



Doc. no.
 Rev. ind. 3
 Date 030922
 Issuer Johan Björklund
 Dept. CNAUS
 Phone
 Fax
 E-mail

To:

Copy:

TEMPLATE:APR-VAS GEN FORM INT, A4 P EN, R1.DOT; FILENAME:DOCUMENT1; PRINTDATE:2004-4-12 20:31; SAVEDATE:4/12/2004 12:57 PM

Subject: Control IT 3.2/7 - Mr Engineer - Hands-on Exercise - Course A

Table of contents

1	导言	3
2	练习	3
2.1	控制器需备	3
2.1.1	任务	3
2.1.2	解决方案	4
2.2	创建系统硬件架构	9
2.2.1	任务	9
2.2.2	解决方案	10
2.2.2.1	创建 AC800C CPU	10
2.2.2.2	设置 AC800C I/O	12
2.2.2.3	设置 AC800 C Profibus	12
2.2.2.4	AC800M CPU 的设置	13
2.2.2.5	AC800 M I/O 的设置	15
2.2.2.6	AC800 M Profibus 的设置	16
2.2.2.7	下载并连线	16
2.3	Function Block Editor 功能块编辑	17
2.3.1	任务	17
2.3.2	解决方案	17
2.4	Control Builder 基础	23
2.4.1	分配 I/O	23
2.4.1.1	任务	23
2.4.1.2	解决方案	23
2.4.2	自动创建文档	24
2.4.2.1	任务	24
2.4.2.2	解决方案	24
2.5	Structure Text 结构文编辑	26
2.5.1	任务	26
2.5.2	解决方案	26
2.5.2.1	模拟项目	28

ABB China Ltd



Doc. no.
Rev. ind. 3
Date 040412

2(53)

2.6	控制和选择执行速度及优先	30
2.6.1	任务控制	30
2.6.1.1	任务	30
2.6.1.2	解决方案	30
2.7	Ladder Diagram 梯形图	31
2.7.1	任务	31
2.7.2	解决方案	31
2.8	Instruction List 指令表	33
2.8.1	任务	33
2.8.2	解决方案	33
2.9	Sequential Function Charts	34
2.9.1	任务	34
2.9.2	解决方案	34
2.10	Function Block Types	37
2.10.1	任务	37
2.10.2	解决方案	37
2.11	PID Loop Setup	42
2.11.1	任务	42
2.11.2	解决方案	42
	2.11.2.1 Function BlocksError! Bookmark not defined.
	2.11.2.2 Control ModulesError! Bookmark not defined.

1 导言

这堂关于 Control Builder M 的课需两天完成。此文件内容基于 Control Builder 3.1/2.

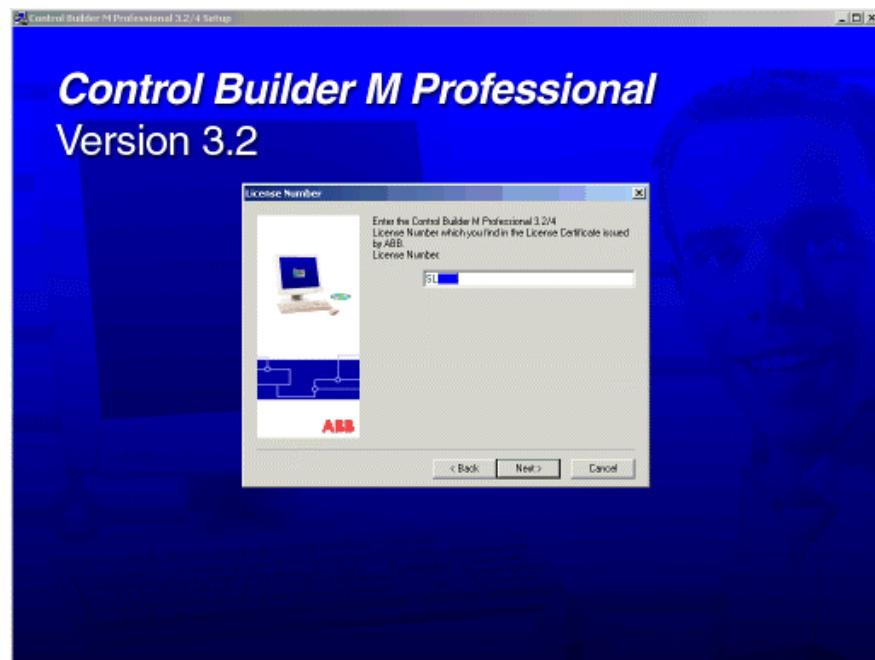
2 练习

2.1 控制器需备

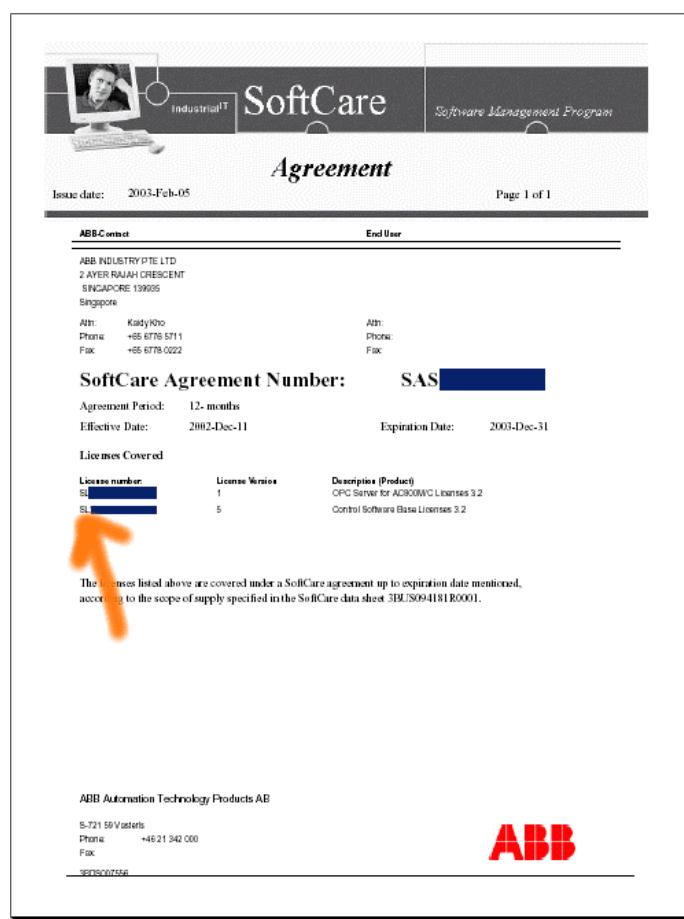
2.1.1 任务

我们首先需要在控制器里加载固件和设置 IP 地址。

如还没安装 Control Builder 软件的话, 请立即安装。

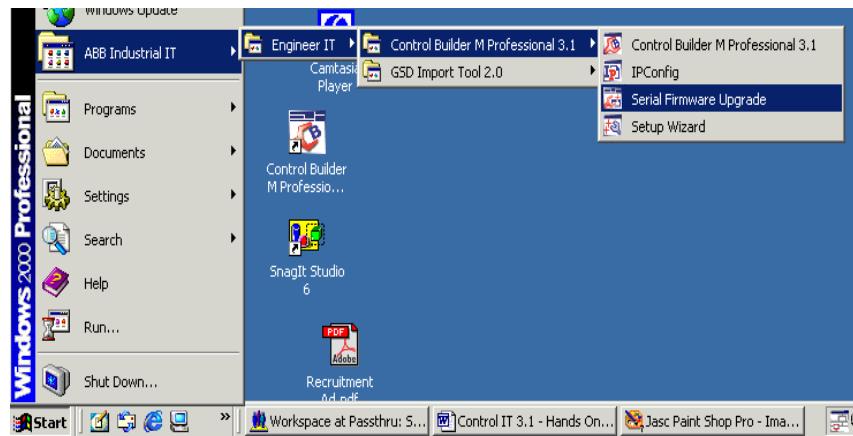


如需相关序列号, 请查看您的 ABB 许可证。



2.1.2 解决方案

我们首先确认该控制器的固件是否需要更新。它必须符合您拥有的 Control Builder 版本。接下来，请在 Windows 启动程序里打开序列号升级固件工具 (Serial Firmware Upgrade) , 见下图。

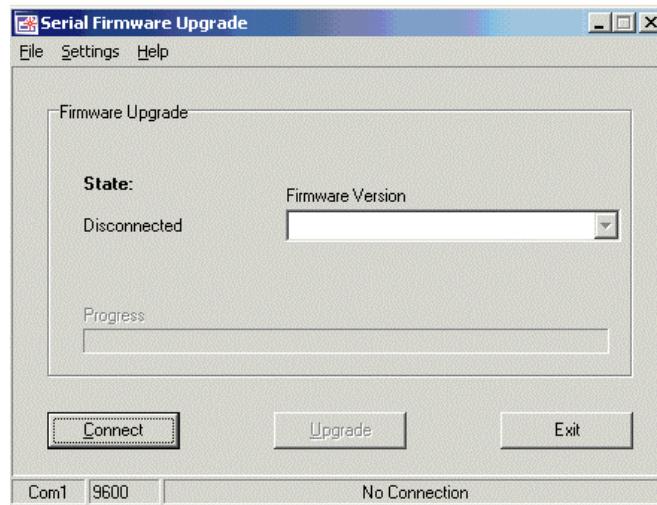




Doc. no.
Rev. ind. 3
Date 040412

5(53)

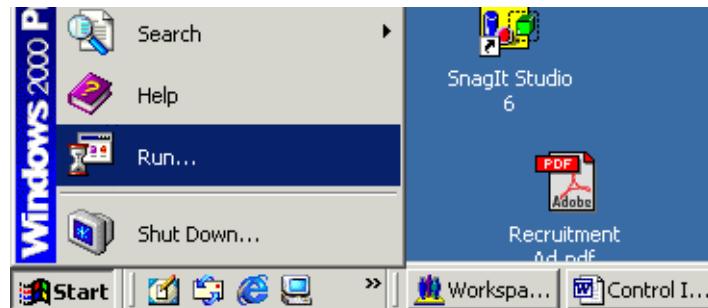
把控制器 Serial 电缆连接到 AC800M 控制器 COM4 上，然后把另一头接到您的电脑（AC800C 上是 COM0）。请按控制器上的复原按钮，再按升级工具里的连接“Connect”键。（注：在 AC800C 上复原，您必须先把控制器背后的转盘调到 0 的位置。如果转盘不在 0 的位置，复原功能只能执行热启。）



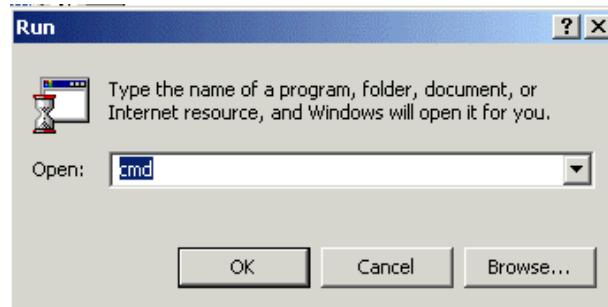
连接上之后，在下移菜单里选择最新的固件版本并按升级“Upgrade”按钮。升级过程需几分钟完成。

下一步是设置控制器的 IP 地址。查询电脑 IP 地址最简便的方式就是在 Command Window 里运行 Ipconfig 命令。

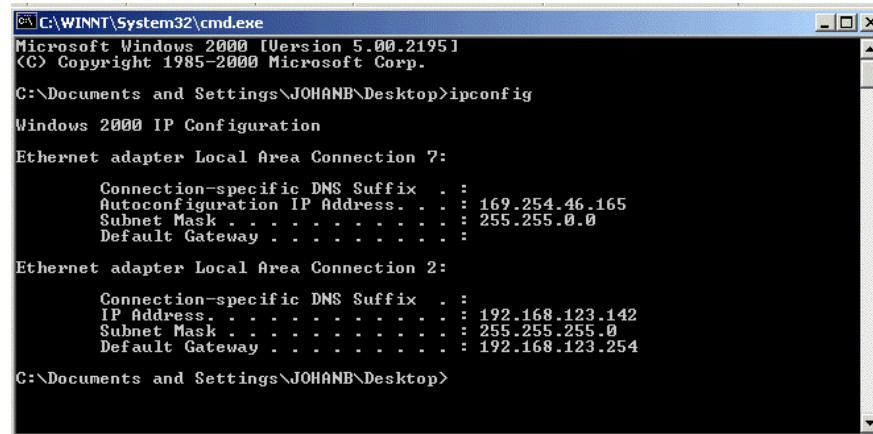
首先，使用运行“Run”功能。



输入“cmd”让后按回车或点击 OK。



运行 IPCONFIG 功能来看您电脑的 IP 地址。



```

C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.21951]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\JOHANB\Desktop>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection 7:

  Connection-specific DNS Suffix  . : 
  Autoconfiguration IP Address. . . . . : 169.254.46.165
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 

Ethernet adapter Local Area Connection 2:

  Connection-specific DNS Suffix  . : 
  IP Address. . . . . : 192.168.123.142
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.123.254

C:\Documents and Settings\JOHANB\Desktop>

```

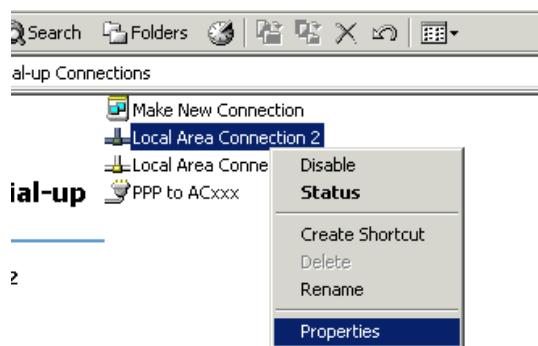
选择一个符合您控制器的 IP 地址。

您也可以自定义 IP 地址。鼠标右击网上邻居 “My Network Places”。

选择属性 “Properties”。

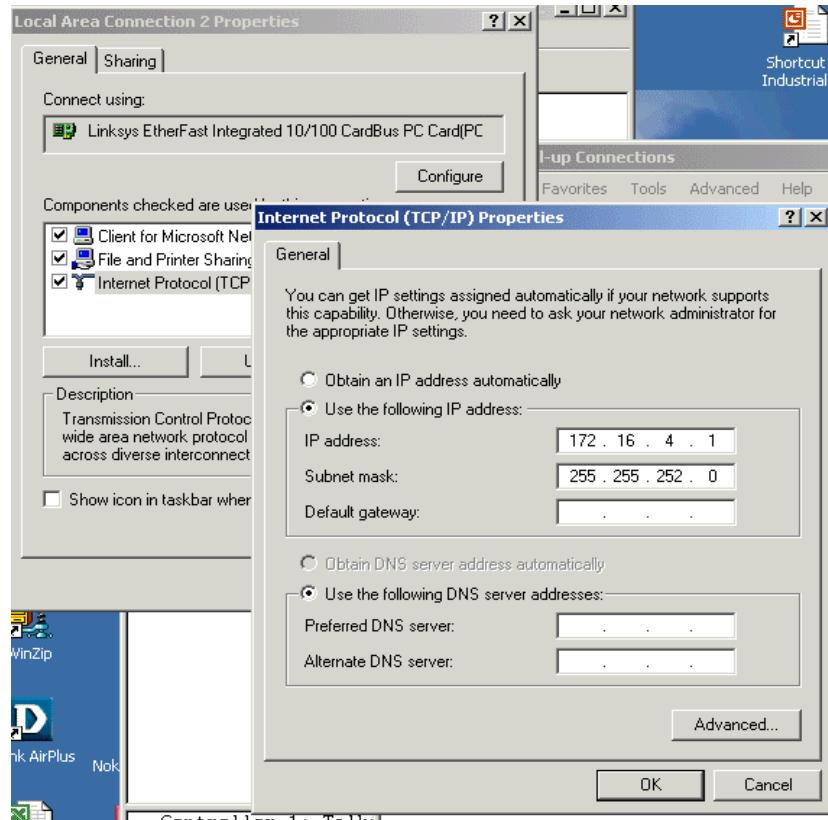


再次选择需组太网卡并鼠标右键点击，选该项的属性 “Properties”。



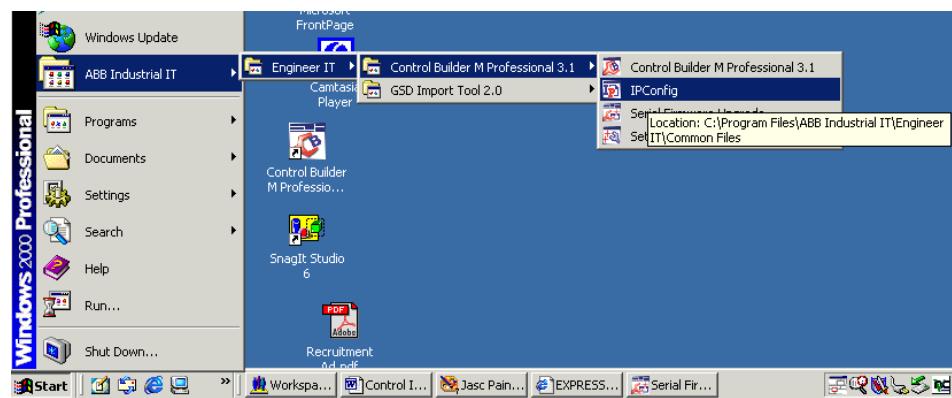
或双击该网卡打开。

在新打开的窗户里，确保 TCP/IP 一项已被打勾并双击该项。



您电脑的 IP 地址现已设定完毕。接下来是设置控制器的 IP 地址了。

请启动 IP Config Tool 工具。

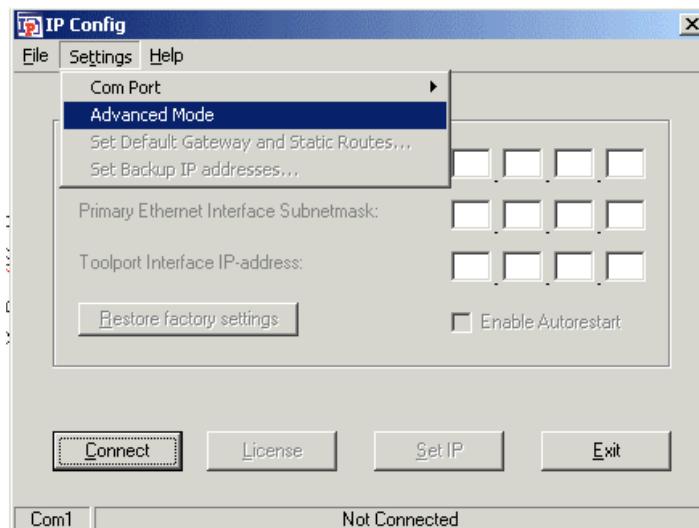


请进入高级模式“Advanced Mode”，这样您就可以任意设置 IP 地址。

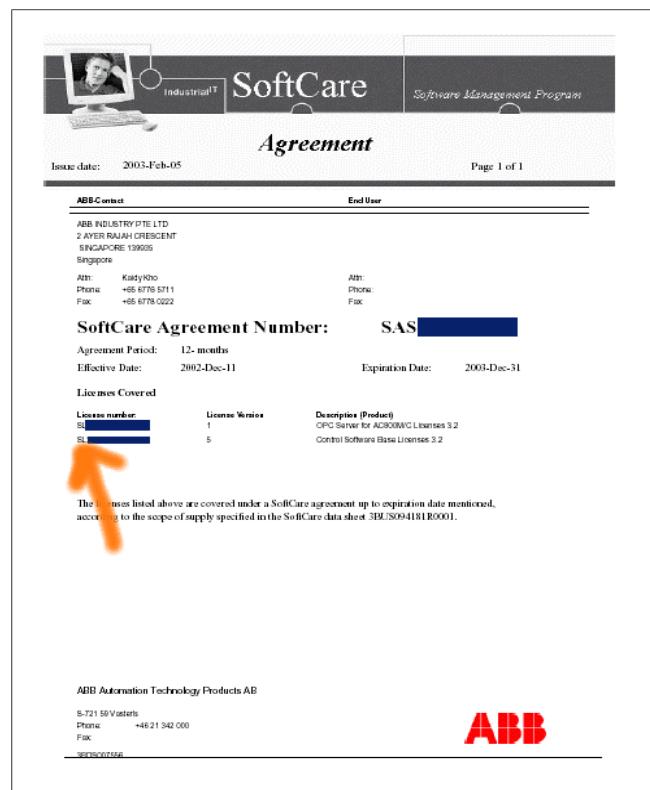


Doc. no.
Rev. ind. 3
Date 040412

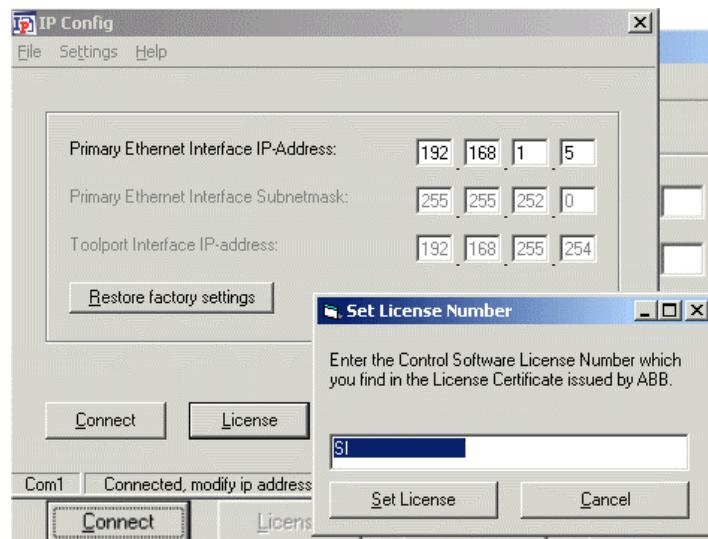
8(53)



在点击接入“Connect”之前必须复原控制器。
您还需要填写序列号，这可以在 ABB 公司序列证书上找到。



见以下序列号填写方式。



输入您需要的 IP 地址和子网络面具 SubNetMask。请按“Set IP”。

您在任何时间都可以用 PING 命令来查询是否正确地连接上控制器。运行 cmd 功能之后，输入 PING 『』；在『』内填写 IP 地址。

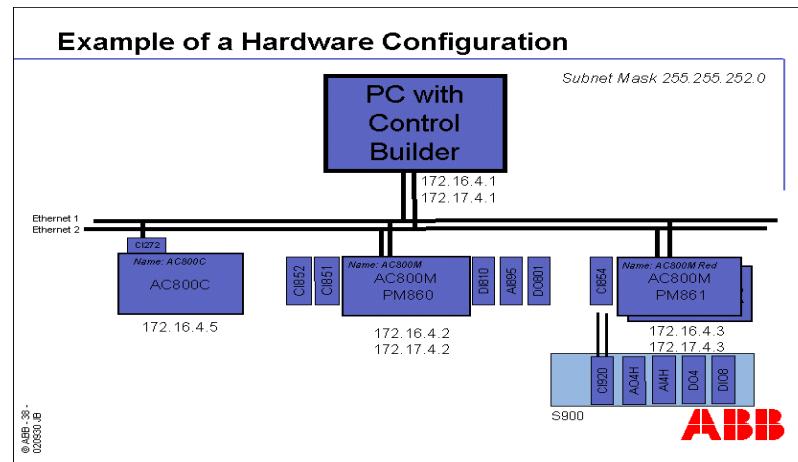
2.2 创建系统硬件架构

2.2.1 任务

这个任务的目的是在系统软件内创建一个同样与您项目的架构。

请您选择控制器 CPU 类型。

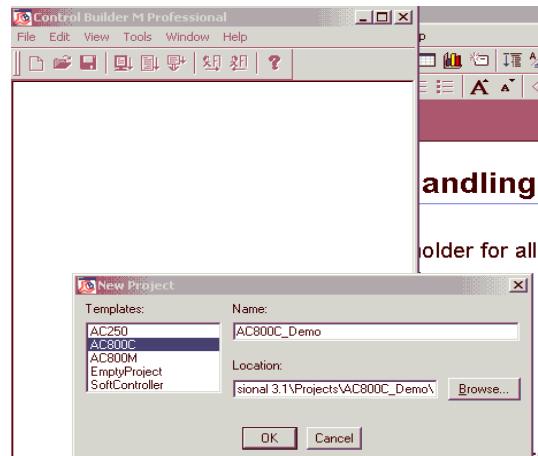
见以下组太案例。



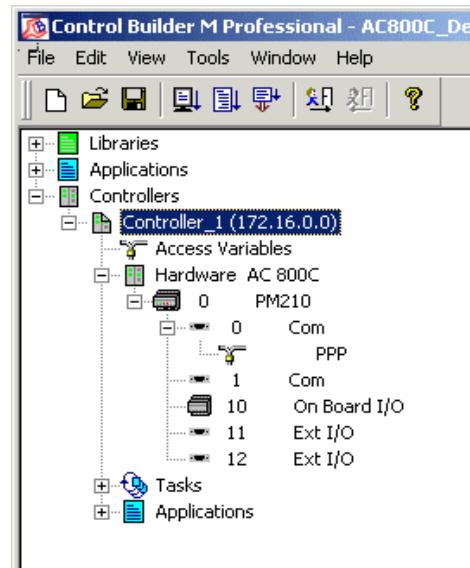
2.2.2 解决方案

2.2.2.1 创建 AC800C CPU

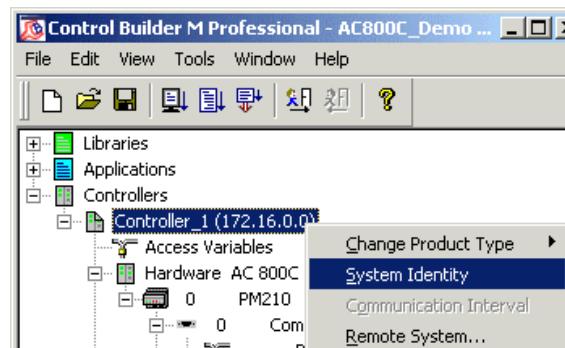
在 File 目录里，选择新项目“New Project”，再选控制器类型。



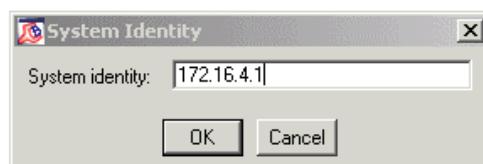
创建项目后，您可以查询控制器下的所有配件。如果是新创项目的话，您可以看到控制器已经配置了 CPU。以后在此项目中再添加控制器，必须自行添加 CPU。



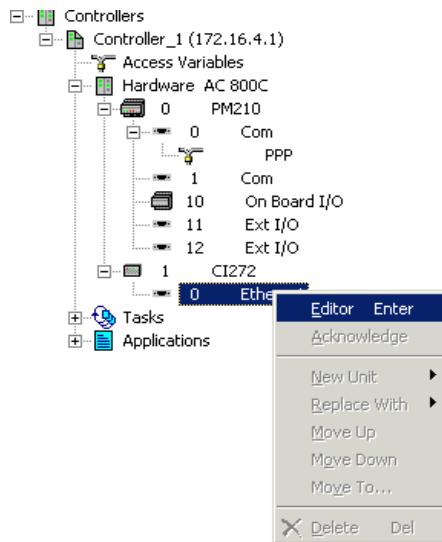
鼠标右击 Controller_1 进行系统身份设置 (System Identity)。

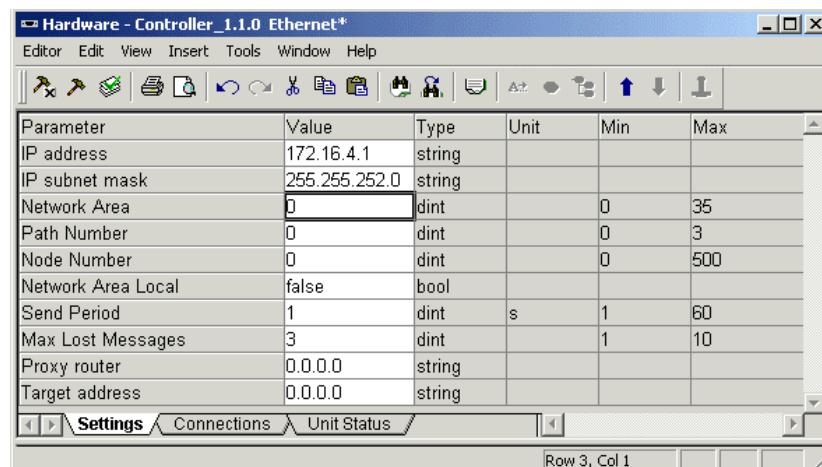


系统身份将是已设定的控制器 IP 地址(The IP address of my controller is 172.16.4.1)。



下一步您可以指定以太网端口的 IP 地址。如果您是用的 AC800C，您需要事先添加以太网卡(右边第一个扩展槽是 1 号槽，旁边的便是 2 号槽等等)。您即可以右击以太网卡图标，也可以双击该项来打开设置窗口。





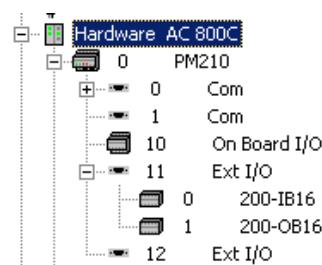
2.2.2.2 设置 AC800C I/O

到添加 I/O 的时候了。请按照您项目的设计图案，连接 I/O 模块。

我的控制器是按以下 I/O 配置的：

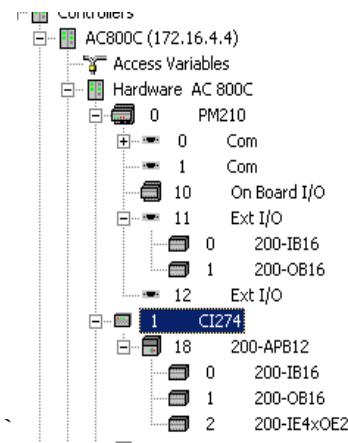
- IB16
- OB16

设置完将可以在 Control Builder 里看到以下的排列方式。直接连接在 CPU 旁的本地 I/O 显示在 11-Ext I/O 栏目里。12-Ext I/O 栏目显示的是 CPU 下方通过 RS232 电缆连接的本地 I/O。



2.2.2.3 设置 AC800 C Profibus

您可能会有一些 Profibus I/O。这里安装的 Profibus 通讯卡件是 CI274。您还需要安装 S200 Profibus 子件；这里是 CI274 下面的 200 APB-12 子件。别忘了把实践模块上的 IP 地址和 Control Builder 里的 IP 地址核实一下。



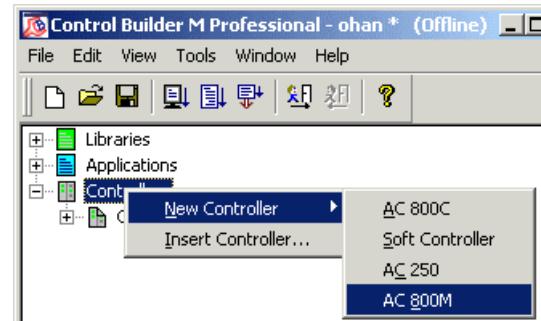
2. 2. 2. 4 AC800M CPU 的设置

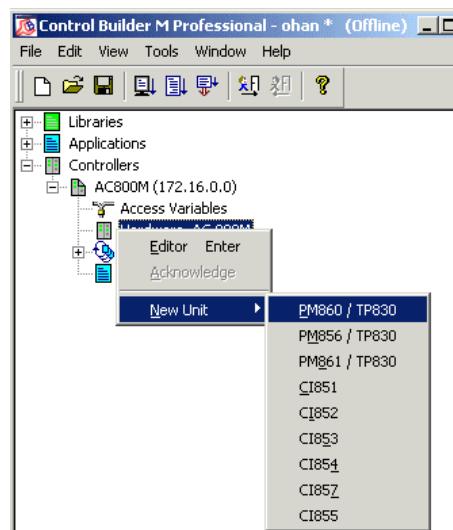
从头开始

在新建项目窗口里选择 AC800 M 控制器，然后点击 OK。

在已创项目里添加新的控制器

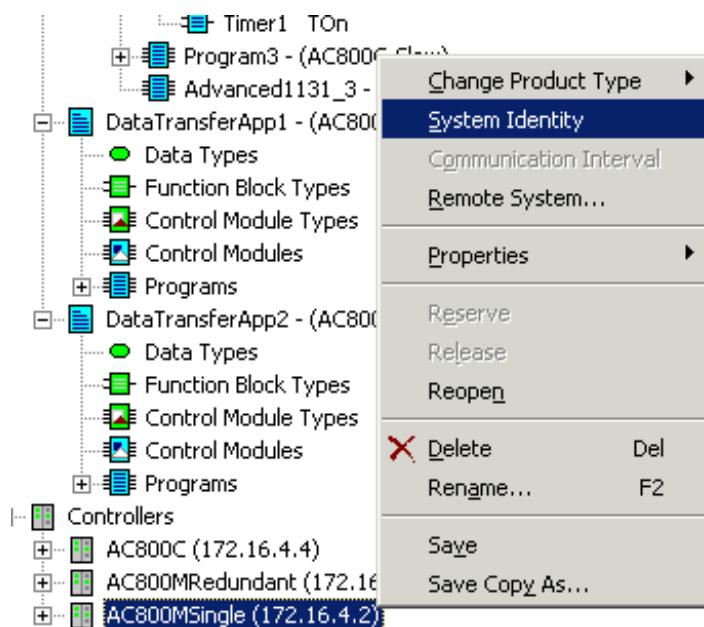
右击控制器 “Controller” ，把鼠标移到 “New Controller =>” 上并选择新的控制器。别忘了添加控制器 CPU(例如 PM856, PM860 or PM861)



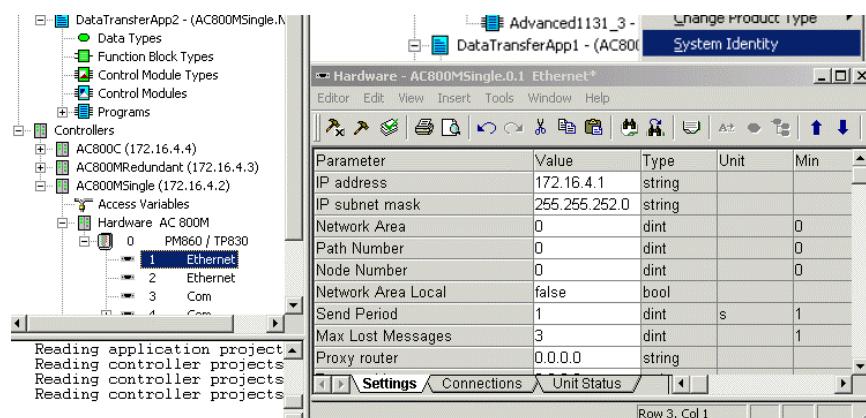


开始组态

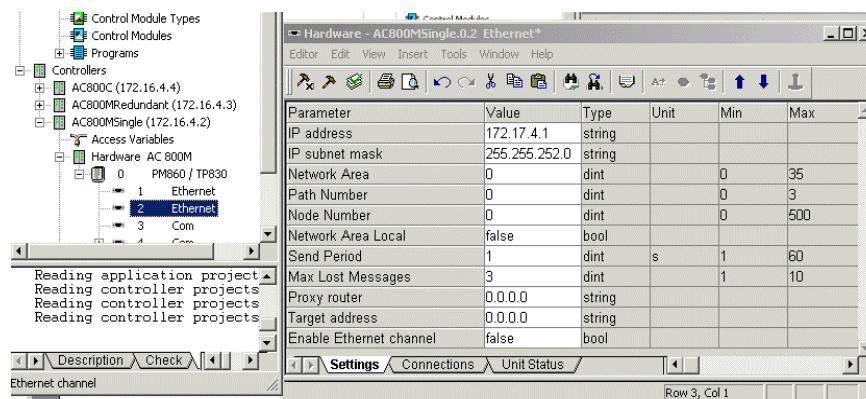
首先按控制器的真实地址组态系统身份“System Identity”。



接下来组态以太网端口（设置 IP 地址和子网掩码）。从第一个开始。



之后再组太冗余端口。此时，您可能还用不到该项功能，但必须进行组太。最简便的方法是在 IP 地址的第二组添加 1，见图。



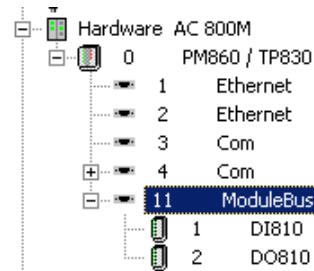
现在您已经可以进行下载了。

2.2.2.5 AC800 M I/O 的设置

又到添加 I/O 的时间了。再次按照您的实际图案来连接 I/O 模块。

我的控制器在本地电缆模块总线 (Electrical Modulebus) 上连接了以下的 I/O (Position 1-12)。

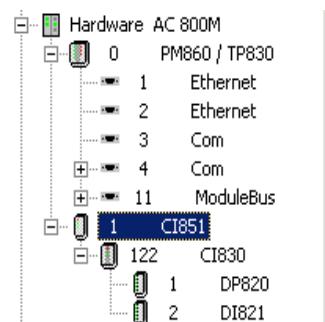
- DI810
- DO810



您可以通过光纤模块总线最多再添加 7 串 12 个 I/O 模块。第一串的标签是 10x，第二个是 11x，等等。

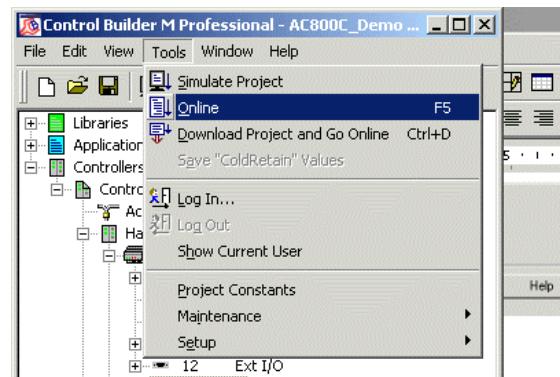
2.2.2.6 AC800 M Profibus 的设置

您可能会有一些 Profibus I/O。这里安装的 Profibus 通讯卡件是 CI851 或 CI854 如需要冗余的话。您还需要安装 S800 Profibus 子件，有 CI830 (或支持冗余的 CI840)。别忘了把实践模块上的 IP 地址和 Control Builder 里的 IP 地址核实一下。

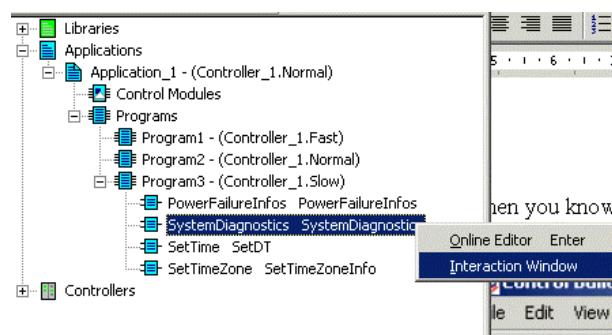


2.2.2.7 下载并连线

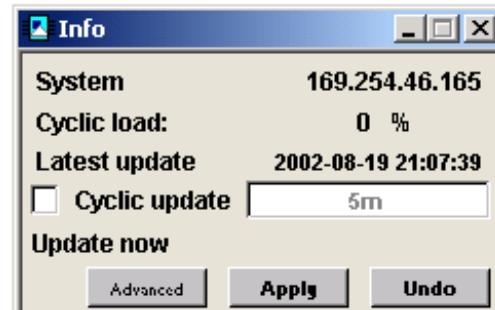
当您确认系统无组态错误，即可下载项目并连线“Download Project and Go Online”。



当您是新建的项目，系统会自动预制系统诊断窗口 (System Diagnostics Window)。右击此功能，并选对话窗口 (Interaction Window)。



这个窗口提供一些 CPU 及应用软件的数据。请点击 Advanced 键看详细内容。

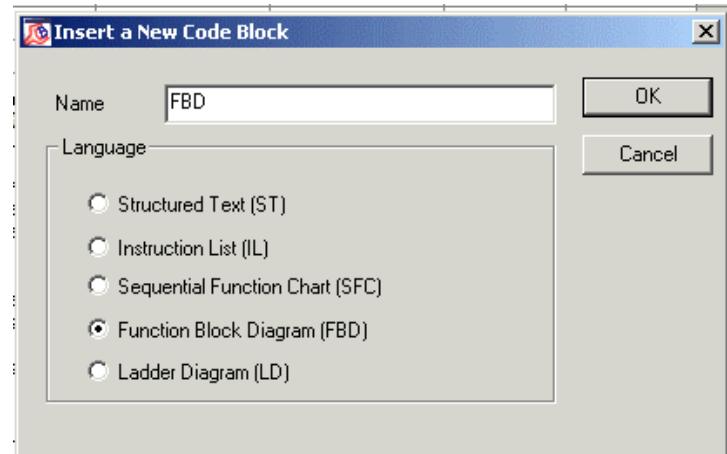
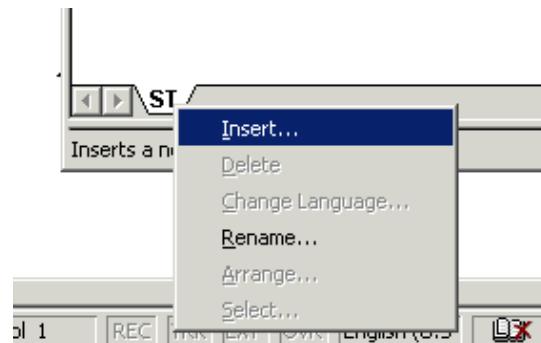


2.3 Function Block Editor 功能块编辑

2.3.1 任务

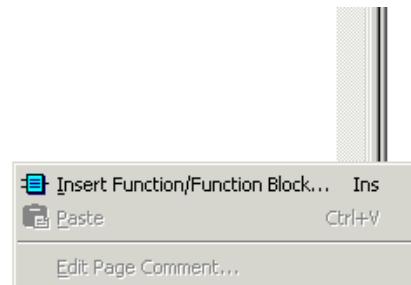
2.3.2 解决方案

建一个新的 FBD 标签。

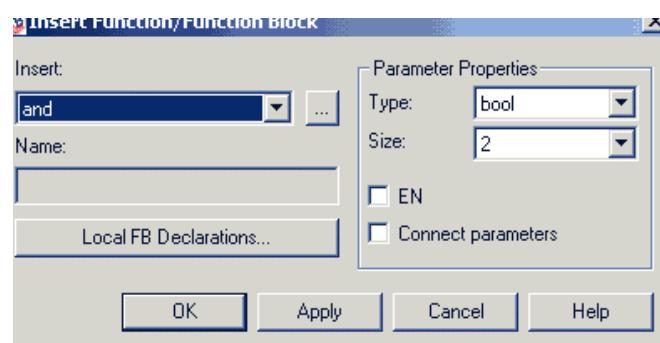


我们来创建一个简单的马达启动器“MotorStarter”。添加一个与门和一个或门。

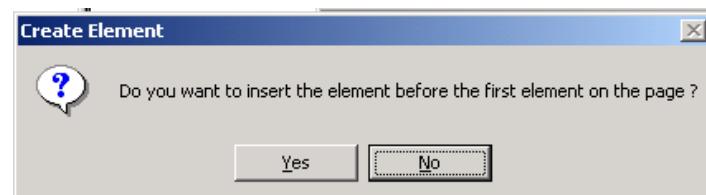
在编辑窗口里，右击任意位置并在打开的菜单内选择插入功能块“Insert Function Block”。



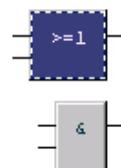
选择一个与门并点击 Apply。再添加一个或门。



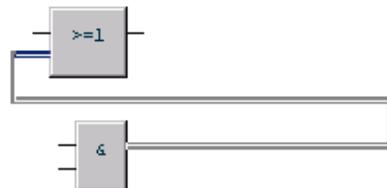
当被提示是否把或门排列在与门之前，请选是“Yes”。



结果

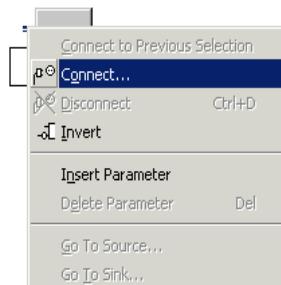


现在来让我们把输入输出信号连接起来。把鼠标标准心移到与门的输出口，按下 Ctrl 键后点击并拉到或门的输入口上。



下一步我们需要给未连接的口分配变量。

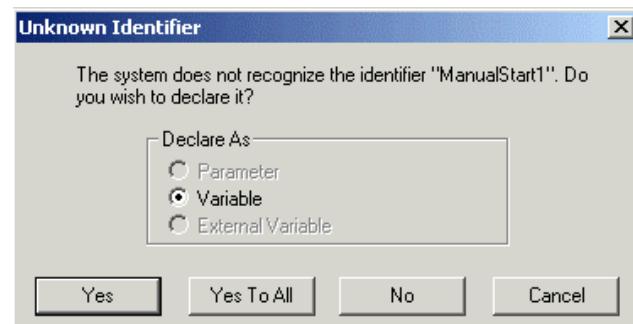
右击或门的第一个输入口并选连接“Connect”。



打入 ManualStart1 并点击 Apply/Next。

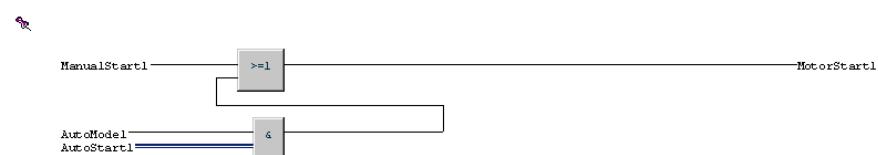


系统会定义此口未声明，按 Yes to All 自动确认声明。

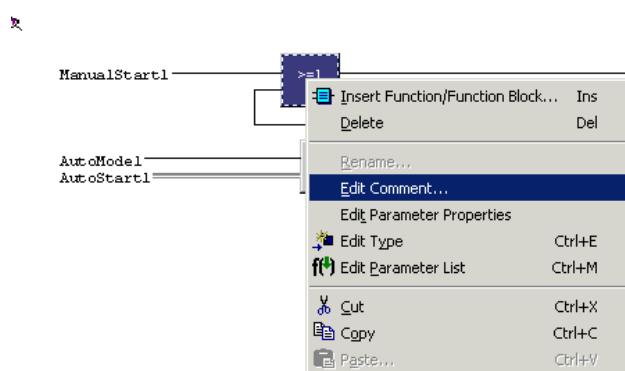


请重复以上步骤直到所有输入输出口已被连接。

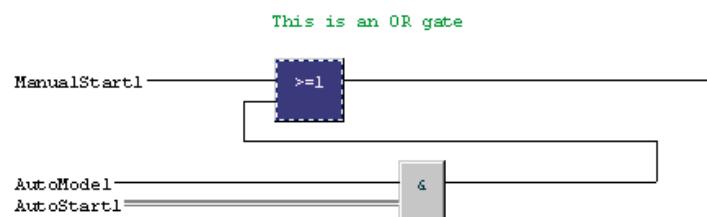
5	ManualStart1	bool	retain
6	AutoStart1	bool	retain
7	AutoMode1	bool	retain
8	MotorStart1	bool	retain
9			



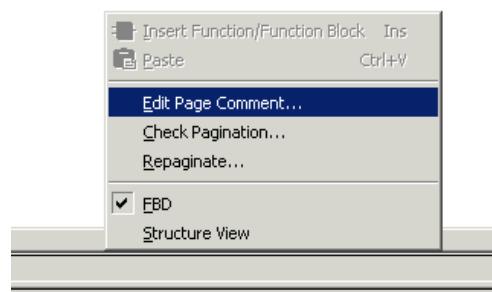
请右击功能块，并选插入注释。

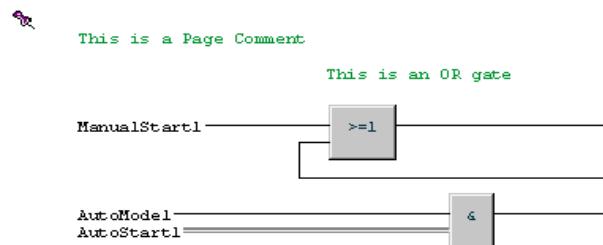


打入任何注释后点击 OK。

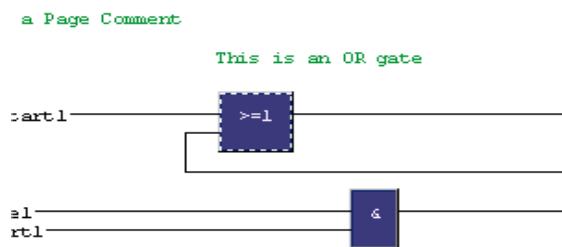


您也可以插入一个页注释。

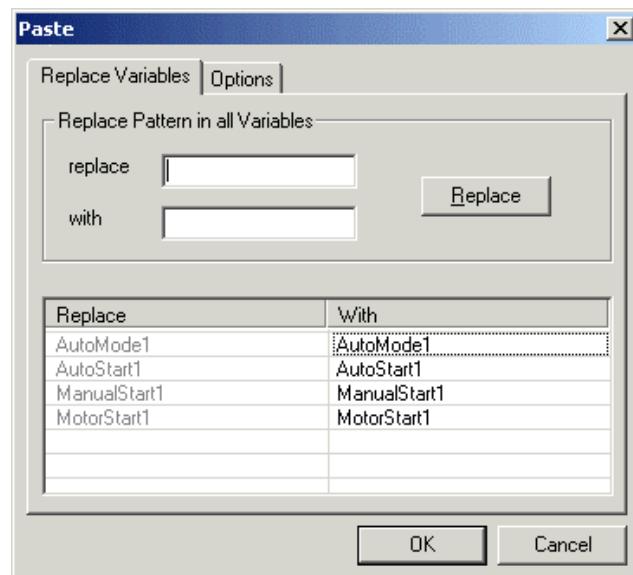




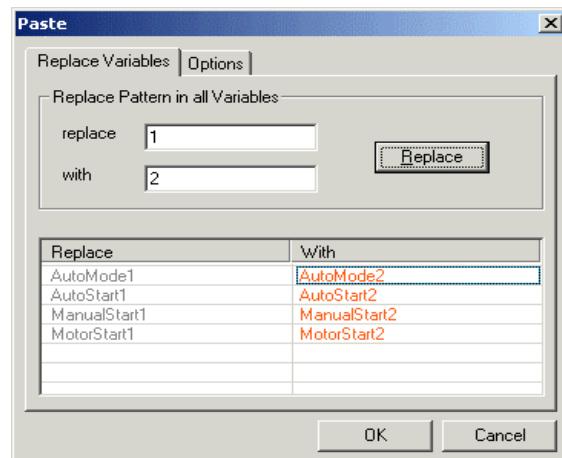
我们确信这个马达启动器非常有用，并想再创建一个同样的。点击与门功能后按下 Shift 键再点击另一个或门。



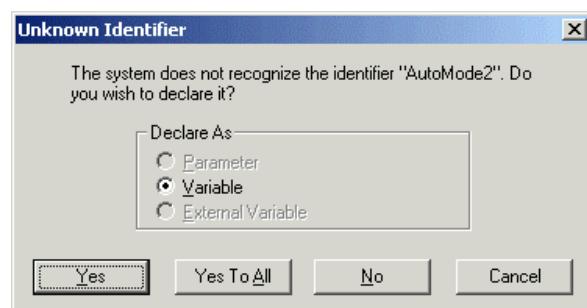
在键盘上同时按下 Ctrl-C 键来复制所选功能块，然后同时再按 Ctrl-V 来执行粘贴功能。此时您的屏幕上会显示一个窗口。您会发现这是个具有强大功能复制工具。



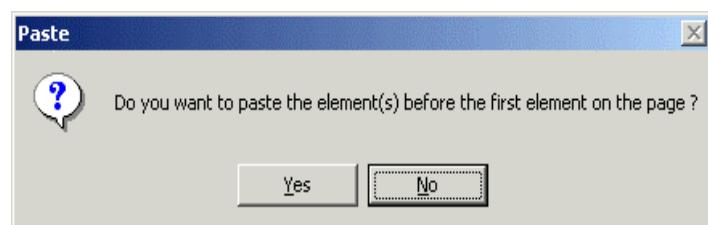
在“Replace”空格里填写“1”，“With”里填写“2”并点击 Replace 按钮。可以看到以下图中四个新的变量已被自动创建



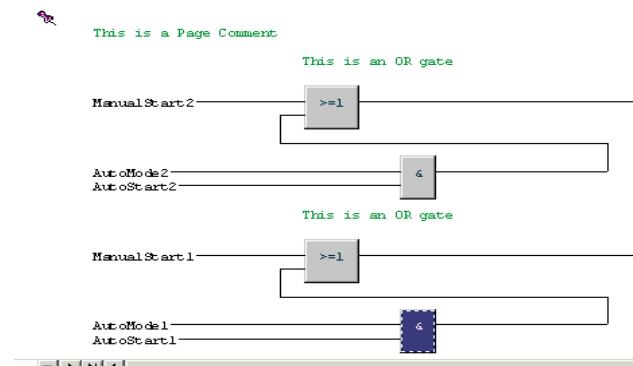
点击 OK。您会被提示是否声明新变量，请按 Yes 或 Yes to All。



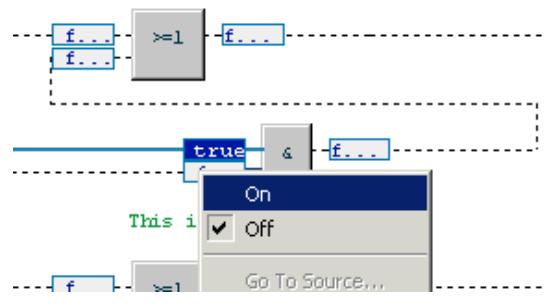
最后的问题是是否把新建的功能块附加到老的之前。回答 Yes。



下面是得到的结果。



再分配两个数字输出分别到 MotorStart1 和 MotorStart2 信号。模拟运行您的应用程序并可强制任何信号开关。



2.4 Control Builder 基础

2.4.1 分配 I/O

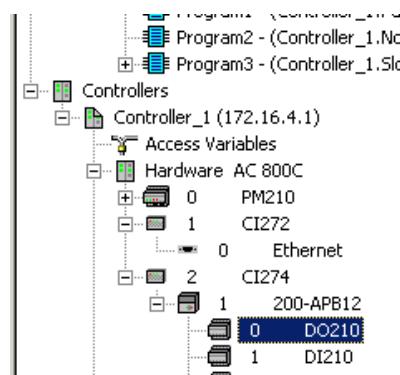
2.4.1.1 任务

分配 I/O 频道到状态变量“Status Variable”。

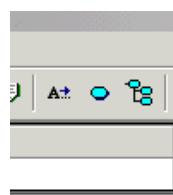
2.4.1.2 解决方案

最好方法是从 I/O 板块（模块）上找到 I/O 并直接分配。

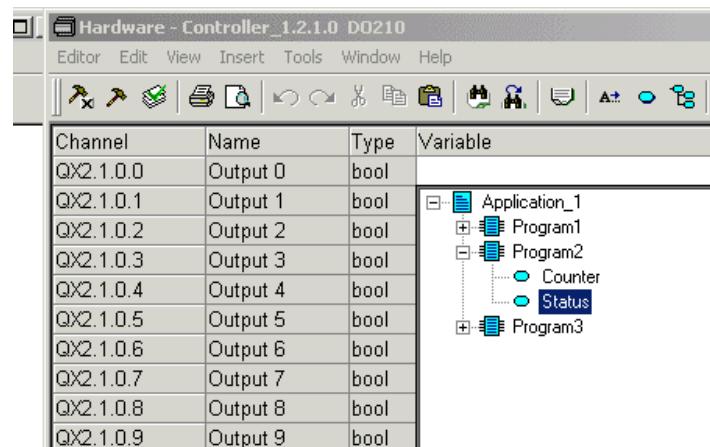
选择您项目内已组态的任何数字输出板块。（地点于名称是根据您的硬件组态定义的）



选中 I/O 频道后点击浏览“browse”。(快击键是一个树型图标, 见图)



浏览到信号位置后, 双击图标并分配它到 I/O 频道。



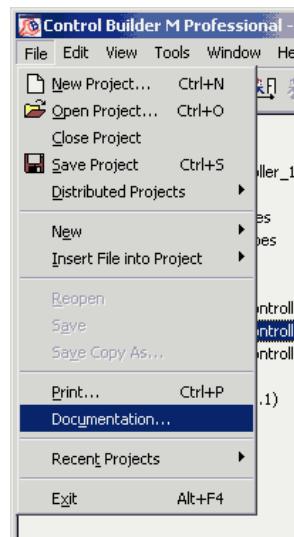
2.4.2 自动创建文档

2.4.2.1 任务

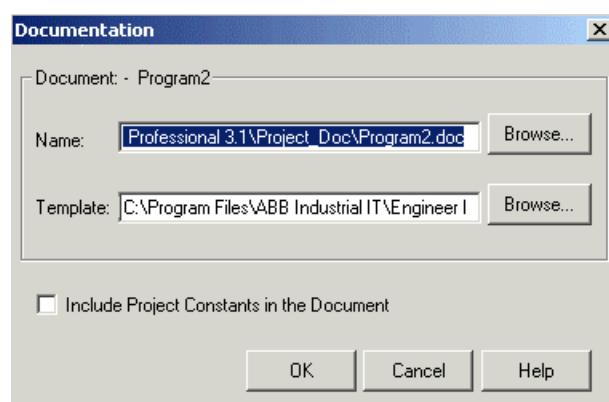
您的任务是记录硬件结构和编辑程序。

2.4.2.2 解决方案

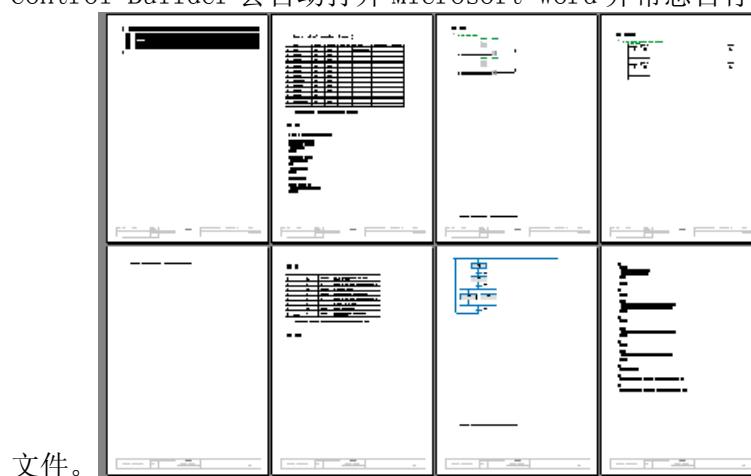
请在项目的树型目录里选定您想记录的区域（点击一次）。您可选数据库, 应用程序或一个控制器。然后在 File 目录下, 选 Documentation。



您当然还需要指定存放文件的地方及文件模板。



Control Builder 会自动打开 Microsoft Word 并帮您自行创建记录项目的



文件。

2.5 Structure Text 结构文

2.5.1 任务

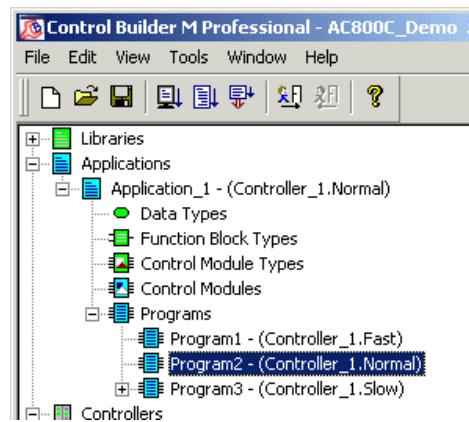
我们想要创建一个计数器程序，变量 Counter 必须在每个循环加 1。如果 Counter 大于 100，我们将它复原到 0。在 Counter 大于 50 以后我们将另一个逻辑变量 Status 改为 TRUE，小于 50 就是 FALSE。

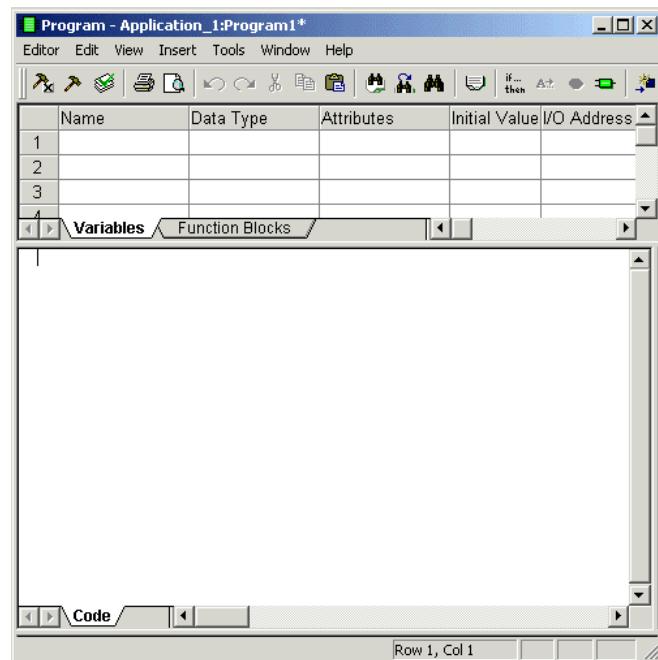
编写完后我们希望进行模拟并测试它的功能。

2.5.2 解决方案

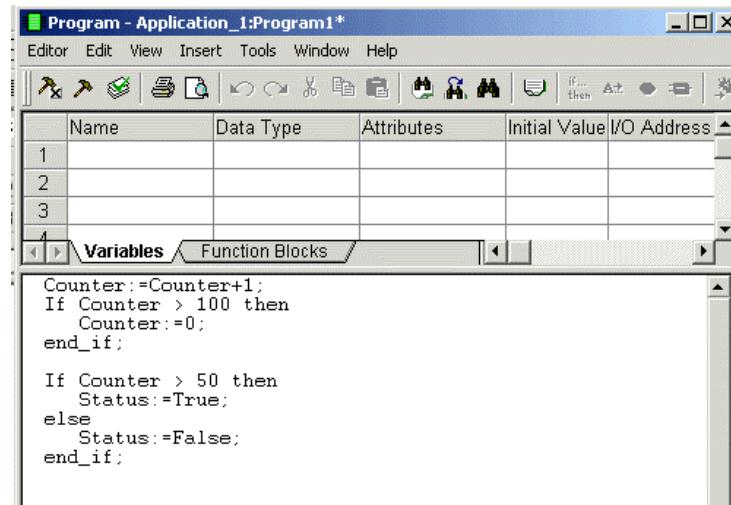
现在是编写你们第一个程序的时候了。应用结构文来编写控制程序。

双击 **Application_1/Programs/Program_2** 打开程序窗口。您也可以右击进入程序编辑工具 (Editor)。

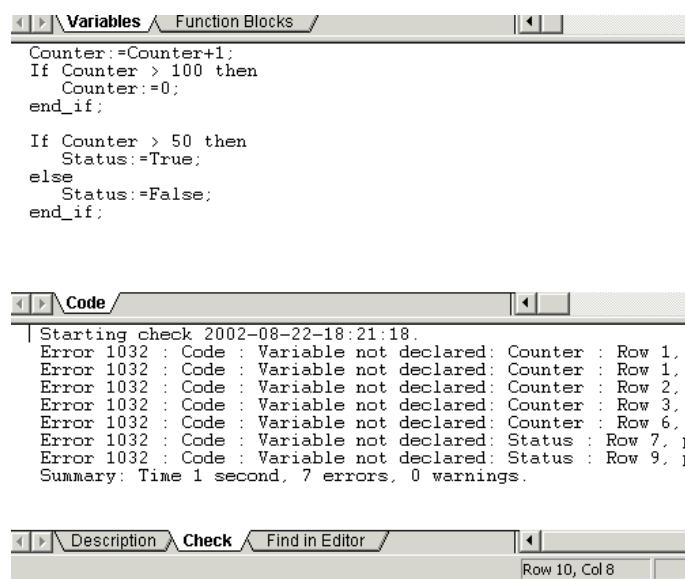




现在可以开始编写程序。请记住结束每个命令句 (statement) 时都得用“；”符号。



随后别忘了声明 (declare) 所有变量。您可以在变量表里填写每个变量并点击“Check”按钮检查输入的变量名称。也可以直接点击“Check”，如有错误，错误信息会自动创建在屏幕下方。



```

Variables
Counter:=Counter+1;
If Counter > 100 then
  Counter:=0;
end_if;

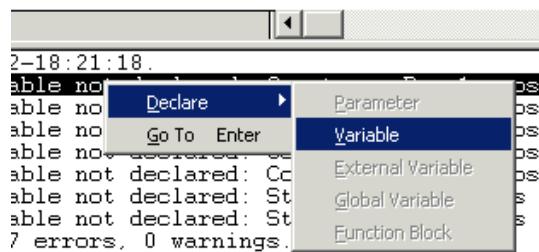
If Counter > 50 then
  Status:=True;
else
  Status:=False;
end_if;

Code
Starting check 2002-08-22-18:21:18.
Error 1032 : Code : Variable not declared: Counter : Row 1,
Error 1032 : Code : Variable not declared: Counter : Row 1,
Error 1032 : Code : Variable not declared: Counter : Row 2,
Error 1032 : Code : Variable not declared: Counter : Row 3,
Error 1032 : Code : Variable not declared: Counter : Row 6,
Error 1032 : Code : Variable not declared: Status : Row 7, p
Error 1032 : Code : Variable not declared: Status : Row 9, p
Summary: Time 1 second, 7 errors, 0 warnings.

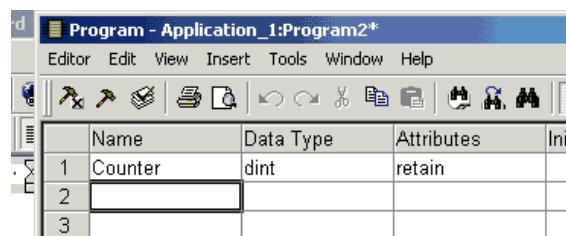
Check
Row 10, Col 8

```

右击错误信息并选择声明变量。



这会自动把变量名称填入表内。之后，还必须填写变量类型。在 data type 处请填写“DINT”。

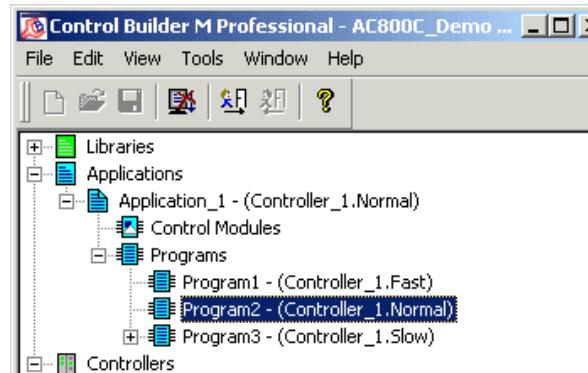


Status 变量的处理方式也是一样的。

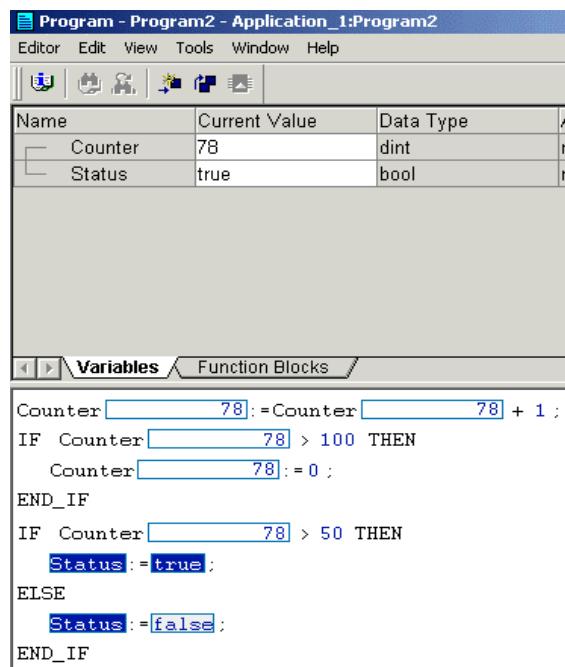
按 Check 检查错误。最后点击 Apply and close 运用并推出编辑工具。

2.5.2.1 模拟项目

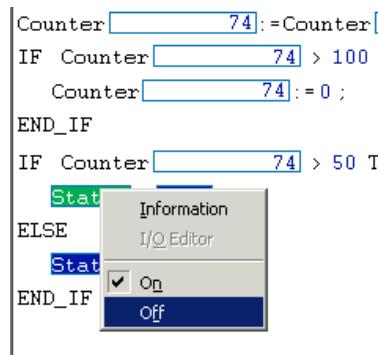
按模拟键 Simulate 并双击 Program2 打开在线编辑工具。



如果一切顺利，您应该看到 Counter 变量在跳动。



现在您可以强制性地改动在线信号。右击变量即可选择。请试验！



您可以随时改变 Status 和 Counter 信号。右击变量或直接在变量表内更改。

2.6 控制和选择执行速度及优先

2.6.1 任务控制

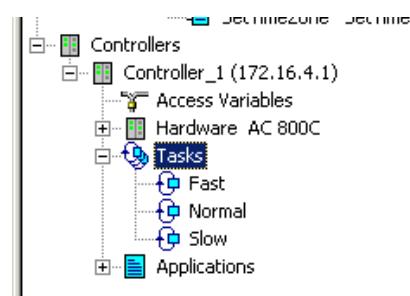
2.6.1.1 任务

学习怎样控制任务进程

2.6.1.2 解决方案

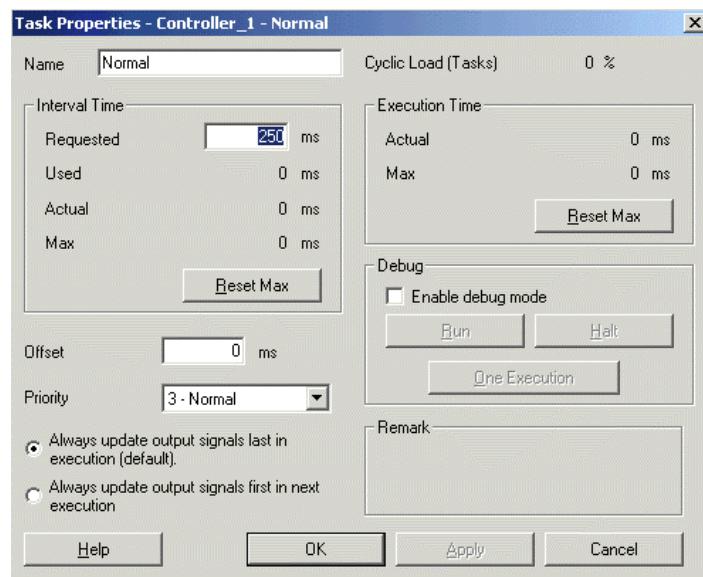
点击模拟按钮开始模拟运行。

扩展正在运行的控制器图标



双击 Normal 任务打开任务属性。

打开 Program2 的程序编辑工具来看运行的变量。



设定循环时为 10ms 并观察变量的循环速度。

下面，我们希望暂停并启动调试状态。点击 One Execution 来设定程序。
按 Ok 结束。

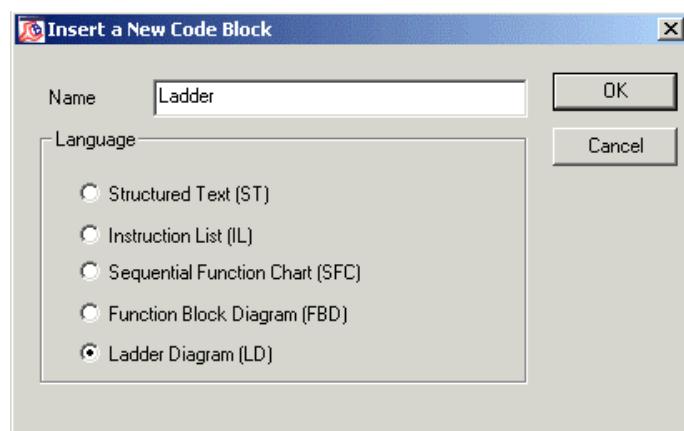
2.7 Ladder Diagram 梯形图

2.7.1 任务

我们希望以梯形图方式来创建一个马达启动器。

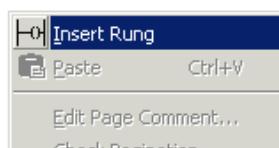
2.7.2 解决方案

打开 Program2，右击页标签并插入新的 Tab。命名为 Ladder 并选择 Ladder Diagram 编程语言。

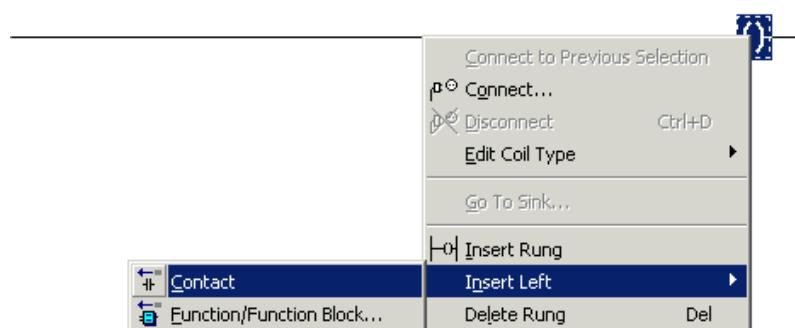


梯形图很接近功能块图。不过这里是用横档“rungs”。

在图中右击并选“Insert Rung”创建第一个横档。



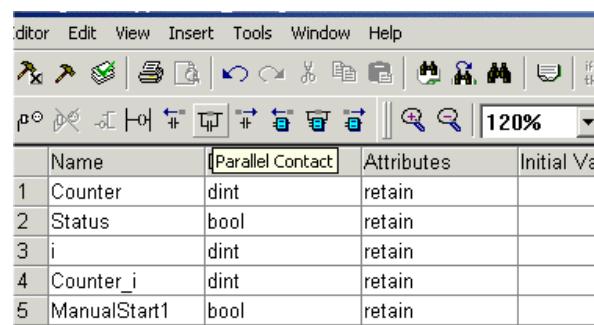
随后再插入一个连接口 Contact。右击 Rung 并选择 Insert Left - Contact。



继续插入另一个连接口。这样我们就成功的编写了一个与门（AND gate）。要再插入或门（OR gate），先选择前两连接口（按 shift 点击即可）。

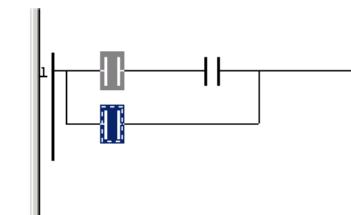


选择插入平行连接口。

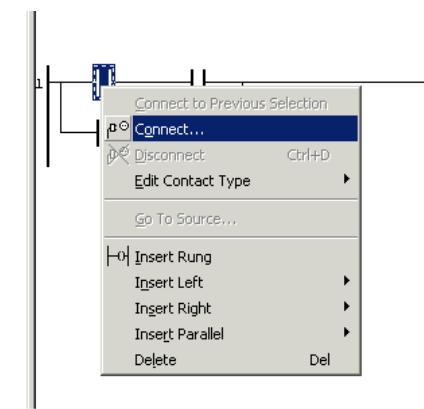


Name	Parallel Contact	Attributes	Initial Va
1 Counter	dint	retain	
2 Status	bool	retain	
3 i	dint	retain	
4 Counter_i	dint	retain	
5 ManualStart1	bool	retain	

结果



还需要把连接口连到变量上去。右击并选 Connect。

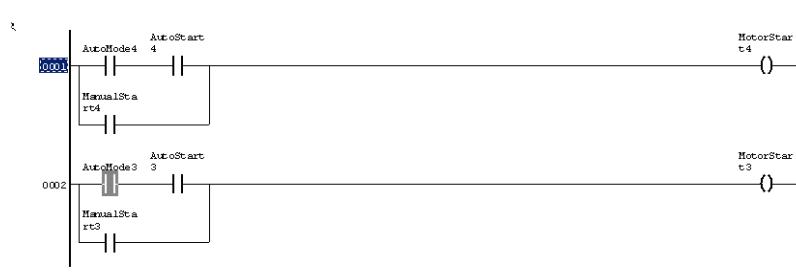


这里的运行方式于功能块编辑工具是一样的。

按下例连接和声明。（AutoStart3, AutoMode3, ManualStart3, MotorStart3）。并且添加两个 Rung 和页注释。



复制并粘贴图案，然后再进行重命名变量 3 到 4。点击最左侧的号码来复制。



连接一些输出到真实 I/O 后进行程序下载并测试组态。

2.8 Instruction List 指令表

2.8.1 任务

我们再次创建一个计数器 Counter，不过这次用指令表 Instruction List。把变量命名为 Counter_IL，也是从 0 数到 100。如果 Counter_IL 大于 50，逻辑变量 Status_IL 定为 FALSE，否则定为 TRUE。

2.8.2 解决方案

相以前一样，创建一个新的 TAB 命名为 IL。指令表于汇编程序 (Assembler) 有相同之处。首先，在每一行都是命令句，并且是基于输入变量到寄存器为主，在进行操作后再导入会变量。

计数器的编程在 IL 里是非常简便的。您只需要掌握软件的若干命令即可开始编写程序。

见以下立案。别忘了声明变量。

Label	Instruction	Operand	Description
	LD	Counter_IL	Load the variable Counter_IL to the register
	ADD	1	Add 1 to the register
	ST	Counter_IL	Store the register in the variable Counter_IL
	GT	100	Check to see if the value in the register is larger than 100
	JMP	SetToZero	If so jump to the label SetToZero
	JMP	NextCheck	If not jump to NextCheck
SetToZero	LD	0	Load 0 into the register
	ST	Counter_IL	Store the register in the variable Counter_IL
NextCheck	LD	Counter_IL	Load the variable Counter_IL to the register
	GT	50	Check to see if the value in the register is larger than 50
	JMP	SetStatus	If so jump to the label SetStatus
	LD	1	If not load 1 into the register
	ST	Status_IL	Store the register in the variable Status_IL
	JMP	Continue	Jump to the label Continue
SetStatus	LD	0	Load 0 into the register
	ST	Status_IL	Store the register in the variable Status_IL
Continue			The label Continue

最后把 Status_IL 连接到一个真实 D0。

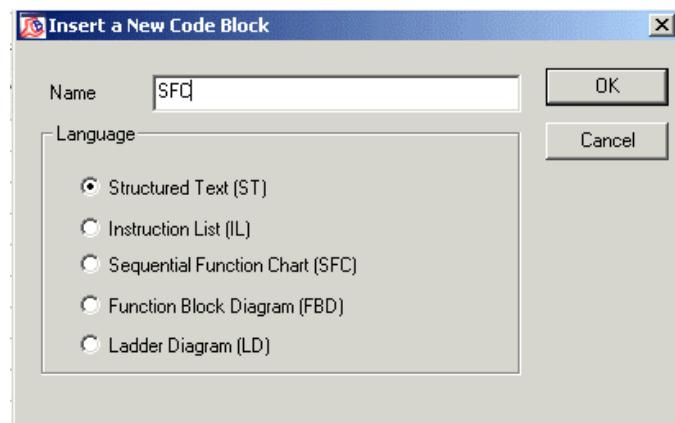
2.9 Sequential Function Charts 顺序功能图

2.9.1 任务

让我们设计一个具有以下所有功能的控制器：

- 一个逻辑启动变量命名为 Start_SFC，原始值为 FALSE；
- 再把两个计数器 SFC_Slow_Counter 于 SFC_Fast_Counter 都复原成 0 并处于等待状态，直到 Start_SCF 变成 TRUE；
- 另一项基本功能是等它们都超越 20；
- SFC_Slow_Counter 计数器应每周期增加 1；
- SFC_Fast_Counter 计数器应每周期增加 2；
- 跳回第一步当双项计数器都超于 100。

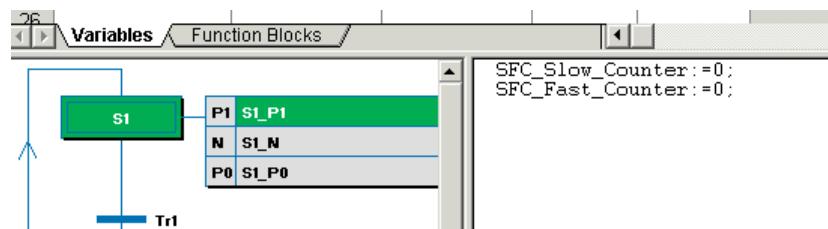
2.9.2 解决方案



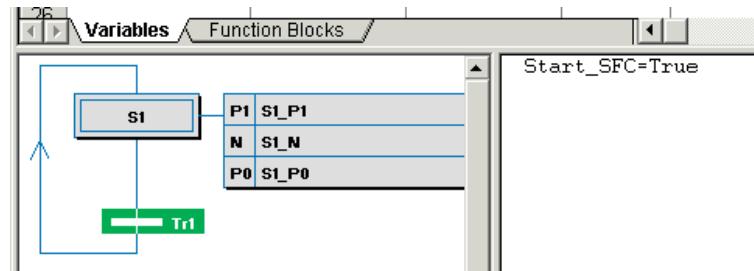
声明 Start_SFC、SFC_Slow_Counter 于 SFC_Fast_Counter 三个变量。

1	Status_IL	bool	retain	
3	Start_SFC	bool	retain	false
4	SFC_Slow_Counter	dint	retain	
5	SFC_Fast_Counter	dint	retain	

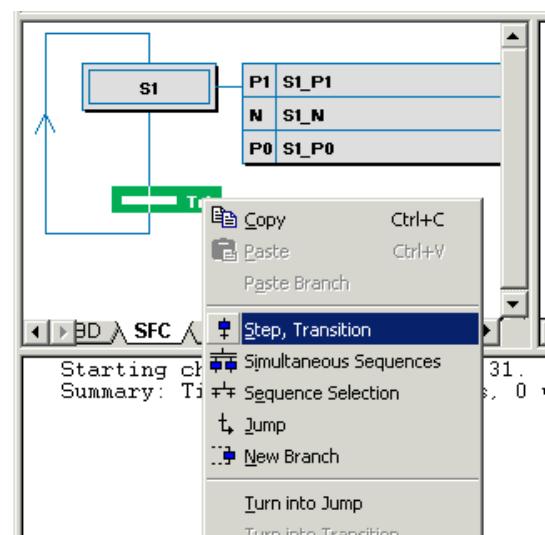
复原双项计数器的 S1_P1。



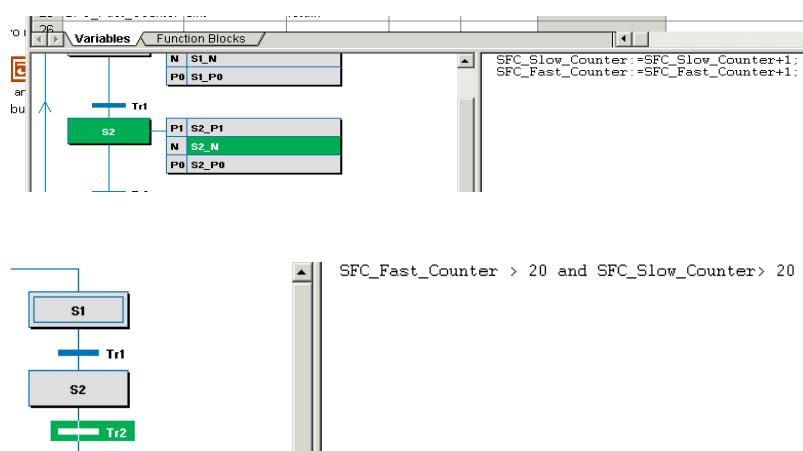
等待 Start_SFC=TRUE。



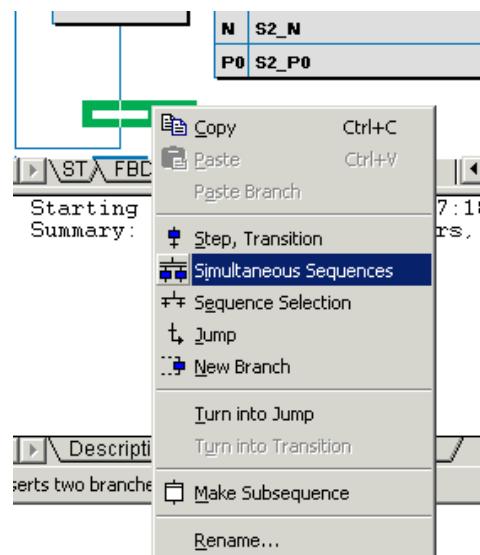
插入一个新步骤 (Step, Transition)。



双项计数器同时增加 1，直到 20。

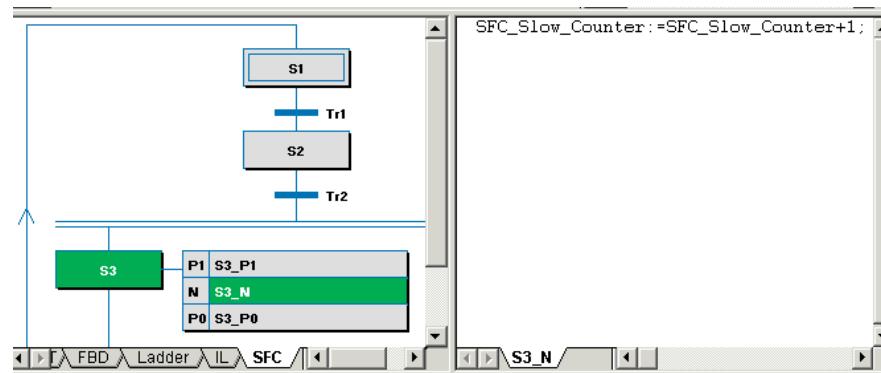


添加一个同时顺序 (Simultaneously Sequence)。

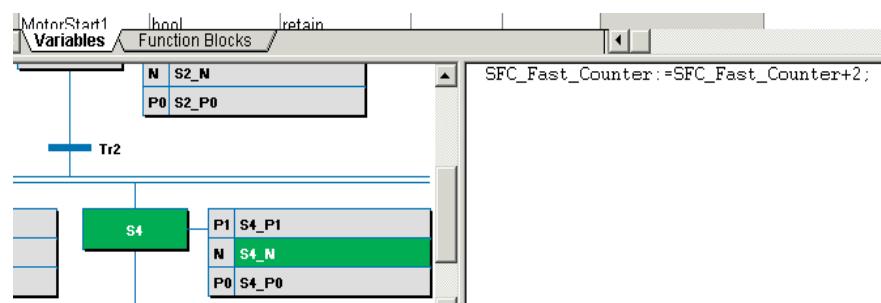


在 S3 中增加 1 到 SFC_Slow_Counter，在 S4 中增加 2 到 SFC_Fast_Counter。

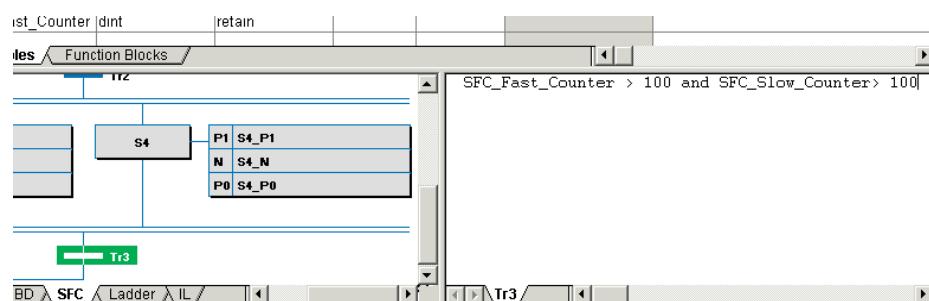
S3 步骤:



S4 步骤:



最后的步骤是当双项计数器都超于 100 后，将走回第一步。



2.10 Function Block Types 功能块类型

2.10.1 任务

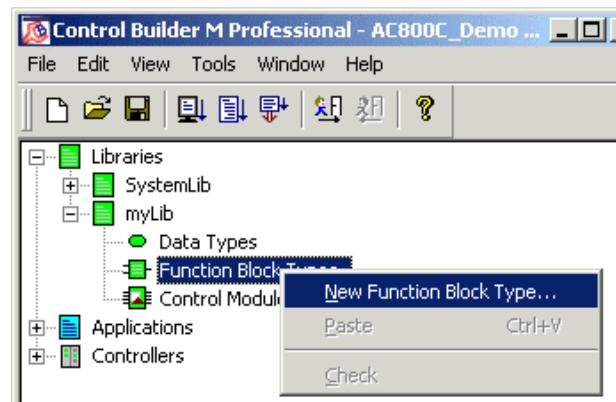
创建我们自己的功能块类型 Counter_FB，它有两个输入参数 Step_Value 于 Reset 和一个输出参数 Status。另外还有一个变量 Counter。Counter 每周期增加 Step_Value 直到 Counter < 100。如果 Counter > 50，Status 参数应改为 TRUE，其他时间都显示 FALSE。

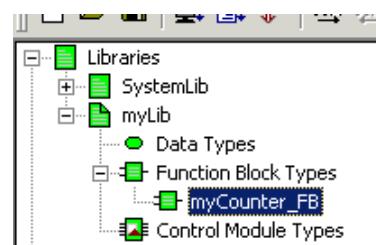
2.10.2 解决方案

创建您自己的数据库 Library 命名为 myLib。



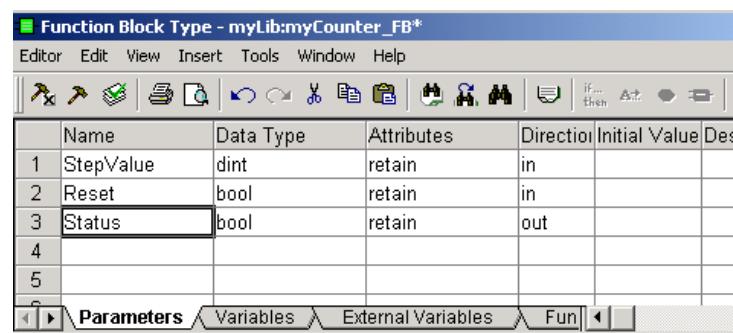
创建您自己的功能块类型命名为 myCounter_FB。



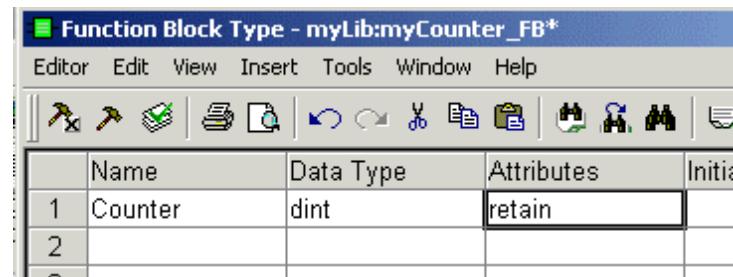


创建后再右击此功能块并选择编程器 Editor。

声明相关的参数。

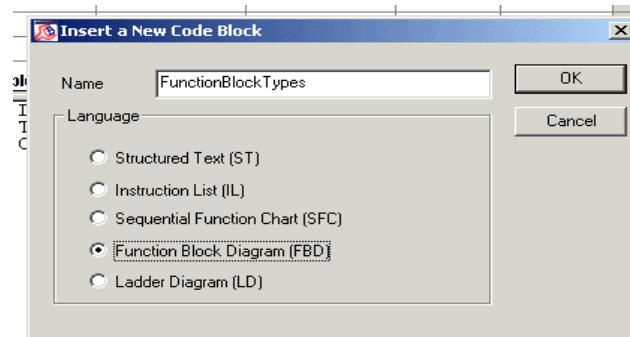


再声明其它变量。

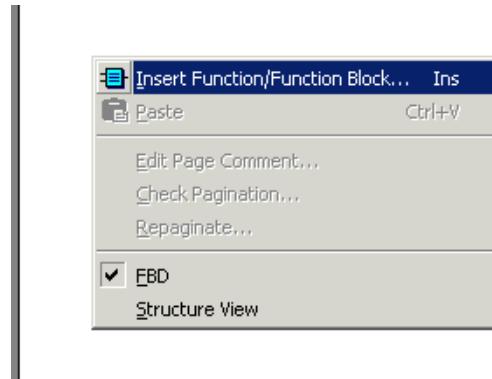


您已经可以运用功能块类型了。不过我们还没这个功能块里编写任何功能。功能块类型是在数据库树 Library Tree 上定义的，不过是在应用树 Application Tree 上运用的。

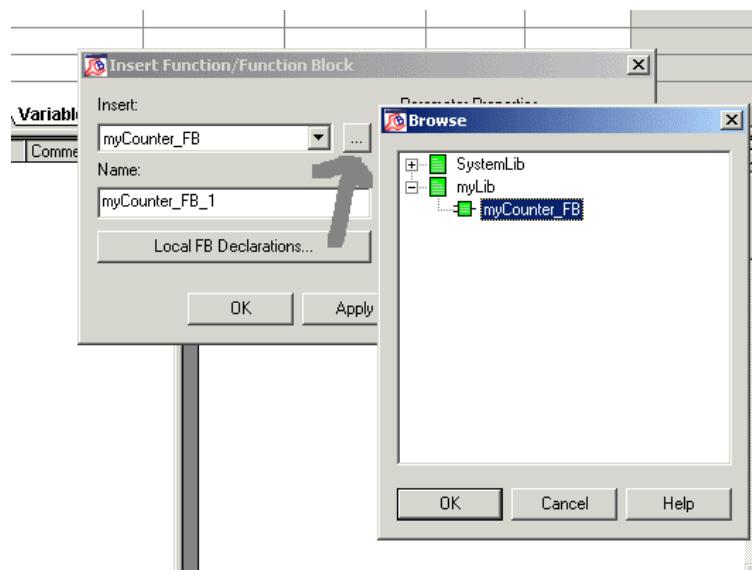
现在我们一起来运用此项新建的功能块。再次在 Application_1\Program1 里创建新的 Tab 命名为 FunctionBlockTypes。



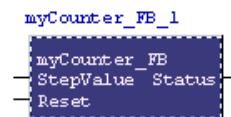
右击程序窗口并选择插入功能块 (Insert Function Block)。



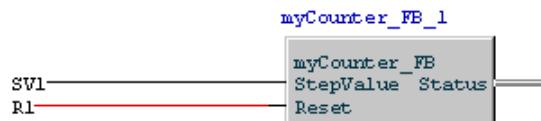
点击浏览并选择 myLib、myCounter_FBF。



结果:



声明需要连接到功能块上的变量: SV1, R1, S1。设置 SV1 原始值到 5。



现在让我们来添加一些功能吧。回到功能块类型 myCounter_FB 并点击打开程序窗口。

```

Counter:=Counter+StepValue;

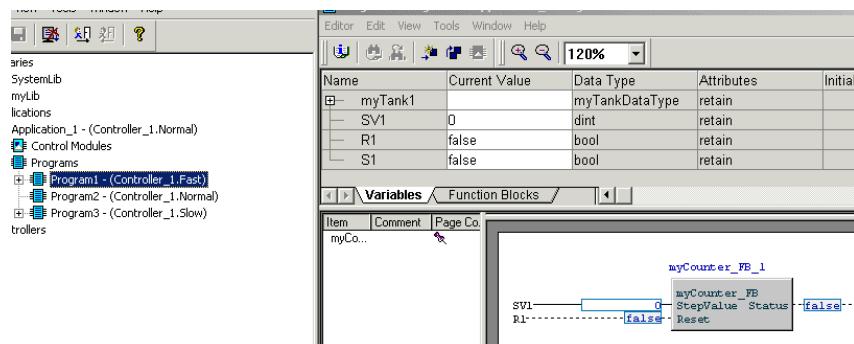
if Reset=True then
  Counter:=0;
end_if;

if Counter > 100 then
  Counter:=0;
end_if;

If Counter> 50 then
  Status:=True;
else
  Status:=False;
end_if;

```

编写完后，执行模拟程序。可以到 Program1 运行程序里看看控制系统是否正常工作。

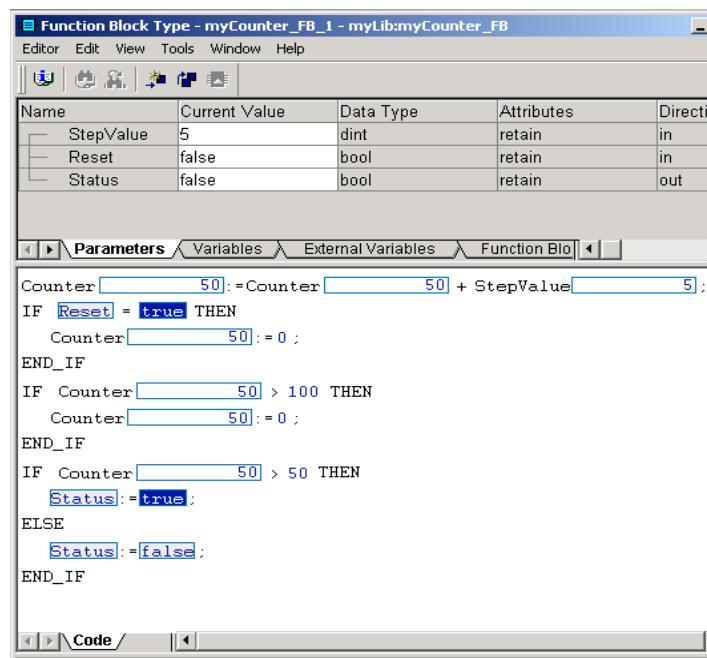


检查 S1 是否在 TRUE 于 FALSE 两值来回跳动。如否，可能是应为您的 StepValue SV1 是设置为 0，试试打入 5。

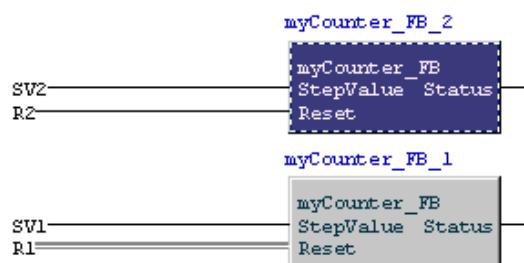
您随时都可以到任何功能块的 On-Line Editor 里查看运行状态。



这将是您现在在屏幕上看到的程序(如果它没被作者所住的话)。

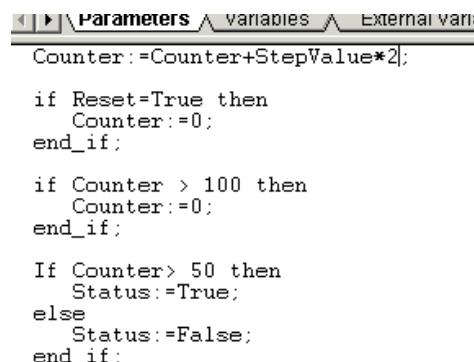


这是一个很有效的解决方案。让我们复制并把所有变量的 1 更换成 2。



可见重复运用此类的解决方案有多么的容易。

如果我们不想再用 StepValue 参数来增加计数器的值了, 我们想用 StepValue*2 值来替代。这将需要更换功能块的类型。当您更换类型, 所有运行状态将自动更新。返回类型。



```

Parameters \ variables \ External var:
Counter:=Counter+StepValue*2;

if Reset=True then
  Counter:=0;
end_if;

if Counter > 100 then
  Counter:=0;
end_if;

If Counter> 50 then
  Status:=True;
else
  Status:=False;
end_if;
  
```

模拟并查实功能块的正常运行。

2.11 PID Loop 设置

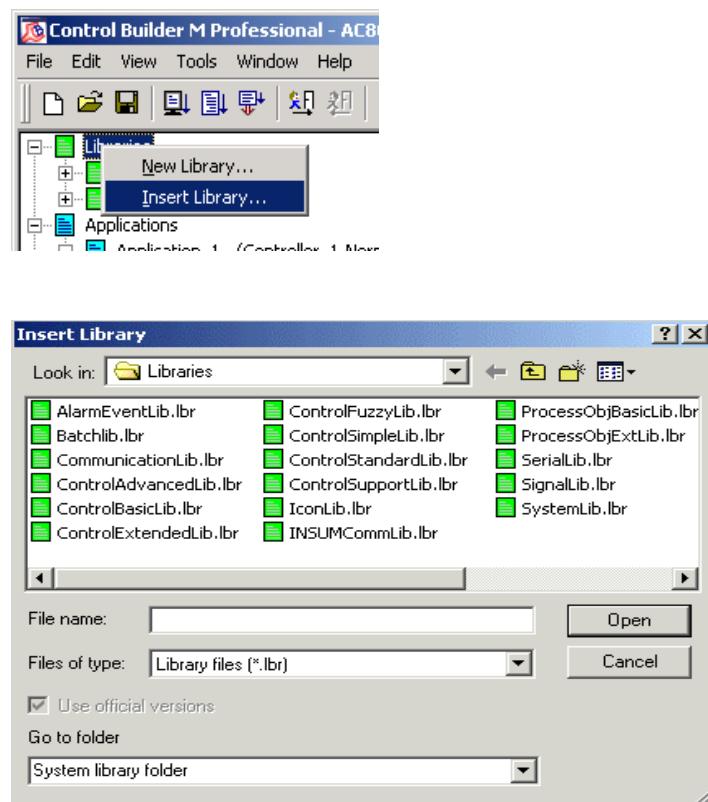
2.11.1 任务

设置一个 PID Loop。

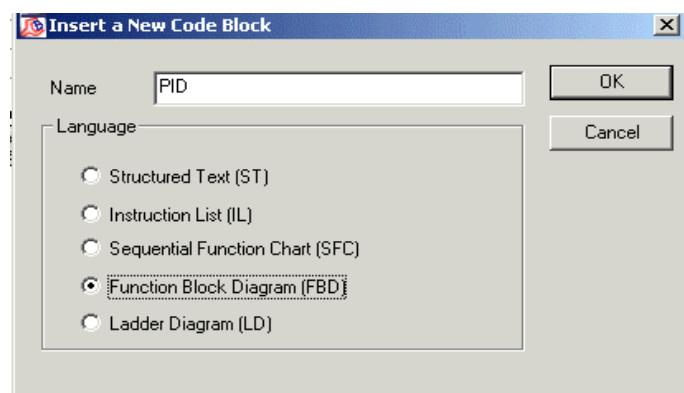
2.11.2 解决方案

2.11.2.1 功能块

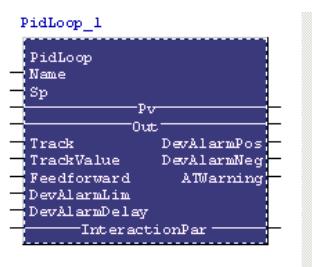
插入 ControlBasicLib 到项目树。右击 CBM 里的 Library 并选 Insert Library。



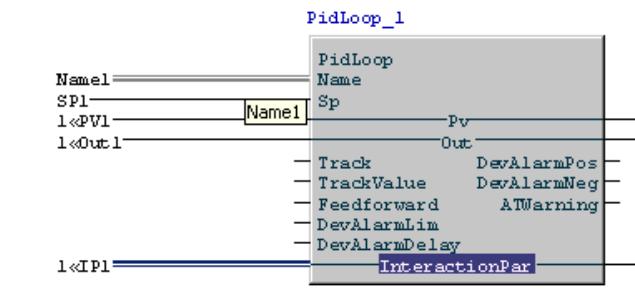
再次创建一个 TAB, 以功能块的方式插入此项 PID。



插入一个 PID loop。



连接以下参数: Name, Sp, Pv, Out 和 InteractionPar。把 Name1 参数改名为 ‘PID1’ 。



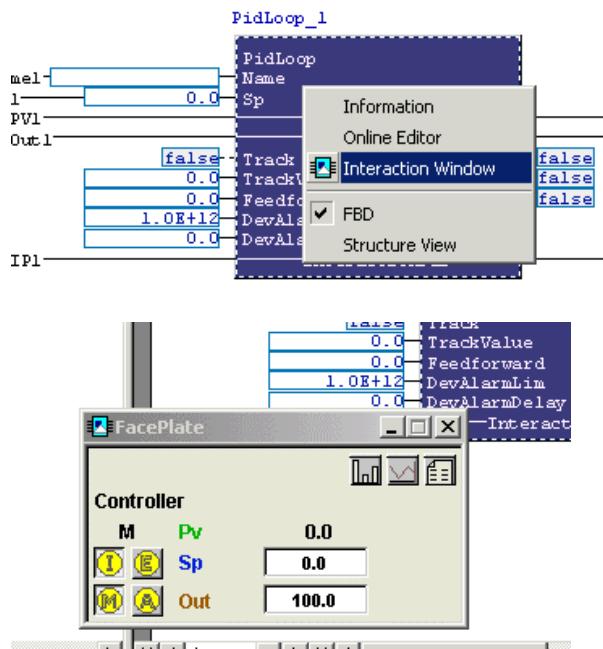
我们没有真实的过程。如果需要得到一些 PV 于 Out 两个参数的关系, 可以完成以下步骤。



查实是否有错误并进入模拟状态 (Simulation Mode)。参考以下变量窗口来命名您的变量。

Name	Current Value	Data Ty
PV1		RealIO
Out1		RealIO
IP1		PidLoop
Gain	1.0	real
Ti	20.0	real
Td	0.0	real
PvFilterTime	0.0	real
DerFilterTime	0.0	real
Offset	50.0	real
DeadZone	0.0	real
FfGain	0.0	real
Direct	false	bool
ControllerTime	2	dint
LimitOutput	false	bool
OutMaxLimit	100.0	real
OutMinLimit	0.0	real
OutIncLimit	1.0	real
OutDecLimit	1.0	real
SpManValue	10.0	real
ExternalSp	false	bool
OutMaxValue	100.0	real

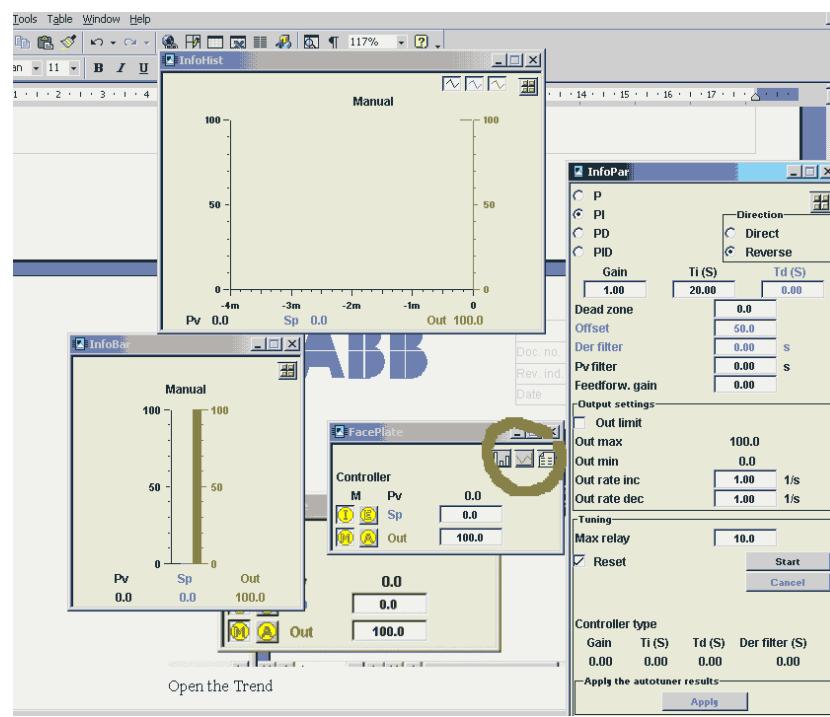
或您可以右击 PID Block 进行 Interaction Window 预制。



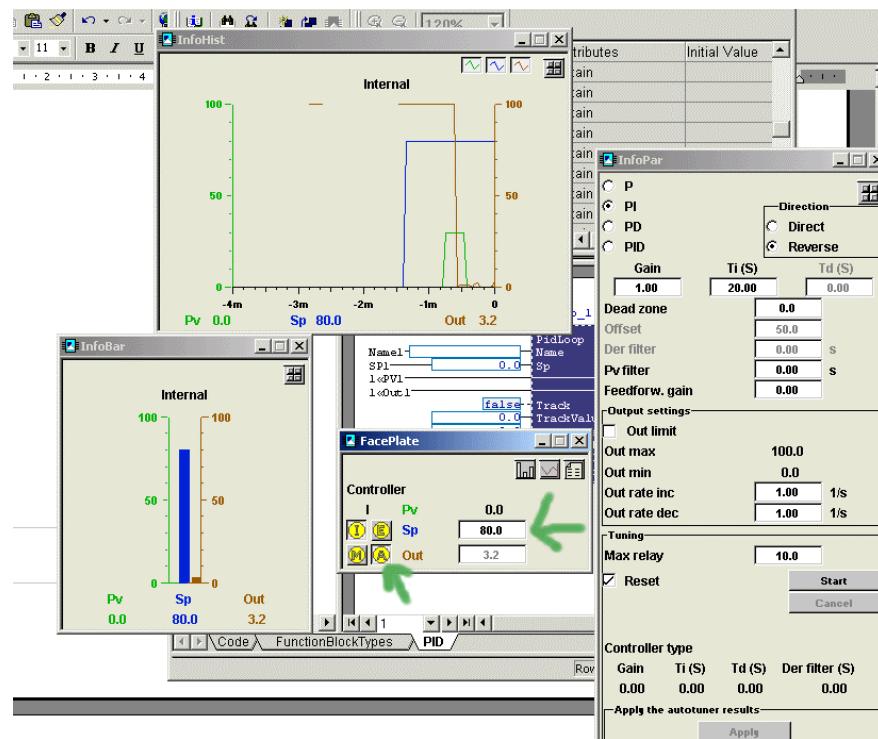
在主面板的右上角，点击图标打开预制的面板。



Doc. no.
Rev. ind. 3
Date 040412



设定点为 80 并选择自动模式 AutoMode。您可以看到 Output 参数应会增加 PV 值，不过这里不是这样。那是应为我们没有一个闭环。



您可以在 InfoPar 窗口里调制和更换 PID Loop 的参数。

当然您也可以连接一些真实 I/O 到 PV 于 Out 参数上，并尝试调制。

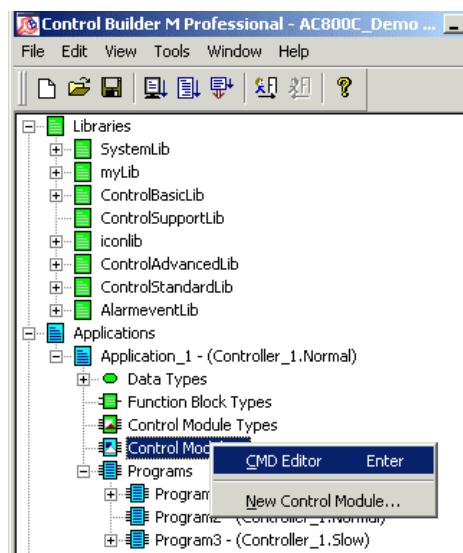
调试过程的时候，把乘数 Gain 改为 1.0 以下。

2.11.2.2 控制模块

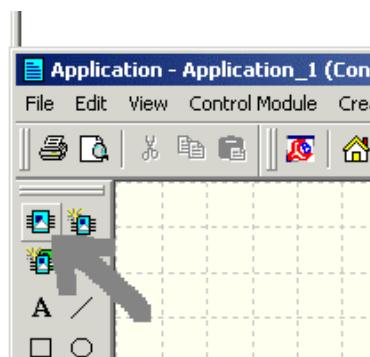
我们想用控制模块概念 (Control Module Concept) 来完成以上任务。

再次导入 ControlAdvancedLib 数据库。

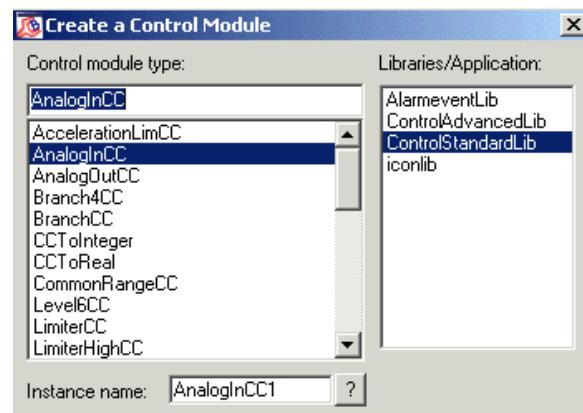
控制模块应创建在应用树上。右击打开控制模块编程器 (CMD Editor)。



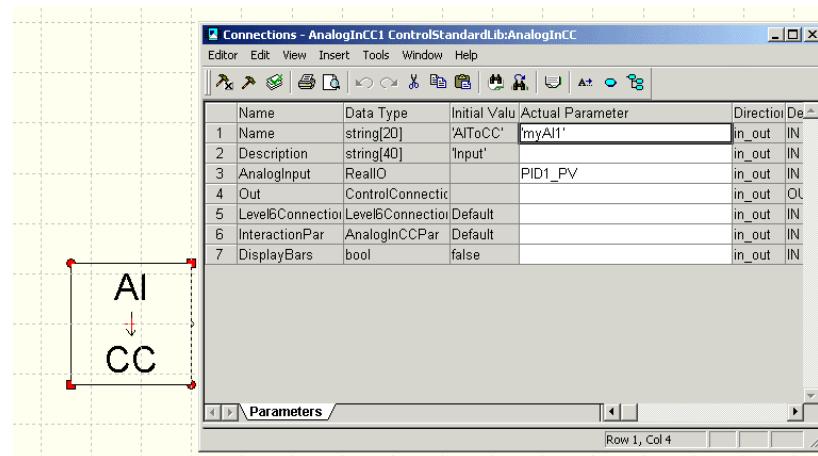
插入一个新的控制模块。



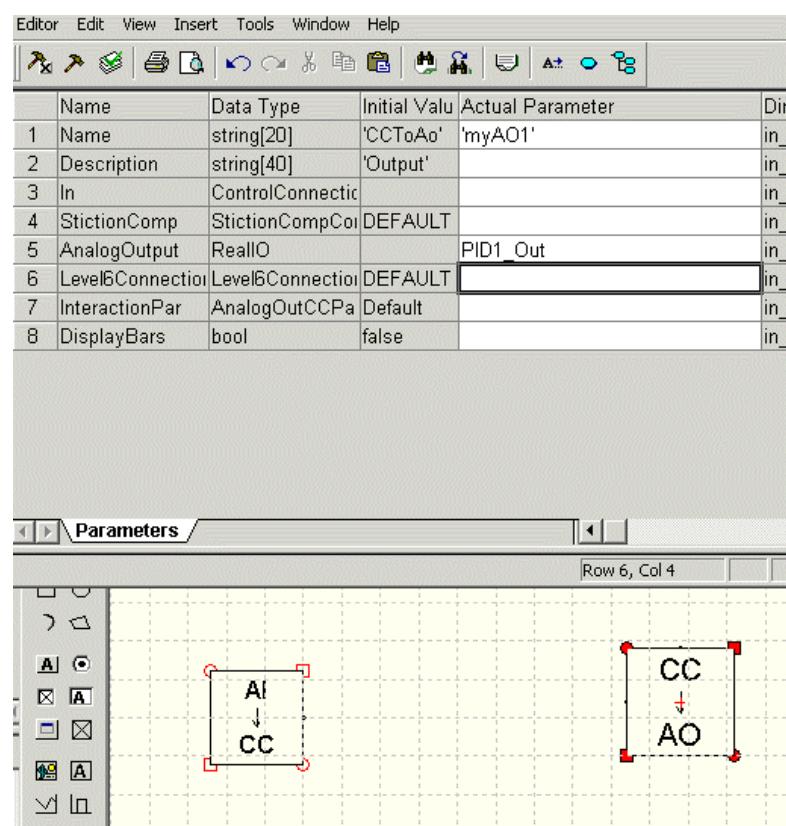
控制模块里的 PIDLoop 只接受从 CC Control Connection 模块输入的信息。这就意味着我们必须把 RealIO 的数据转换为 CC 的数据。再导入 AnalogInCC Control StandardLib。



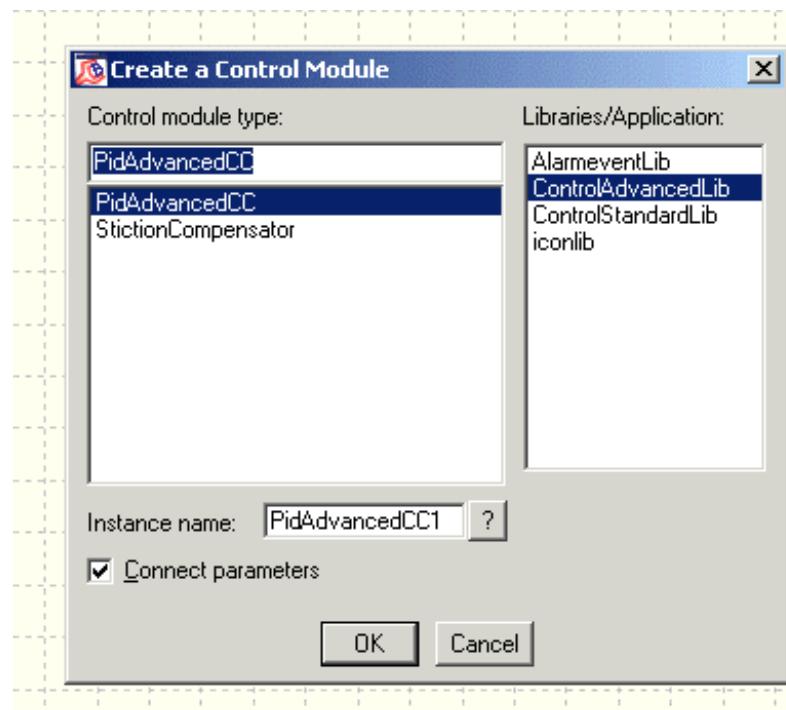
点击选择编辑器里的输导入点。鼠标左击按住并拉放表示 CC Connection 功能块的大小。您之后需要给 AnalogInCC 添加一个名字并从现场提供输入信号。我命名它为 PID1_MV。



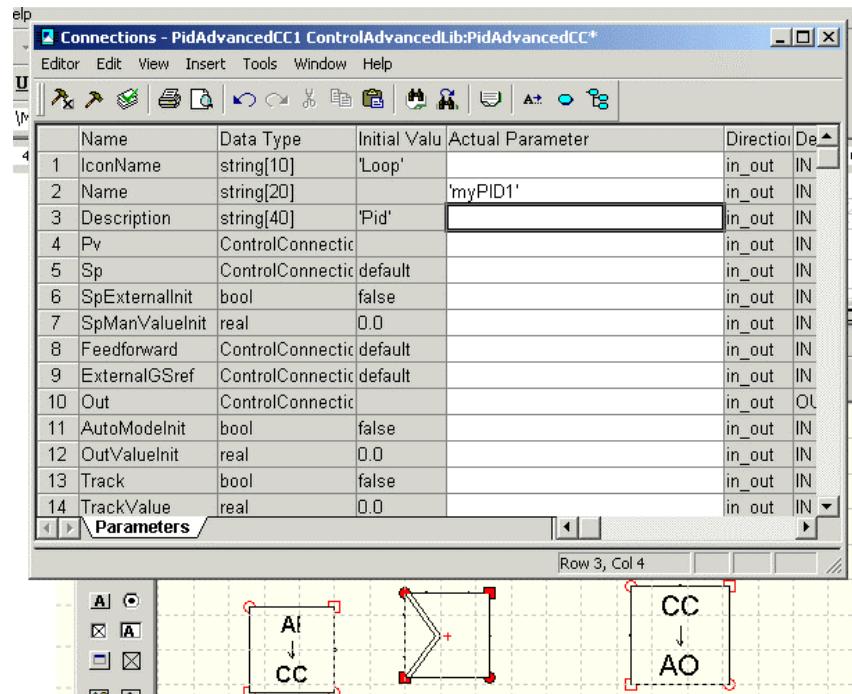
继续同样的任务完成 AnalogOutCC 的设置。



现在可以导入 PID Loop 了。

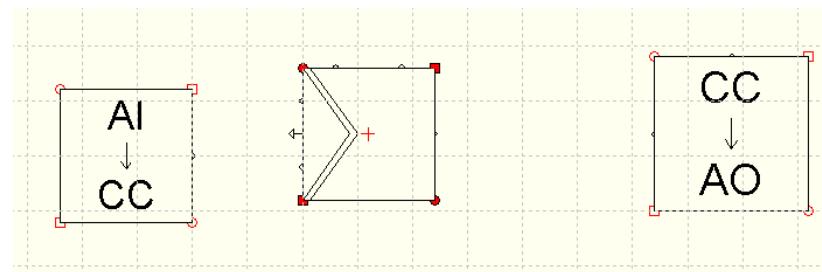


唯一需要做的就是命名您的 PIDLoop。

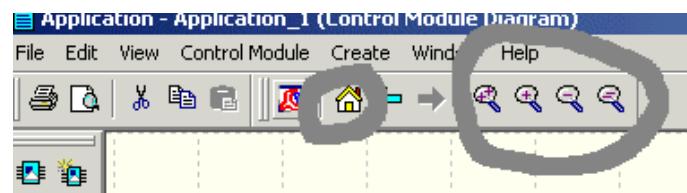


下一步是连接 PV 和 Out 两个参数到 CC 功能块上。

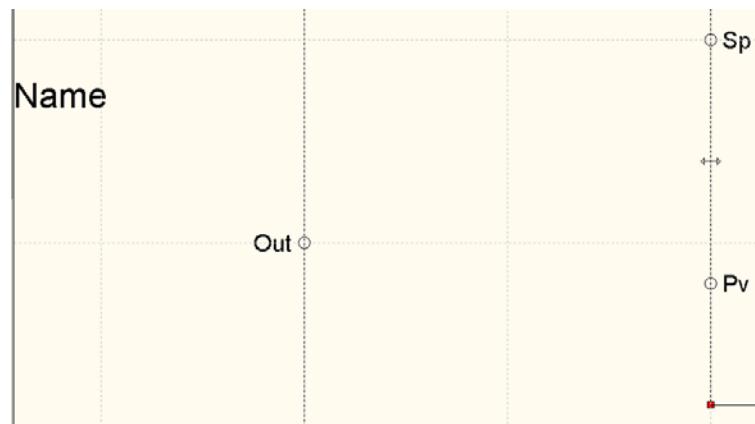
见以下案例：



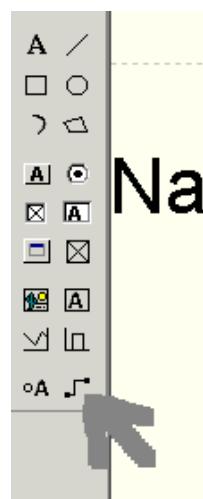
放大窗口里的模块，这样您就可以看到各模块的输入输出口的名称了。可以用工具栏上的快击键。



放大查看 AICC 和 PID Loop 的输入输出。AICC 功能块的左侧是 Output 口，右侧下面是 PV。

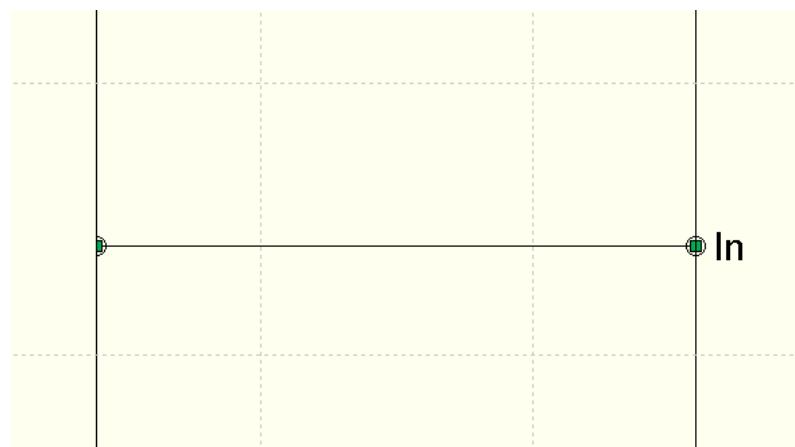


用图像连接 Graphical Connection 创建它们之间的连接。

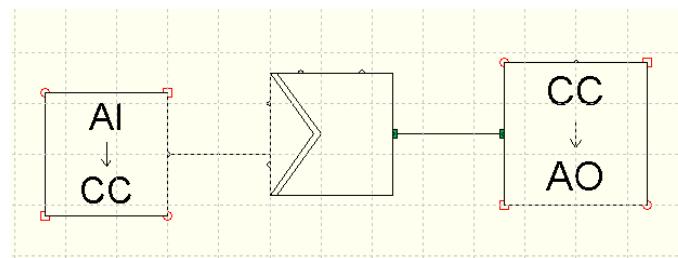


点击 Out 后再双击 PV 建立连接。

重复以上步骤来连接 AOCC 和 Out 变量。

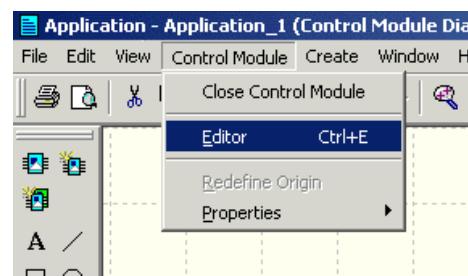


您最后的图像显示符合下图吗？



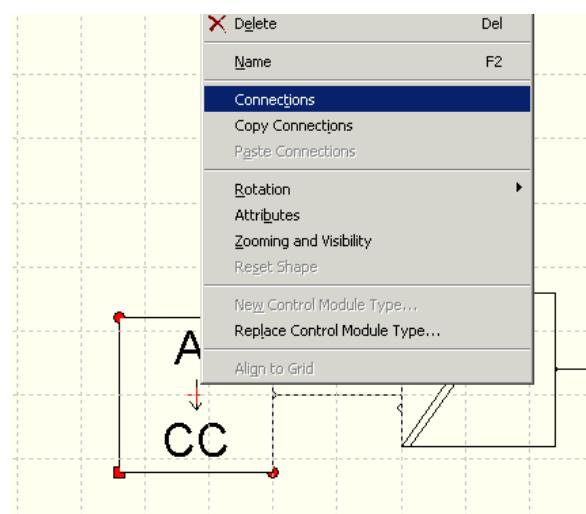
连接已创建。您还需要声明真实 I/O: PID_Pv1 和 PID_Out1。

在应用编辑器 Application Editor 里定义两个通用变量 Global Variables。



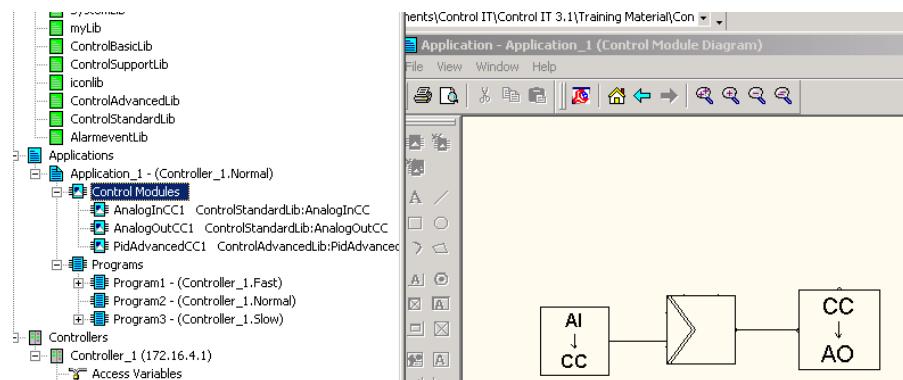
	Name	Data Type	Attributes	Init
1	PID1_PV	RealIO	retain	
2	PID1_Out	RealIO	retain	
3				
4				

如许查看控制模块的连接，右击功能块并选择其连接口。

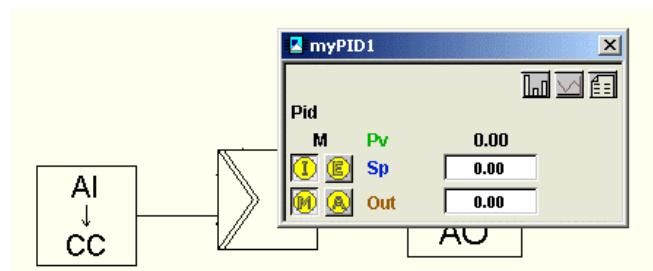


如有真实 I/O, 现在可以分配一些变量。

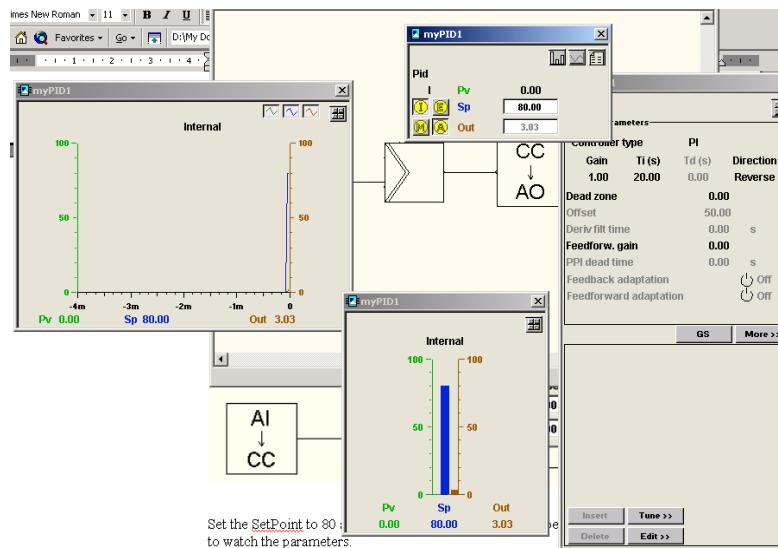
现在可以模拟您的解决方案了。双击控制模块打开 CMD Editor。



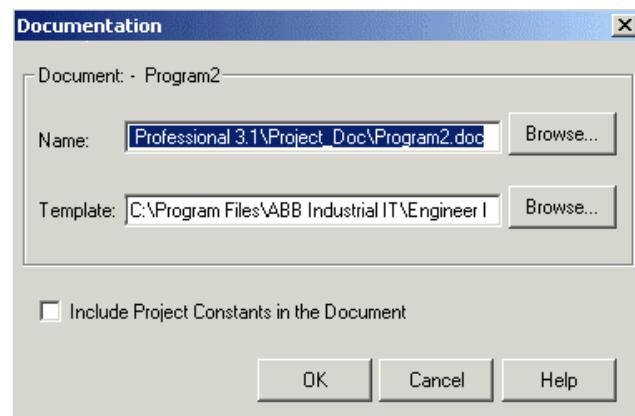
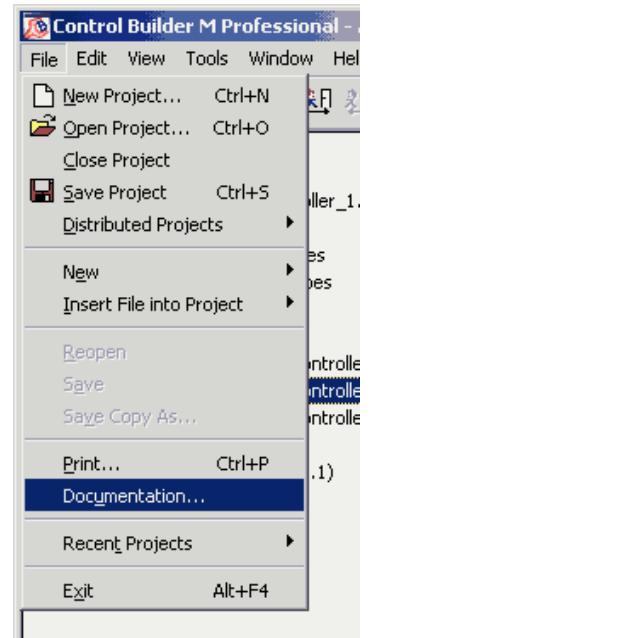
点击 PID Loop 打开 Interaction Window。



设置定点为 80 和 PID Loop 为 AutoMode。打开面板调试于查看参数。



在结束练习之前，记录 (Document) 以上的硬件配置于控制系统逻辑。



练习结束！