T5 中，并组态到与跟踪任务名相同为TRAX01。
5. 打开Symtrace 软件，点击图标 组态其通讯参数。其中192.168.0.1 是TDC 的TCP 协议。01090 为端口号。CTS 连接 CP51M1 的以太网接口即硬件组态中D2000C.X01。需要设置该SER 功能块的NPG 参数为1。





CP51M1 的IP 地址，1090 是SER 功能块组态的端口号，1025 是@SYMTC 功能块组态的端口号。

coupling parameters

Serial coupling

Coupling using serial interface / DUST1

Interface COM1

Baud rate 19200

Channel service 1

Channel @SYMTC 129

CPU local coupling

TCP/IP coupling

Coupling using TCP/IP connection

IP-Adress of SIMATIC 192.168.0.1

TCP/IP port SER 1090

TCP/IP port @SYMTC 1025

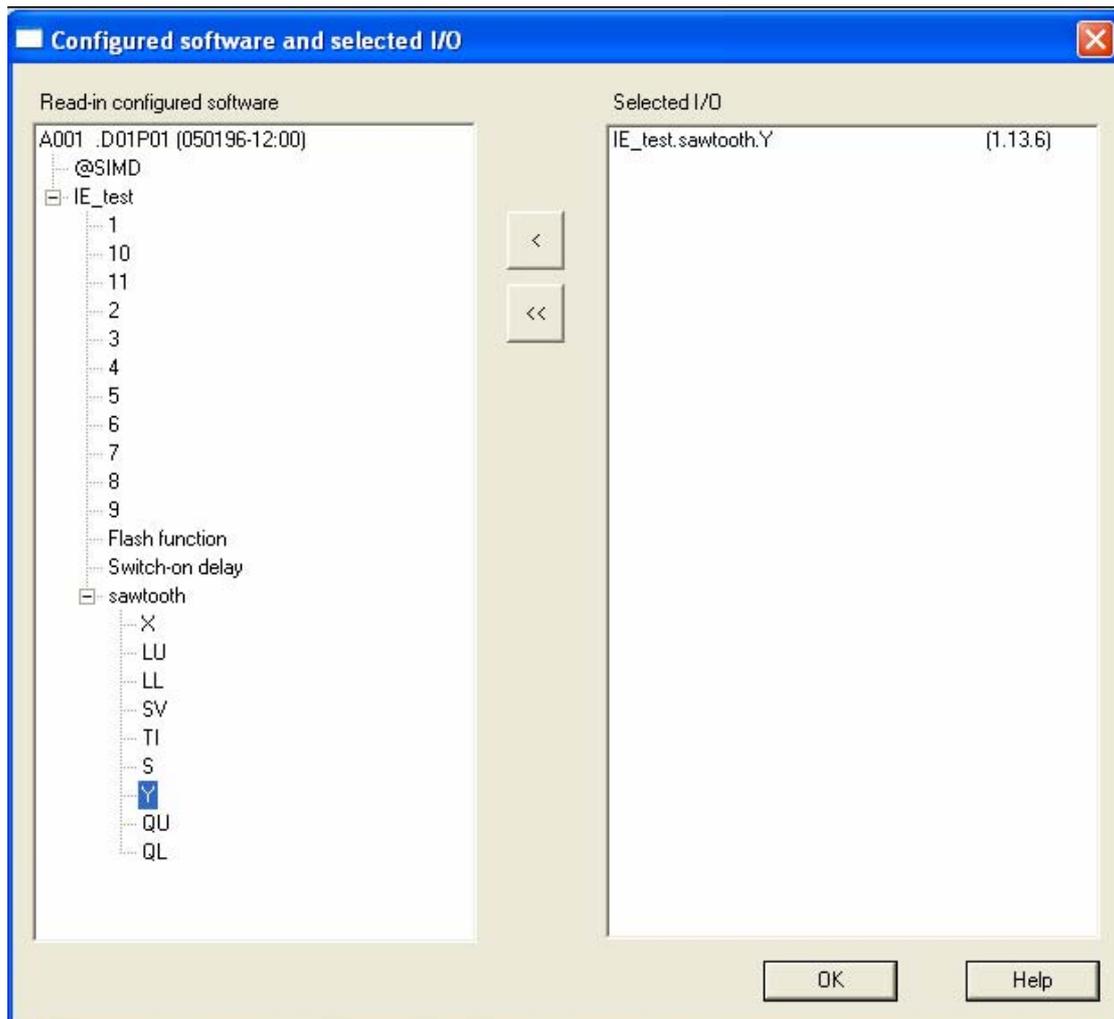
General

Read level Chart

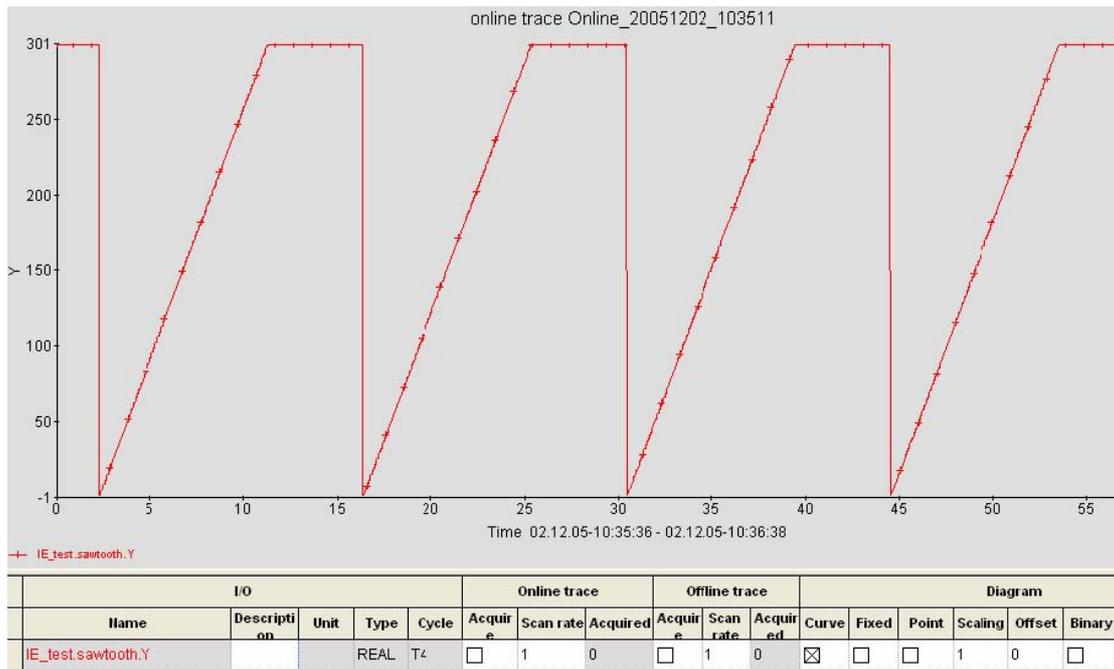
Timeout [s] 5

OK Cancel Help

6. 点击  设置要跟踪的数据。



7.察看最后的结果。



需要注意的是使用CP5100 模板来进行数据跟踪，其组态的数据与CP51M1 略有不同，请参照Symtrace 帮助文件或TDC 的在线帮助查阅即可。关于Symtrace 软件想了解更多信息可以查阅其帮助文件或登陆[www.symtrace.de](http://www.symtrace.de)。

## 附录一推荐网址

### AS

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/>

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>

### NET

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/>

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

Net常问问题: <http://support.automation.siemens.com/CN/view/zh/10805868/133000>

Net更新信息: <http://support.automation.siemens.com/CN/view/zh/10805868/133400>

“找答案”Net版区: <http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1031>