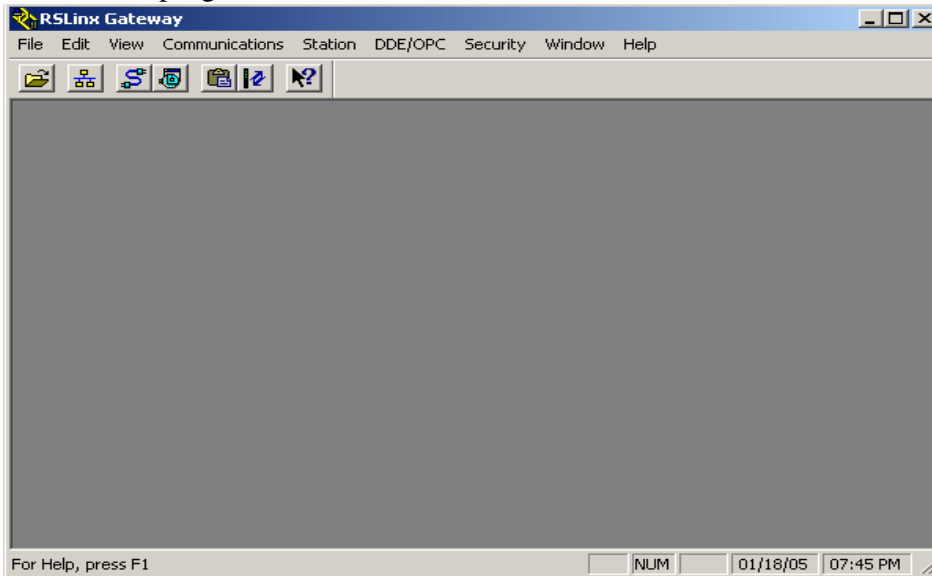


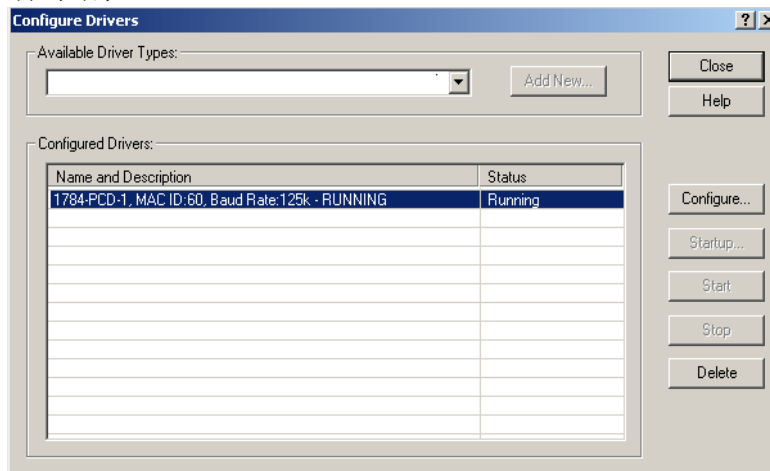
实验一 用 RSLinx 创建通讯路径

在开始 SLC500 的实验之前，创建通讯路径是必要的。通讯路径的创建对后面完成 IO 的自动配置以及程序的下载都是必须的。

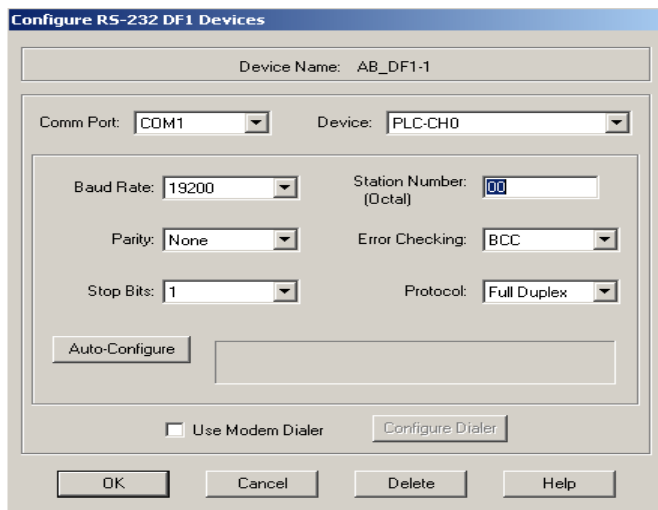
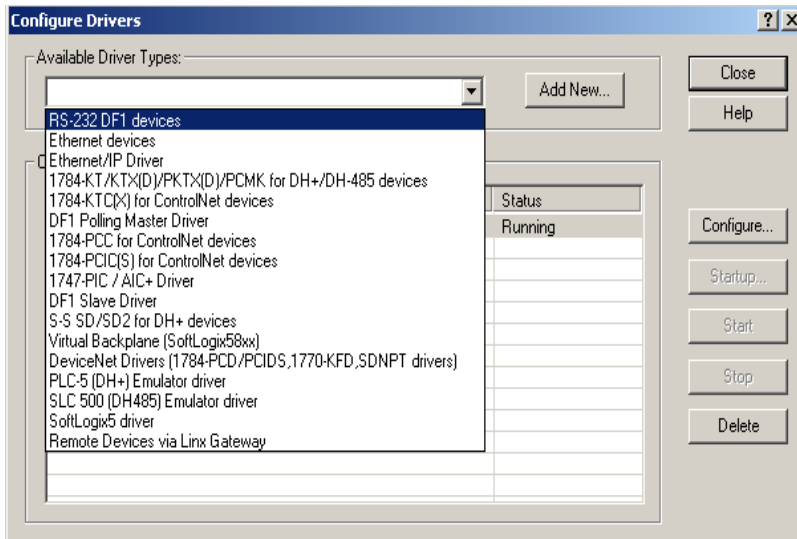
找到 Start->program->Rockwell software-> RSLinx，运行 RSLinx,出现画面如下：



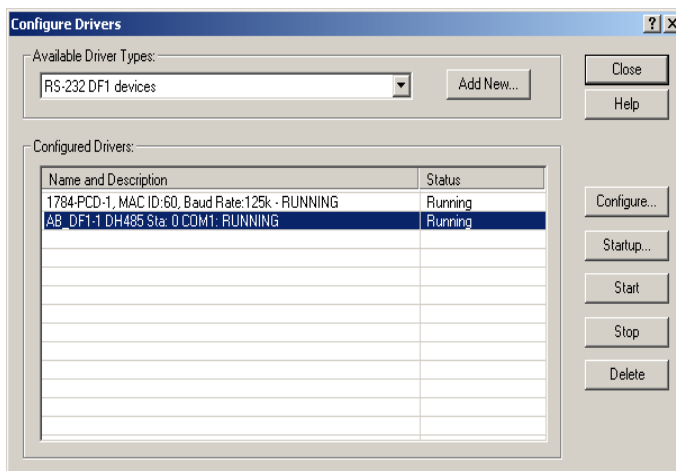
在主菜单中点击 Communications 后，在弹出的下拉菜单中点击 Configure Drivers,出现配置驱动对话框



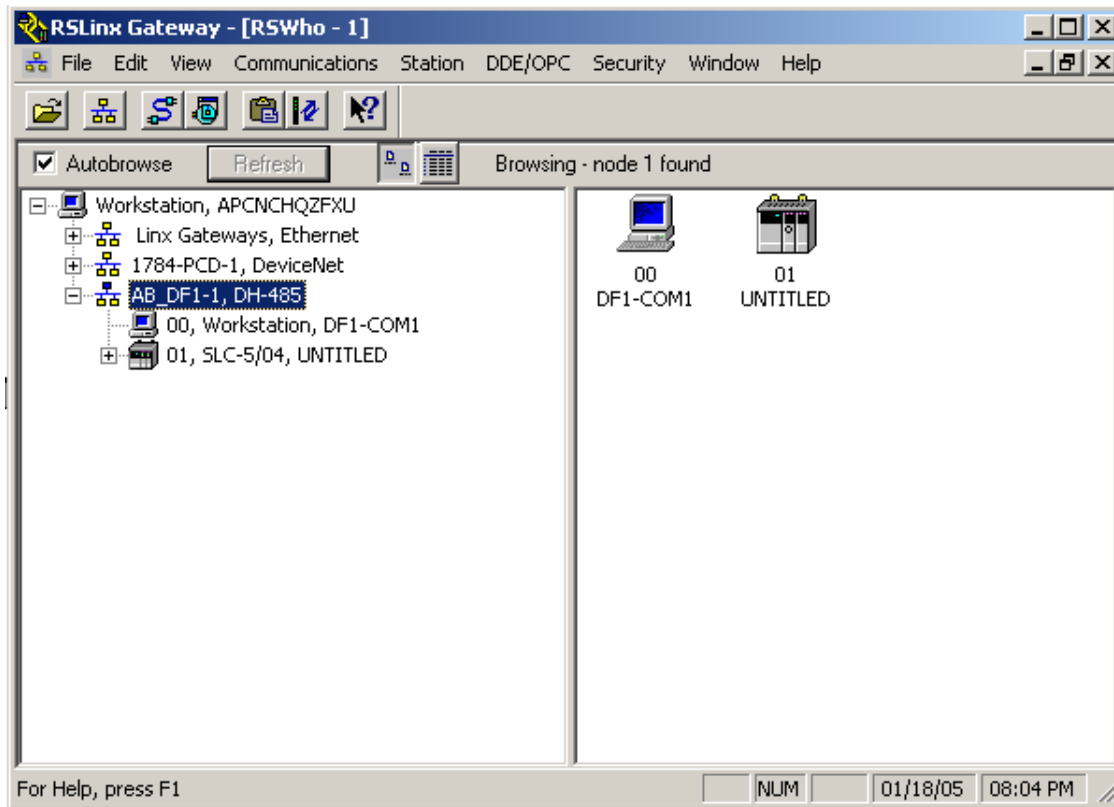
在 Available Drivers Types 处点住下拉按钮，在出现的驱动中选择 RS232 DF1 devices, 按下 Add New 按钮，在弹出的对话框上点击 OK 保留默认的名称后出现如下画面



选择正确的 PC 通讯口，点击 Auto_Configure，将会自动完成通讯配置。




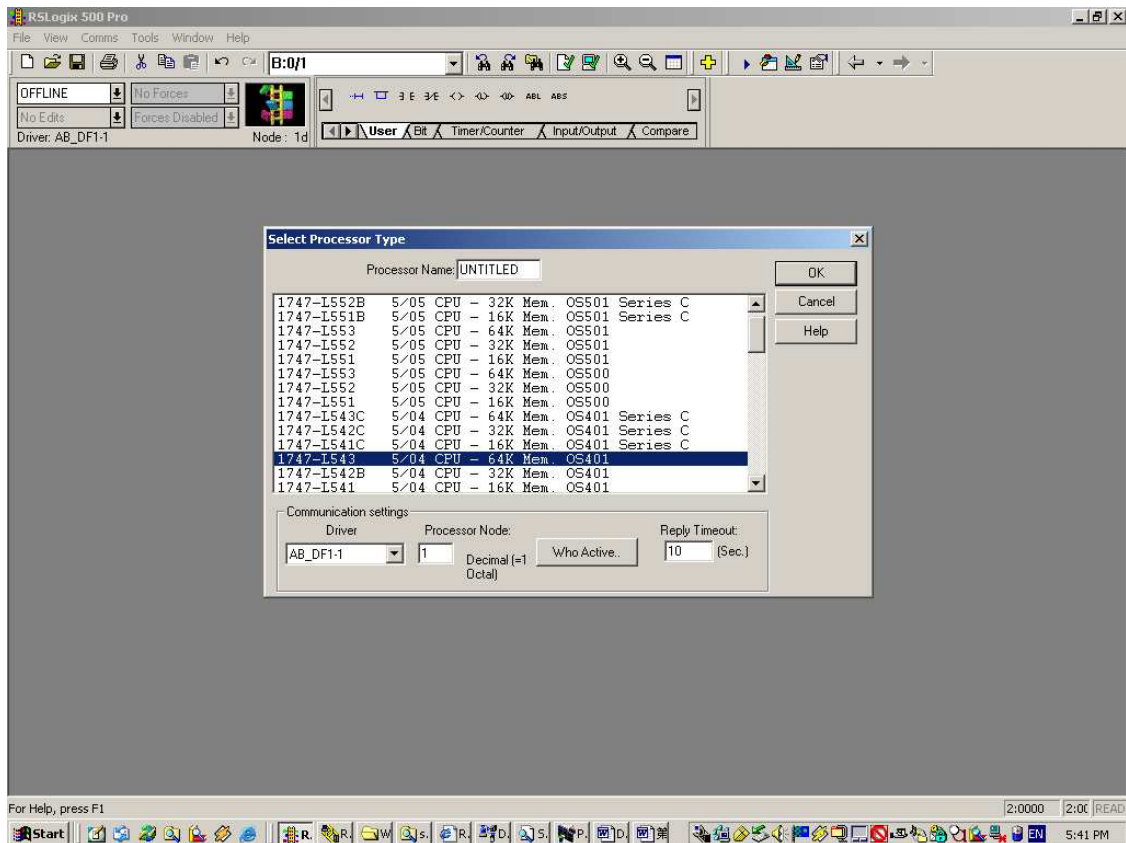
好了，我们已经创建好啦 DF1 通讯，点击图标 ，在线浏览设备。



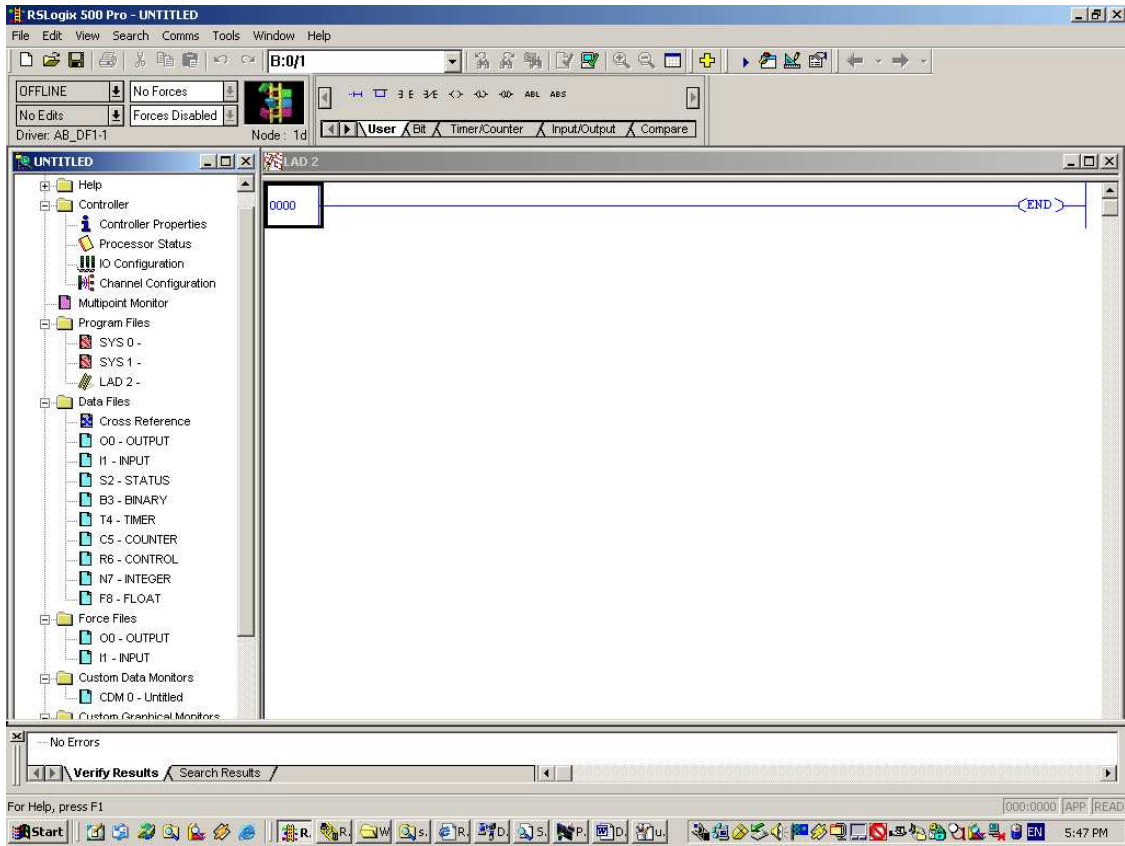
到此，实验一就大功告成。

实验二 应用 RSLogix500 软件创建新的工程应用

1、运行 RSLogix500， Start->program->Rockwell software-> RSLogix500. 点击  出现如下画面

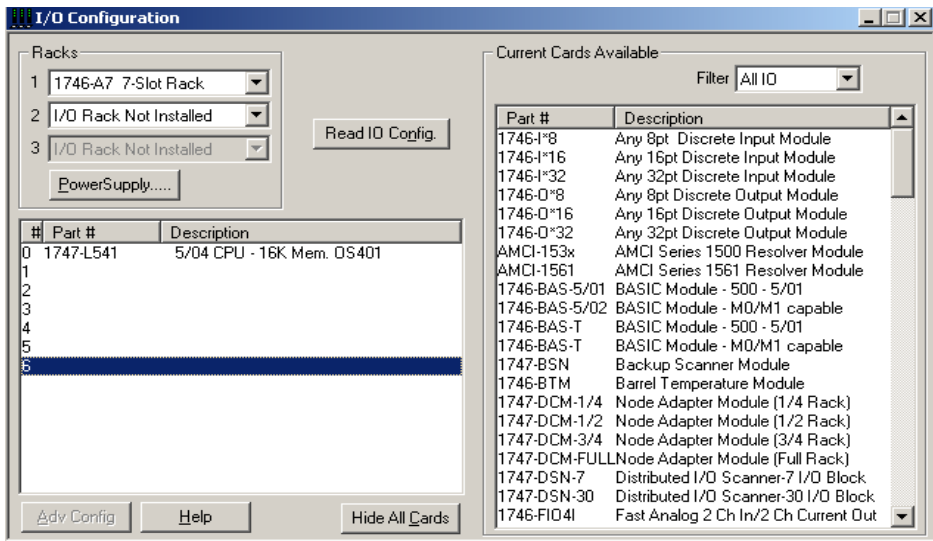


选择处理器类型（本次实验处理器为 1747-L541， 点击 OK 按钮， 进入工程画面。

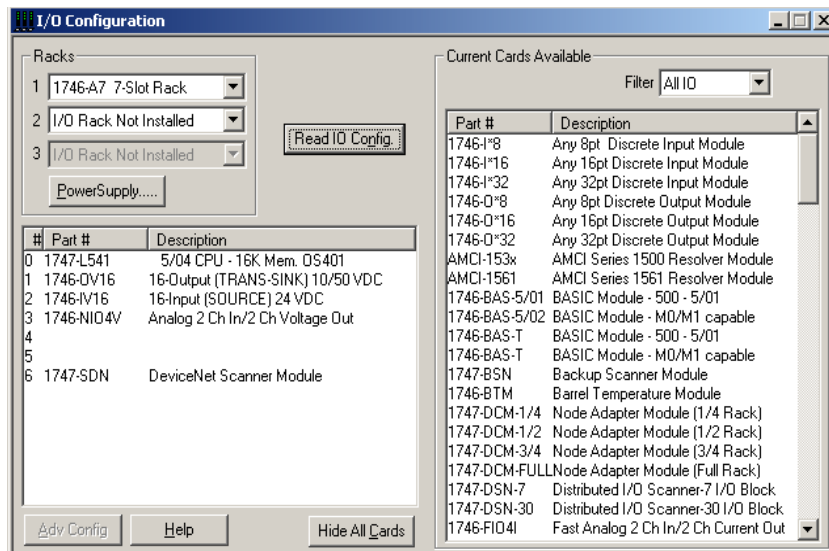
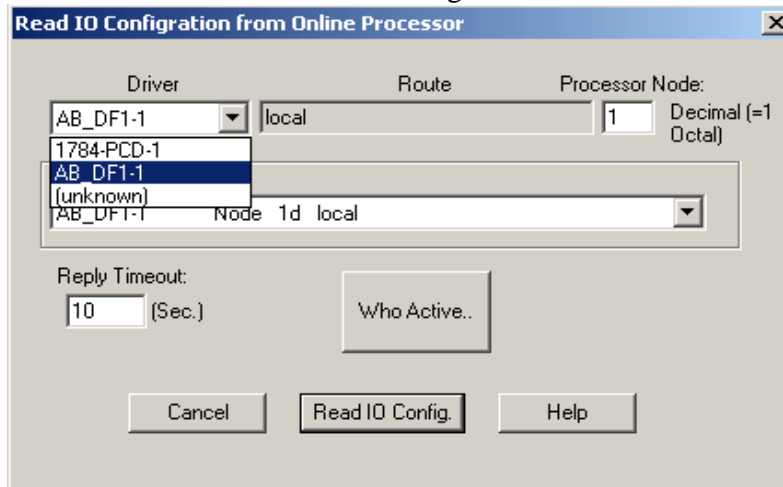


2、I/O 配置

SLC500 支持 I/O 的自动配置，在工程树下，找到 Controller 文件夹，双击 IO Configuration, 出现画面



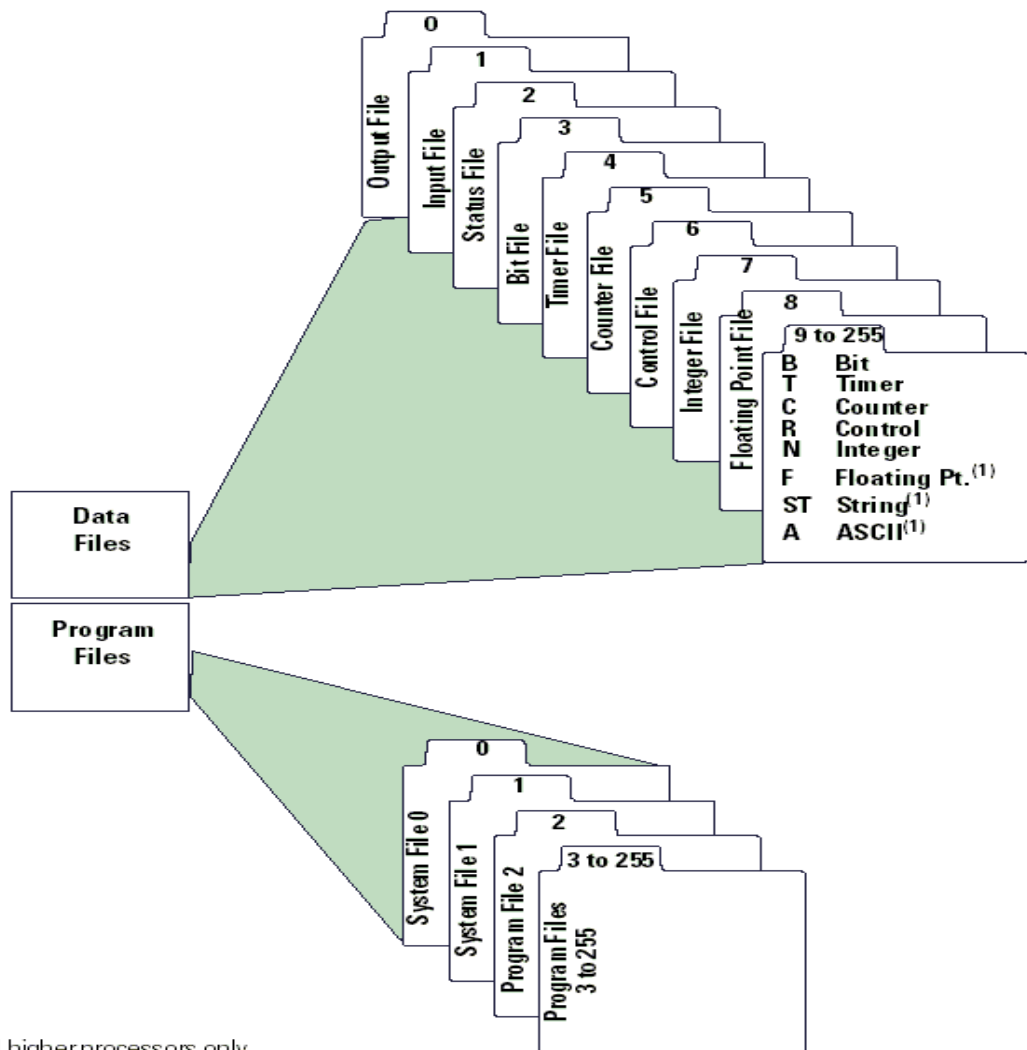
此时，点击 Read IO Config 按钮，出现一个选择通讯路径的画面，选择实验一创建的 DF1 驱动，然后按下 Read IO Config 按钮，将会完成 IO 的自动配置。



3、了解 SLC500 的内存、数据文件及其寻址表达方式

内存

- 1) 程序文件
- 2) 数据文件



¹ higher processors only.

程序文件 program files

- 1) 系统文件:
- 2) 主控程序: 只能有一个 LAD2
- 3) 通用子程序: 被主程序或其它子程序调用, LAD3----255

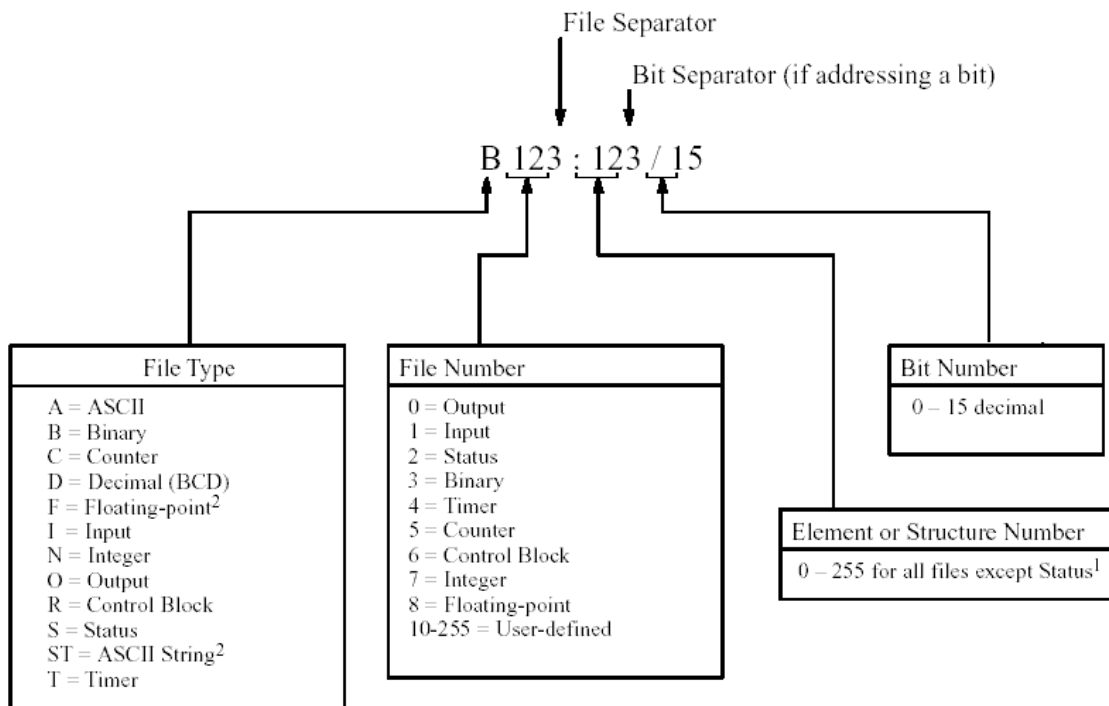
SLC500 只支持 LADDER

数据文件 Data files

File Number	File Type	Logical Address
0	OUTPUT IMAGE	\$O:0 to \$O:30 输出
1	INPUT IMAGE	\$I:0 to \$I:30 输入
2	STATUS	\$S:0 to \$S:n 状态
3	BINARY	\$B3:0 to \$B3:255 BIT 位
4	TIMER	\$T4:0 to \$T4:255 定时器
5	COUNTER	\$C5:0 to \$C5:255 计数器
6	CONTROL	\$R6:0 to \$R6:255 控制
7	INTEGER	\$N7:0 to \$N7:255 整数
8	FLOATING-POINT	\$F8:0 to \$F8:255 浮点数
9	NETWORK	\$x9:0 to \$x9:255 DH485 通讯
10 thru 255	USER DEFINED	\$x10:0 to \$x255:255 自定义

- O0、I1、S2、B3、T4、C5、R6、N7、F8
- 文件 0-8 为系统定义，不能更改、删除
- 文件 F8 仅 SLC5/03 以上
- 当非 SLC500 的 DH-485 设备存在时，文件 9 用作网络通讯
- 文件 10-255 可以自由定义为 T/C/B/N/F/R/ST/A 文件类型

数据文件地址表达方式



地址包括：文件类型、文件号、：元素号或数据结构/位号

- 1) **O0 O:e.s/b** e:槽号 s:字号 b:位号
I1 I:e.s/b

例如 O:3/15 DO, 3 槽的 bit15
 O:5.1 DO, 3 槽的 word1
 I:7/8 DI, 7 槽的 bit8
 I:7 DI, 7 槽的 word0

- 2) **S2**

S:1/15 Element 1, bit 15. 上电“first pass” bit
S:6 故障代码

- 3) **B3**

B3:3/14 Bit 14, element 3
B3:252/00 Bit 0, element 252
B3/62 Bit 62 = B3:3/14

- 4) **T4 定时精度1ms/10ms/1s**

T4:0/15 or T4:0/EN 使能位
 T4:0/14 or T4:0/TT 正在计时位
 T4:0/13 or T4:0/DN 完成位
 T4:0.1 or T4:0.PRE 预置值
 T4:0.2 or T4:0.ACC 累积

5) C5,计数范围 (-32,768, +32,767)

C5:0/15 or C5:0/CU 加计数使能位
 C5:0/14 or C5:0/CD 减计数使能位
 C5:0/13 or C5:0/DN 完成位
 C5:0/12 or C5:0/OV 溢出位
 C5:0.1 or C5:0.PRE 预置值
 C5:0.2 or C5:0.ACC 累积

6) R6,控制寄存器 PID等指令使用

7) N7,整数 (16bit) 可寻址到字或位

N7:2
 N7:2/8
 N25:22

8) F8 浮点数, 2word

**** 寻址方式:

直接
 变址
 间接

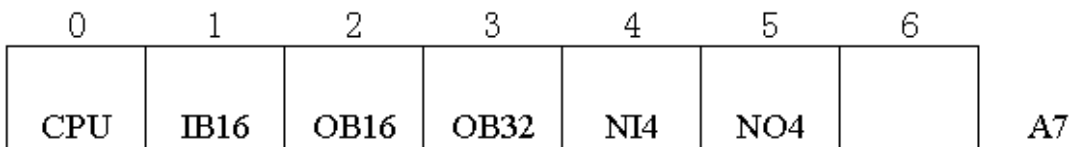
Addressing Mode	Example
Direct	N7:0
Indexed Direct	#N7:0
Indirect	N7:[N10:3]
Indexed Indirect	#N7:[N10:3]

I/O模块的寻址

O0 O:e.s/b e:槽号 s:字号 b:位号
I1 I:l.e.s/b

1: 本地机架

1#

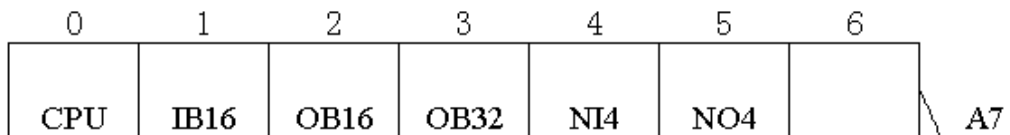


eg:

IB16	I:1.0/0—15	
OB16	O:2.0/0—15	
OB32	O:3.0/0—15	O:3.1/0—15
NI4	I:4.0—3	
NO4	O:5.0—3	

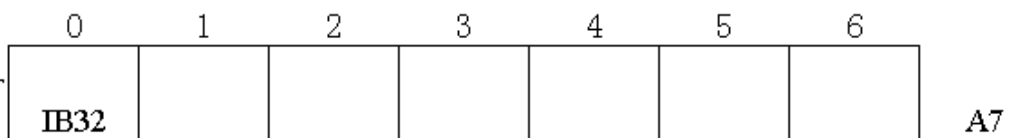
2: 本地I/O扩展机架

1#



1747-C16

2#



7 8 9 10 11 12 13

1#机架同上

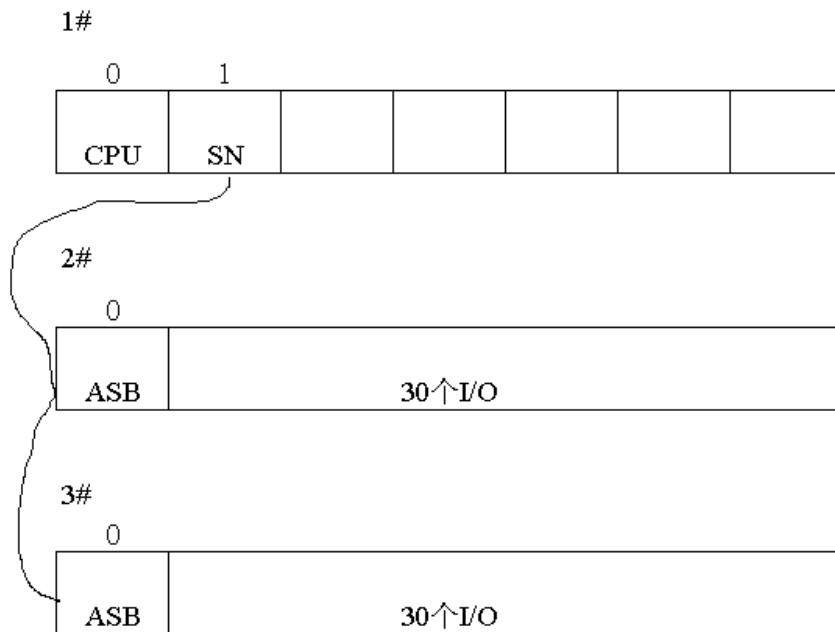
2#机架的槽号从1#的 6开始，其它类推

IB32 **I:7.0/0—15** **I:7.1/0—15**

**：最多可以扩展 3 个 local

**：槽号数最大为 30

3: 远程I/O

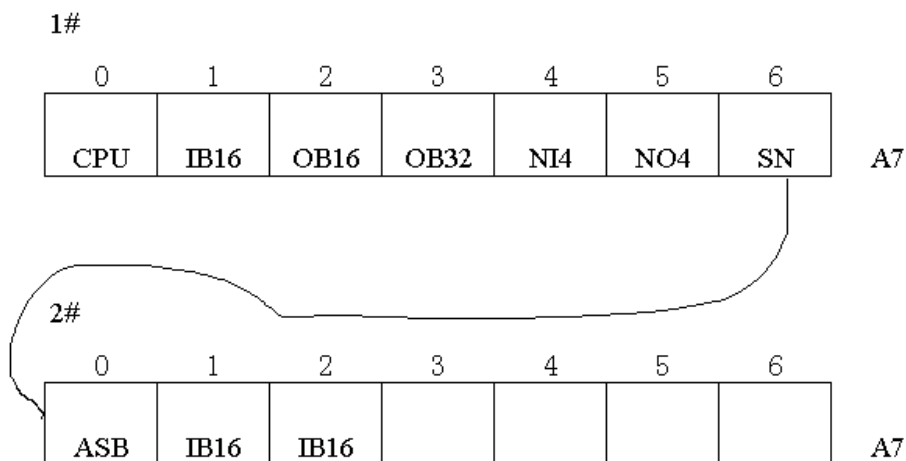


RIO的扩展

** 每个扫描器SN支持32个RIO站，每个RIO站最大30个I/O

** 每个扫描器SN在处理器有32字的输入输出I/O映像区

RIO的寻址



2# IB16 I:e.X/0—15; e代表SN在本地机架的槽位。

下面确定 X 的写法

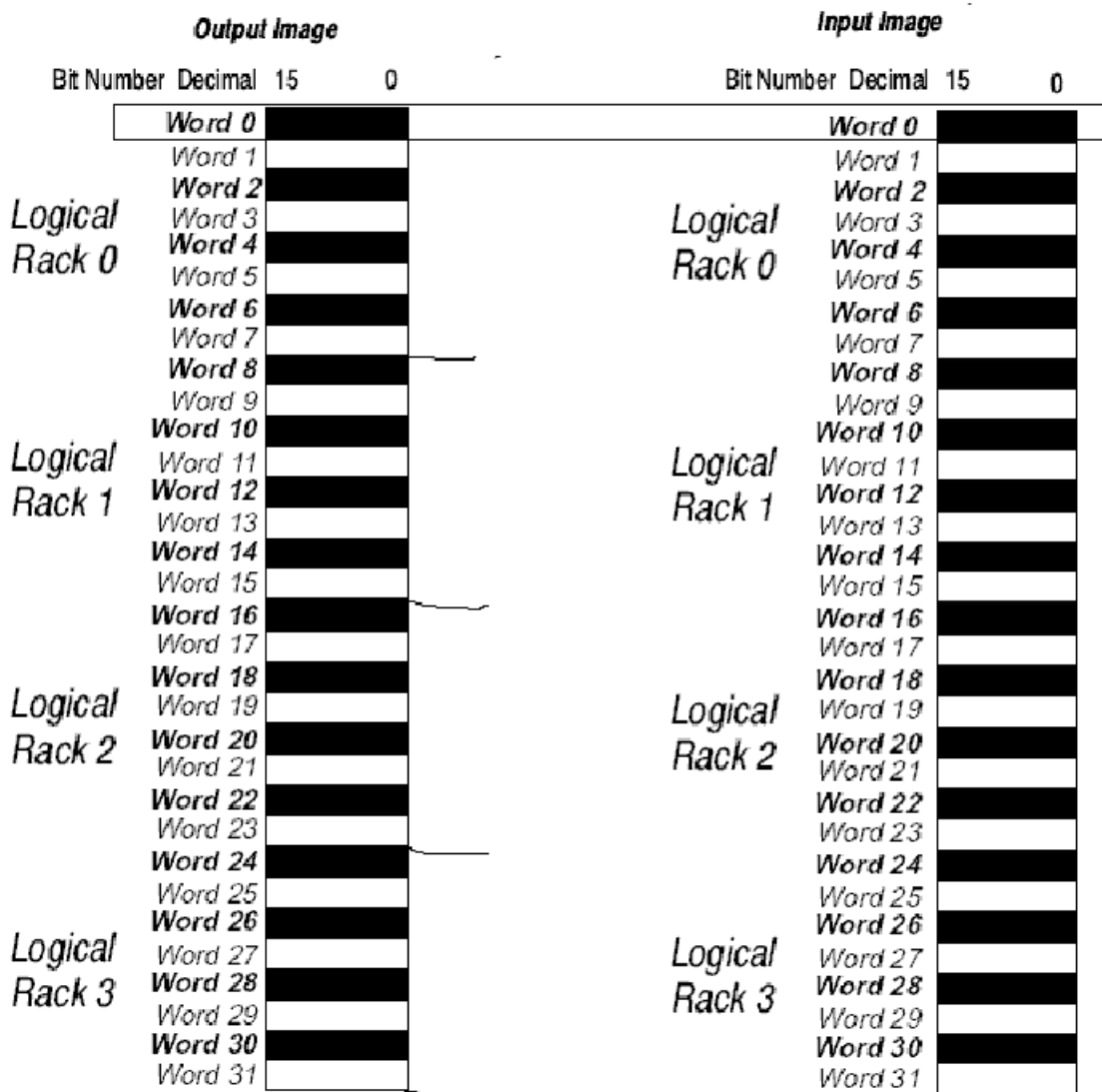
定义:

- 1: SN在CPU中占用32个I和O字, 即32个I/O组(group)
- 2: 一个机架 RACK占8个 group
- 3: addressing 机架寻址方式 (物理机架和逻辑机架的对应关系), 在ASB的DIP开关设置寻址方式、起始RACK和I/O号。
 - 1槽寻址 1槽对应1个I/O组 *****最常用
 - 2槽寻址 2槽对应1个I/O组
 - 1/2槽寻址 1槽对应2个I/O组

1槽寻址模式下:

1 RACK==8 I/O 组 == 8槽硬件

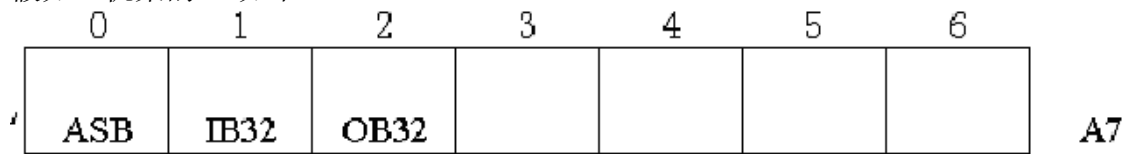
SN= 4 RACK = 32 I/O组= 32 槽硬件



2#的第一块IB16 I:6.0/0—15

第二块IB16 I:6.1/0—15

假如2#机架的I/O如下



则I/O地址为：

IB32: I:6.0/0—15 I:6.1/0—15
OB32: O:6.0/0—15 O:6.1/0—15

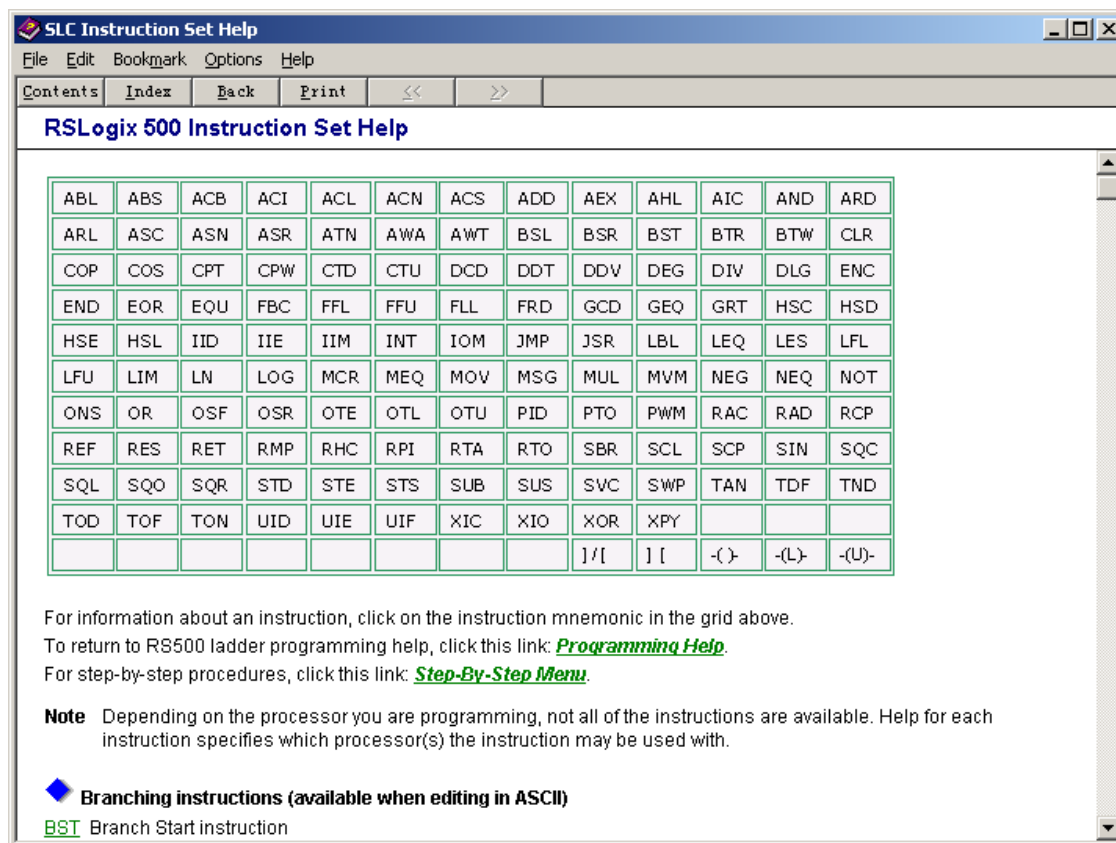
I字和O字互相借用,必须是32点的I和O配对安装, 否则浪费地址空间

实验三 基本指令介绍

重点介绍指令：

- 位操作（继电器）
- 定时器/计数器
- 运算
- 传送
- 文件操作

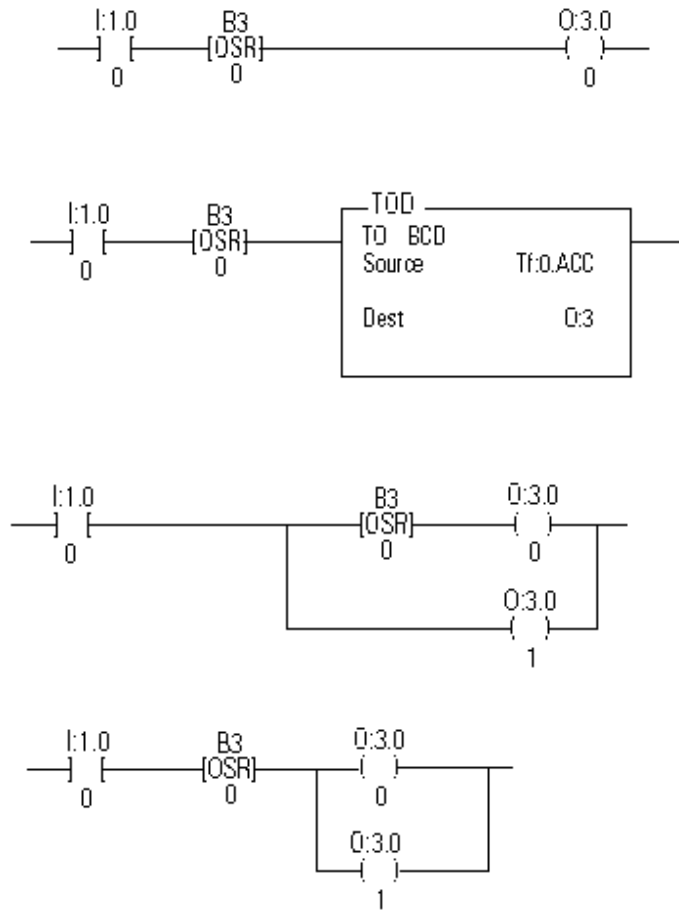
指令集帮助：



1:位操作指令

- **XIC --[]**— 对bit进行ON状态检测 (常开)
- **XIO --[/]**— 对bit进行OFF状态检测 (常闭)
- **OTL -- (L)** — 输出锁存 (置位)
- **OTU -- (U)** — 输出解锁 (复位)
- **OSR --[OSR]**— 一次响应指令，输入条件由加变真，输出保持一个周期 (上升沿触发)

对OSR举例：



2:定时器指令

	15 14 13	
Word 0	EN TT DN	Internal Use
Word 1	Preset Value	
Word 2	Accumulator Value	

Addressable Bits

EN = Bit 15 Enable
 TT = Bit 14 Timer Timing
 DN = Bit 13 Done

Addressable Words

PRE = Preset Value
 ACC = Accumulated Value

定时器地址由三个字元素(3 word)组成的数据结构。

Word0 为控制字: EN、TT、DN

Word1 为预置值 PRE

Word2 存储定时累加值 ACC

输入参数： PRE、Time Base 定时时基(1ms,10ms,1000ms)

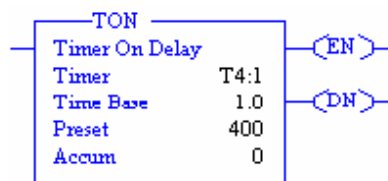
地址结构：

T4:0/15 or T4:0/EN	使能位
T4:0/14 or T4:0/TT	正在计时位
T4:0/13 or T4:0/DN	完成位
T4:0.1 or T4:0.PRE	预置值
T4:0.2 or T4:0.ACC	累积

定时器类型：

- **TON** 延时导通
- **TOF** 延时断开
- **RTO** 保持计时
- **RES** 计时器和计数器复位

TON 延时导通



状态位	置位条件	保持置位直到
DN(Bit13)	ACC » PRE	梯级条件为假
TT (Bit14)	梯级条件为真且 ACC « PRE	梯级条件为假； DN=0
EN (Bit15)	梯级条件为真	梯级条件为假

1) 正在计时而未到达 PRE 值时，如停电或由 RUN==》PRO 后：

EN 和 TT 置位，ACC 保持不变

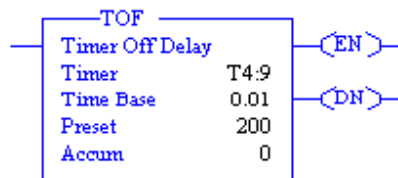
2) 再次回到 RUN 状态时，计时器继续：

梯级条件为真：重新计时（EN 和 TT 置位，ACC=0）

梯级条件为假：全部复位

时序图(****)

TOF 延时断开



状态位	置位条件	保持置位直到
DN(Bit13)	梯级条件为真	梯级条件为假而且 ACC \gg PRE
TT (Bit14)	梯级条件为假且 ACC \ll PRE	梯级条件为真；DN=0
EN (Bit15)	梯级条件为真	梯级条件为假

RTO 保持计时

当条件由真变假时，计时器不复位，当条件又由假变真时，继续计时。

其状态变化情况和 TON 一样。

RES 计时器和计数器复位

-- (RES)

3:计数器指令

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Word 0	CU CD DN OV UN UA								Internal Use							
Word 1	Preset Value															
Word 2	Accumulated Value															

Addressable Bits

CU = Count up enable
 CD = Count down enable
 DN = Done bit
 OV = Overflow bit
 UN = Underflow bit

Addressable Words

PRE = Preset
 ACC = Accum

计数器地址由三个字元素(3 word)组成的数据结构。

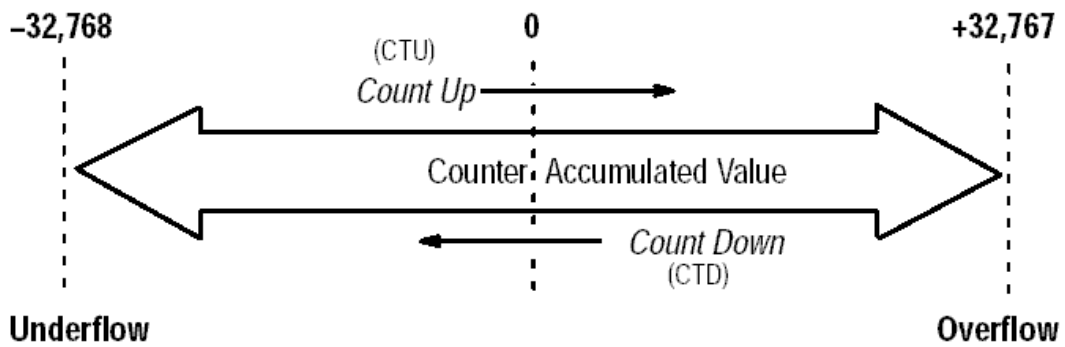
Word0 为控制字：CU、CD、DN、OV、UV

Word1 为预置值 PRE

Word2 存储计数累加值 ACC

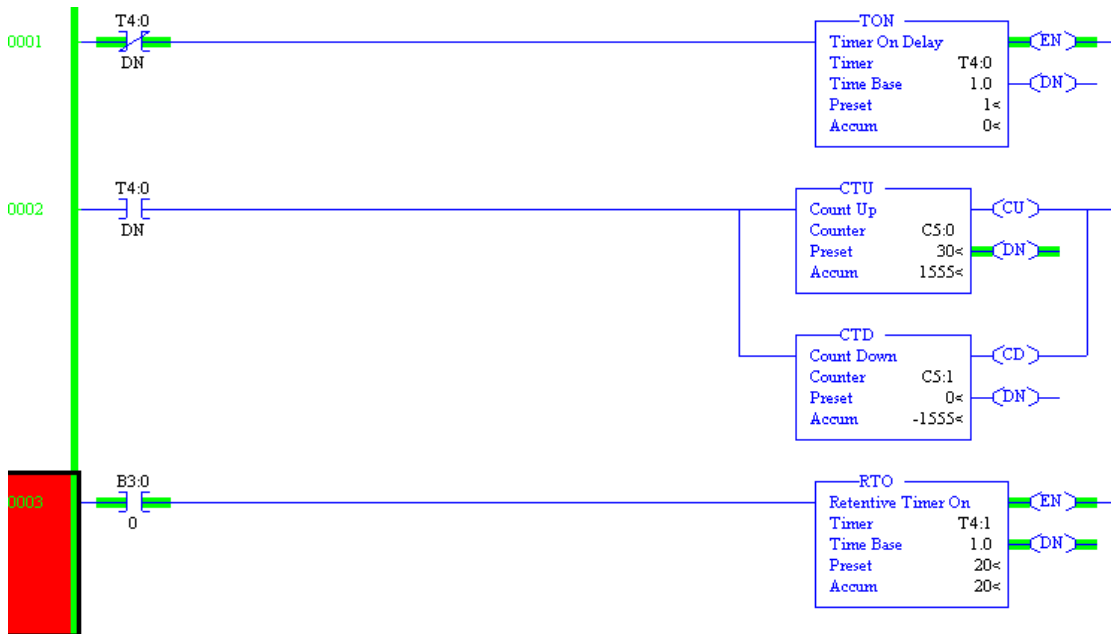
C5:0/15 or C5:0/CU	加计数使能位
C5:0/14 or C5:0/CD	减计数使能位
C5:0/13 or C5:0/DN	完成位
C5:0/12 or C5:0/OV	溢出位

C5:0.1 or C5:0.PRE 预置值
 C5:0.2 or C5:0.ACC 累积



CTU/CTD 加计数和减计数器

****: 输入条件由假变真时执行一次加/减



4:比较指令

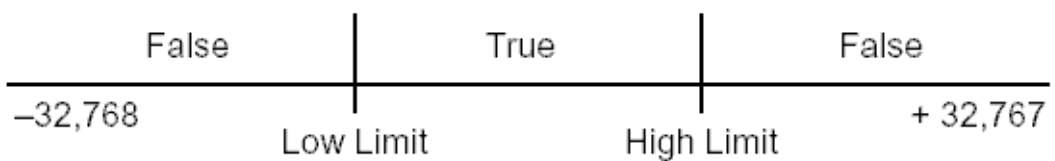
- EQU ==
- NEQ <>
- LES <
- LEQ <=
- GRT >
- GEQ >=
- **MEQ 屏蔽比较**
- **LIM 极限测试**

** 比较指令缺省用 DEC，也可以用 0101B， 0FAB3H

** 源 A 必须是地址，如 N7:0 或#N7:0

LIM 极限测试

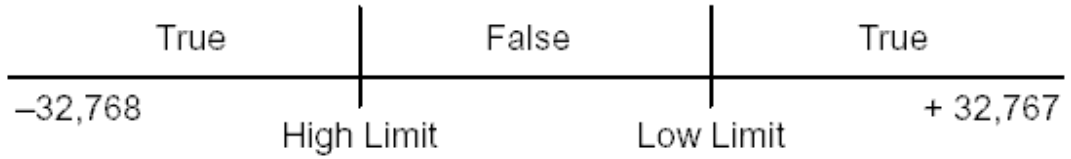
1)



Example, low limit less than high limit:

Low Limit	High Limit	Instruction is True when Test value is	Instruction is False when Test value is
5	8	5 through 8	-32,768 through 4 and 9 through 32,767

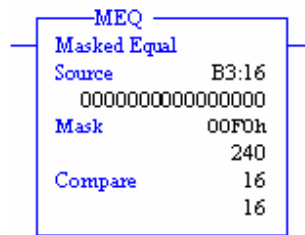
2)



Example, low limit greater than high limit:

Low Limit	High Limit	Instruction is True when Test value is	Instruction is False when Test value is
8	5	-32,768 through 5 and 8 through 32,767	6 and 7

MEQ 屏蔽比较



- 源为比较的地址
- 屏蔽: bit=1,通过该数据; bit=0,被屏蔽
- 比较: 数值或地址

eg: Sourec 为 N7:0 (xxxx xxxx abcd xxxx)

屏蔽: 00F0h(0000 0000 1111 0000)

比较: N7:1 (0000 0000 mnop 0000)

只对要 B3:16 的 bit4—Bit7 进行比较, 当 abcd=mnop 比较结果为 TRUE; 否则为 FALSE

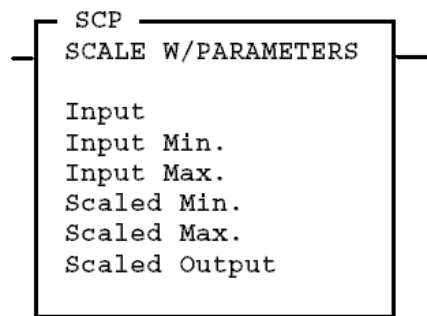
5:算术指令

- 数学运算 ADD, SUB, MUL, DIV, NEG, SQR, SCP, ABS
- 计算指令 CPT
- 三角函数 sin,cos,tan,atn
- 高级运算 XPY, LN, LOG, RAD, DEG

SLC5/03 以上处理器支持浮点数运算

1) SCP 工程参数整定

SLC5/03/04/05 支持



$$y = mx + b$$

y = scaled output

m = slope (scaled max. – scaled min.) / (input max. – input min.)

x = input value

b = offset (y intercept) = scaled min – (input min. × slope)

例如 AI 标定：0-32767-----》0-50kpa (工程单位)

AO 标定 0-100%-----》0-32767

AI 直接工程标定：输入源 0(min) 32767(max)

定标后 0(min) 50(max)

6:逻辑指令

AND, OR, NOT, XOR

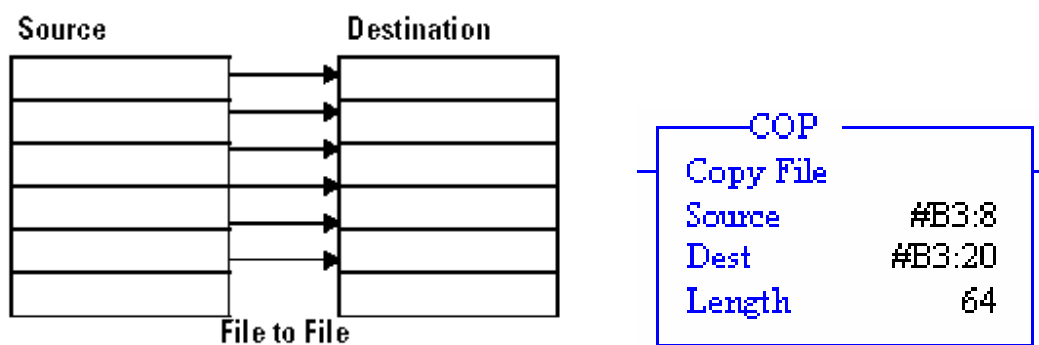
7:转换指令

- TOD, FRD
- DEG,RAD
- DCD,ENC 4-1decode,1-4encode

8:复制和拷贝

- COP、FLL、CLR

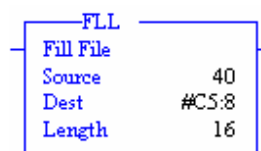
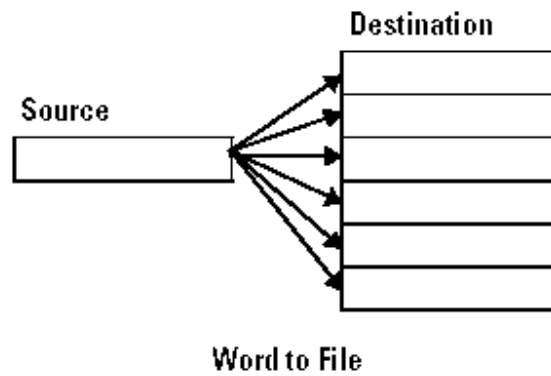
1)COP



文件对文件的复制

- Source 和 Dest 必须有地址符号#, 表示起始位置
- 长度 Lenth 最大 128

2) FLL

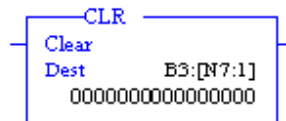


- Source 为常数或某个寄存器
- Dest 必须有地址符号#, 表示起始位置
- 长度 Lenth

Destination filetype	Maximum length
Output	128
Input	128
Bit	128
Timer	42
Counter	42
Control	42
Integer	128
String	3
Float	64
Long	64

3) CLR

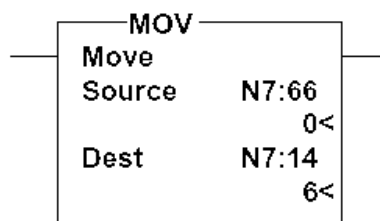
对某 word 清零操作。



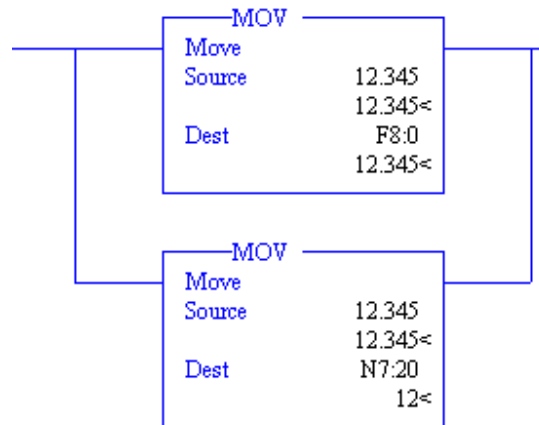
9: 传送指令

- Mov
- MVM

1) Mov

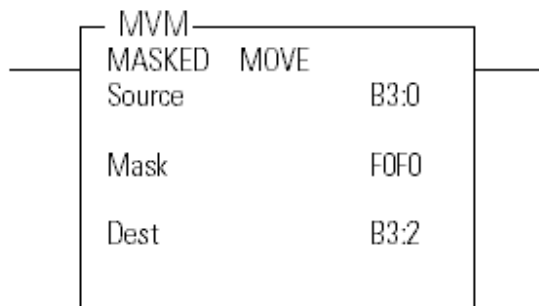


- source 为常数或地址
- Dest 必须为地址
- 实现一个 16bit 的整数或 32bit 的浮点数
- 当源为浮点，目的为整数地址时，结果四舍五入取整
- 多于 1 个地址数据传送时，采用 COP 指令



2) MVM 屏蔽传送

部分目的数据被屏蔽(取决于 mask,bit=0 屏



B3:2 before move

1111111111111111

source B3:0

0101010101010101

Mask F0F0

1111000011110000

B3:2 after move

0101111101011111

蔽)

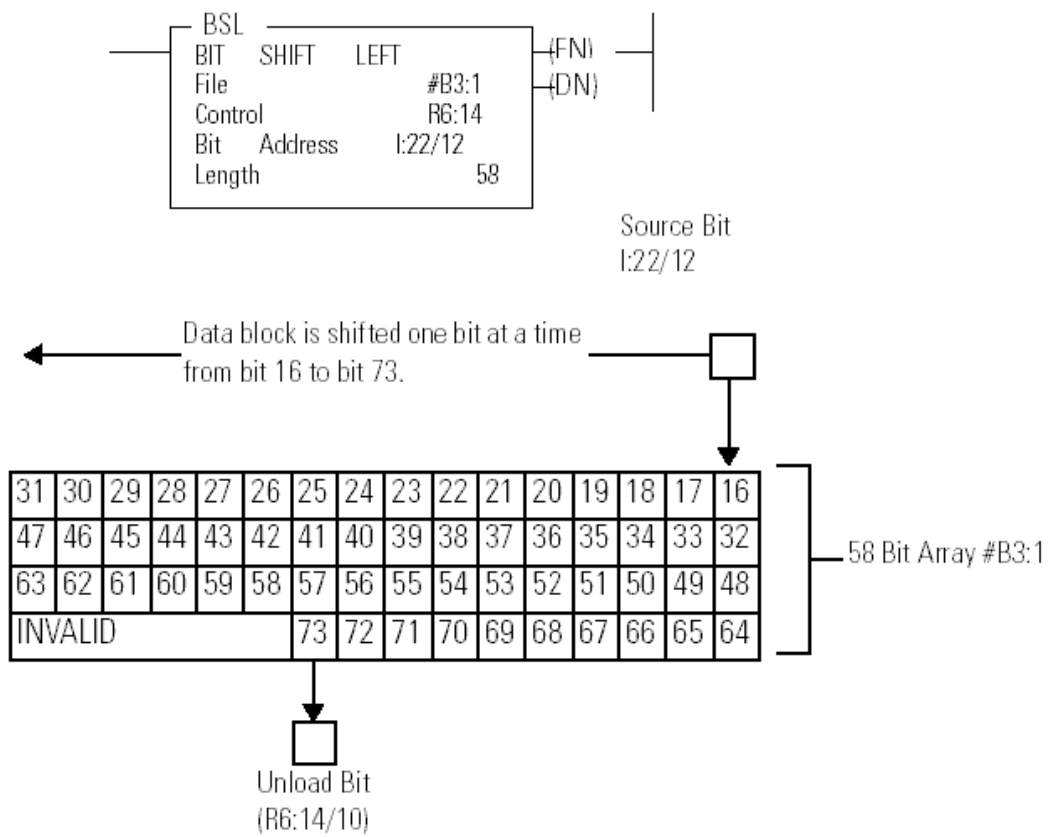
该例表示 b3:2 的 bit0-bit3 以及 bit8-bit11 的值不因执行将 b3:0 的数据传送到 b3:2 而改变。

10:移位指令

- BSL
- BSR
- SQO
- SQC

当梯级条件由假变真时，BSL/BSR 指令装入一位进入队列，在整个队列中的数据左移/右移以为，同时卸出一位进入到 UL 卸出位

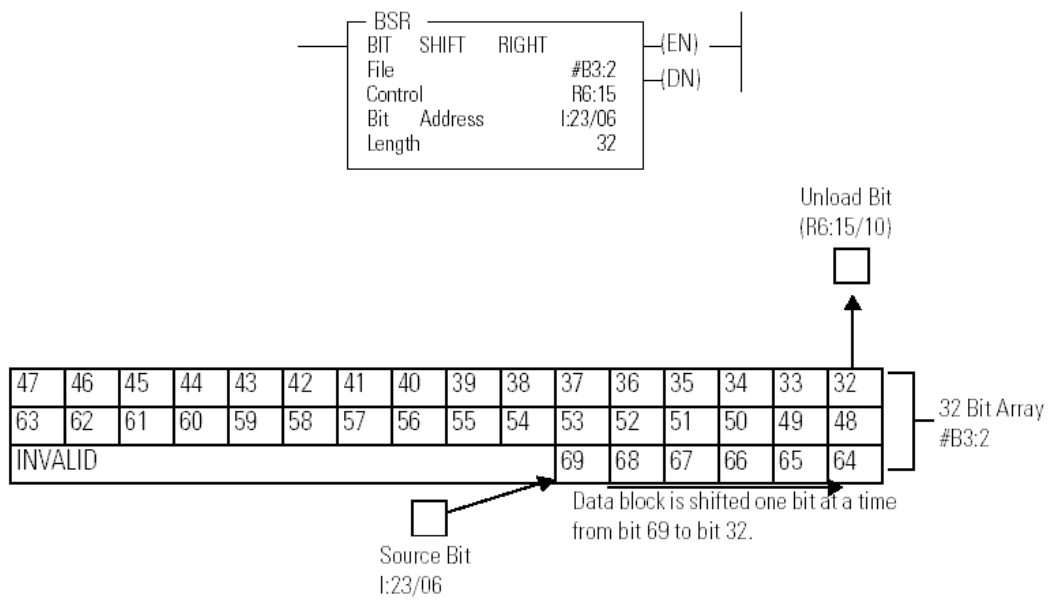
1) BSL



- length 最大 2048,128word
- 当需要实现循环移位时，将 Bit Address 变为 UL 位即可，本例为 R6:14/10

2) BSR

与 BRL 类似



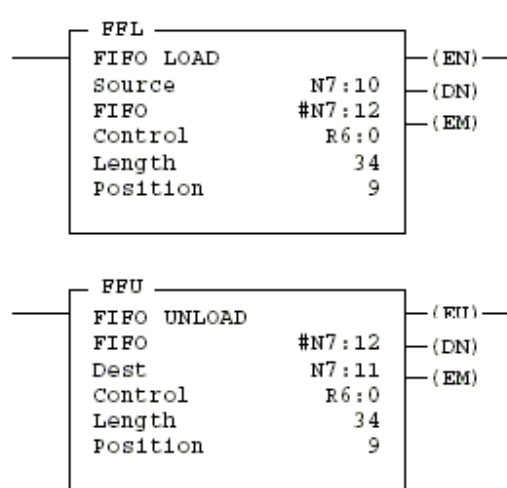
3) SQQ 顺序器比较指令

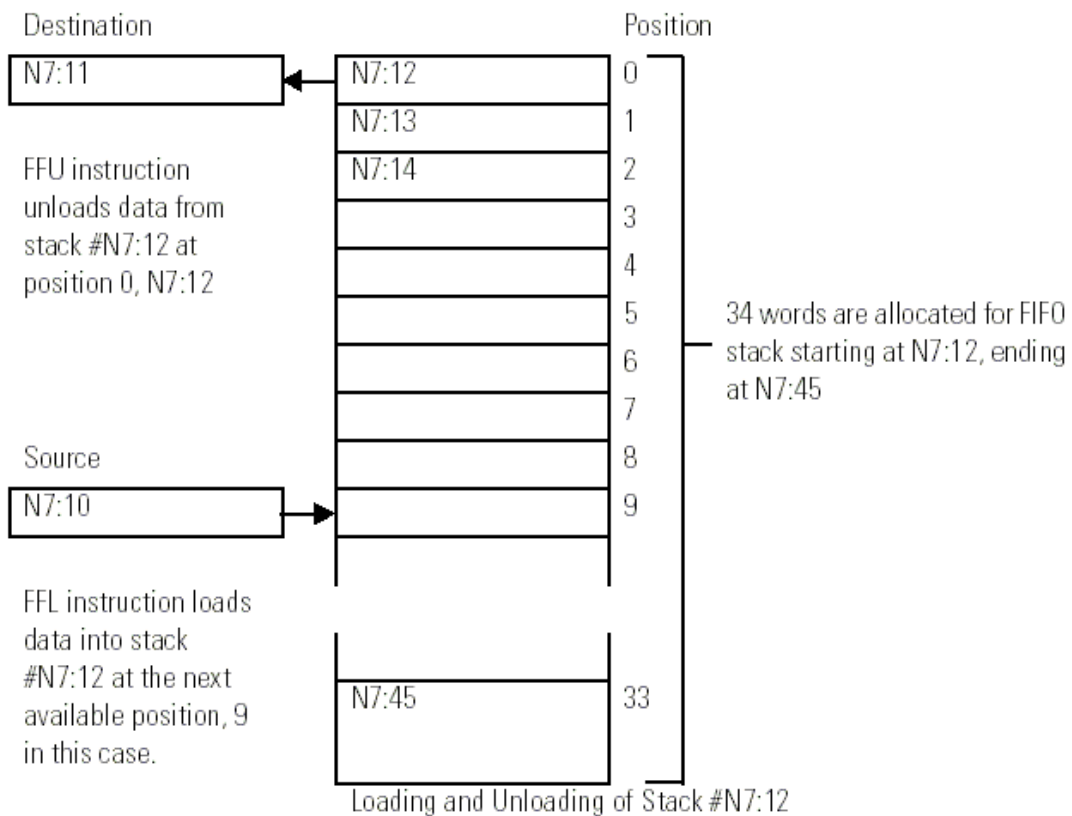
11:缓冲区指令 FIFO,LIFO

FIFO 指令把字装入文件并且按照他们被装入的顺序卸出；LIFO 后入先出正好相反。

(可以将 FIFO 和 BSL 做比较理解，一个是 Bit，一个是字)

- **FFL、FFU FIFO 装入、FIFO 卸出**
- **LEL、LFU LIFO 装入、LIFO 卸出**
- 使用时必须装入和卸出配对使用
- 梯级条件由假变真时执行一次
- FFL 触发条件为上升沿
- FFU 触发条件为 R6:0.DN





当 R6:0.POS 为 33 时，每次将 N7:10→N7:45

FFL:当上升沿到来时，FFL 的 EN=1，N7:10→N7:9，POS 加 1；

直到堆栈满为止（34 个元素），DN=1，禁止源装入。

FFU:当上升沿到来时，FFU 的 EN=1，堆栈中位置 0 的元素移出→N7:11, POS 减 1。直到栈空为止。

12:程序控制指令

JMP、LBL、**JSR**、RET、**SBR**

1) JMP, LBL

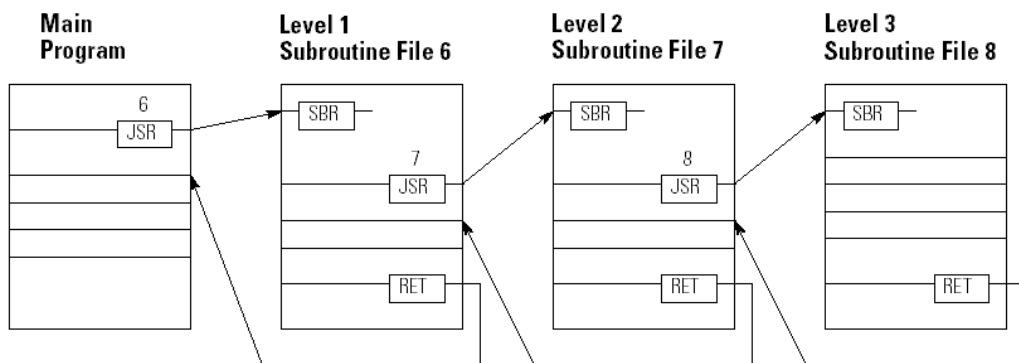
- 配对使用
- 标号可达 0-999

Jump (JMP) and Label (LBL)

—(JMP)—

——|LBL|——

2) JSR, SBR, RET



- SLC5/01 四级子程序
- SLC5/02 以上八级子程序
- STI、DDI、Fault、HSC 中断，只能三级
- JSR 中的 file number 为 3-255

13:MSG 通讯指令

SLC5/03/04 通讯通道

When using an SLC 5/03 or SLC 5/04 processor the message instruction:

- initiates reads and writes through RS-232 Channel 0 when configured for the following protocols:
 - DF1 Full-Duplex (Point-to-Point)
 - DF1 Half-Duplex Master/Slave (Point-to-Multipoint)
 - DH-485
- initiates reads and writes through:
 - DH-485 channel 1 (SLC 5/03 processors only)
 - DH+ channel 1 (SLC 5/04 processors only)

可选的通讯方式

- Peer-to-Peer Read/Write on a Local network to another SLC 500 processor
- Peer-to-Peer Read/Write on a Local network to a 485CIF
- Peer-to-Peer Read/Write on a Local network to a PLC-5^①
- Peer-to-Peer Read/Write on a Remote network to another SLC 500 processor
- Peer-to-Peer Read/Write on a Remote network to a 485CIF (PLC2 emulation)
- Peer-to-Peer Read/Write on a Remote network to a PLC-5 processor^①

参数组态

- **Read/Write** – Read indicates that the local processor (processor in which the instruction is located) is receiving data; write indicates that it is sending data.
- **Target Device** identifies the type of device which will receive data. Valid options are:
 - 500CPU, if the target device is another SLC processor
 - 485CIF, if the target device is a non–SLC device on the DH-485 network
 - PLC-5, if the target device accepts PLC-5 commands
- **Local or Remote** identifies if the message is sent to a device on a local DH-485 or DH+ network, or to a remote device on another network through a bridge. Valid options are:
 - Local, if the target device is on the local network
 - Remote, if the target device is on a remote network
- **Control Block** is an integer file address that you select. It is a 14-word integer file, containing the status bits, target file address, and other data associated with the message instruction.
- **Control Block Length** is fixed at 14 elements. This field cannot be altered.

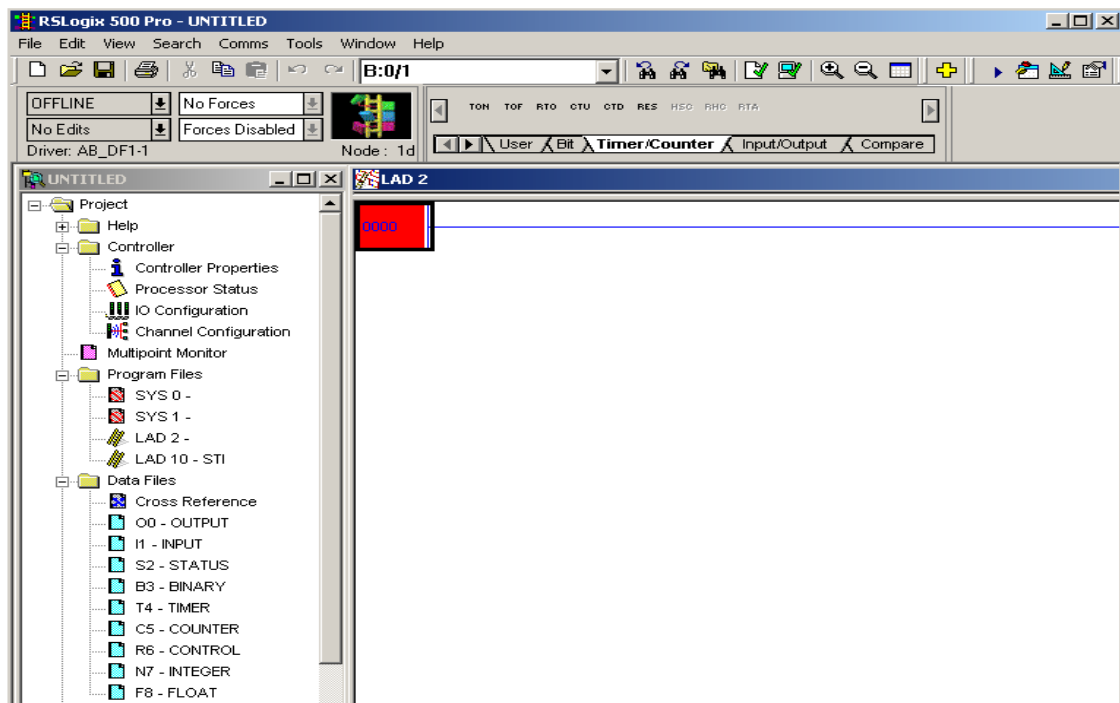
组态菜单

Type:	Peer-to-Peer	
Read/Write:	READ	ignore if timed out: 0 TO
Target Device:	500CPU	to be retried: 0 NR
Local/Remote:	Local	awaiting execution: 0 EW
Control Block:	N10:0	continuous run: 0 CO
Channel:	1	error: 0 ER
Target Node:	2	message done: 0 DN
		message transmitting: 0 ST
		message enabled: 0 EN
		waiting for queue space: 0 WQ
Destination File Addr:	N7:0	
Target Source File Address:	N7:50	
Message Length In Elements:	10	
Message Timeout (seconds):	5	
ERROR CODE: 0		control bit address: N10:0/8
Error Code Desc:		

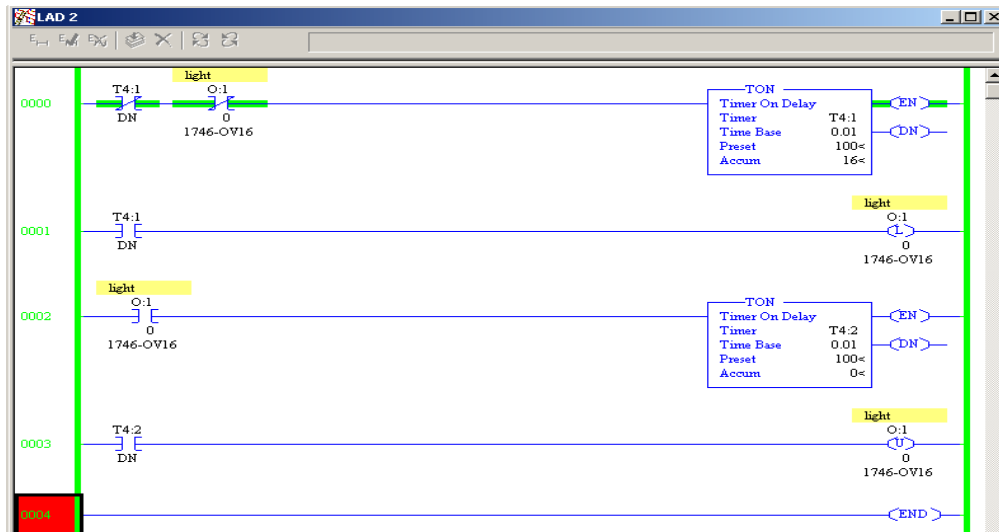
实验四 程序的编辑、保存和下载

本实验作一简单的程序：PLC控制一灯的闪烁，每隔1秒钟闪烁一次。

1、双击 LAD 2，进入程序的编程状态





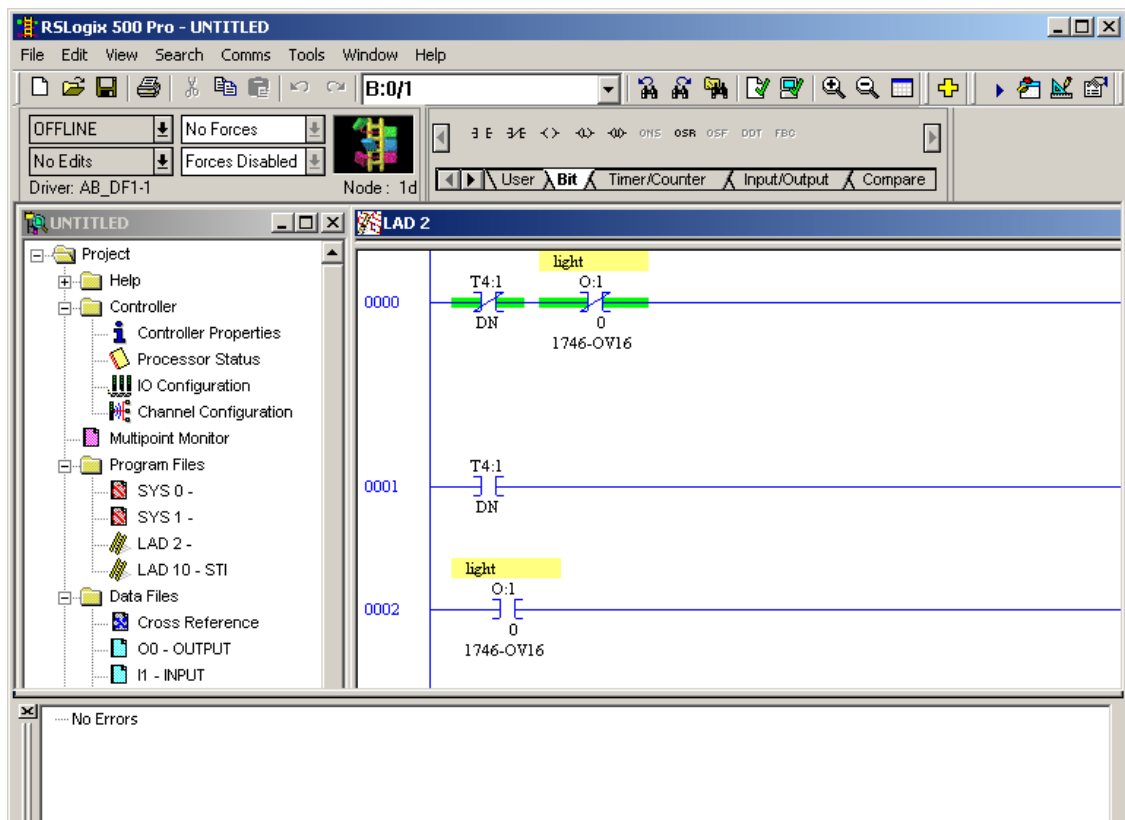
2、输入梯形图如图所示



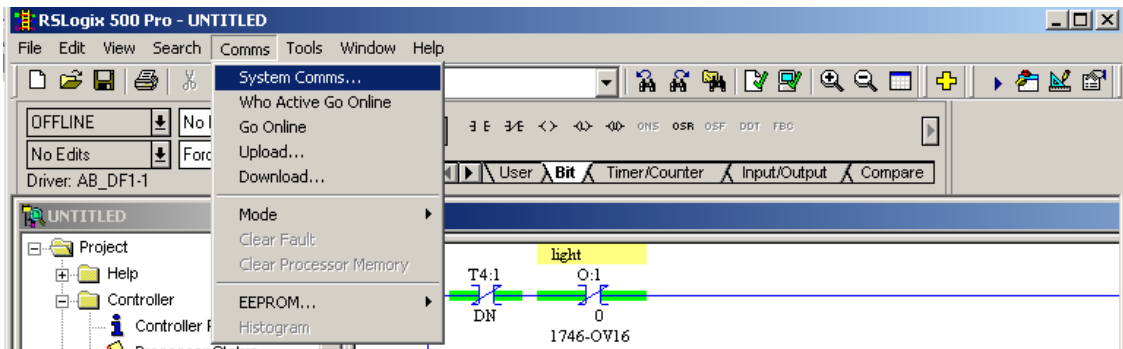
其中，T4:1，T4:2 是两个计时器，O:1/0 是输出指示灯。关于计时器的参数，在上一实验中作了介绍，这里就不在说明。

3、程序的下载

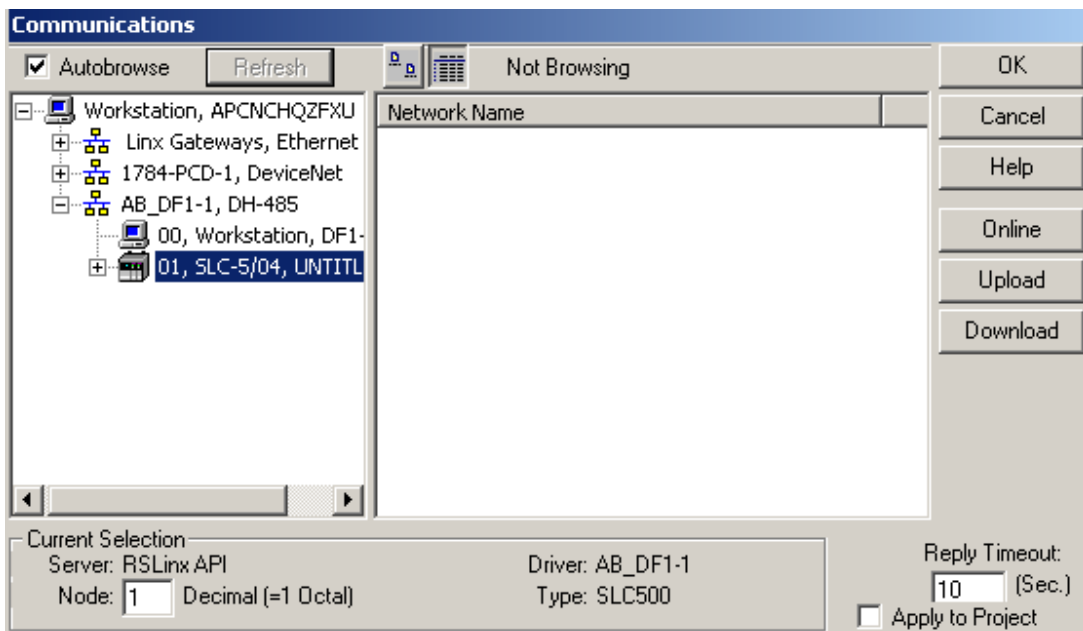
我们在实验一已经创建了 DF1 通讯路径。程序下载前，在工具栏内点击图标 进行工程校验，确定程序的正确。点击图标 保存文件。



接下来把程序下载到处理器。在菜单中找到 Comms->System Comms,点击



出现选择路径对话框，在 DF1 通讯下，鼠标点到 SLC-5/04 处理器，然后点击 Download 按钮，开始程序的下载。

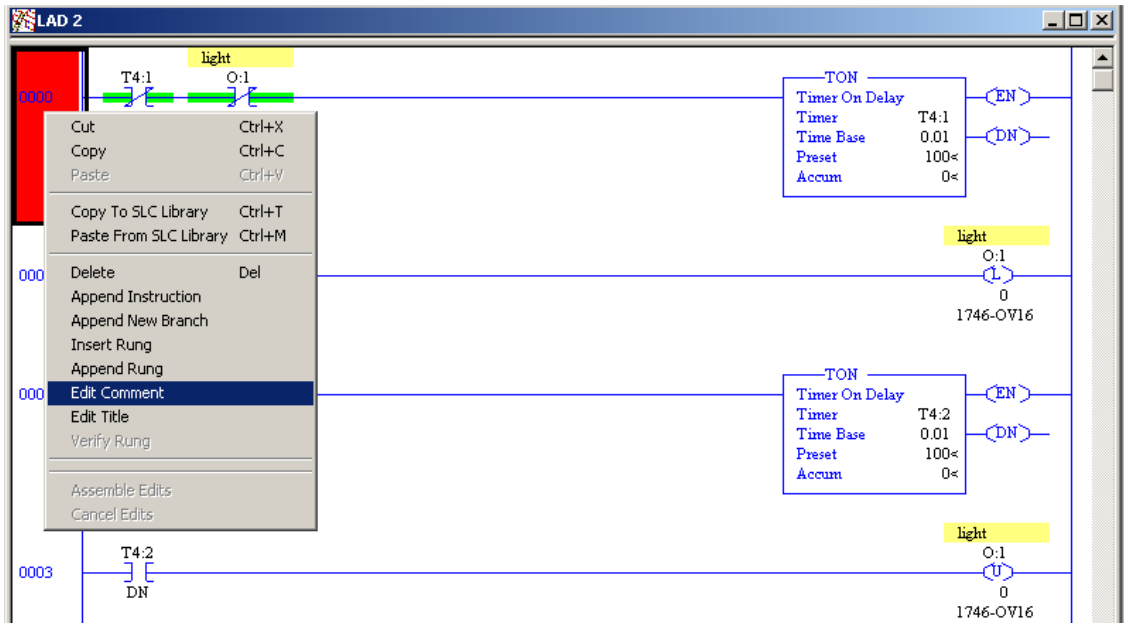


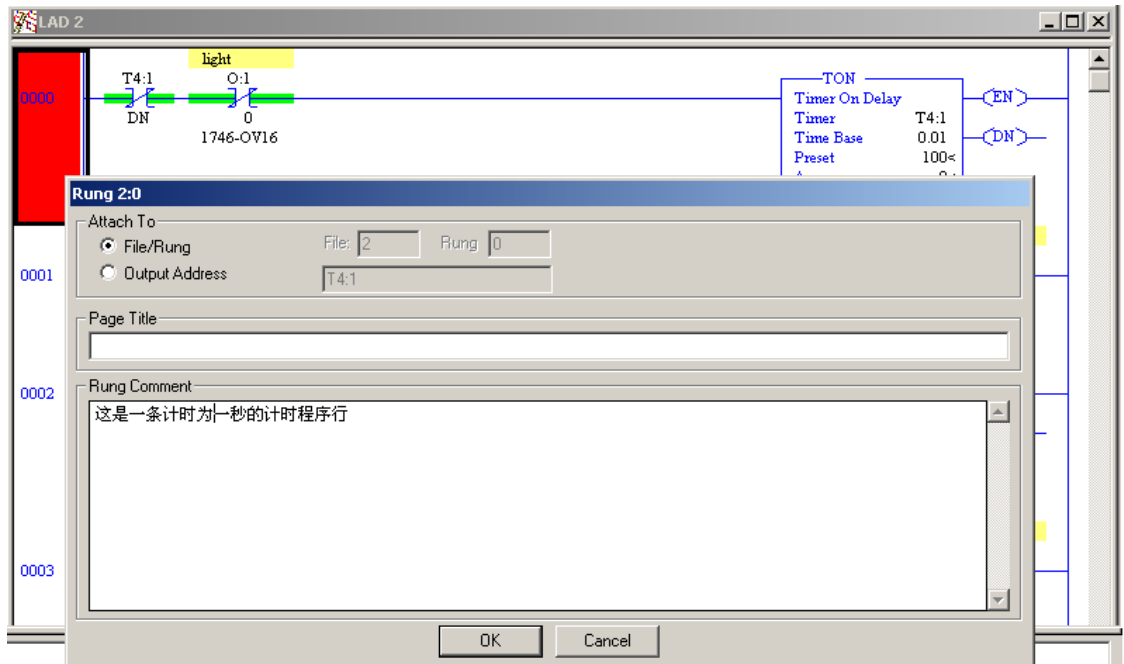
好啦，在处理器运行时，你就可以看到间隔一秒闪烁的灯啦。恭喜你，实验成功！

实验五 如何编辑程序注释

我们继续实验四的内容，对程序进行注释。

- 1、对程序 RUNG 进行注释。在程序编辑状态下，鼠标右点到要注解的行，选择 **Edit Comment**



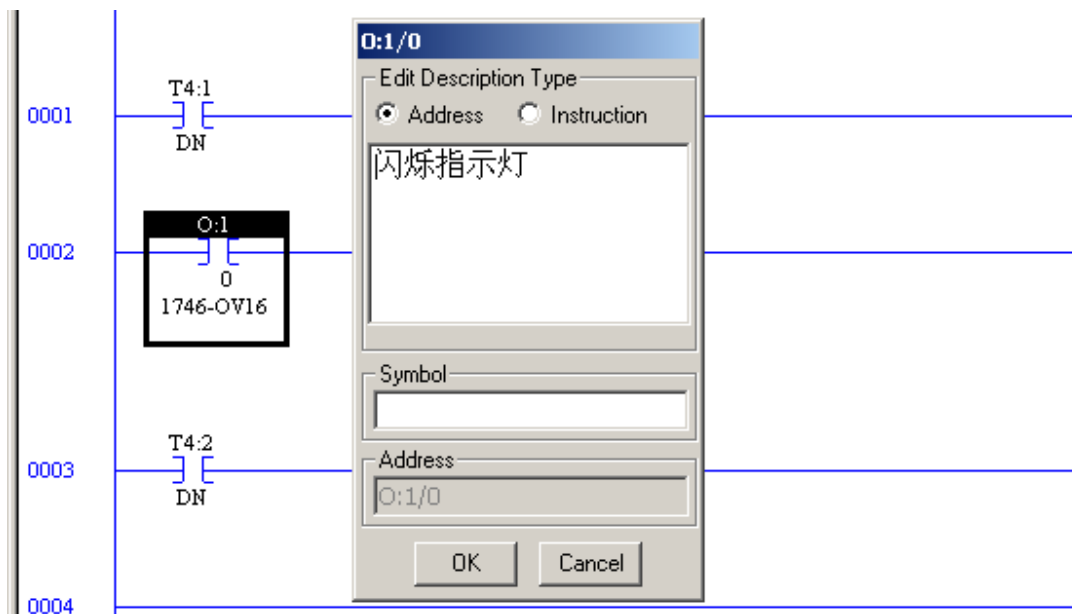
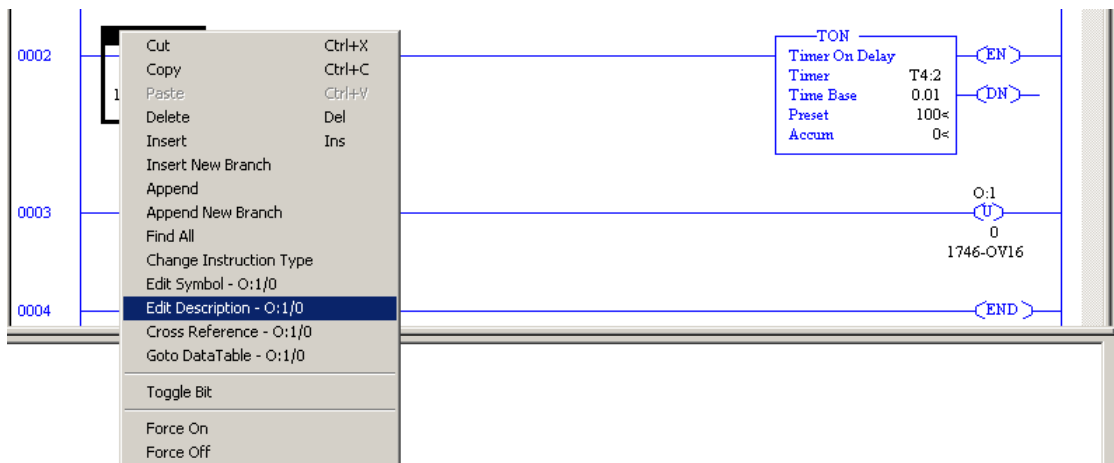


在 Rung Comment 下输入注解，按 OK 按钮，程序行上将出现刚输入的文字注解。

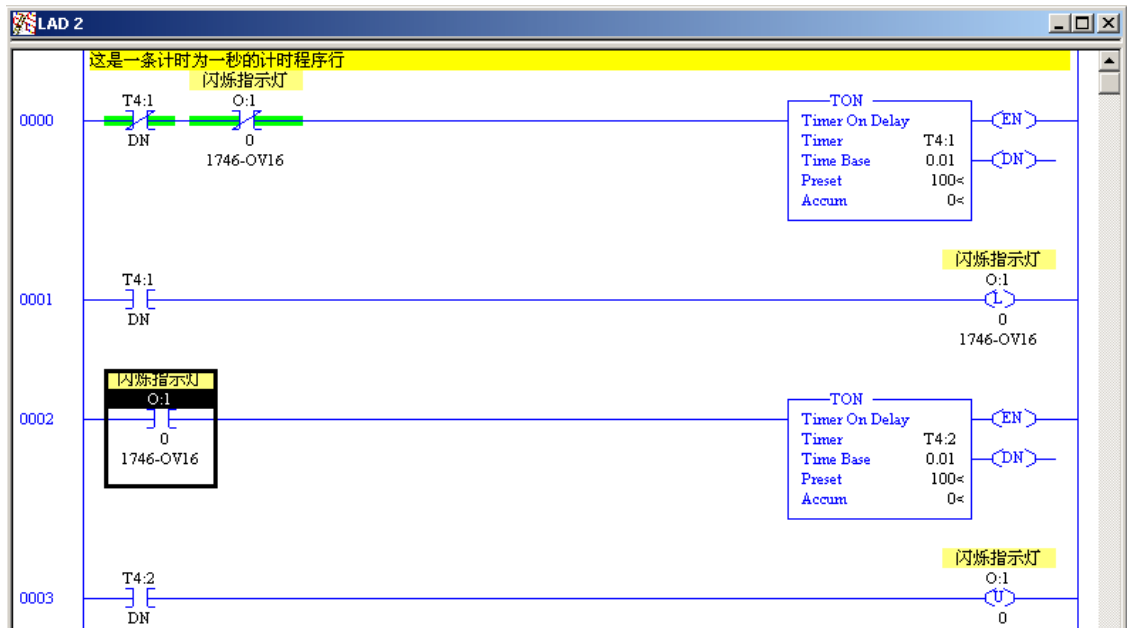


2、对程序的地址进行注解

鼠标右击想要进行注解的地址，找到 Edit Description, 点击进入对话框



输入描述性的文字注解，点击 **OK** 按钮，你就可以在程序中看到你输入的文字注解啦，程序中同一地址都会出现你输入的注释。



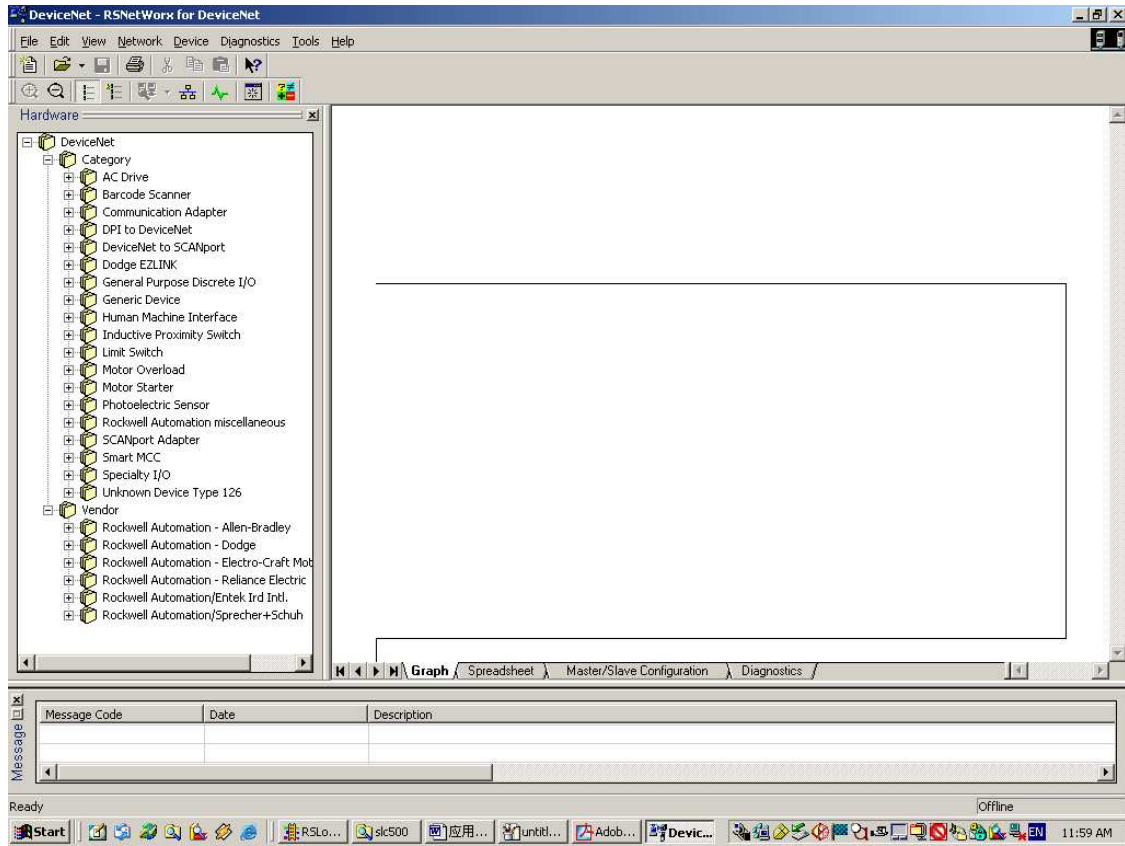
OK,这个实验就完成了。



实验六 SLC500 通过 DeviceNet 和 Panelview600 通讯

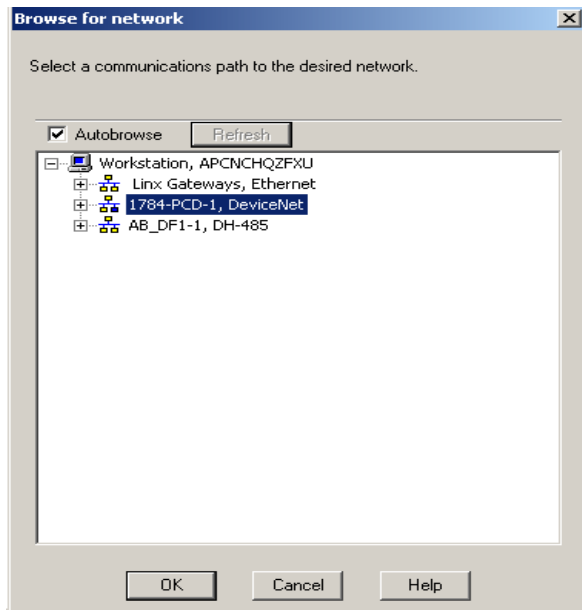
我们这次的内容里，需要 SLC500 通过 DeviceNet 和 Panelview600 通讯，在进行 PV 的实验前，先进行 DeviceNet 的准备工作。

实验中会用到 RSNetworx for DeviceNet 软件进行网络的组态。本次实验通过 1784-PCD 卡从 PC 机连到设备网，网络上连接 1747-SDN 和带 DeviceNet 接口的 PV。实验前我们已经完成了网络的连接，已经在 PC 中创建好 1784-PCD 的通讯驱动，在实验中只进行网络的软件规划。

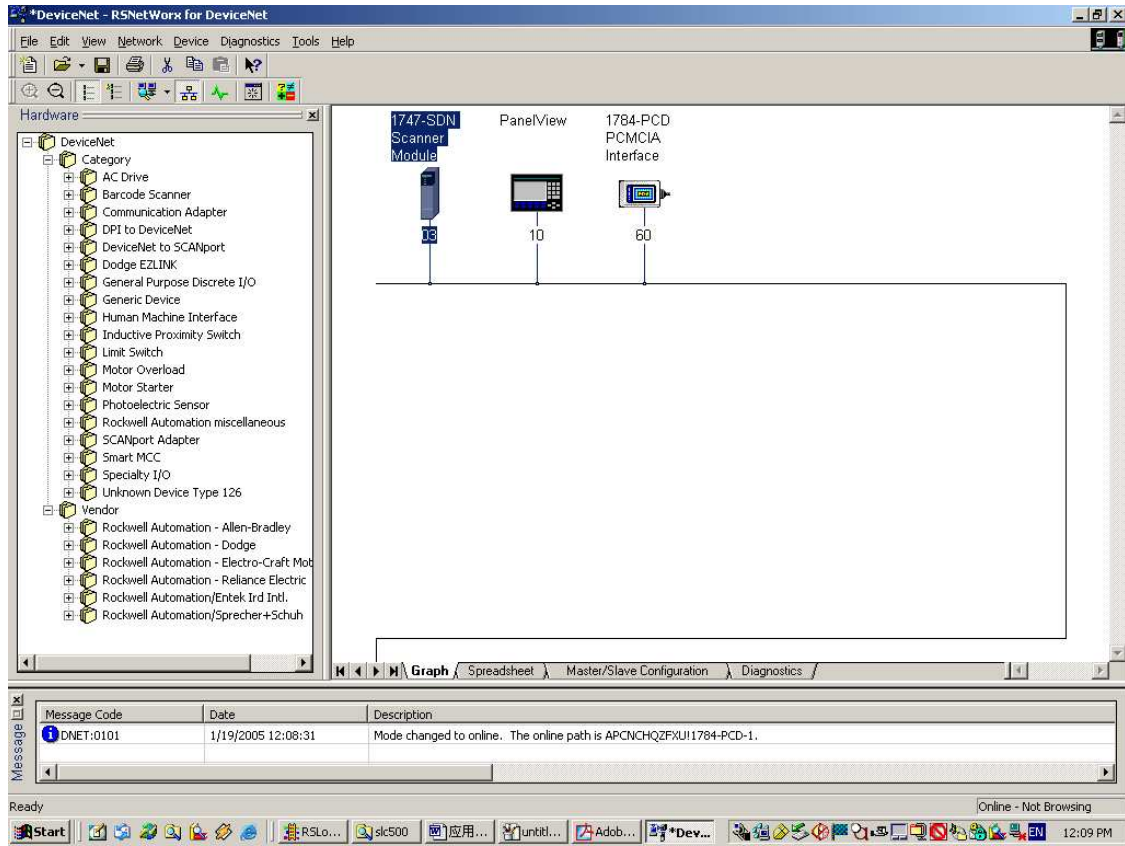
- 1、Start->program->Rockwell Software->RSNetworx->RSNetworx for DeviceNet, 运行 RSNetworx for DeviceNet



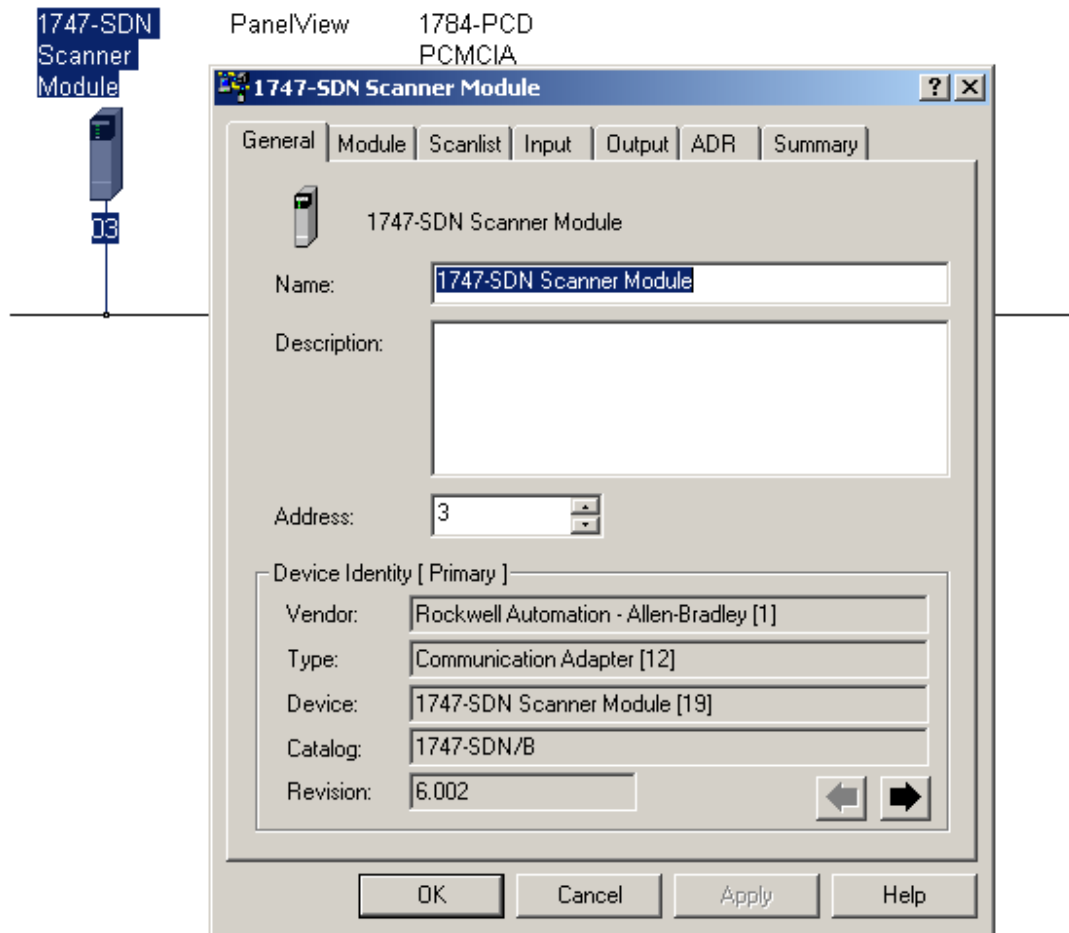
点击图标 ，建立新文件。点击在线图标 ，选择 1784-PCD,按 OK 按钮，在出现的提示信息对话框上点击 OK 按钮，进入网络节点扫描。



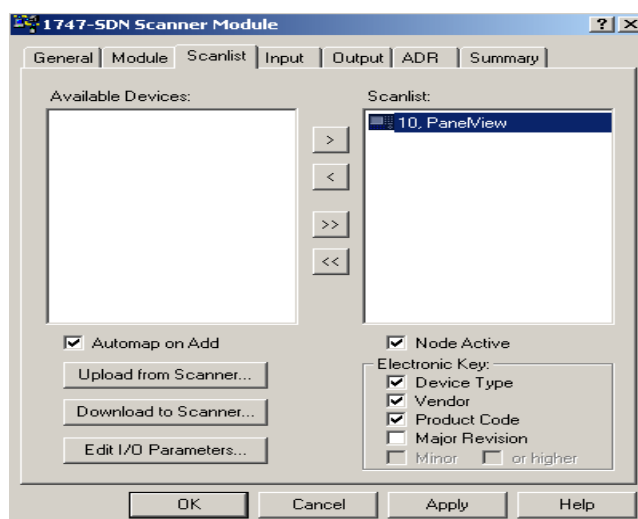
将会出现如下的画面，网络上现在只有三个节点。



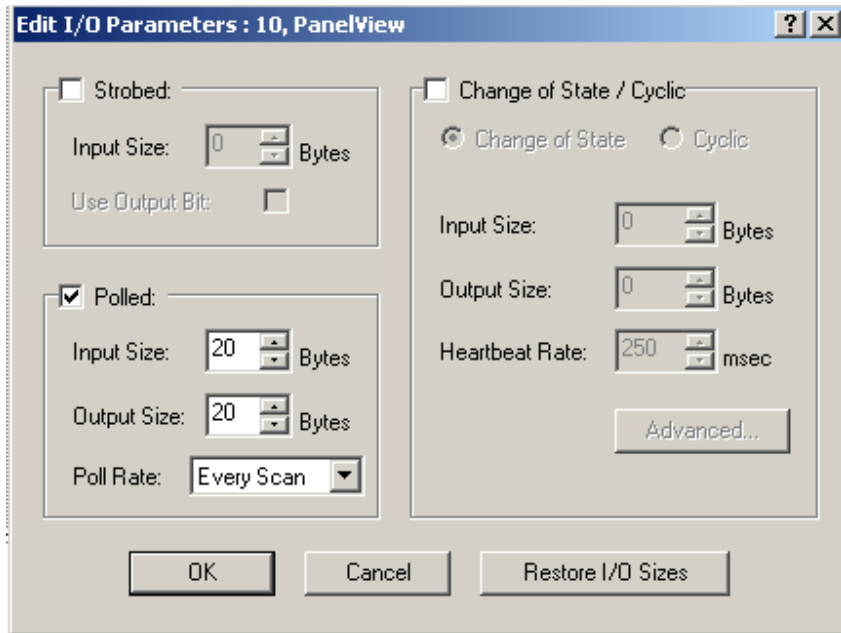
右键点击 1747-SDN 扫描器，找到 Properties,点击进入 1747-SDN 的配置



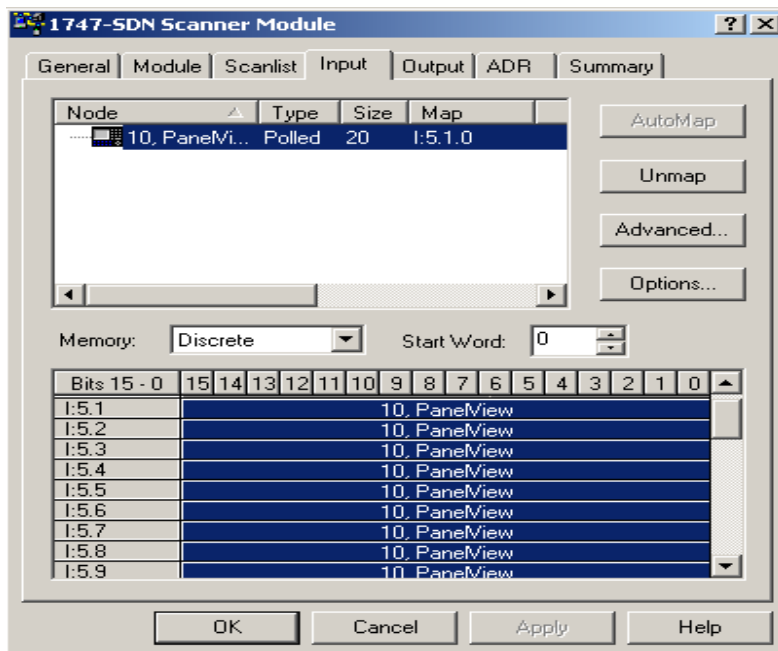
在 General 下，可以看到模块的信息，地址（在这里可以修改站地址）。接下来，我们进入 Scanlist 配置。点击 Scanlist，然后 upload



由于 PV 不含任何的 I/O，因此需要进行 I/O 配置。点击 Edit I/O Parameters 按钮，进行 I/O 配置

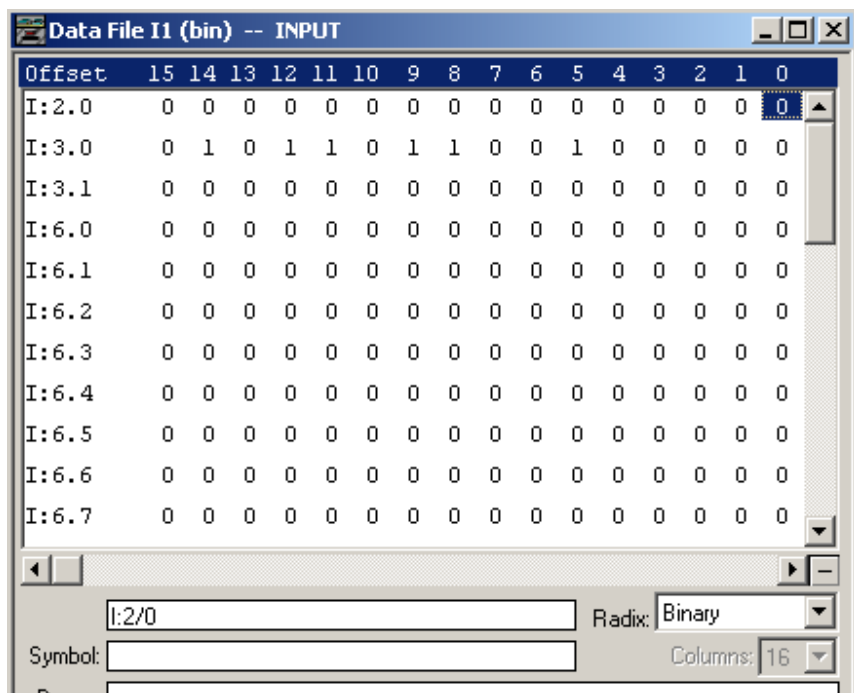


我们选择 POLLED 类型，输入输出各 10words（输入输出字节树要和 PV 相同）。在 input 和 output 下可以看见分配给 PV 的 I/O 地址



好啦，我们现在完成了网络的配置和 I/O 的分配。

现在，我们回到编程软件，看一下 I/O 数据文件，从 I:6.0 起的输入和从 O:6.0 起的输出就分配给 PV，各 10 个字。它们用作数据交换区来使用。



和 PV 的通讯，我们放在 PV 一节来具体介绍。另外，要通过 DeviceNet 来通讯，一定要激活它哟。方法是使扫描起的输出命令寄存器的运行位（RUN）为 1，它位于输出第 0 字的第 0 位。

实验七 熟悉 Panelbuilder32

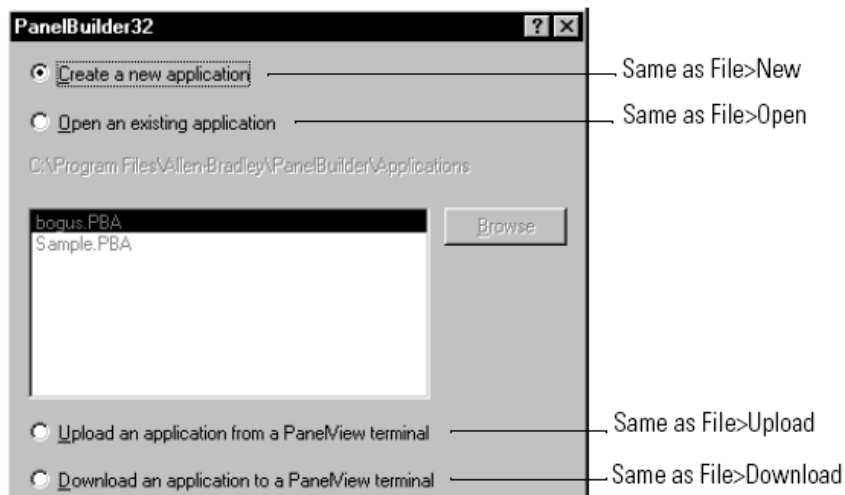
内容：

- 启动对话框

- 应用窗口
- 画面窗口
- 工具条
- 状态条
- 右击菜单
- 设置参考和默认值
- 在线帮助

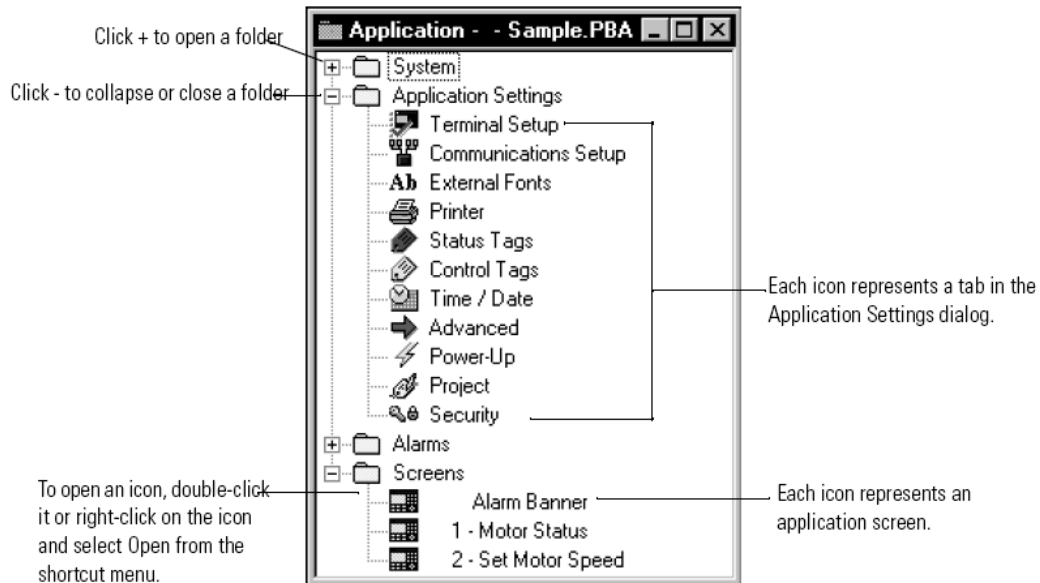
1、启动对话框

在运行 PANELBUILDER32 时，出现如下画面：

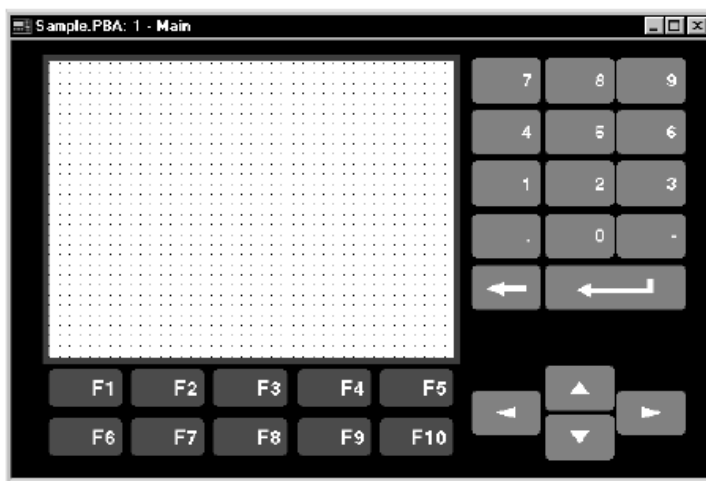


Hint: You can disable this dialog on the **General** tab of the **Tools>Options** dialog.

2、应用窗口



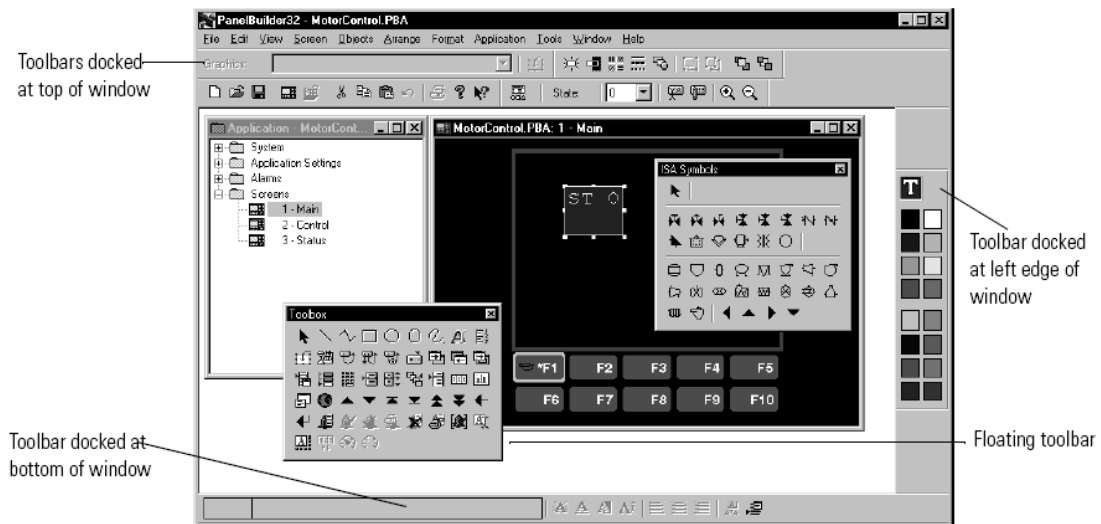
3、画面窗口



Hint: For keypad terminals, you can show or hide the keypad from the View menu.

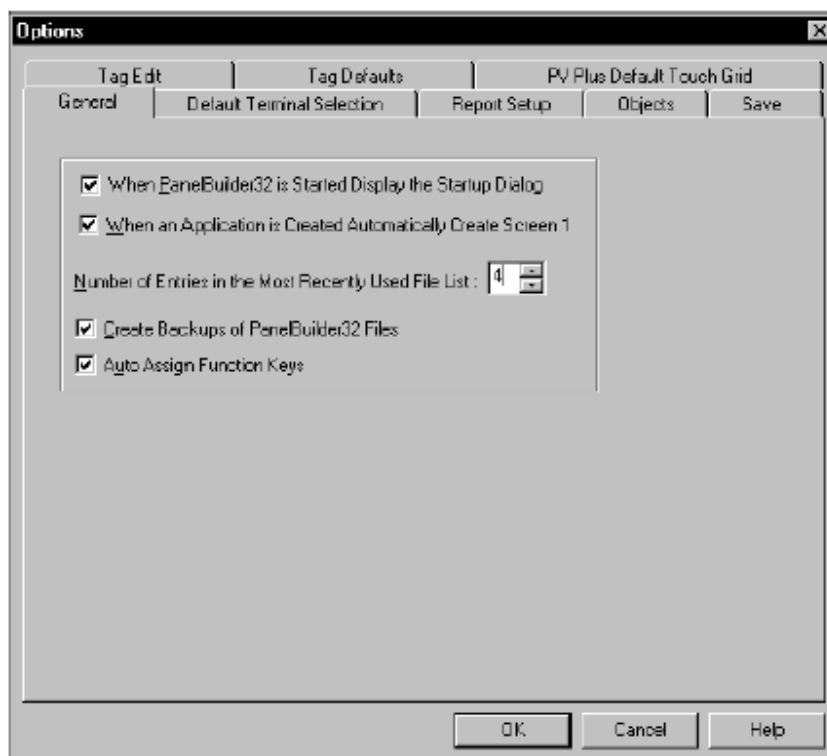
4、工具条

共有 9 个工具条。



5、设置参考和默认值

TOOLS—>OPTIONS




实验八 应用和画面

内容:

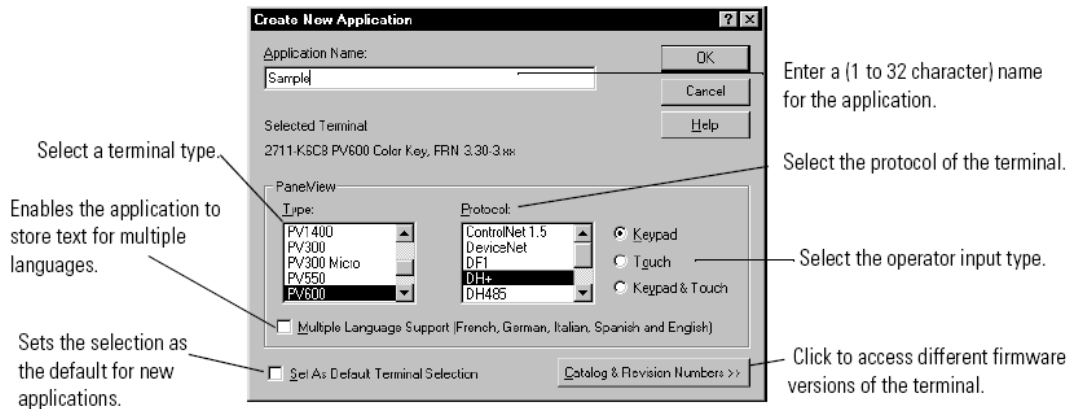
- 生成新的应用
- 使用应用画面
- 生成应用报告

1、生成新的应用

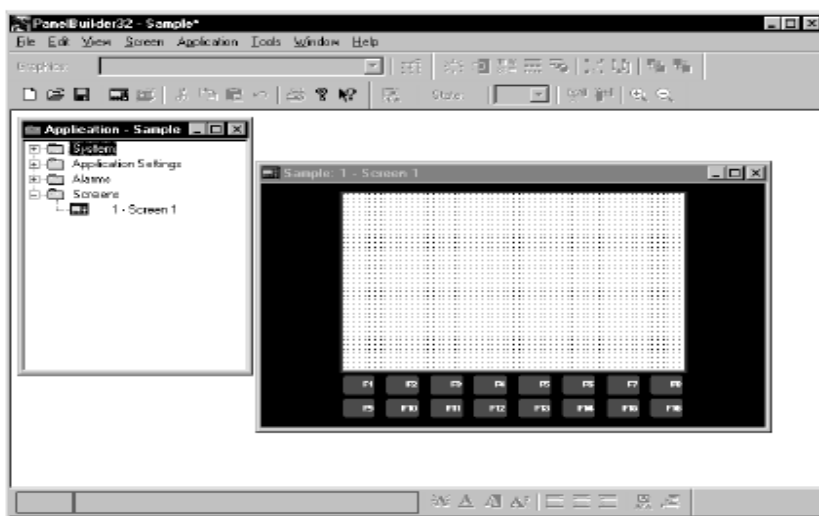
在创建新应用时，根据具体的实验设备，选择 PV 的类型、通讯协议及操作员输入类型。

- Select **File>New** or click the **New** tool on the standard toolbar. 

The **Create a New Application** dialog opens.



应用打开时，带有一个空白的画面。

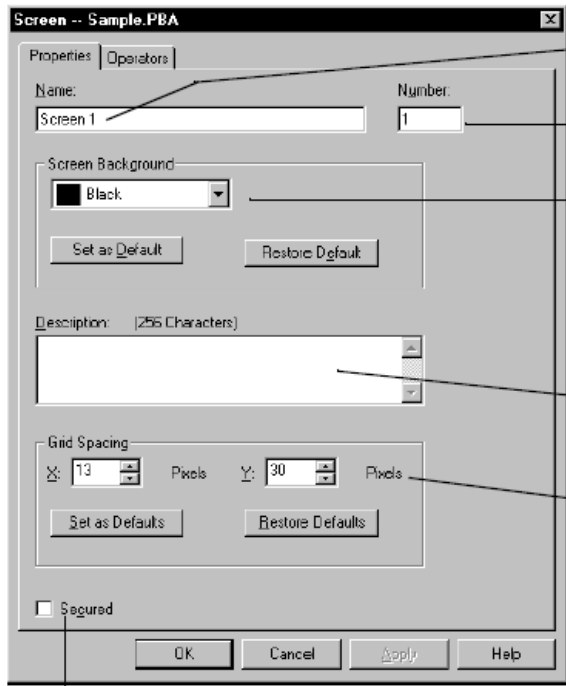


2、使用应用画面

每个画面具有下列属性：

- 名称(默认为 Screen1,Screen2.....)
- 画面序号(1-255)
- 背景
- 描述

右击 Screens→New,打开画面对话框



The screenshot shows a dialog box titled "Screen -- Sample.PBA" with two tabs: "Properties" and "Operators". The "Properties" tab is active. It contains the following fields and controls:

- Name:** A text field containing "Screen 1".
- Number:** A text field containing "1".
- Screen Background:** A dropdown menu showing "Black". Below it are "Set as Default" and "Restore Default" buttons.
- Description:** A text area with "(255 Characters)" above it.
- Grid Spacing:** Two spinners for "X" (set to 13) and "Y" (set to 30), both labeled "Pixels". Below them are "Set as Defaults" and "Restore Defaults" buttons.
- Secured:** A checkbox that is currently unchecked.
- Buttons:** "OK", "Cancel", "Apply", and "Help" at the bottom.

Annotations with arrows point to the following elements:

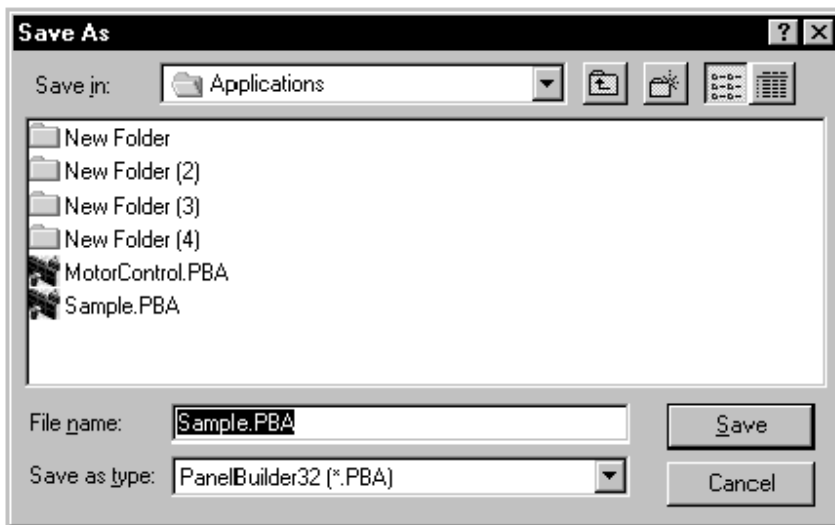
- Line to the "Name" field: "Enter a descriptive name (up to 32 characters)"
- Line to the "Number" field: "Sequential number assigned to new screen."
- Line to the "Screen Background" dropdown: "Appears only for color or grayscale terminals. You can change the color on a per screen basis. To apply setting to all new screens, click Set as Defaults. To restore last defaults saved, click Restore Defaults."
- Line to the "Description" text area: "The description appears in the application report."
- Line to the "Grid Spacing" spinners: "You can change the grid spacing on a per screen basis. To apply the settings to all new screens created for a terminal type, click Set as Defaults. To restore the last defaults saved, click Restore Defaults."
- Line to the "Secured" checkbox: "Allows only authorized operators to access screen."

3、保存应用

PanelBuilder32 使用三种文件类型：

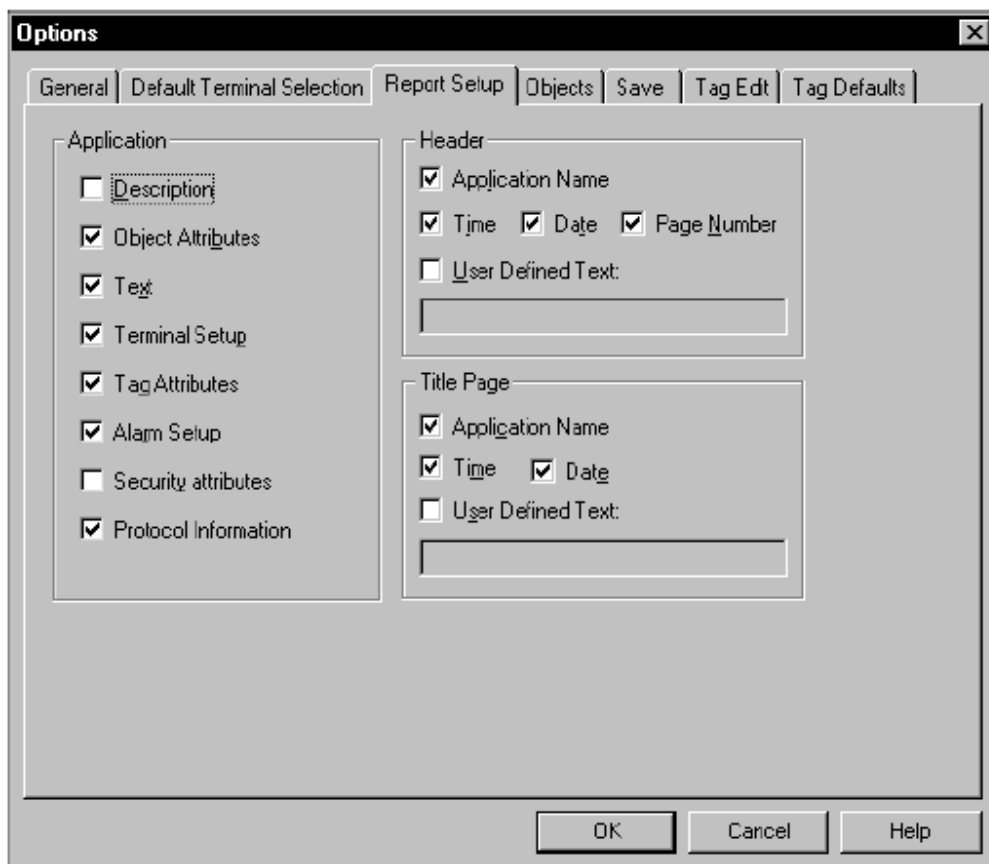
- .PBA - refers to an application configured and saved in PanelBuilder32.
- .PVA - refers to an application saved in a PanelView Terminal format.
- .BAK - refers to a backup file automatically created by PanelBuilder32.

File→Save 或 Save as



4、生成应用报告

在 **TOOLS**→**OPTIONS** 下，设置报告信息，可以根据需求，设置报告格式，并将应用信息打印成文本。

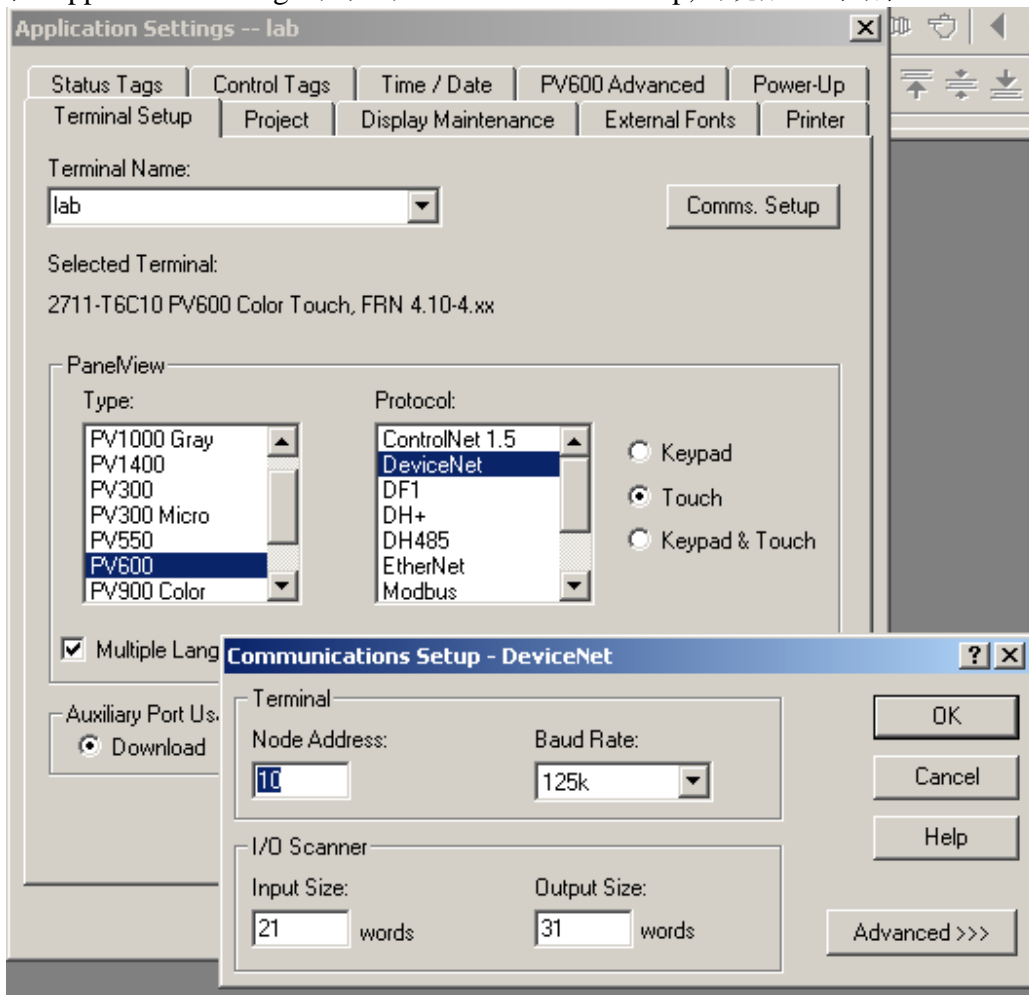


实验九 使用画面对象

1、配置通讯

PanelView 和控制器及远程设备可以以多种方式通讯，DH-485, DH+, Remote I/O, Ethernet/IP, DeviceNet, DF1, ControlNet, Modbus 等，本实验为 DeviceNet。

在 Application Settings 下双击 communications setup, 出现配置对话框



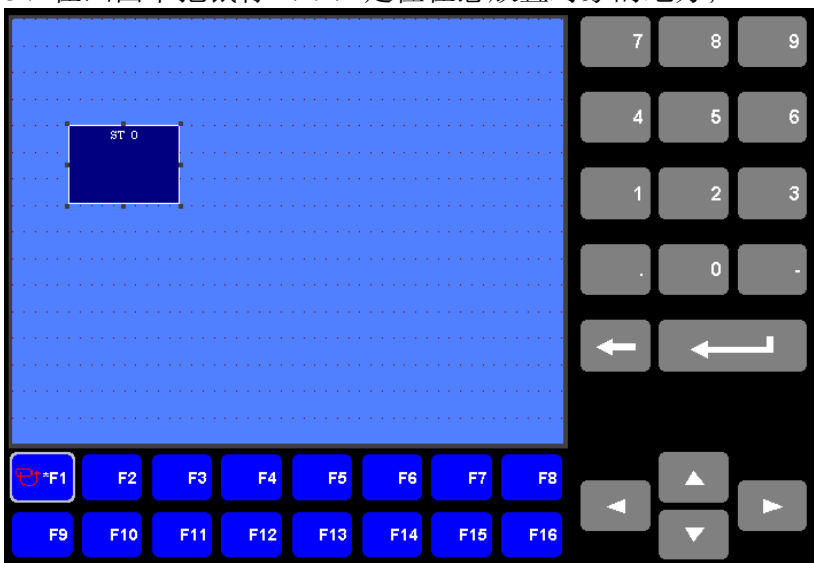
输入节点地址，选择波特率，I/O 输入输出字数量和我们作 DeviceNet 网络规划时相同。

2、生成画面对象

从 Toolbox 或从 Object 选择一个对象，如：瞬动按钮

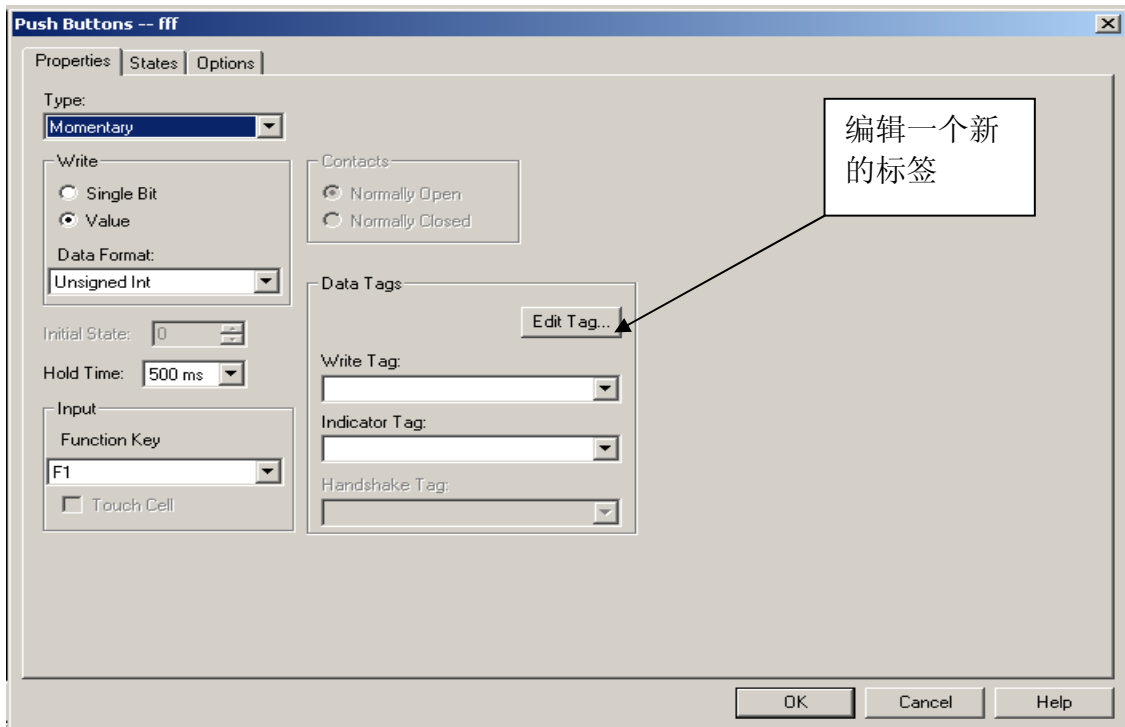


3、在画面中把鼠标（+）定位在想放置对象的地方；

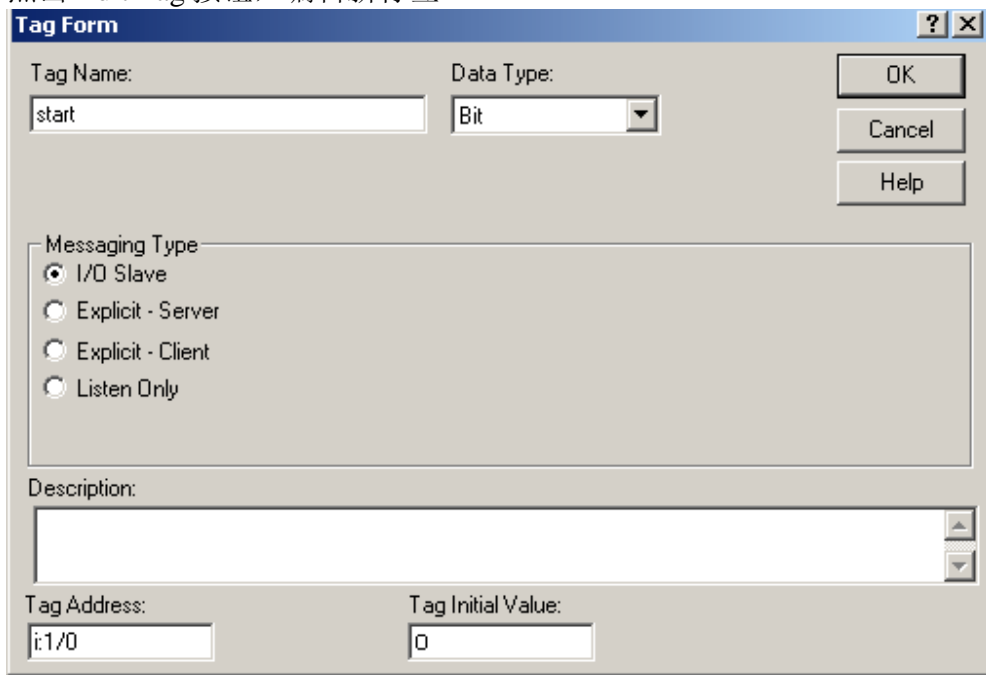


4、配置对象属性

右击添加的按钮对象，找到 object properties,出现如下画面



点击 Edit Tag 按钮，编辑新标签



或从 Tag Editor 编辑标签

	Tag Name	Data Type	Description	Address	Initial Value	Array Size	Swap Bytes	Swap Word
1	alarm	Bit		α:4/0	0	1	<input type="checkbox"/>	<input type="checkbox"/>
2	fill	Unsigned Integer		α:3	0	0	<input type="checkbox"/>	<input type="checkbox"/>
3	start	Bit		i:1/0	0	0	<input type="checkbox"/>	<input type="checkbox"/>
4	stop	Bit		i:1/1	0	0	<input type="checkbox"/>	<input type="checkbox"/>
5	timer	Signed Integer / Int		α:2	0	0	<input type="checkbox"/>	<input type="checkbox"/>

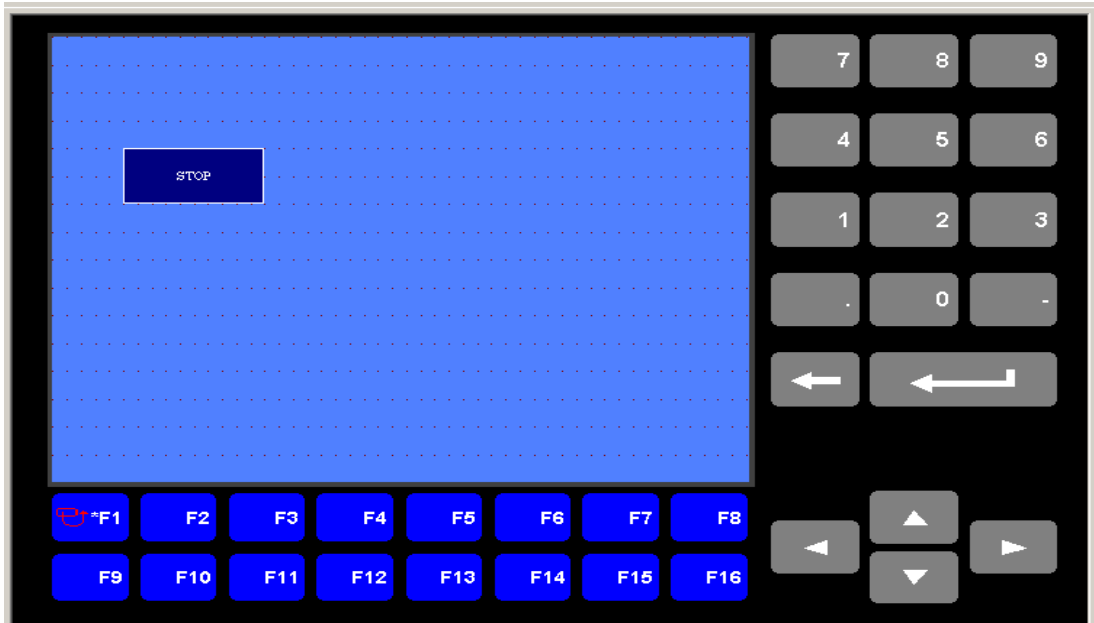
值得

注意的是：对于 PV，I/O 地址从 I:0/0 和 O:0/0 开始，而在 SLC500 中是从 I:6.1/0 和 O:6.1/0 开始的（6 是 1747-SDN 的槽号），I:6.0 和 O:6.0 背占用，为只读。

好了，对应的控制器标签就定好啦。接下来，然我们看一下 STATE 表，可以分别给 0、1 值输入信息文本，譬如，0 值为 “STOP”，1 为 “START”

	Message Text	Value	Graphic	Blink	Fill	Object Background
0	STOP	0	None	<input type="checkbox"/>	<input type="text"/>	Blue
1	START	1	None	<input type="checkbox"/>	<input type="text"/>	Blue

到此，就在画面中做好一个按钮

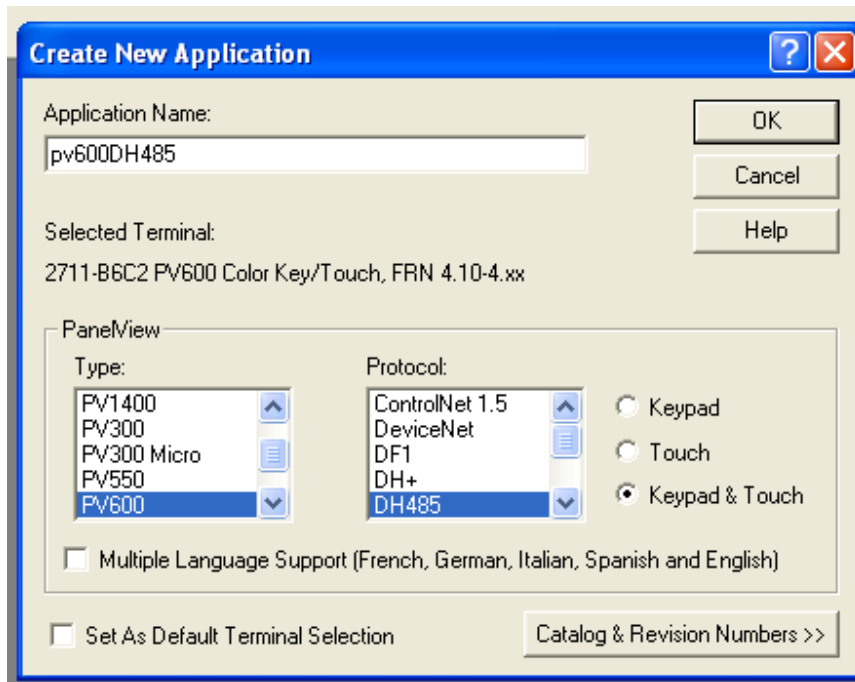


本实验在此只给出对象定义的一般步骤，其他画面对象设置雷同。
可以尝试在画面中添加如下的画面对象：按钮、数字输入输出、状态选择、画面选择器等。

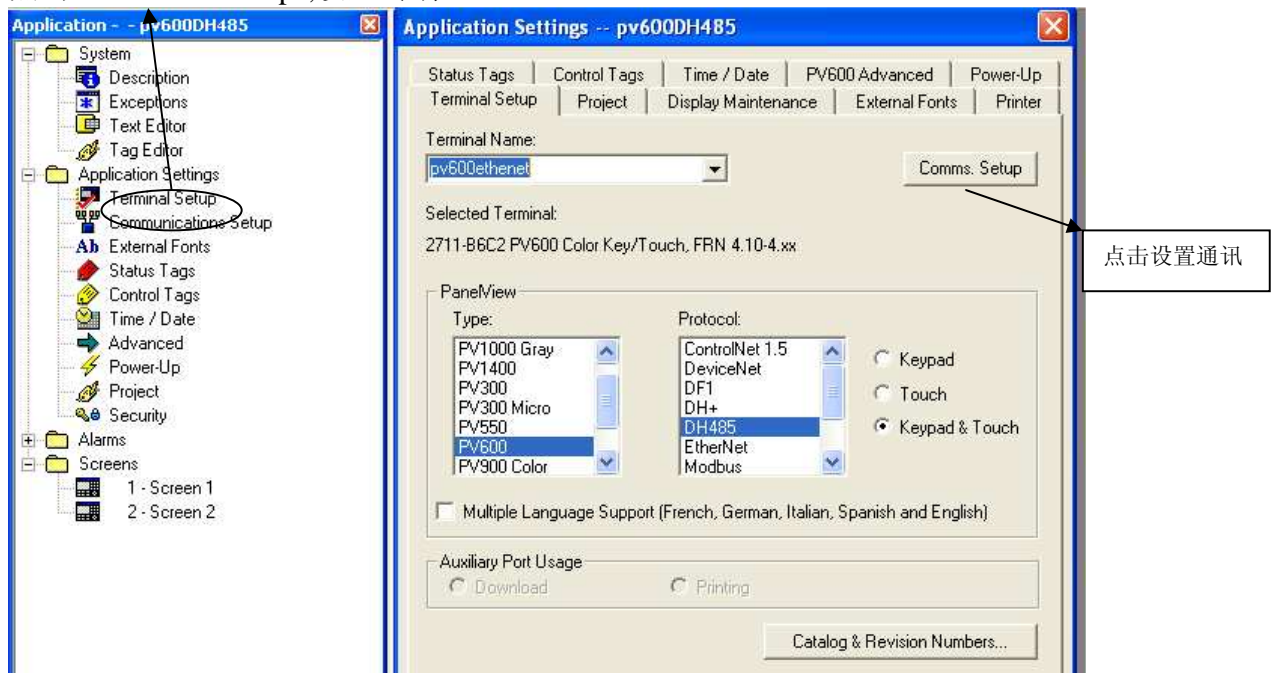
实验十 PanelView 与 SLC 通过 DH+, DH485, EtherNet 通讯建立

1、DH-485 通讯路径建立

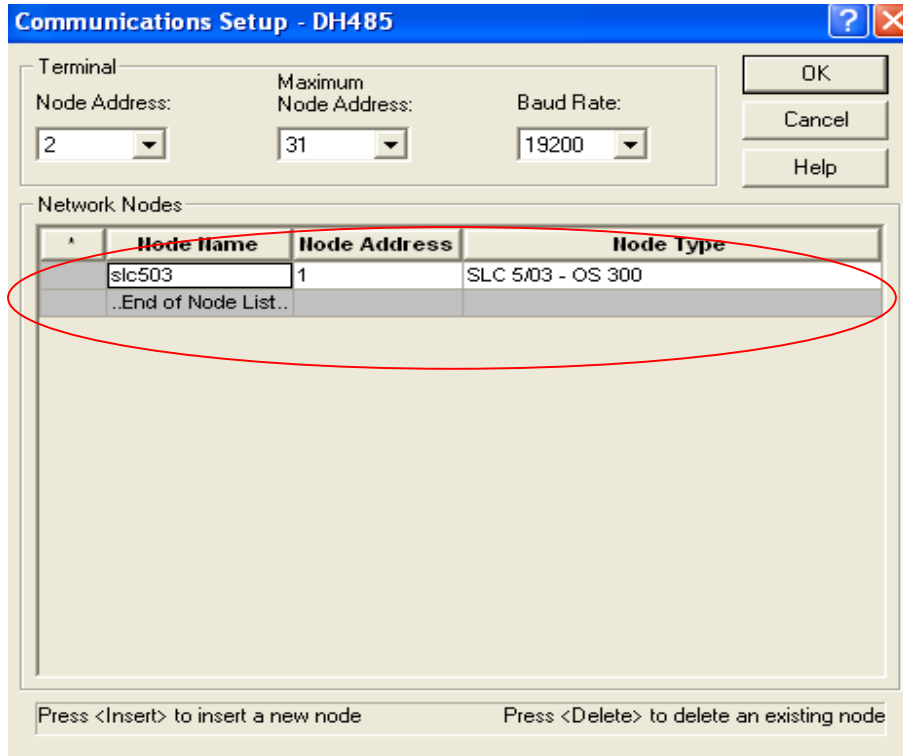
新建项目，选择操作员终端类型，点击 OK



点击“Terminal Setup”，设置终端，



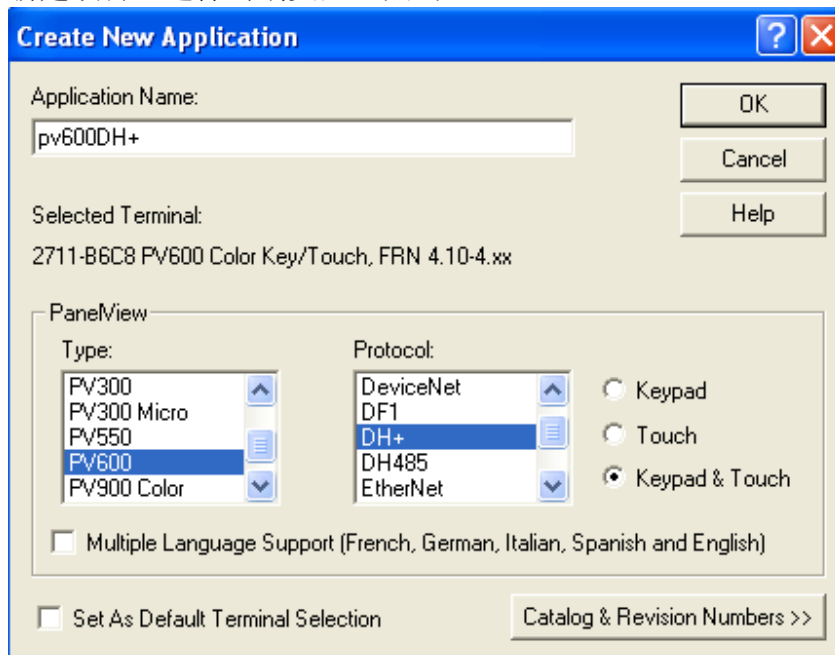
与所要通讯的 PLC 连接，默认终端地址位为 2，输入节点名及 PLC 类型、地址。



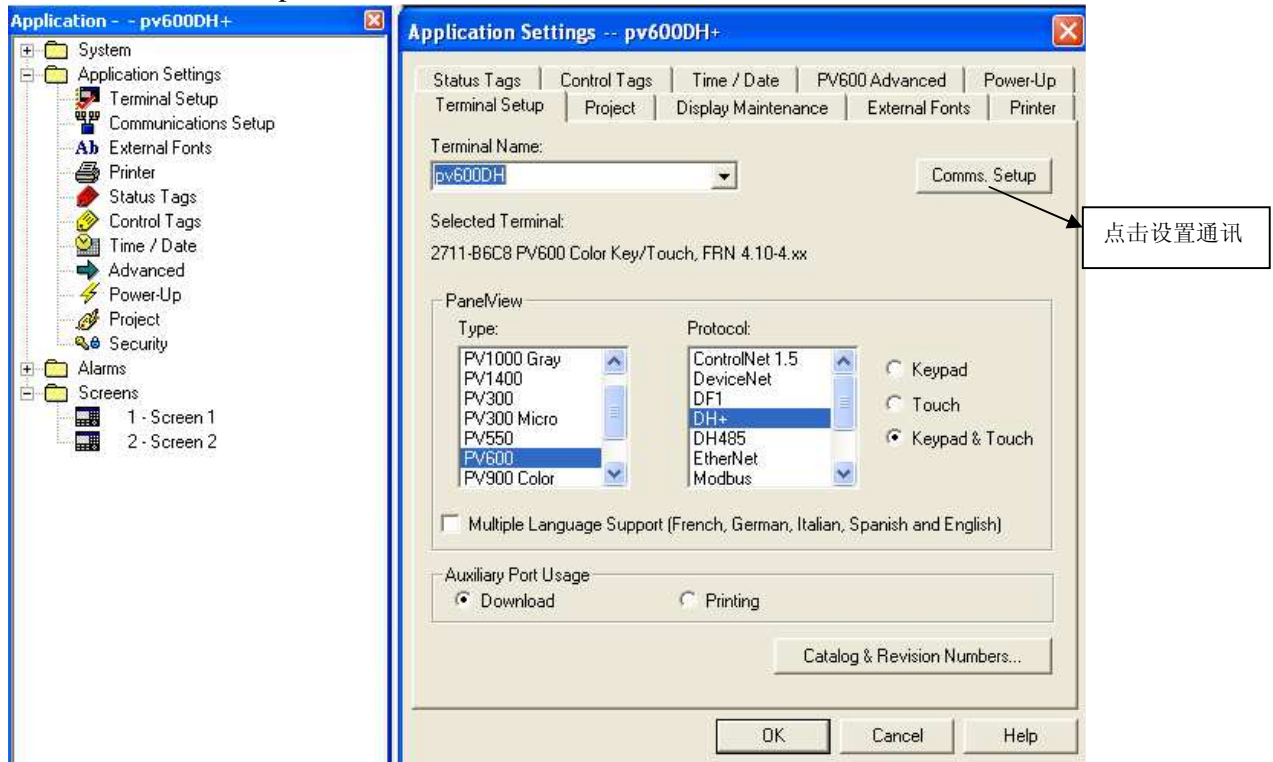
点击 OK，通讯设置完毕。

2、DH+通讯路径建立

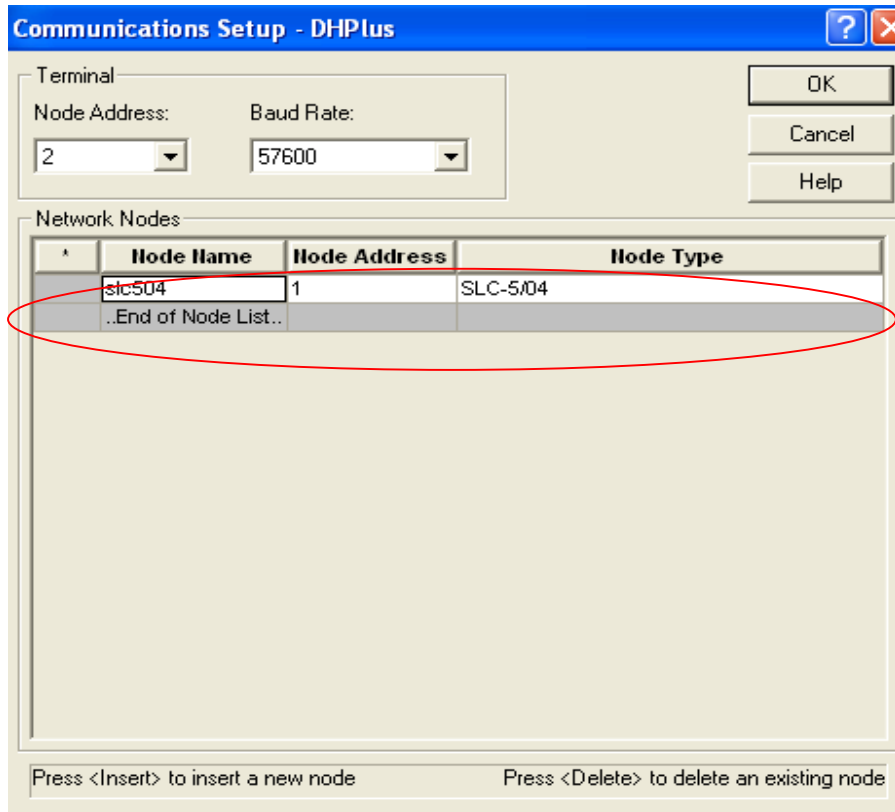
新建项目，选择终端类型，点击 OK。



点击“Terminal Setup”，设置终端，



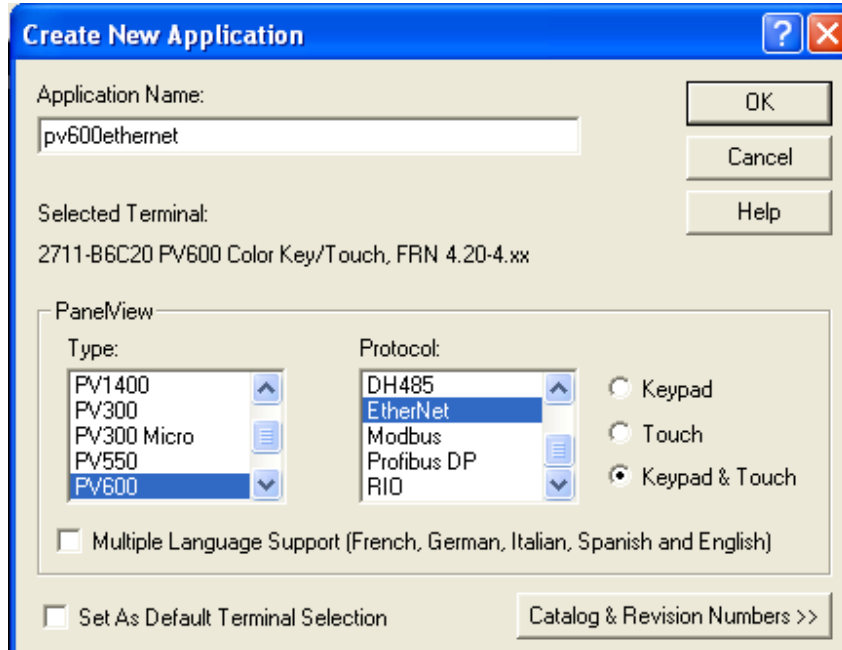
默认终端地址，设置与其通讯的 PLC 信息，



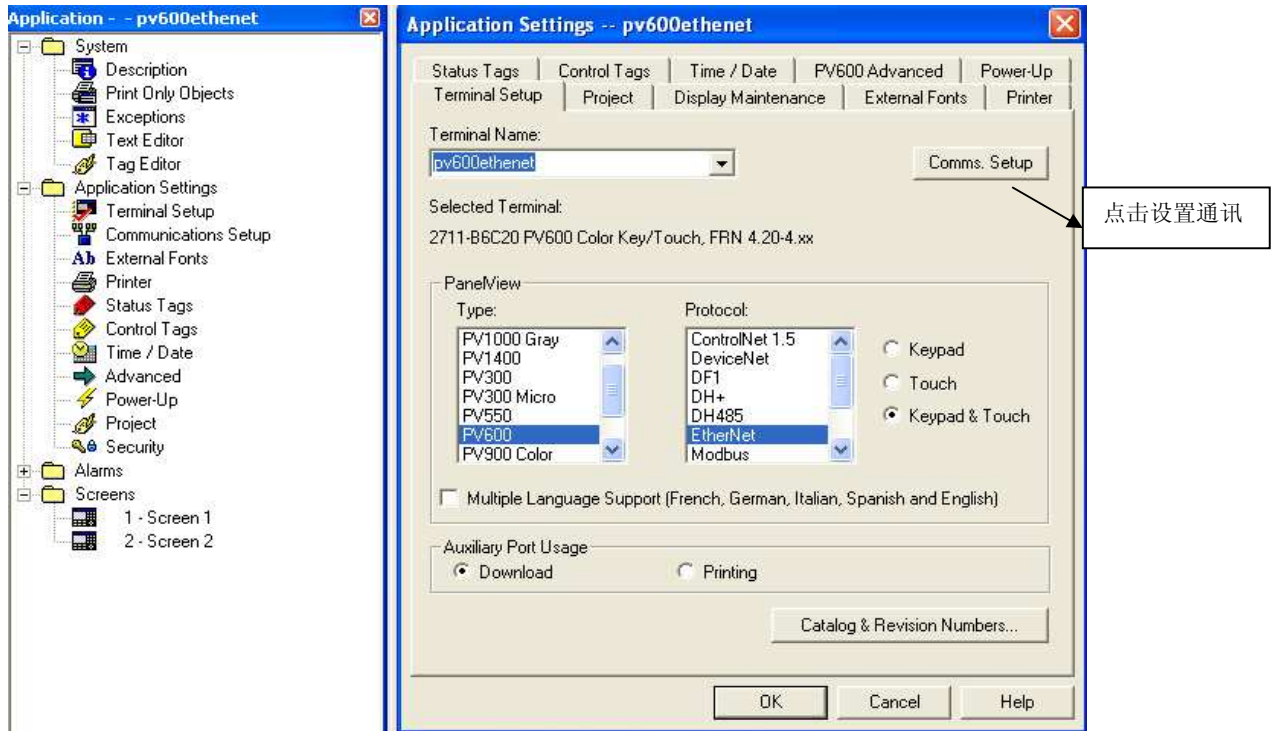
点击 OK，通讯设置完毕。

3、Ethernet 通讯路径建立

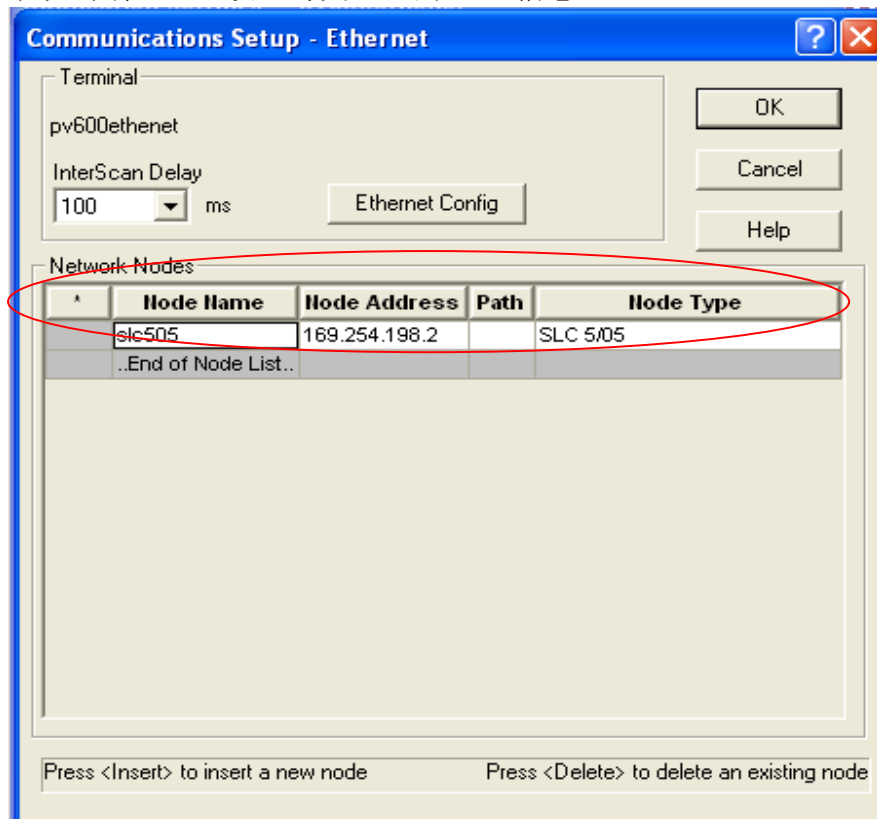
新建项目，选择终端类型，点击 OK



点击“Terminal Setup”，设置终端，



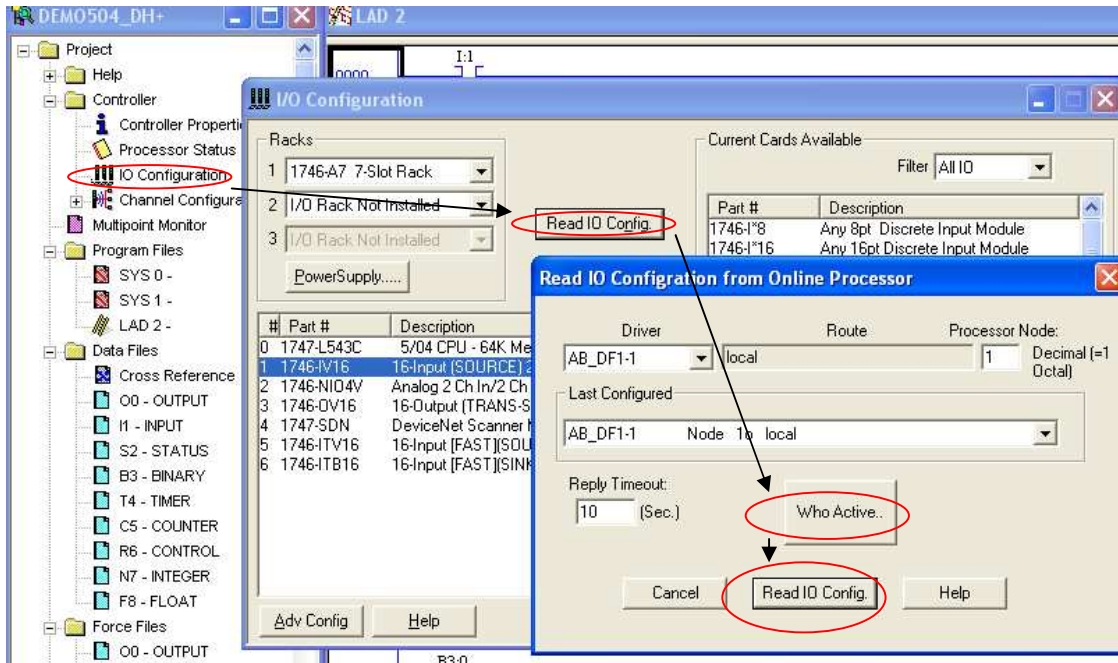
默认终端地址，设置与其通讯的 PLC 信息，



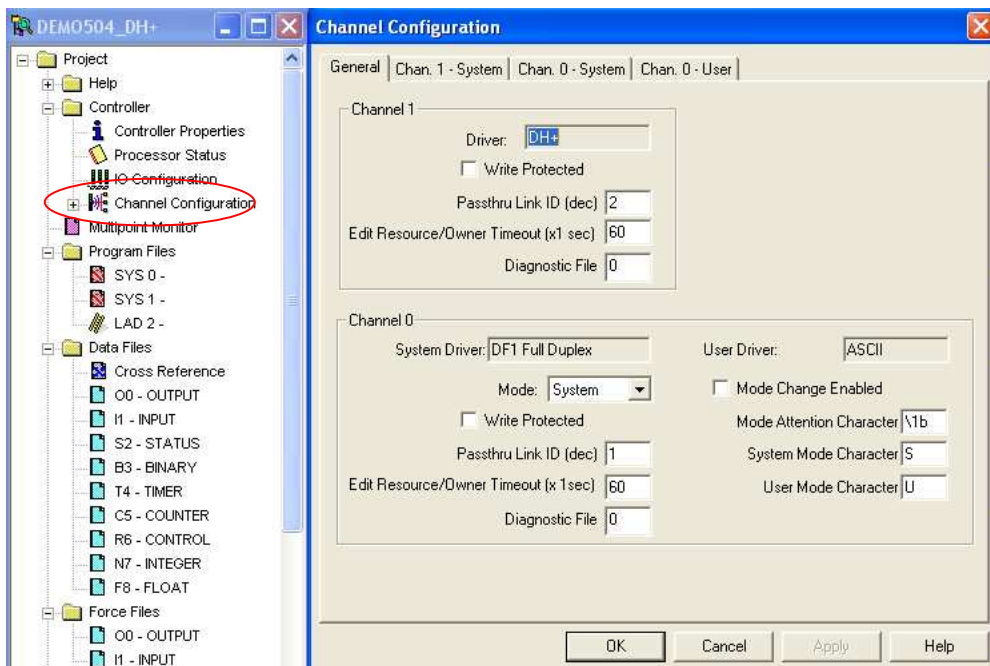
点击 OK，通讯设置完毕。

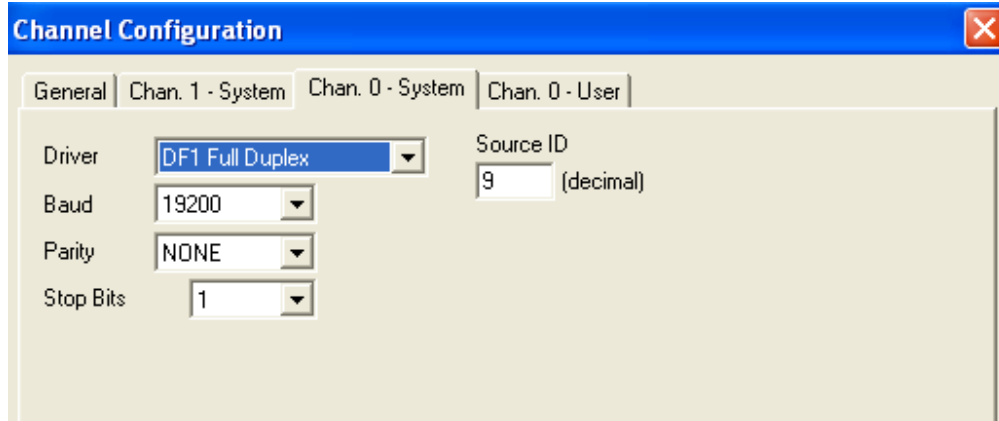
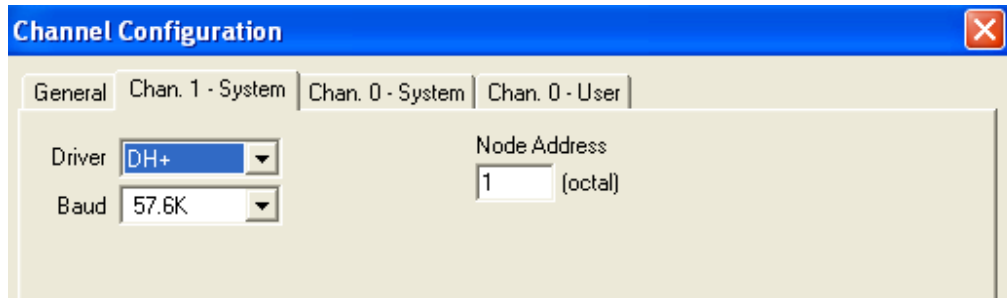
实验十一 PanelView 与 SLC 通过 DH+ 通讯程序示例

1、SLC 程序建立，如前面所讲，新建项目，在 RSlinx 内建立 SLC 下载路径，通过 RS232 串口下载程序，自动上传 I/O 组态。

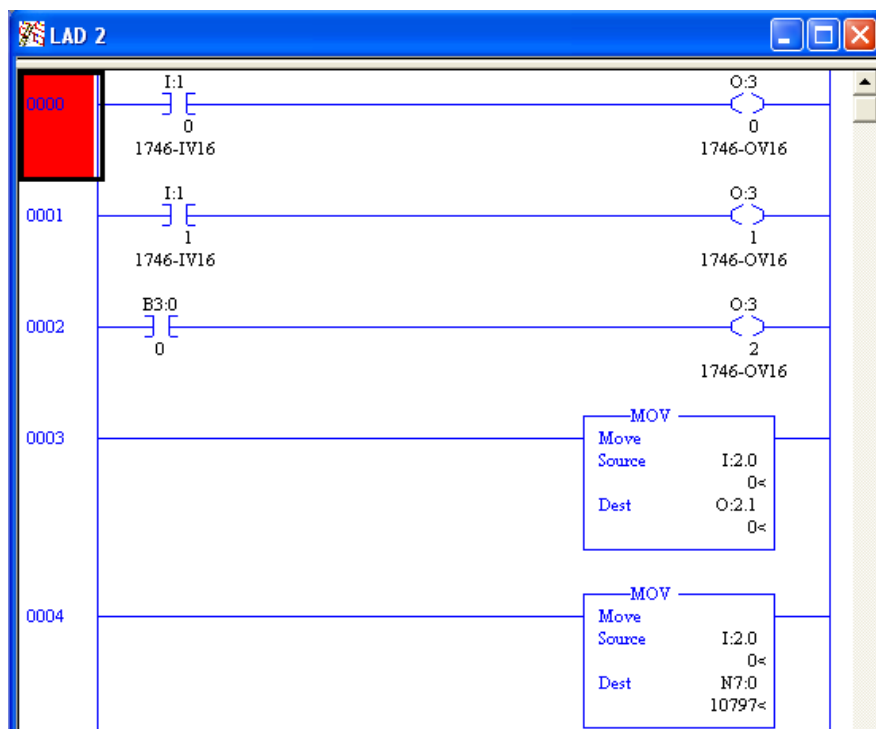


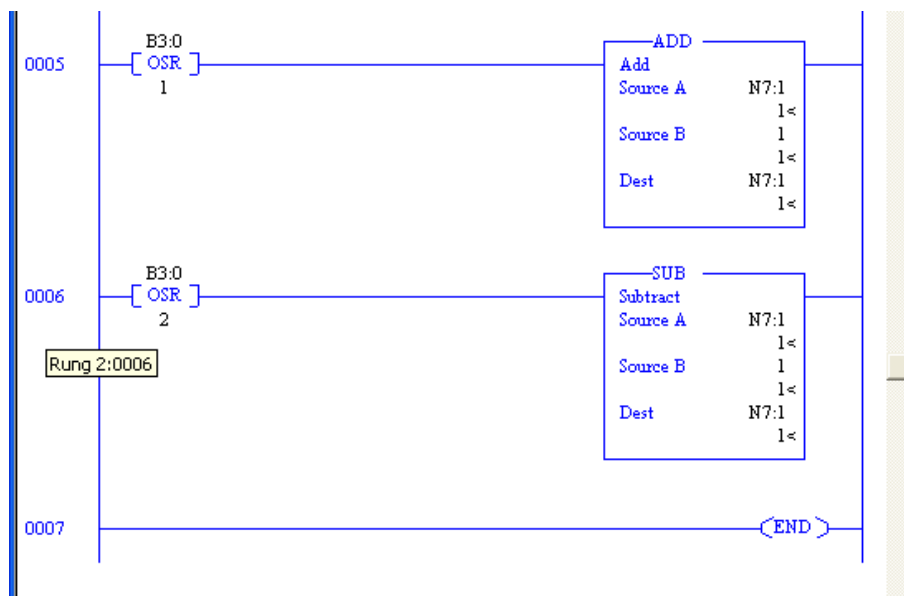
2、检查 SLC CPU 通讯口设置，利用默认值，



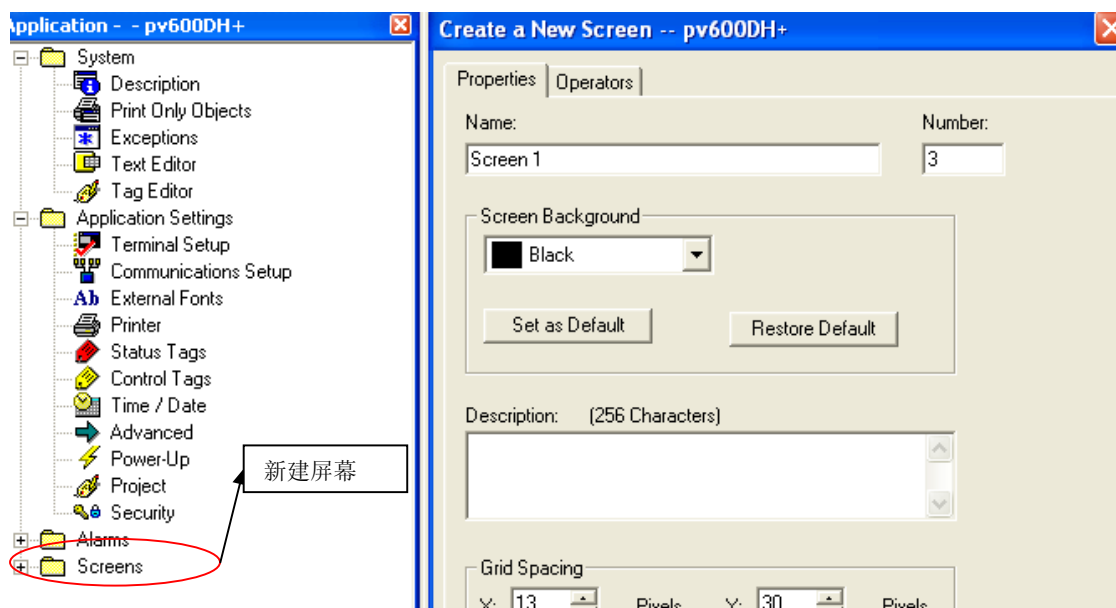


3、用梯形图编写测试程序



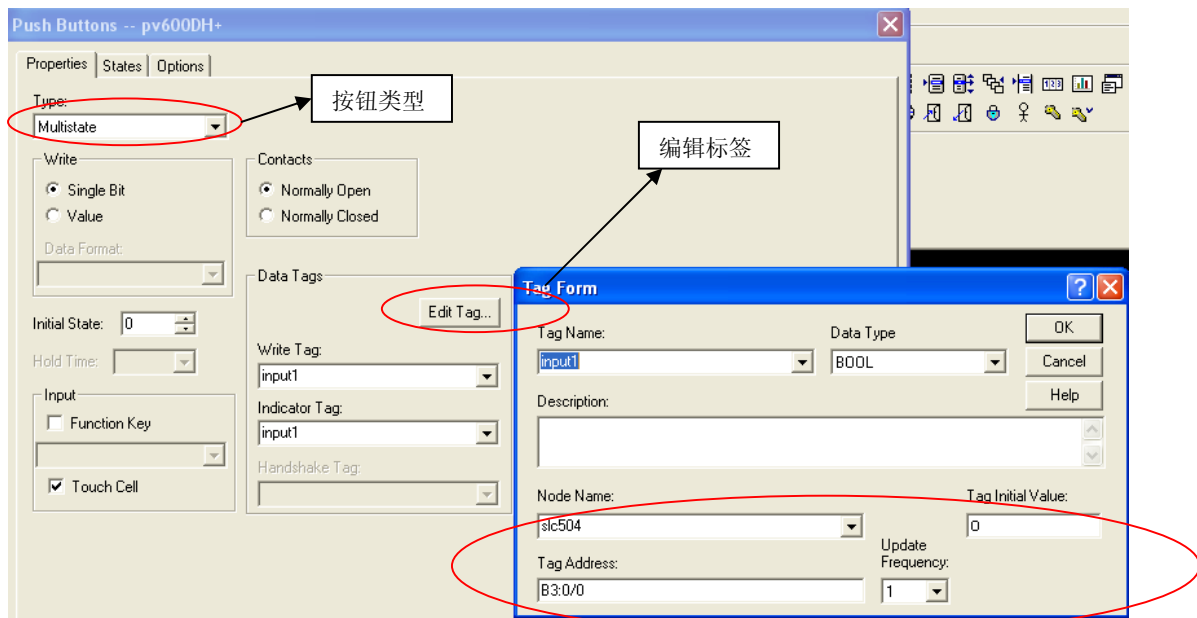
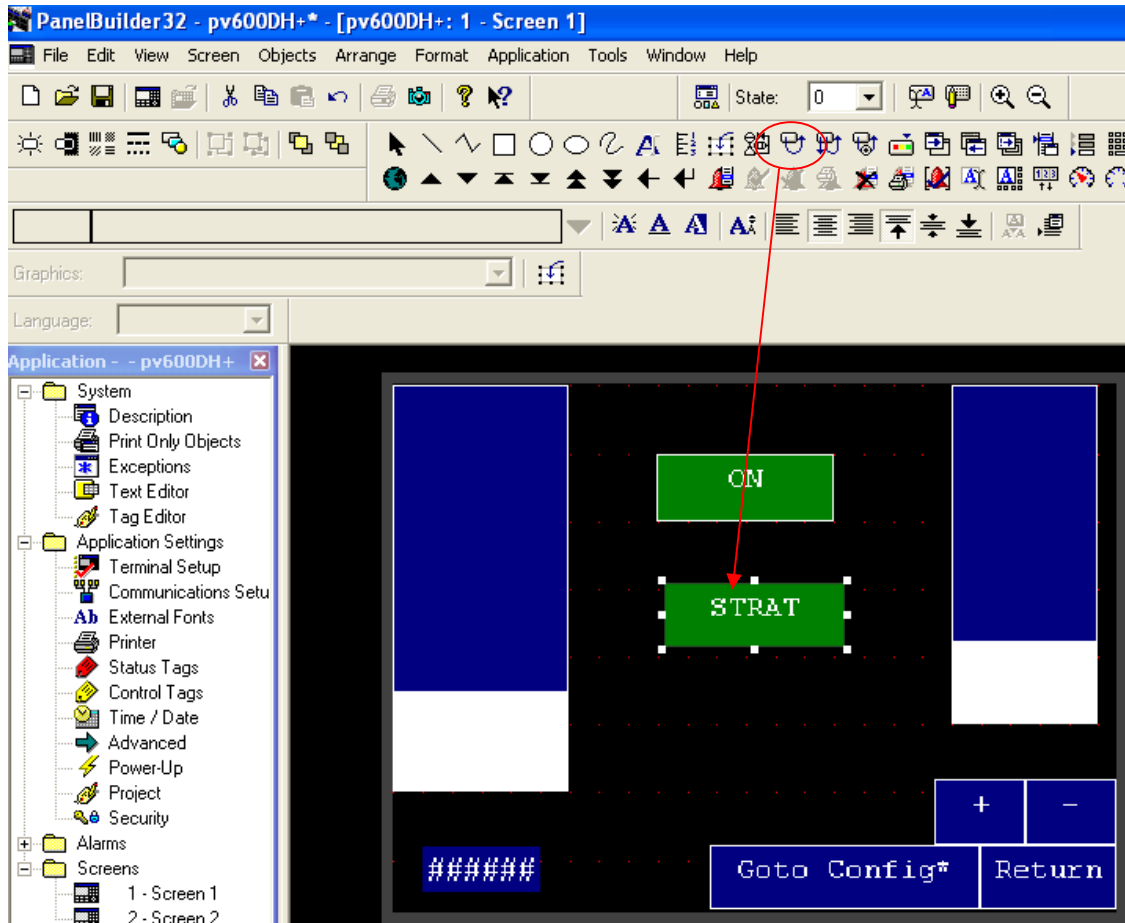


4、组态 PanelView 画面，利用前面所讲通过 DH+ 与 SLC 通讯例子建立新项目，组态通讯路径。建立新画面。



5、在画面上添加按钮、指示灯、数值显示、棒图显示等功能画面。

1) 添加按钮，双击按钮编译按钮功能，

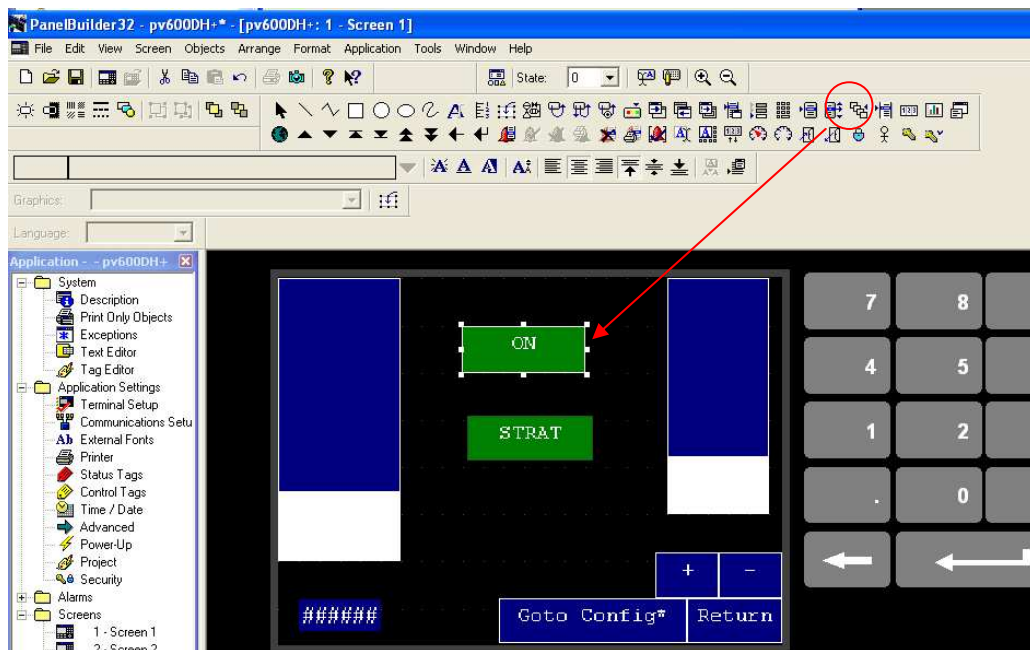


编辑状态

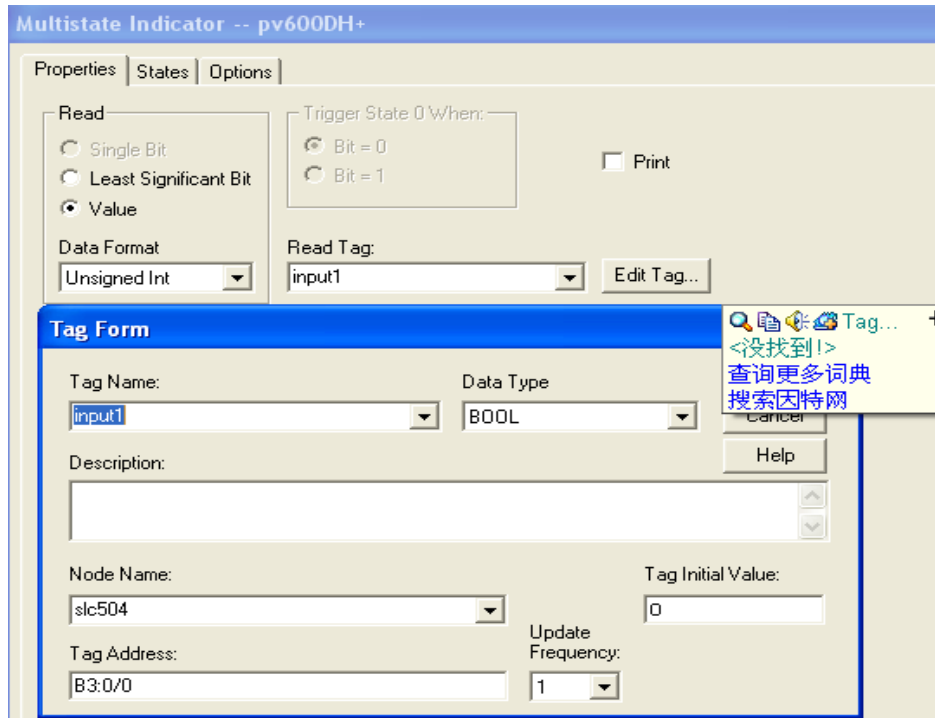
	Message Text	Graphic	Blink	Fill	Object Background	Object Fo
0	STRAT	None	<input type="checkbox"/>		Green	Forest Green
1	STOP	None	<input type="checkbox"/>		Red	Red
E	Error	None	<input type="checkbox"/>		Blue	White

点击 OK 按钮制作完毕。

2) 添加指示灯



双击指示器编译属性，编辑标签变量



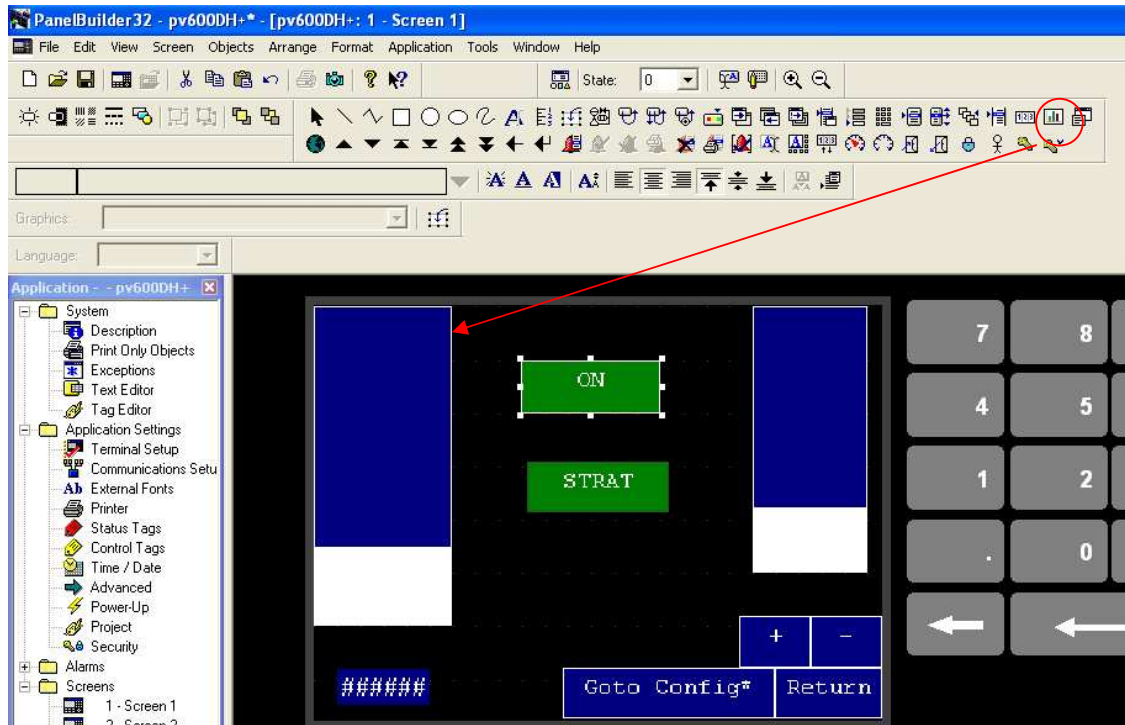
Multistate Indicator -- pv600DH+

Properties States Options

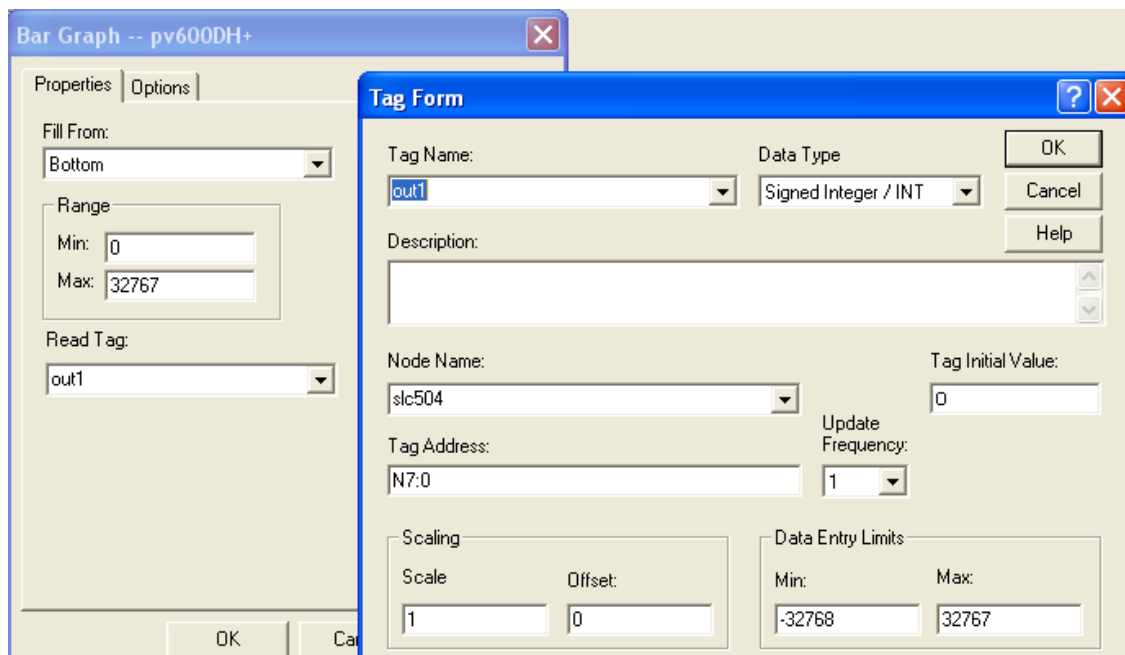
	Message Text	Value	Graphic	Blink	Fill	Object Background	
0	ON	1	None	<input type="checkbox"/>		Green	<input type="checkbox"/> Wt
1	OFF	0	None	<input type="checkbox"/>		Red	<input type="checkbox"/> Wt
2	ST 2	2	None	<input type="checkbox"/>		Blue	<input type="checkbox"/> Wt
3	ST 3	3	None	<input type="checkbox"/>		Blue	<input type="checkbox"/> Wt
E	Error		None	<input type="checkbox"/>		Blue	<input type="checkbox"/> Wt

指示器设置完毕

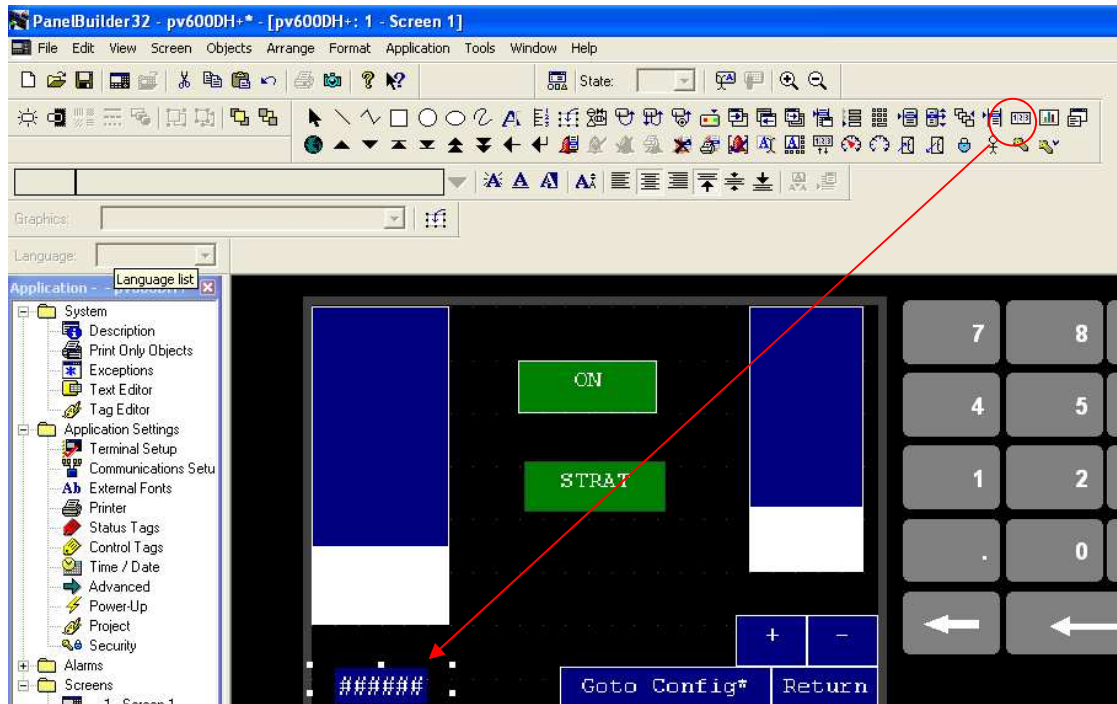
3) 添加显示棒图



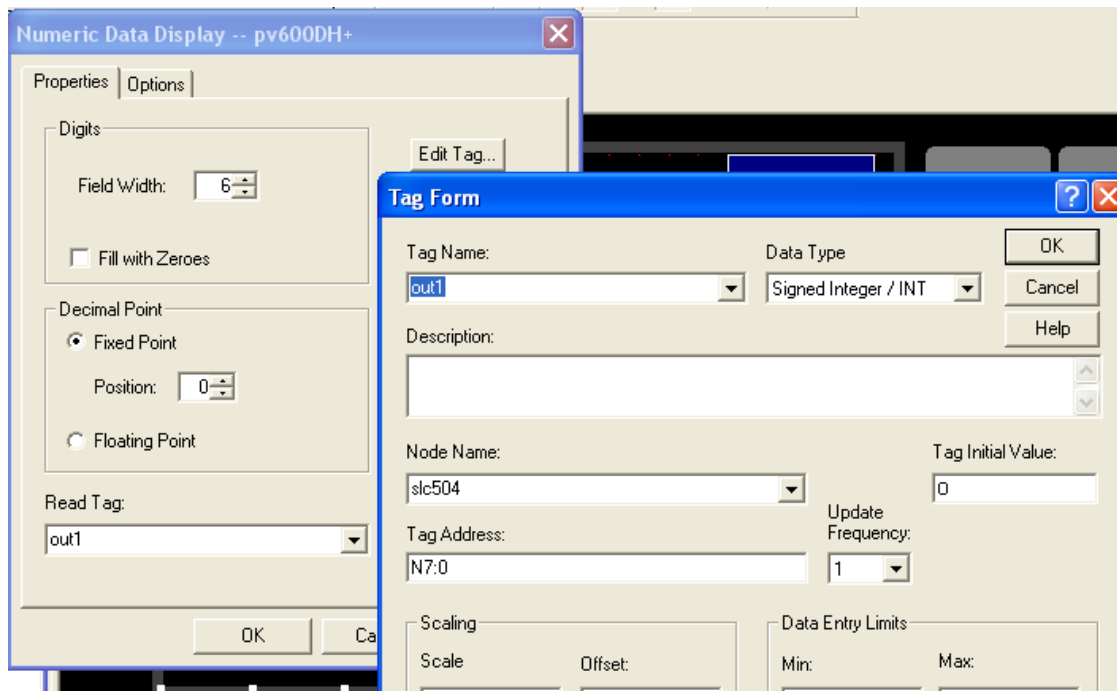
双击编译属性



4) 添加输出数值显示



双击编译属性



数值显示编译成功。

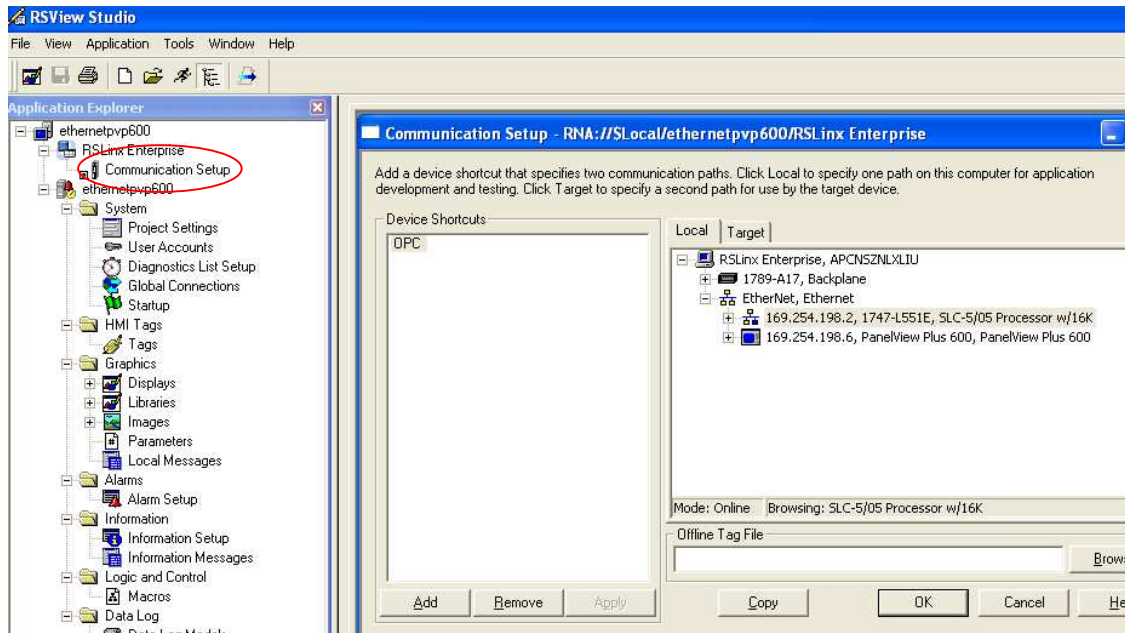
当然，您可以添加其它功能，测试程序，注意标签变量来源、类型匹配。

不要忘记添加配置 Go to configuration 按钮。

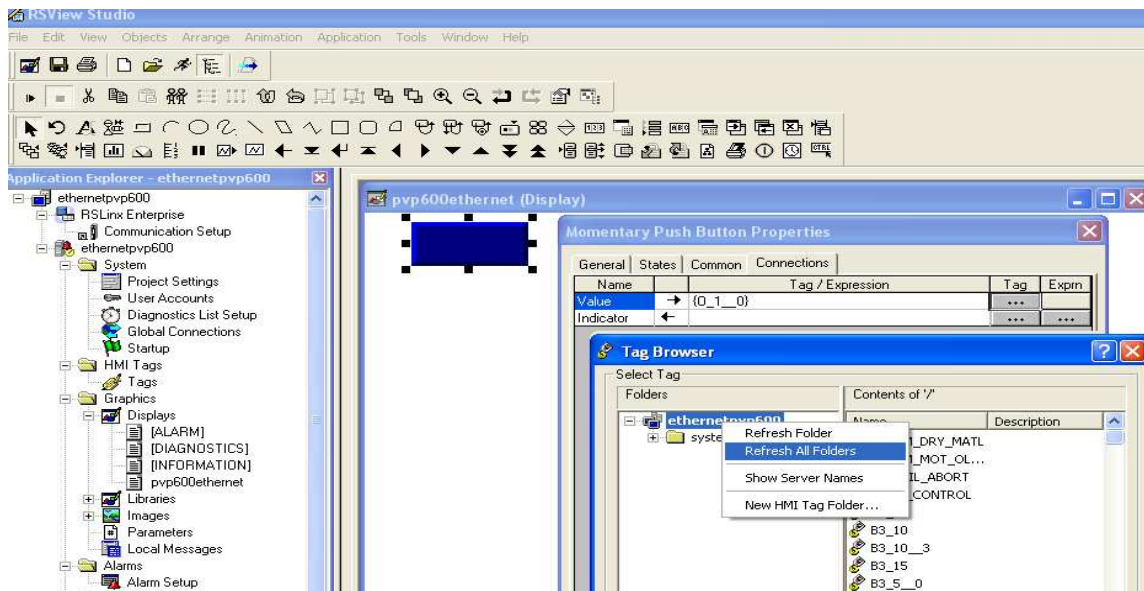
实验十二 SLC5/05 与 PVplus 利用 EtherNet 通讯

1、RSlinx 设置 Ethernet，组态 IP 地址

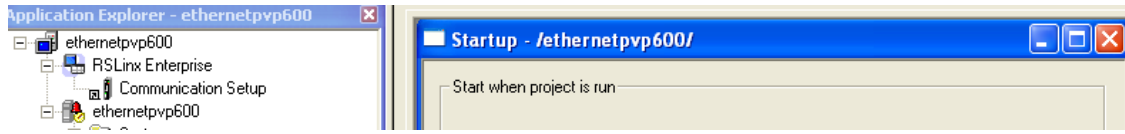
2、打开 RSview Studio 软件，“ME”建立新项目，配置通讯路径。



3、建立新画面，编译标签变量，右键点击“Tag”“Refresh All Folders”刷新变量，找到要连接的变量，组态 OK。



4、设置启动画面



ERROR: ioerror
OFFENDING COMMAND: image

STACK: