Value & Technology

GC/EA7E_AIP 系列触摸屏

（画面编辑软件 SCREEN CREATOR 5）

入 门 手 册

（第二版修订 1）

光洋电子(无锡)有限公司

前　言

感谢您选用光洋电子 GC/EA7E_AIP 系列工业触摸式显示器。

GC 系列触摸屏为早期 4：3 普通液晶屏系列工业触摸屏；EA7E_AIP 系列为新型 16： 9 宽屏系列工业触摸屏， 2 系列产品合用一个画面编辑软件—Screen Creator 5。该软件从Ｖ 2 .25 版本开始支持 EA7E-AIP 系列产品，以前版本的软件仅支持 GC 系列产品。

本手册以及 CG 系列的用户手册均同时适用于 CG 及 EA7E_AIP 系列产品。

我们致力于使我们的资料正确完整，但也因为我们的产品在不断更新和改进，所以我们不可能保证资料完全最新。并且，我们对您使用本产品作如下声明：

1） 我们有权在未经用户允许的情况下对本手册的任何部分进行修改。

2） 我们热忱欢迎用户对本手册中错误和不当之处提出修改意见，对您表示感谢！

3） 光洋电子对正确和不正确使用本手册及 Screen Creator 5 软件所产生的一切直接和间接后果不承担任何法律和经济责任！

4） 在使用本手册和 GC 或 EA7E_AIP 产品时有任何疑问可与本公司当地负责人联系，或直接与我们联系。我们的联系方式是：

地址：江苏省无锡市滨湖区建筑西路 599 号 1 栋 21 层
光洋电子（无锡）有限公司
联系电话：0510－85167888－2075/2055
传真：0510－85161393

GC/EA7E_AIP 专用名词一览：

OIP = Operator's Interface Panel	**触摸屏**
project = system	**工程**
screen	**画面**
part ＝ component	**部品**
control ＝ primitive	**控件**
Texture = a collection of figures	**构件**
Text	**文本**
Device	**设备**
property = setting = attribute	**属性**
Figure	**图形**
pattern	**图案**

目　录

第一章．概述…… ..1

第二章．画面（Screen）……5

第三章．部品（Part） ……9

第四章．基本操作 ……12

第五章．Screen Creator 5 操作……...14

第六章．简单作画 ……17

6．1　开关的设置 ……18

6．2　指示灯的设置 ……22

6．3　数据显示器的设置 ……26

6．4　设置画面选择部品 ……30

6．5　画面保存 ……31

第七章．创建方便画面 ……32

7．1　数据输入画面 ……33

7．2　通过 PLC 内部数据进行画面切换 ……36

7．3　报警显示 ……39

第八章．画面数据下载 ……42

第九章．画面测试的准备 ……45

第十章．画面测试……48

第十一章．K—Basic 控制程序基本 ……50

第一章．概述

１）ＯＩＰ介绍

・ＯＩＰ是显示器和触摸开关一体型的可编程序显示面板，简称触摸屏。

与一般的操作面板和控制面板不同，OIP 具有如下功能：

- ●紧凑型设计风格
- ●对不同规格具有灵活的适应性
- ●连接简单，用一根电缆与 PLC 相连
- ●作图方便，自带丰富的部品配置
- ●基于Ｋ－Ｂａｓｉｃ的灵活的用户接口功能
- ●多通道的串口通讯功能

由于具有以上功能，OIP 成为新一代的智能型操作面板。

ＯＩＰ（EA7EAIPC 系列）的画面是用专用作图工具「SCREEN CREATOR 5」做成，通过个人电脑的 USB 通讯口,利用专用下载电缆（自制)下载到 OIP 的通道 1(USB 通道)。SCREEN CREATOR 5 是一种先进的专用作图工具，运行在 Windows9X/NT/XP 环境下，用户只需在各种标准部品库中选择配置各种部品就能产生多种画面。

・ＯＩＰ支持多种通信方式。

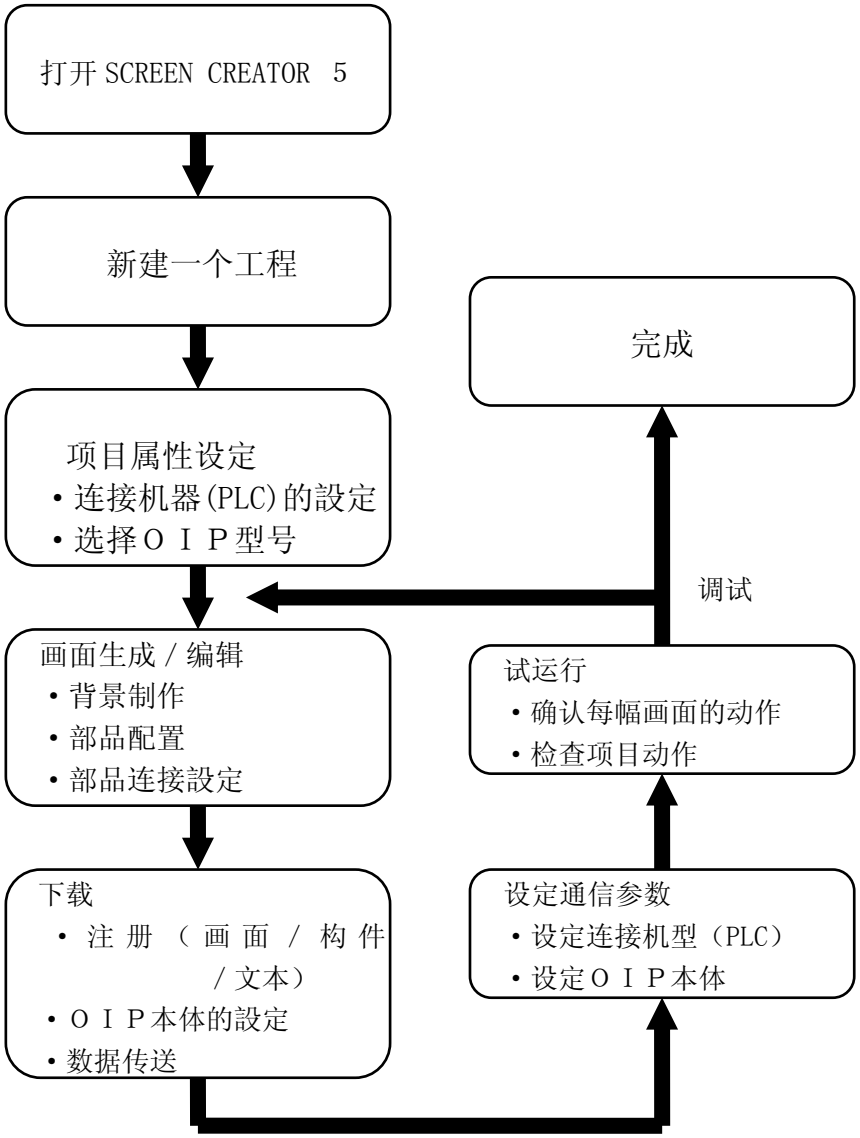
支持同时使用三个通道进行通信，因此也支持同时与个人电脑和 PLC 进行通信。

- ●直接访问通信（对象：ＰＬＣ等）
  如果 OIP 的串行口与ＰＬＣ的ＣＰＵ模块或计算机通信模块相连，那么 OIP 能够读写 PLC 的 WORD/BIT 功能存储器。用户不必要对 PLC 的通信进行深入的了解。
- ●存储器连结通信（对象：个人电脑等）（Memory link 方式）
  个人电脑传送数据给 OIP 的存储器表，OIP 根据自身存储器表内容显示。用户能够在屏幕上访问存储器表并使用各种部件，就像与 PLC 通信时一样。
- ●上位机命令通信（对象：个人电脑等）
  个人电脑采用专用通信命令和 OIP 进行通信，用户传送数据就像个人电脑之间的串行通信。
- ●无协议通信（对象：MODEM、专用控制器等）
  OIP 可与通信协议固定不变的设备进行通信，例如 MODEM、温控器等，这个功能使系统配置适应性更广。

注：ＯＩＰ为 EA7EAIPC 系列工业触摸式显示器的另一种名称。

2）制作画面的流程

SCREEN CREATOR 5 按以下次序制作画面。
操作过程将在下面内容中详细描述。

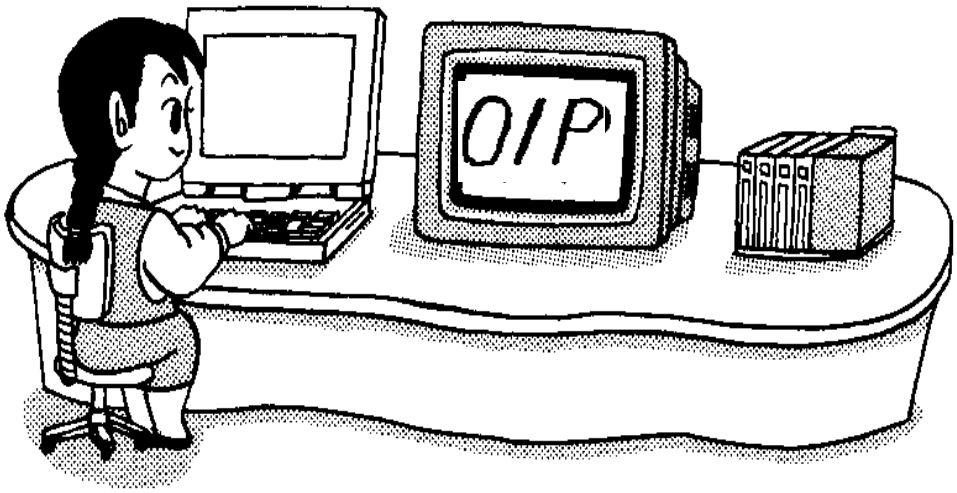


３）软件安装

▪开发环境

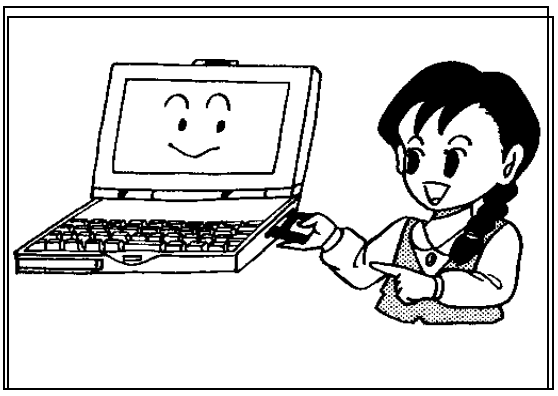
为使用「SCREEN CREATOR 5」，需要一台运行 Windows9x/NT/XP 操作系统的个人电脑，推荐的硬件配置如下：

①２５６色，ＶＧＡ显示器
②内存３２ＭＢ以上
③８０ＭＢ以上的硬盘空间



▪开始安装

首先把 SCREEN CREATOR 5 的光盘放入个人电脑的光驱中。



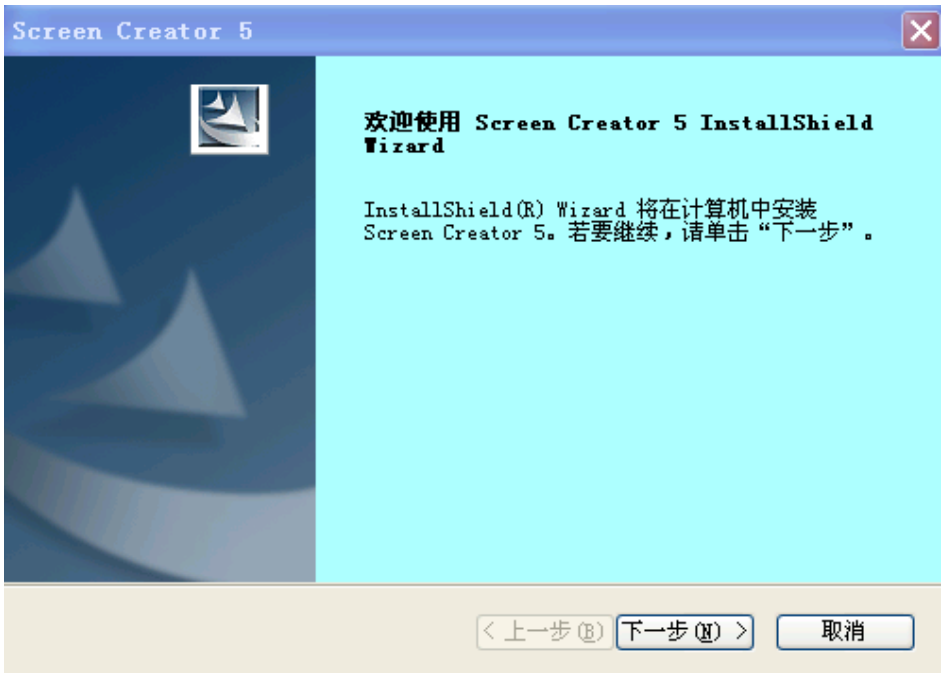
运行 Windows9x/NT 的「explorer」或指定文件名执行光盘的 SETUP.EXE 文件。

Setup.exe 文件在 setup 目录下 Disk1 子目录中。

例）将 CD 放入电脑光驱中

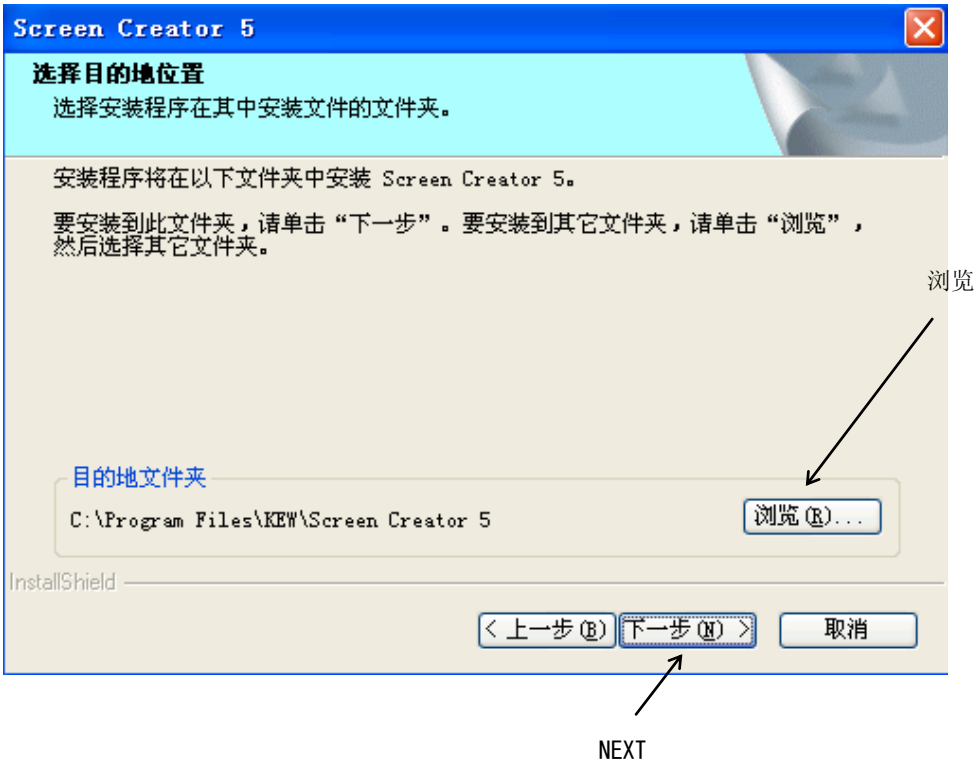
执行如下指定可执行文件：

```
D:Setup\Disk1\Setup.exe
```



- 开始设定安装对象

执行 SETUP.EXE 后将会出现程序安装窗口,按 NEXT 按钮打开下一个窗口。按 BROWSE 按钮可以改变 SCREEN CREATOR 5 的安装目录。



在此窗口下点击 NEXT 按钮就开始安装。
也可点击“浏览”改变安装路径。

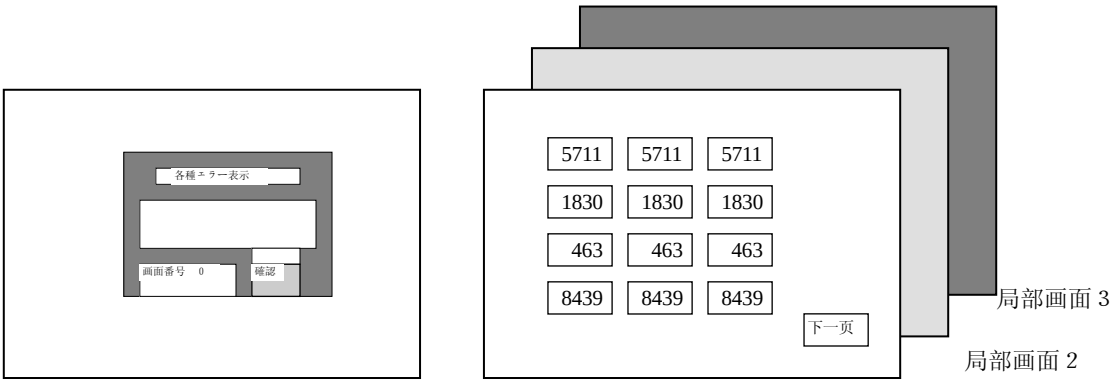
第二章. 画面

1）画面构成

画面是构成工程的组成单位，画面由部品、构件、背景图形、画面控制程序等组成。
ＯＩＰ画面由两种类型的画面组成，它们是：

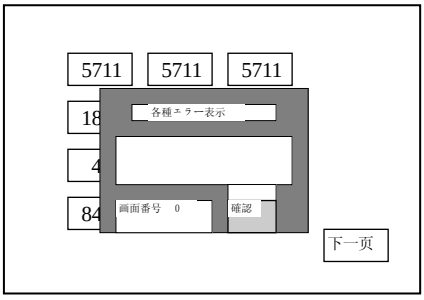
- 全局画面（每个项目有且仅有一幅）
- 局部画面

实际显示的画面是以上两种的组合。



全局画面　　局部画面 1

ＯＩＰ运行时全局画面和某局部画面重叠显示
例如：当局部画面 1 在 0IP 上显示时，首先显示的是由全局画面和局部画面 1 叠加成的画面（见下图）。用户可以通过全局画面的透明玻璃板效果看到局部画面 1。



全局画面＋局部画面的合成显示

● 全局画面 ..... 每个工程都有一个全局画面。当用户需要创建一个新的工程（项目）时， SCREEN CREATOR 5 会自动生成一幅全局画面。一般在全局画面上安排了如下部件：

①常用的通信部品（如报警显示）

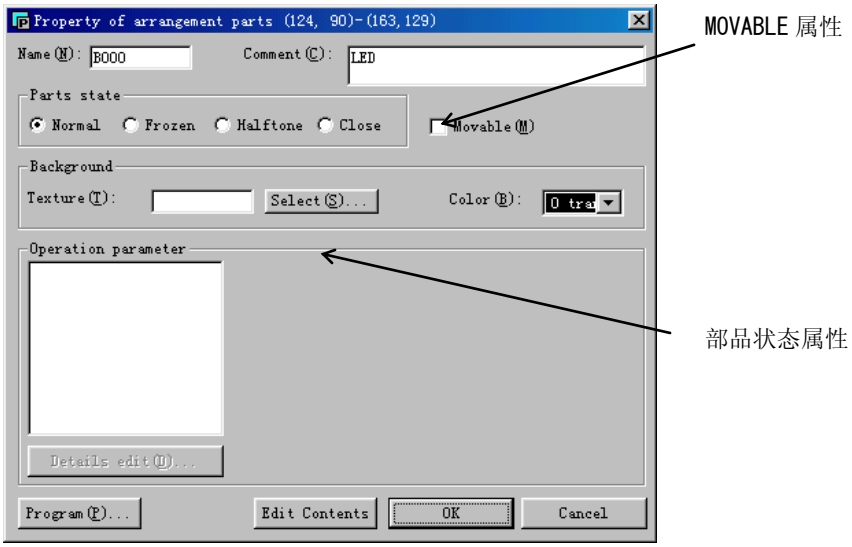
②常用的显示部品（如时钟）

③每个局部画面都调用的通用部品（数字键盘等）

● 局部画面 ..... 局部画面由用户创建并可以根据所连结的外部设备来选择，如 PLC 内部寄存器、内部继电器或触摸开关。

用户可以选择在屏幕上显示的每个部品的状态：NORMAL(开)、冻结（FROSEN)、HALF TONE（半色调）和关闭（CLOSE)。部品处于 NORMAL（开）状态时可以看见，处于关闭状态的部品是看不见的，当 OIP 运行时，只有具备“MOVABLE”属性的部件能够改变打开（OPEN）与关闭(CLOSE)的状态。

例如数字键盘部品的属性。数字键盘由其它部件或外部设备调用时，就从关闭变成 NORMAL（开）状态，因此该键盘可以在屏幕上预置为关闭状态。



要将一个部品预置为关闭状态，就必须选择该部品的”MOVABLE”属性。

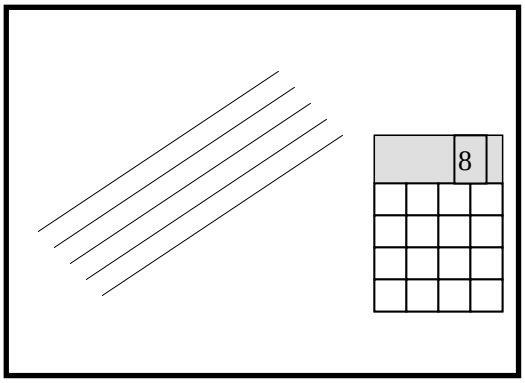
在屏幕上可以将具有“MOVABLE”属性的部品叠加，但不能将不可移动的（固定）的部件重叠。

注意！

打开的部品重叠时会有优先级别。低级别部品禁止刷新和响应开关输入。

- 新打开的部品优先于以前打开的部品，全局画面上显示的部品总是优先于局部画面上的部品。

2）全局画面



不管局部画面是否变化，全局画面的部品总是显示的，因此要合理安排这些部品，不要影响局部画面的显示。

可以将部品设置为关闭(close)状态，必要时打开，只有具有“MOVABLE”属性的部品才能配置在全局画面上。

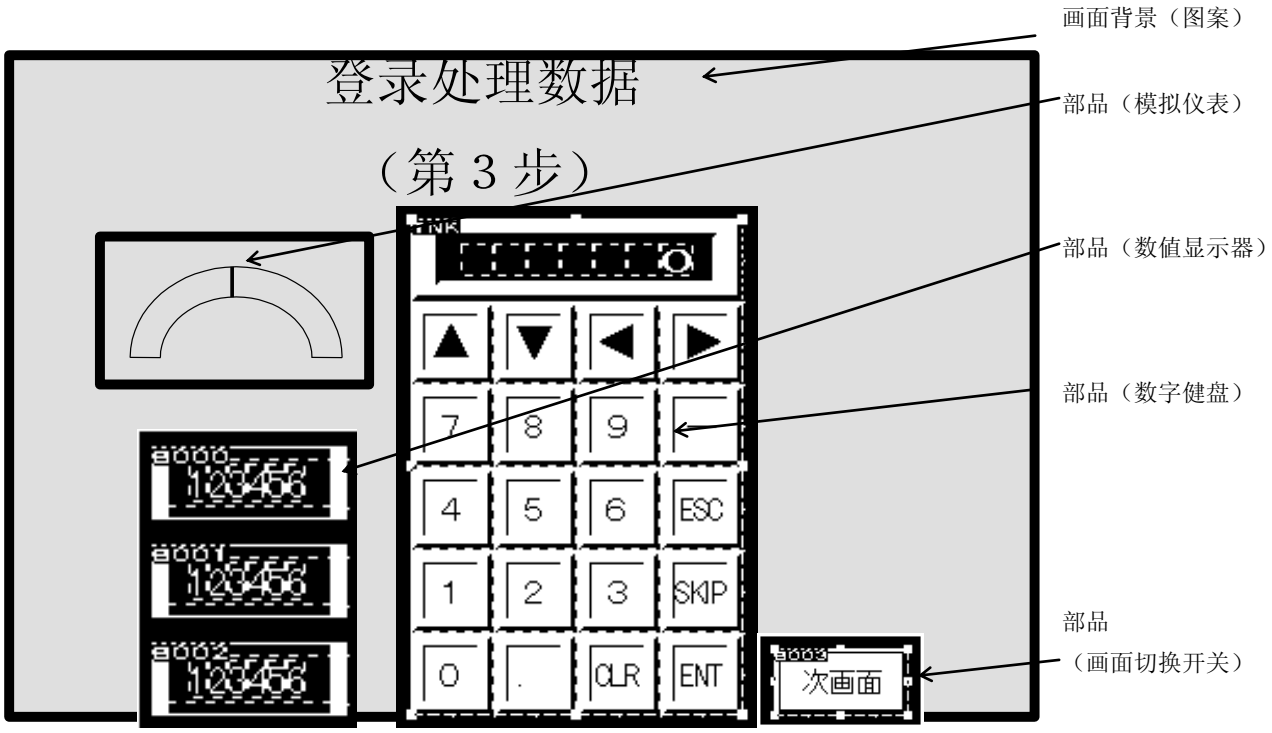
处于关闭状态时也可以安排部品，然后在必要时打开。只有属性为“Movable”的部品才能安排在全局画面上。

注意！

- 在全局画面上只能配置部品，绘制的图形只能在局部画面中使用。
- 在全局画面中放置的部品总是在屏幕上显示。
- PLC 和 OIP 之间的通讯只能在当前显示的局部画面或全局画面中被激活。因此，需要经常通信的部品可以安排在全局画面中。

3）局部画面

每个局部画面由各种背景模式的部品组成，象全局画面的部品一样，局部画面的部件也有开放和关闭两种状态。



局部画面可以由按键或 PLC 发出的命令来改变。
通过局部画面的改变可以显示各种画面，由于画面上的部品有开放（open）/关闭(close)两种状态，因此一幅画面可以发挥各种功能。
一幅画面上可配置的部品数量最多为１０２４个，但随着部品增加，OIP 负担也在增加，为了不影响程序执行，请将画面拆分，从而减少每幅画面上部品数量。

注意!

构件（TEXTURE）和部品（PART）都可以被放置在局部画面上，构件也称背景画面，总在部品的后面。如果部品被放置在构件绘制的地方，则构件被部品隐藏。 部品总是处于绘制的各种背景图形、文字之上，此时所绘制的内容作为显示部品的背景。PLC 和 OIP 之间的通讯只能在当前显示的局部画面和全局画面中被激活，尽可能连续分配功能定义号（device），这样通信效率将大为提高，OIP 运行速度会更快。

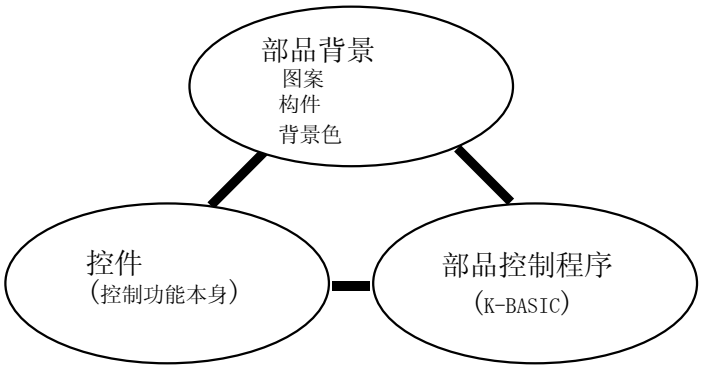
第三章．部品（Parts）

部品是配置到画面上的最小功能部件，系统提供 13 大类好几百种部品，供用户在制作画面时直接调用。普通的画面，其功能通过直接调用部品一般都能实现。

1）部品结构

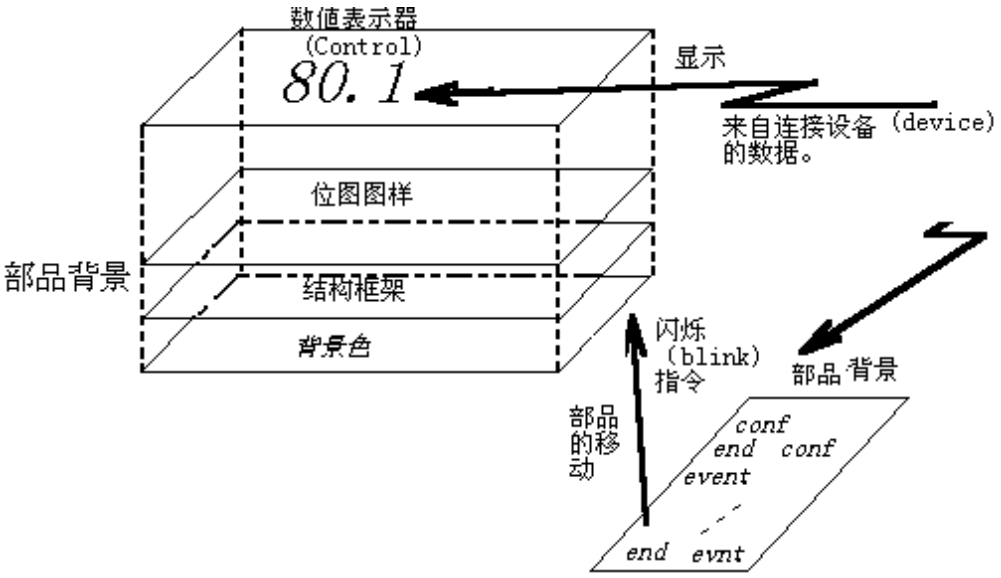
每个部品有三个元素组成：控件(Control)、部品程序和部品背景（Background)。

- 控件（Control）
  控件用于显示所连接的 PLC 功能存储器的数据以及将相关数据写入 PLC 的功能存储器中，可通过设定动作参数使部品与指定的 PLC 功能存储器相连。
- 部品程序（Program）
  部品程序的功能是获得控件功能所不能实现的部品功能，通过创建部品程序，既可减轻 PLC 的负担，又能够通过无协议通讯方法来连接那些原本不能连接的设备。部品程序又称为控制程序。
- 部品背景（Background）
  部品背景决定部品的视觉效果，由图案（绘制的各种图形)、(构件）Texture 和背景(background)色组成。用户可以利用软件提供的构件结合图案产生多种视觉效果。



每个部品包含有这三个元素。由这三个元素随意变化组成的部品可以登记到部品库中，这样就可以方便地进行调用。

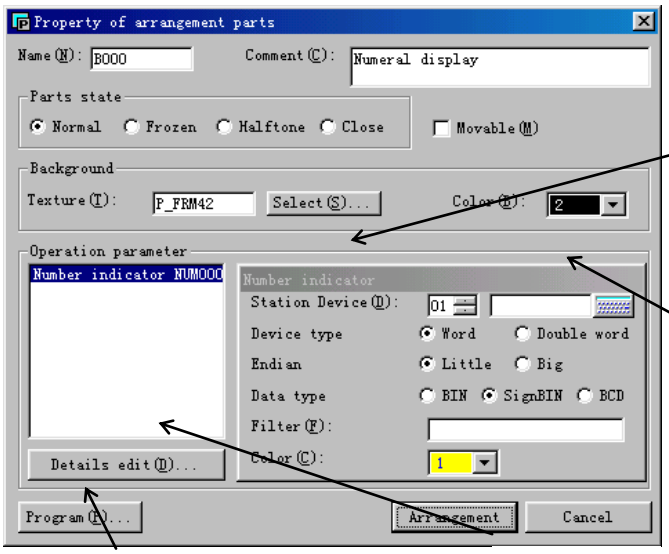
例如：可以把一个数字显示器修改成一个这样的显示器，当超过上限或低于下限时闪烁，此时控件功能用来显示所连接设备的数据，部件程序检查上下限并当超出上下限时进行闪烁。



2）控件

控件是用于构成部品的最小功能单位，在同一部品中可重叠放置多个控件。系统提供 2 大类 16 种控件。

控件可以直接与 PLC 功能存储器相连。因此它能直接显示 PLC 功能存储器内的数据，该功能存储器在部品属性窗口中设置。点击部品属性窗口的详细编辑（“Detailed edit”）按键可打开控件设置窗口，这时可以详细设置控件的属性。但有些项目不能在这设置，此时，可通过点击部品属性窗口的内容（“contents”）按键，打开部品详细修改画面进行详细的修改。如果要起用控制程序而不用控件动作参数来控制部品的动作，需要将控件动作参数属性设置为无效。



数字显示控件的动作参数设置窗口

输入所连接的 PLC 功能定义号

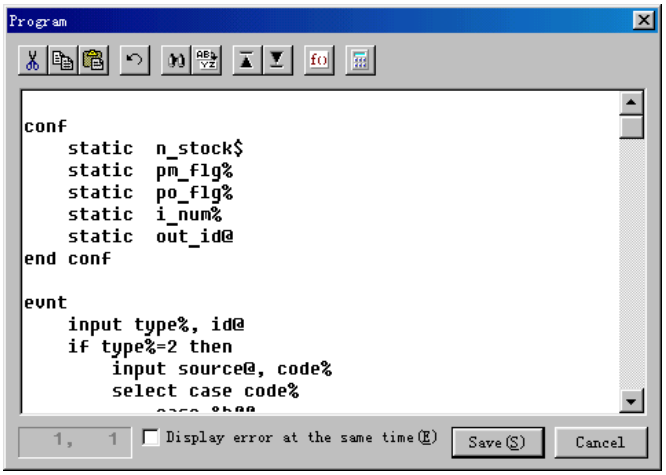
详细编辑按钮

控制程序编辑按钮

3）部品程序

部品程序中可以有大量的显示控制，如打开和关闭部件，通讯等，当然也包含控制数据显示、闪烁选择、开关输入等的控件（Control），一些标准部品已经包含操作程序。按“Program”按钮开始进行程序编辑。在这个编辑窗口中可以写入部品程序或改写标准部品原有的“K-Basic”控制程序。但要注意，如果一些标准部品的控制程序被改变了，则其动作可能会出错，或数据下载可能会无效。

想要详细了解如何编写控制程序，请参考《 K-Basic 控制程序参考手册》。



点击部品属性的“Program”按钮激活程序编辑器，可以发现很多标准部品没有程序，这些标准部品只是通过控件获得它们的功能，因此没有必要为那些功能编写程序。

4）部品背景

一个部品背景由图案（“pattern(drawn picture element)”），构件（“texture”）和背景颜色（“background color”）三项组成，构件指已登录的图形，尽管每个标准部件已经设计了最合适的背景色和构件，用户仍然可以选择其它背景色和构件。同样，用户也可以写入文字，就象修改部品的背景颜色一样。

部品最底下一层的颜色就是背景色。背景色是根据部品的属性来设置的，这是将部品设置成期望颜色的功能。在背景色上，指定的构件显示在整个部品区域，背景色可以透过构件的透明效果看见，部品的图案可以在背景色和构件上显示。

与标准部品一样，构件可以存放到构件库中。打开“library”菜单的新创建功能就能创建新的构件（texture）并登录到库中。如果有必要就可以创建并登录这些内容。

注意：

当改变了部品库中的某个部品时，在此之前已配置在画面上的部品不随着改变；但构件不同，只要构件库的内容改变了，显示在画面上或部品中的构件随着改变。

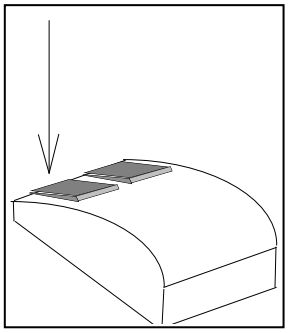
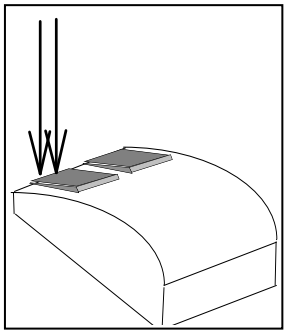
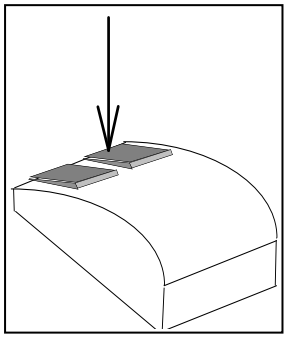
第四章．基本操作

本节描述使用 SCREEN CREATOR 5 所必需的基本操作。

1）基本操作

●单击鼠标和双击鼠标

单击或双击鼠标进行图标选择、选择功能生效、快捷方式等操作，如下所示：

单击左键 （选择操作等）	双击左键 （选择、确定操作等）	单击右键 （快捷菜单或取消操作等）
		
按一下左键	快速按两下左键	按一下右键

●「ＥＳＣ」键

按下 ESC 键将依次取消所选的功能。

2）基本的工具条

SCREEN CREATOR 5 有一个进行基本操作的工具条。当工具条没有显示时，可以打开工具（tool）菜单选择“Display”。

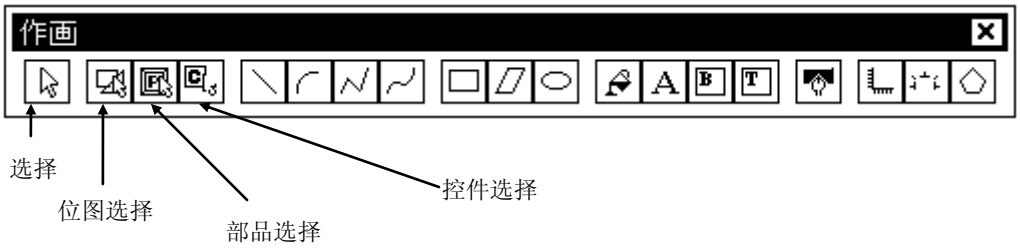
用户也可以把工具条从窗口边上移到窗口中央，这时作为一个工具箱使用，你可以随意选择喜欢的显示形式。

标准工具条



标准工具条包含基本操作的各种图标如：读写文件、拷贝、粘贴等等。

绘画工具条



绘画工具条的图标可以用来安排位图，还可以安排屏幕画面和部品。工具条最左边四个图标用来选择在窗口中使用的对象类型。位图（绘画元素）和部品覆盖在屏幕上，这样可能很难选择想要选择的对象，此时可以点击上述图标限制选择的对象类型。

部品工具条



部品工具条上的图标用来安排屏幕上的部品。对应于绘图和部品选择的工具条在屏幕上打开，参见绘图和部件选择。

3）方便的操作过程

●「ＳＨＩＦＴ」键

在移动鼠标的同时按下 SHIFT 键可以强制画垂直线。

●「↑」「↓」「→」「←」键

按下上述各键可以代替鼠标来移动光标。

●「Ｑ」键

绘画时按下 Q 键可以把光标移动到最近的图形或控件的末端或中央。

第五章. Screen creator 5 操作

本节描述实际画面产生的过程，这里将创建一个名为“DEMO”的新工程。

1）启动 SCREEN CREATOR 5

SCREEN CREATOR 5 安装完毕后，在 Windows 9x/NT 的桌面上会出现快捷图标，双击该图标启动 SCREEN CREATOR 5。

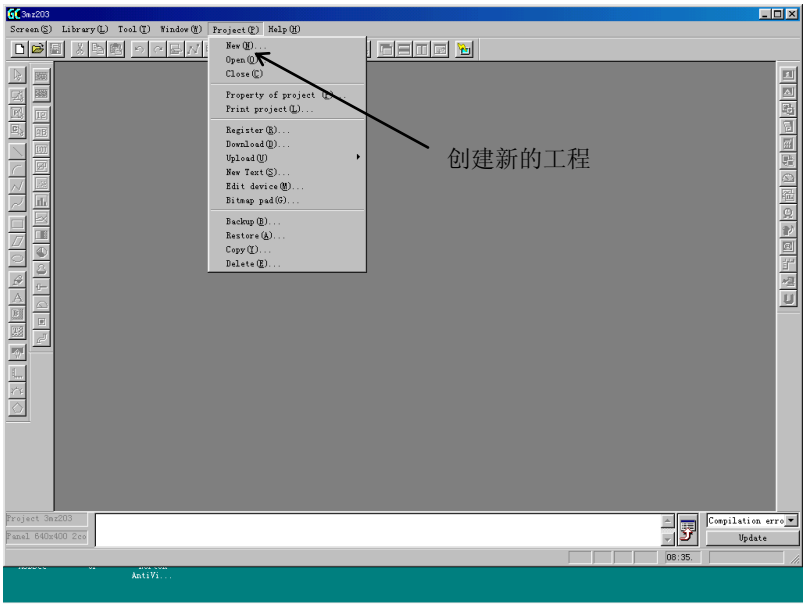


Screen Creator 5.lnk

快捷图标

2）创建一个工程

一个工程聚合了一个全局画面和一个或多个局部画面的屏幕文件，是一组将要下载到 OIP 的屏幕画面数据。选择“Project(P)”下拉菜单下的“New(N)”。

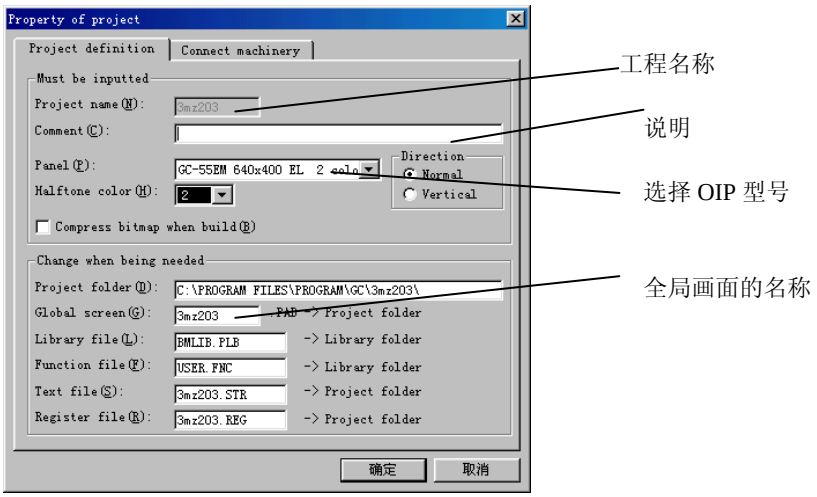


要编辑一个已存在的工程，选择“Open”打开工程。SCREEN CREATOR 5 默认为最近使用的工程，如果使用的是相同的工程，那么继续下面的操作。

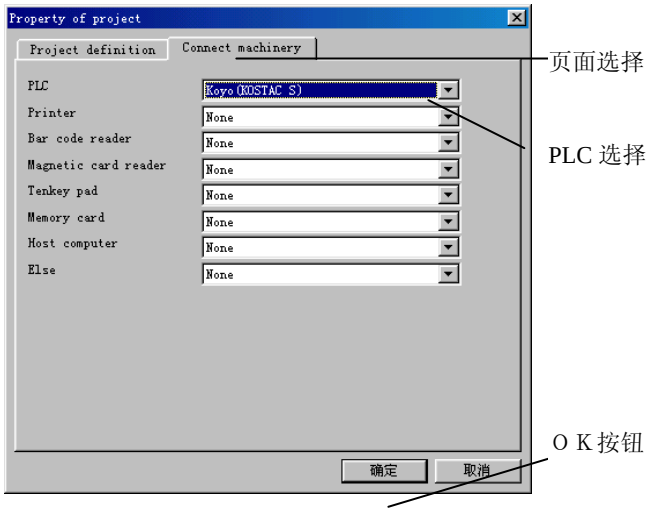
然后，对工程作一些必要设置。

3）设置工程的属性

设置方法如下：



选择所要连接的 PLC 型号。



a. 输入工程名（特别注意：必须以字母开头）

b. 输入工程注释
c. 选择一个 OIP 型号
d. 全局画面名默认为“DEMO”。

其它的不需改变。

e. 点击页面标签改变页面，选择所要连接的设备。

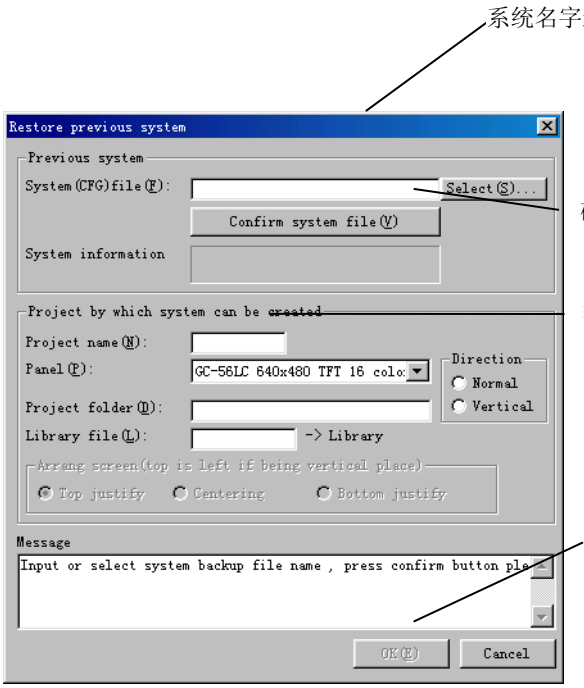
f. 设置所要连接的设备。
选择 KOYO(KOSTAC S)，如果连接的是其他设备请预先找到它。
由于 OIP 能够连接三个通道，因此最多连接三台设备。
注意：只能选择一个通道与 PLC 连接。
g. 按下 OK 键

至此工程设置完成。

4）恢复老型号的数据（参考）

SCREEN CREATOR 5 能够识别由 EA7EAIPC-SGP3 创建的画面并能将它们下载到 EA7EAIPC5x 系列。选择工具菜单（“TOOL”）的老版本系统恢复（“Restore previous system”）打开下面的窗口，将 EA7EAIPC-SGP3 数据文件转换成 SCREEN CREATOR 5 格式的数据文件。

使用如下的 EA7EAIPC-SGP3 数据。
在工具菜单上选择老型号数据恢复。



a. 输入备份的 EA7EAIPCx 系列数据的系统名称，用 Select（S）按钮选择。
b. 确认在所要查找的目标中存在该系统文件。
c. 输入新的工程名称。
其它设置不用变更。
d. 按下 Execute（E）按钮完成。

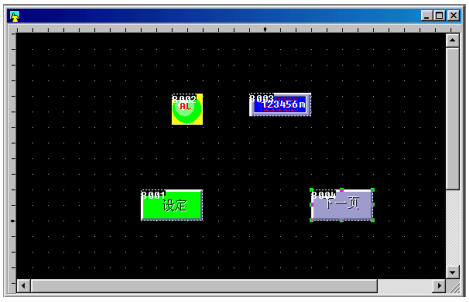
5）退出 SCREEN CREATOR 5

选择“Screen”菜单上的“Exit Application”或直接按下 Close ☒ 按钮
注意！

请在画面数据保存后再退出。

第六章. 创建简单画面

首先我们建立一个很简单的画面，
体会一下部品的基本配置和画面的创建过程。
下面是一个简单画面的例子，包括一个带监视的开关、一个指示灯灯、一个数字显示器。

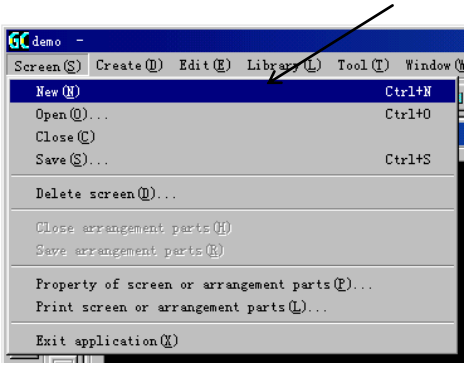


这个简单画面的创建过程如下：

1）创建一个新画面

现在着手创建一个新画面窗。

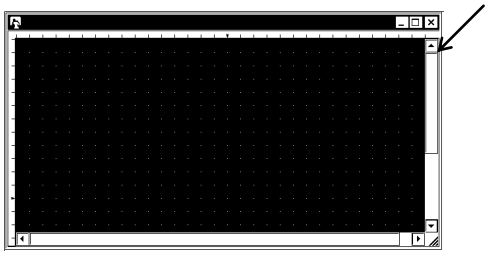
建新画面



在 Screen(S) 菜单中选择 New(N)。

新画面的基础建立了。

创建新画面的窗口



现在开始在新画面上配置各种组成要素。

6．1 配置开关

1）配置一个带监视指示灯的开关

试着配置一个带监视指示灯的开关，这是一个点动(momentory)开关，按下则所连接设备对应位置 ON，手指离开则复位。开关带有监视指示，一般与带有自锁回路的设备相连。下面的例子中，指示灯和开关连接相同的设备。

在部品工具箱中选择开关图标。

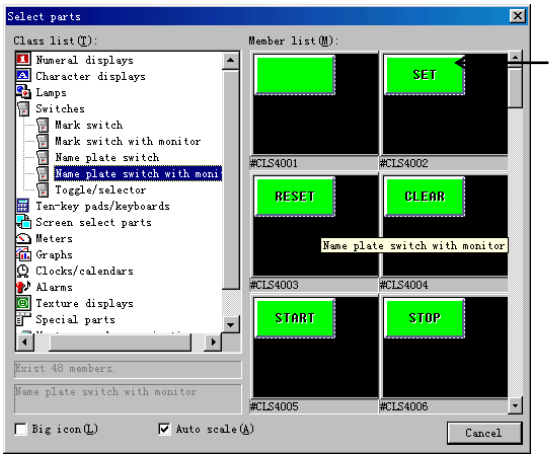
a. 点击开关图标。



开关图标

选择开关图标，打开部品选择窗口，选择部品的格式和功能。

在部品选择窗口中选择必要的部品。

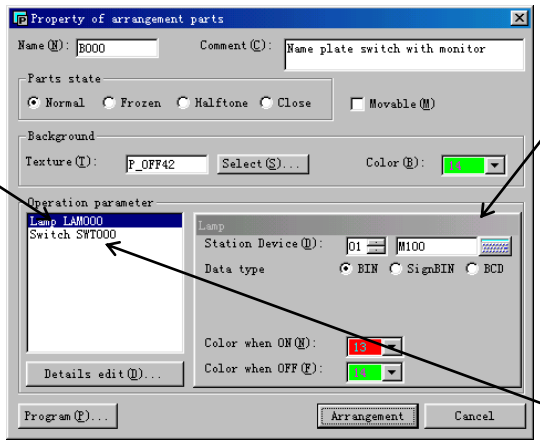


备选的带监视的开关

b. 本例中选择#CLS4002“SET”。

所选部品的属性如下。

首先选择指示灯。



指示灯部品参数选择

输入所连接PLC功能存储器的名称

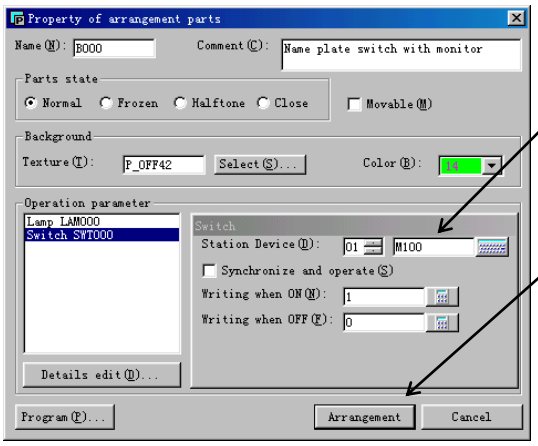
开关部品动作参数设置选择

c. 输入与指示灯连接的 PLC 功能存储器的名称。
本例中为 M100。

然后改变操作参数设置页，设置开关的动作参数。

d. 点击 SWT000 动作参数，选择开关控件的动作参数设置页。

部品属性页切换到开关设置页



输入所连PLC功能存储器的名称

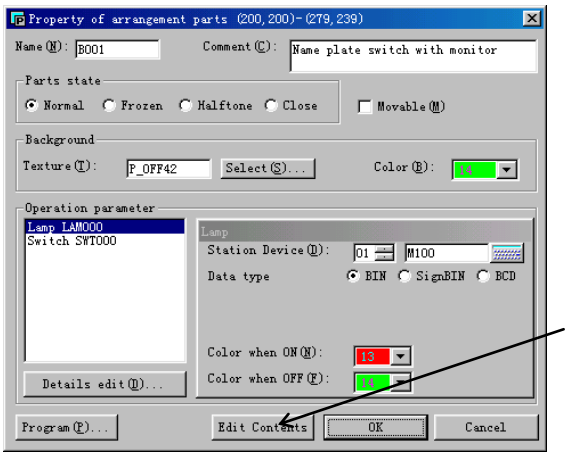
‘放置’按钮

把所选择的带监视的带铭板开关配置在画面上。



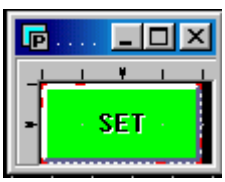
2）改变铭板上的字符

打开部件属性窗口。



内容编辑按钮

部品内容编辑窗口打开。



显示的字符

e. 输入与开关相连的 PLC 功能存储器的名称。本例中输入 M100。

f. 点击放置按钮

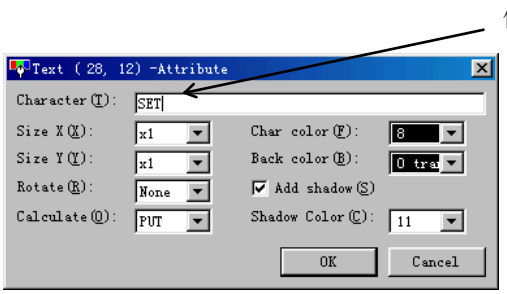
g. 点击画面后，开关的框线就会显示。在要放置开关的位置再点击一下，开关就被放置在那儿。

h. 双击刚放置好的部件。

i. 点击内容编辑按钮。

j. 双击所显示的字符。

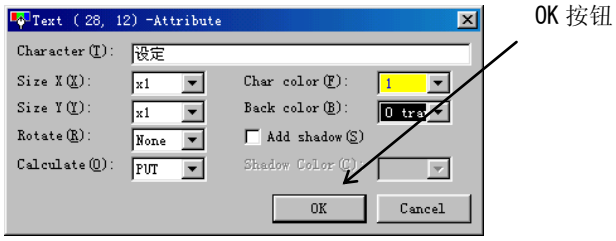
打开字符特性窗口。



k. 点击现在显示的字符“SET”，移动光标。

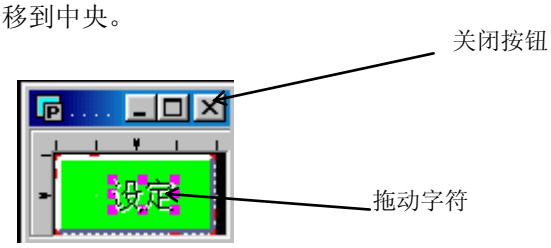
l. 删除“SET”字符，输入新的字符。

改变字符。



m. 按 OK 键完成。

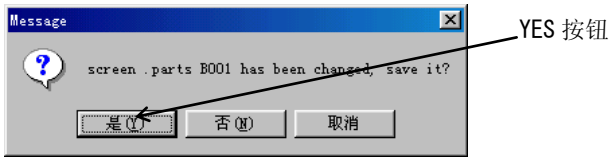
将字符移到中央。



n. 拖动新的字符至部品的中央。

o. 点击部品内容编辑窗口的关闭按钮。

对话框出现。



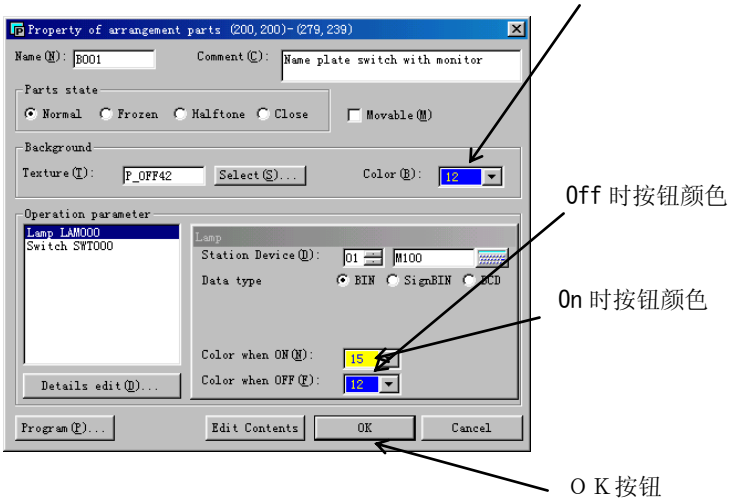
p. 在对话框中按“YES”键。

字符修改完成。



3）改变带监视指示灯的开关颜色

修改方法如下。
本例中，将绿色的(14)OFF 颜色改为蓝色(12)，将红色的(13)ON 颜色改为黄色(15)。打开部件属性窗口。



q. 点击背景颜色按钮，选择颜色，本例中为蓝色(12)。

r. 选择 off 颜色按钮，它必须与背景颜色相同，因此在本例中选择蓝色(12)。

s. 选择 on 颜色按钮，本例中选择黄色(15)

改变 ON 和 OFF 的颜色。

t. 点击 ok 键完成颜色修改。

修改完成。

注意！

- 一些标准部品的指示灯颜色可以改变，其它的标准部品的指示灯颜色用另外的方法改变。

6．2 放置指示灯

1）放置一个 LED

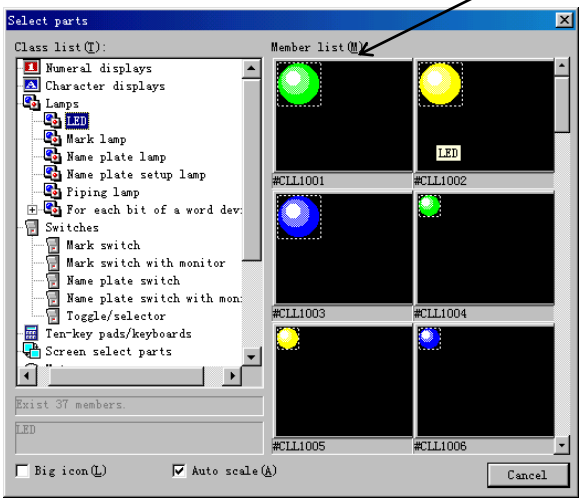
试着放置一个指示灯，当所连接 PLC 功能存储器对应位置位时点亮。



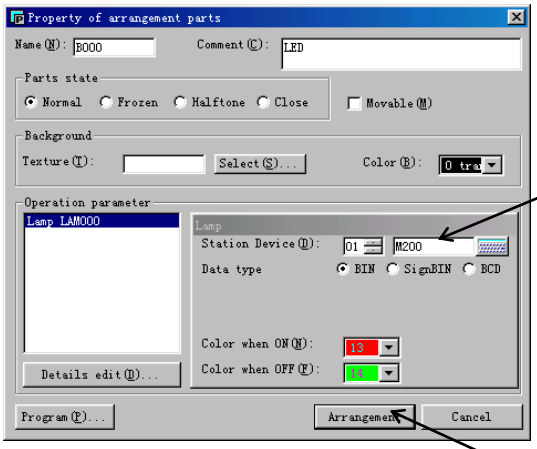
指示灯图标

选择指示灯图标，出现部品选择窗口。
在此窗口中选择一个部品的格式和功能。

想要选择的 LED



打开所选部品的属性窗口。



输入所连接 PLC 的功能存储器的名称

放置按钮

在部品工具箱中选择指示灯图标。

a. 点击指示灯图标。

在部品选择窗口中选择必要的部品。

b. 本例中点击“#CLL1001” LED。

设置指示灯属性。

c. 输入与指示灯连接的 PLC 功能存储器的名称。
本例中输入 M200。

d. 点击放置按钮。

放置 LED 指示灯。



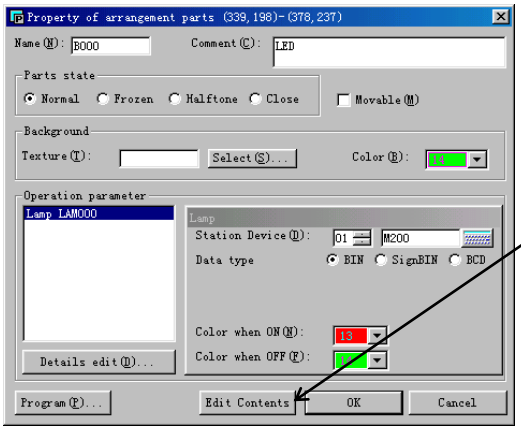
e. 在画面上点击一下，指示灯的方框将出现。再在设定区域点击一次，指示灯就出现在指定位置。

2）在 LED 上添加字符

在 LED 上写入字符。

如下所示，打开部品属性窗口。

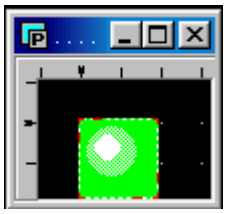
f. 双击刚放置的部品。



内容编辑按钮

g. 点击内容编辑按钮。

内容编辑窗口出现。



在窗口内写入字符

h. 点击工具栏的 A 字符图标。

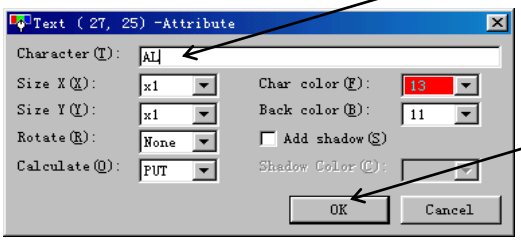
字符图标



字符项目

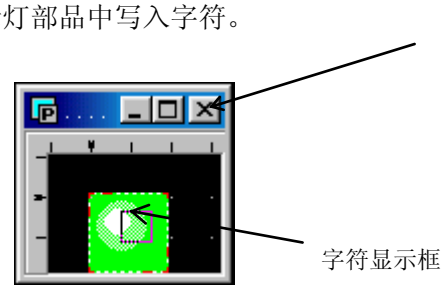
i. 在字符项目中输入“AL”。

点击 OK 按钮。



O K 按钮

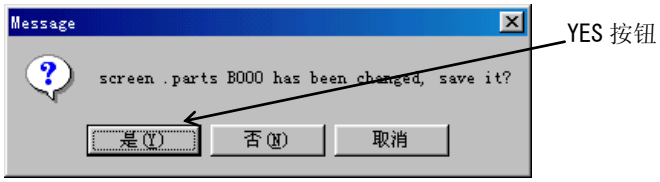
在指示灯部品中写入字符。



j 点击指示灯，出现字符显示框。

k. 选择合适的位置点击，写入字符。

出现一个信息条。



l. 点击 YES 按钮。

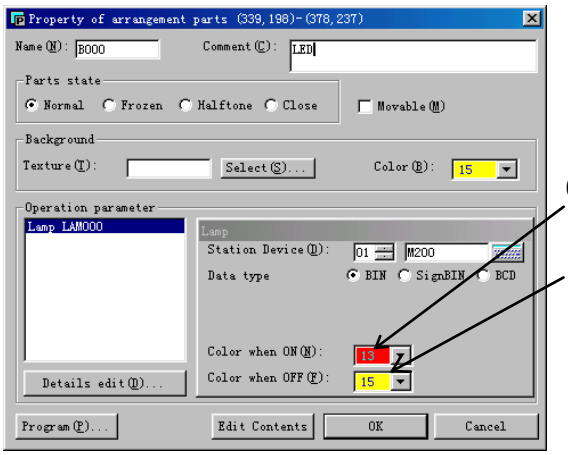
字符被写入。



3 ）改变 LED 颜色

试着改变 LED 指示灯的颜色，如下所示。本例中将指示灯蓝色(14)更改为黄色(15)。

打开部件特性窗口。

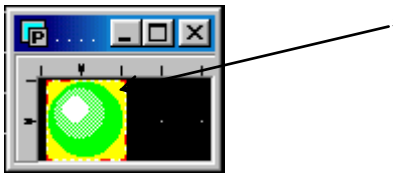


m. 双击要放置的部品。

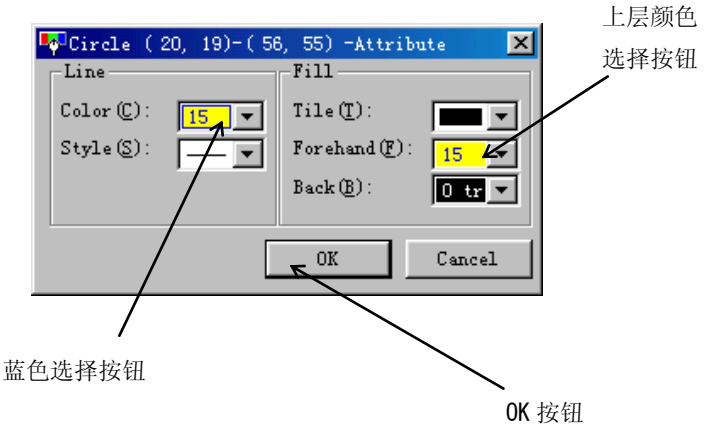
n. 选择 OFF 颜色按钮，本例中为黄色(15)。

o. 点击内容编辑按钮。

将部品这个圆形图案的背景色从绿色(14)改变为黄色(15)。



打开圆形图案的特性窗口。



指示灯 OFF 的颜色已被更改。

注意

.• 要改变指示灯 ON 的颜色，在部品特性窗口 ON 颜色选择按钮中选择新的颜色。
• 一些部品在圆形上覆盖着蓝色圆形，这时同样可以改变其它圆形的颜色。
• 本例中改变的是蓝色指示灯，其它颜色的指示灯同样方法改变。

p. 双击绿色的圆圈以显示圆的特性窗口。

q. 将圆圈特性窗口里面的前面色（fore-color）由绿色（14）改为黄色（15）。

r. 将圆圈特性窗口里面的直线颜色由绿色(14)改为黄色(15)。

s. 点击 ok 键以使设置生效。

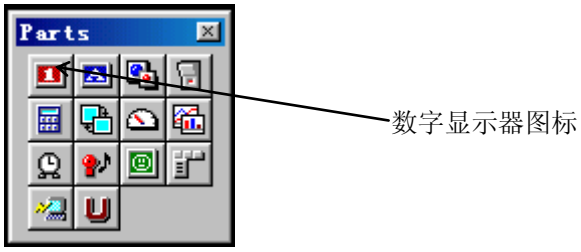
6．3 放置数字显示器

1）放置数值显示器

在数字显示器中显示被选字功能存储器的值，当然能够显示一个双字内容。

在部品工具箱中选择数字显示。

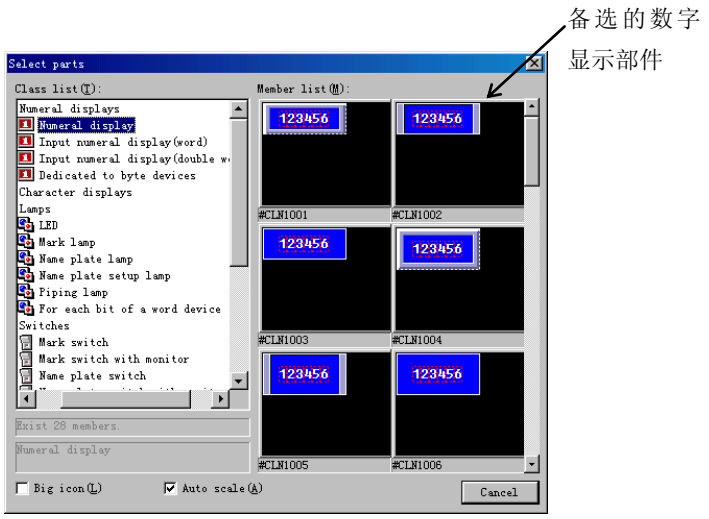
a. 点击数字显示器图标。



在部品选择窗口中选择必要的部品。

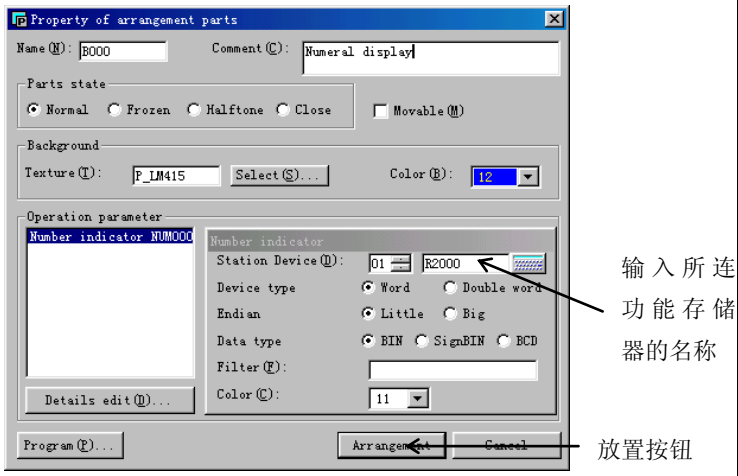
选择数字显示器图标，打开部品选择窗口。
在此窗口内选择合适的格式和功能。

b. 点击数字显示器，本例中选“CLN1002”。



设置数字显示器的参数。

出现所选部品的特性窗口。



c. 输入与数字显示器连接的功能存储器的名称，本例中为R2000。

d. 选择存储器类型和数据类型，本例中分别为 WORD 和 BIN。

e. 点击放置按钮。

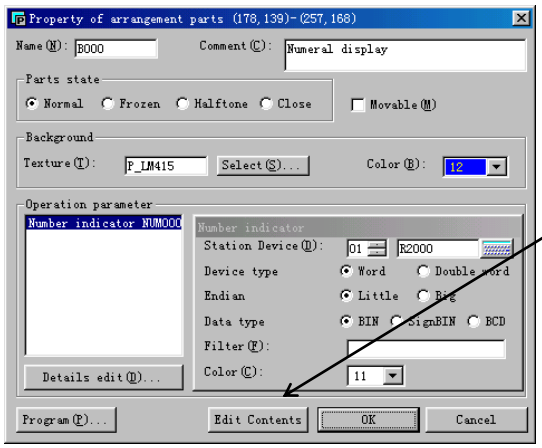
在画面上放置数据显示部品。



f. 在画面上点击一下鼠标，数字显示框出现，在想要安放的位置点击一下，数字显示部品就出现在该位置。

2）在数字显示器上输入字符

打开部品属性窗口。

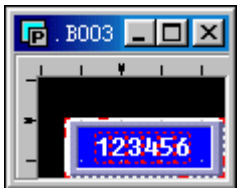


内容编辑按钮

g. 双击刚放置的部品。

h. 点击内容编辑按钮。

打开内容编辑窗口。



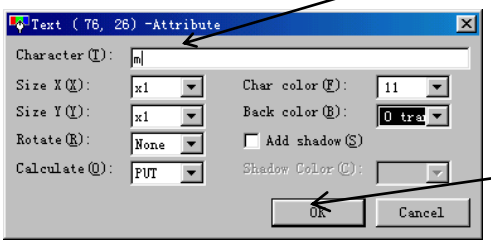
输入字符。

字符图标



i. 点击字符图标。

字符项目

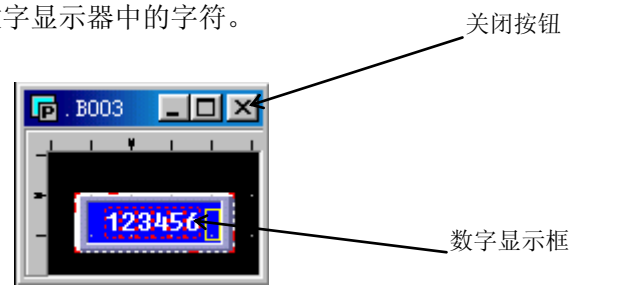


OK 按钮

j. 本例中输入“m”字符。

点击 ok 按钮确认。

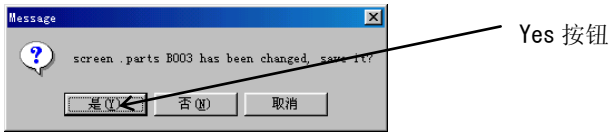
输入数字显示器中的字符。



k. 点击数字显示器以显示字符显示区域。
必要时，调整部品中数据数字显示控件的放置位置。

l. 再次点击确认要放的位置。

对话框出现。



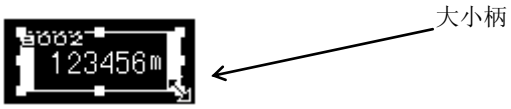
m. 点击 YES 确认。

字符被输入。



3）改变数字显示器的数位和大小

改变数字显示器的大小。



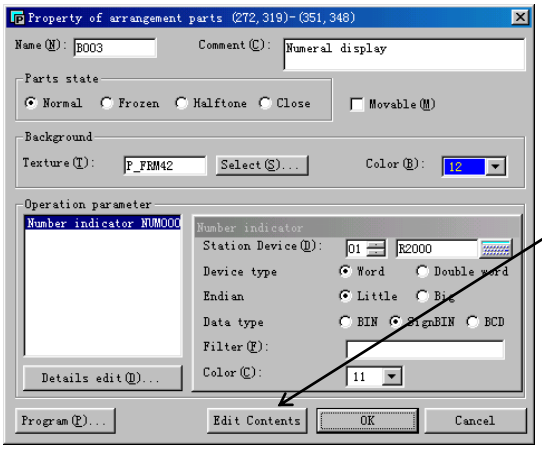
n. 点击一下刚放置的部品，出现大小调节柄。

拉大数字显示器。



o. 调整大小，按下左键并拖动鼠标。

改变数字显示的数位。

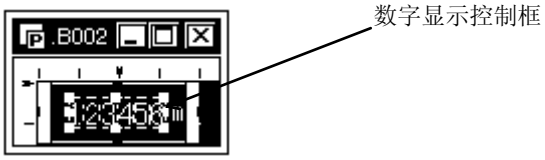


p. 点击内容编辑按钮。

内容编辑窗口出现。



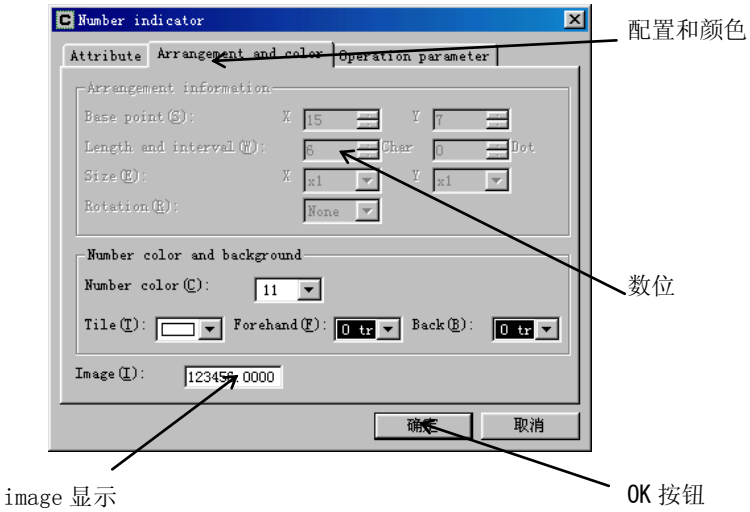
改变数字显示上显示的数位值。



q. 双击数字显示器控件框，打开数字显示控制特性窗口。

数字显示控件的属性窗口出现。

r. 选择配置和颜色页。



s. 改变为 5 个数位。

t. image 显示改为 12345.000000。

u. 点击 OK 完成修改。

完成修改！

注意！

- 如果没有改变 image 显示则显示“e”，这只与 Screen Creator 的处理有关，与 OIP 上的显示无关。
- 如果增加数位，那么它们不能在原部品区域内被显示，这时，在内容编辑结束之后控制功能可能会丢失。

6．4 放置画面选择部品

1）放置一个画面选择开关

请试着在画面上设置一个画面选择按键。
画面选择开关安排好后，用户只需点击一下就能切换到期望的某幅画面。

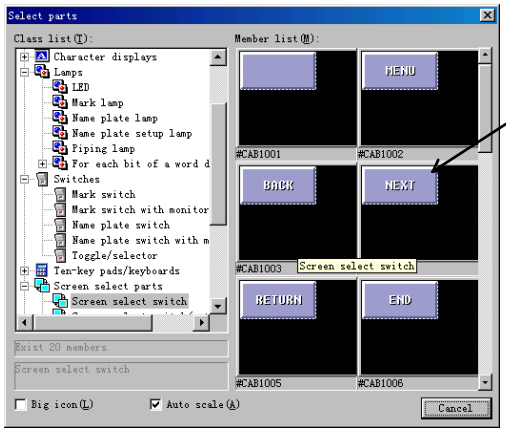
选择部品工具箱中画面选择部品的图标。



画面选择部品图标

a. 点击画面选择部品图标。

选择开关图标，部品选择窗口打开。
在窗口中选择格式和功能。

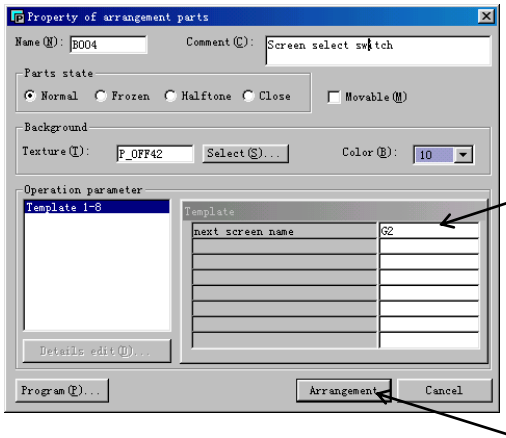


要选的 NEXT 画面开关

在部品选择窗口中选择必要的部品。

b. 本例中点击下幅画面开关“#CAB2004”。

被选部品特性窗口打开



输入下一幅画面名称

设置下一幅（next）画面的画面名称。

c. 在 template 中输入下一幅画面名称，本例中输入 G2。

放置按钮

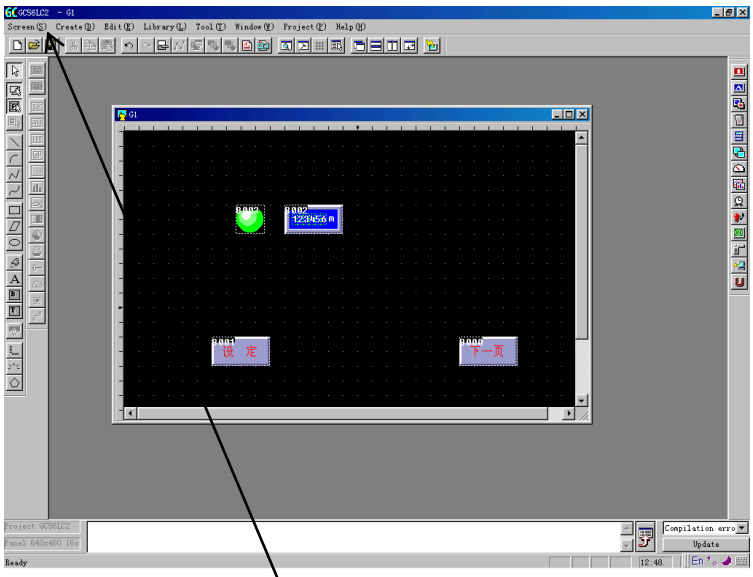
d. 点击放置按钮。

放置 next 画面选择开关。



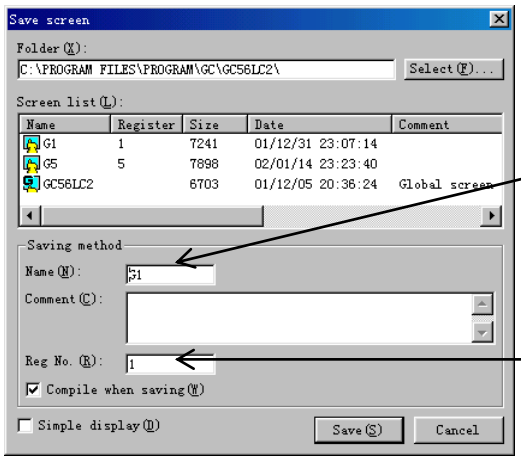
e. 在画面上点击一下鼠标，下幅画面开关框显示，在要放置的位置再点击一下即放置完成。

6．5 画面的保存



点击“保存”快捷按钮。或选择画面菜单的保存项

点击“save”，弹出如下对话框



输入文件名称，注意：不能以数字开头，否则会出错。

输入注册（登记）号。注意：注册时必须有注册号为“1”的画面。

点击“save”以保存画面。

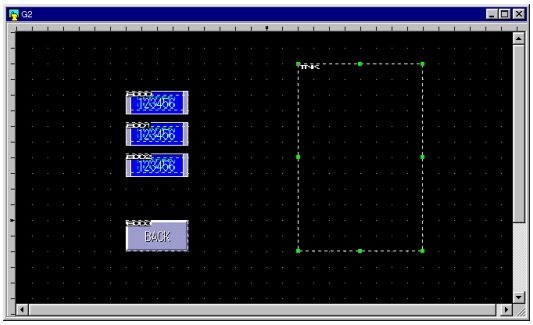
注意：

每个工程必须有一个注册号为 1 的画面，其为 EA7EAIPC 系统上电时显示的初始画面。

没有分配注册号的画面，将不下载到 EA7EAIPC 中。

第七章 7. 简易画面的创建

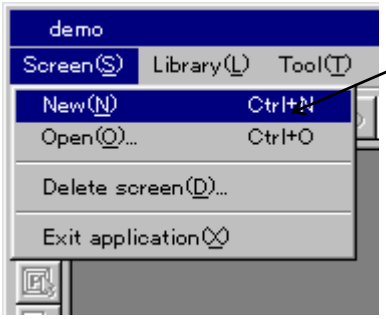
下面让我们来创建一幅画面，以练习部品的拷贝和使用数字键盘来进行数值输入。该画面由三个输入数据显示单元、一个数字键盘和一个屏幕切换按钮组成。本章还将介绍如何使用用于监视错误位的报警显示部品。创建的画面如下：



创建画面的步骤如下：

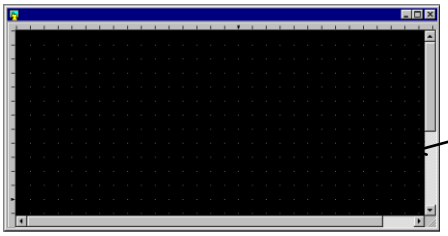
(1) 创建新画面

同第六章一样，选择 Screen 菜单下的 New 选项：



创建新画面

这样就出来一幅新的空画面。



新画面窗口

接下来，就可以开始做新画面了！

a. 选择“Screen(S)”菜单下的 "New (N)" 选项。

7.1 创建数值设置画面

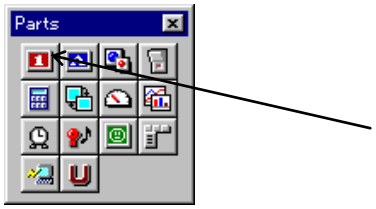
下面介绍如何创建一幅画面，该画面的功能是通过数字键盘向 PLC 单元里输入数值。 每次的数值都通过该键盘输入，而光标总是在上一个值输入完之后自动跳到下一数值显示单元。

本画面由一个数字键盘和 3 个数值显示单元组成。

(1) 输入数值显示单元

在画面上放置输入数值显示单元，当触摸这些输入数值显示单元时，会自动弹出数字键盘，通过该键盘可以往 PLC 内部单元里输入数值，并通过显示单元在画面上显示出来。同样可以输入并显示双字的数值。

在部品工具箱里选择数字显示部品。

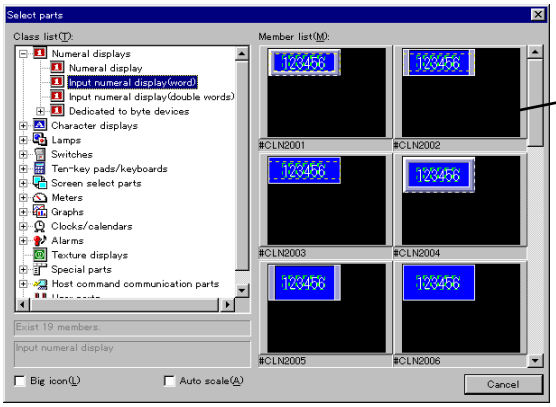


数字显示单元部品

选择数值显示部品项，打开部品选择窗口。

选择所需格式和功能的部品。

a. 点击数据显示部品项



要选的输入数据显示单元

在窗口里选择所需部品

b. 本例中，点击 "#CLN2002" 项。

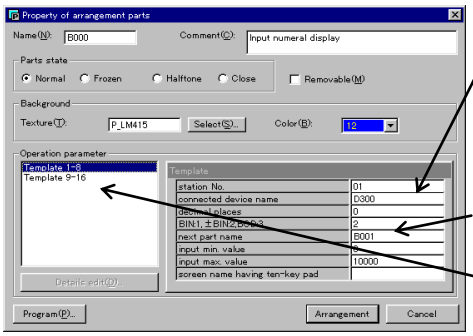
然后就会显示所选部品的属性对话框。

设置输入数据显示单元的属性

c. 输入要连接的功能存储器名称，例如 R2100

d. 输入要转跳到的部品名称，既然本部品的名称为 B000，则下一个为 B001。

e. 选择 "template 9-16" 设置第二页的参数。

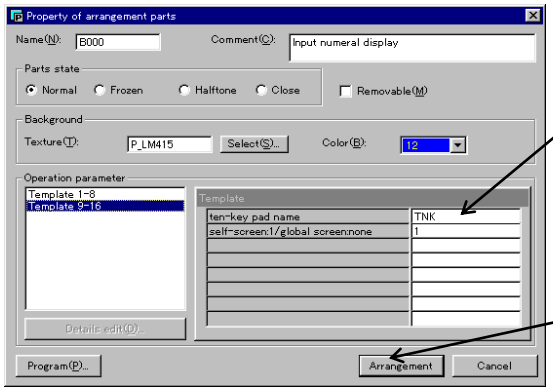


输入要连接的功能存储器名称

输入要转跳到的部品名称

Template 9-16

设置第二页的参数



f. 输入数字键盘名称，这里假设其部品名称为“TNK”。

g. 点击放置键

放置输入数据显示部品



h. 在屏幕上点击鼠标左键，就会出现出现一个虚框架（框架可以随着鼠标移动）然后在期望的位置再点击一下左键，部品就会放置在该位置。由于本数据显示单元带有开关控制功能，所以移动时只能以 20 点为单位。

(2) 部品拷贝

按如下方式拷贝部品



i. 点击已经放置的部品，然后点击复制快捷键。

j. 点击复制快捷键，就会出现输入数据显示单元属性。 输入部品名称 "B001", next item part（下一部品）名称 "B002",连接设备名 R2101". 点击 "Arrangement" 键放置部品。然后用上述同样的方法放置部品。

拷贝两次



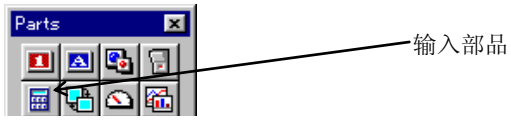
k. 用上述同样的方法复制下一部品。输入部品名称为"B002"。使 next item part（下一部品）为空。 输入连接设备（linked device）名称为 R2102，点击 "Arrangement" 键。

这样就在画面上安排了 3 个输入数值显示单元。

(3) 放置数字键盘

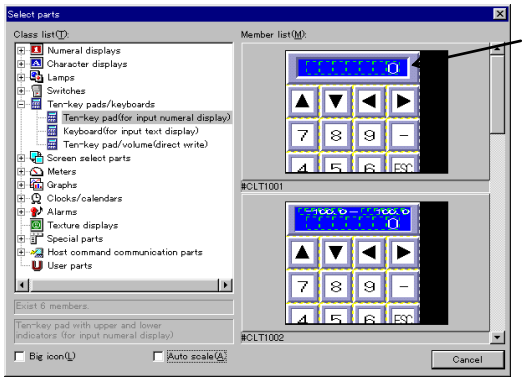
根据相应的输入数据显示单元安放数值输入用十键键盘。

在部品工具箱里选择数据输入部品



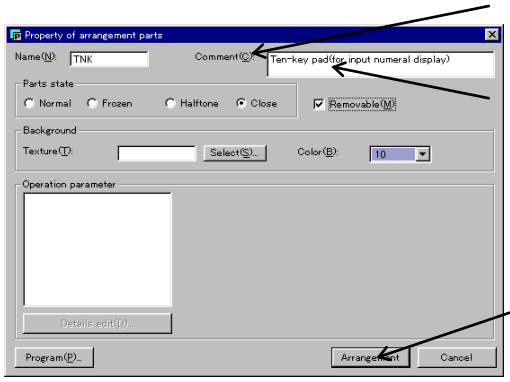
l. 点击输入部品项

选择输入部品项，打开部品选择窗口。
选择所需模式和功能的部品。



需要的十键键盘

出现部品属性对话框

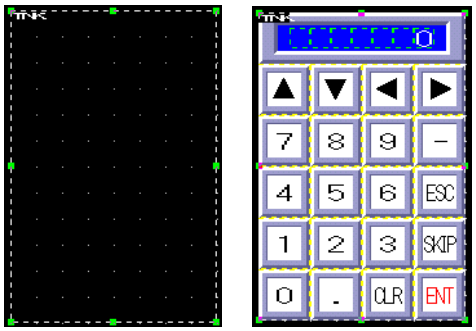


"Close" 补品

可移动控制

放置按钮

放置数字键盘
关闭(Closed)状态 打开(Open)状态



在运行时，当你触摸输入数据显示单元时，会弹数字键盘。在输入完数据并按“ENT”后，光标会自动跳到下一输入数据显示单元。如果没有下一单元，在按“ENT”后，键盘会自动关闭。

m. 选择与输入数据显示单元相对应的数值输入键盘。

设置键盘状态

n. 选择 "Closed" 状态

o. 在 "Movable" 前打“√”， 如果不打“√”，在“Closed”状态下将不能放置。

没有参数需要设置！

p. 点击“Arrangement”按钮。

q. 用同样的方法将键盘放在屏幕上。

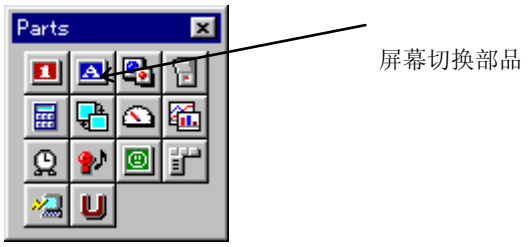
如果部品是以 Closed 状态放置在屏幕上，我们将只能看见键盘的虚框。

7.2 根据PLC的内部状态进行画面切换

根据指定的 PLC 内部单元数据，进行画面切换。同时也可以使用放置在屏幕上的画面切换开关进行。切换的目标画面由 PLC 内部单元设定的数据决定。

(1) 放置画面切换开关

在屏幕上放置一个画面切换开关，当触摸该开关可以将新画面写入规定的设备（单元）里面

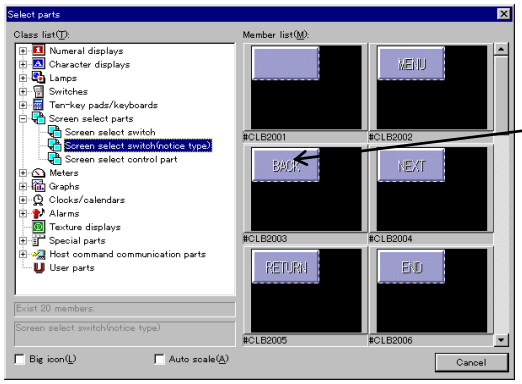


在部品选择工具箱中选择画面切换部品选项

a. 点击画面切换部品选项

选择画面切换部品选项，会出现画面切换部品选择对话框。

选择所需功能和模式的画面切换部品。

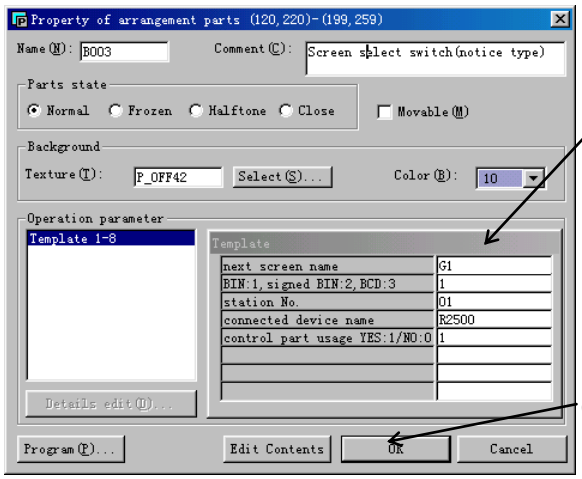


选择所需的画面切换部品选项

b. 本例中，选择前一幅画面选择部品 "CLB2003" 。

出现所选部品属性对话框

设置该部品的相关数据



c. 输入下一幅画面的名称，在这里为 G1。

d. 输入所连接的设备名称(也就是画面注册号所需通知的寄存器号) 这里我们假设为 R2500.

e. 将 control part usage(画面选择控制部品使用)设置为"Use: 1"，即使用！

f. 点击“OK”键

放置切换到上一幅画面按钮

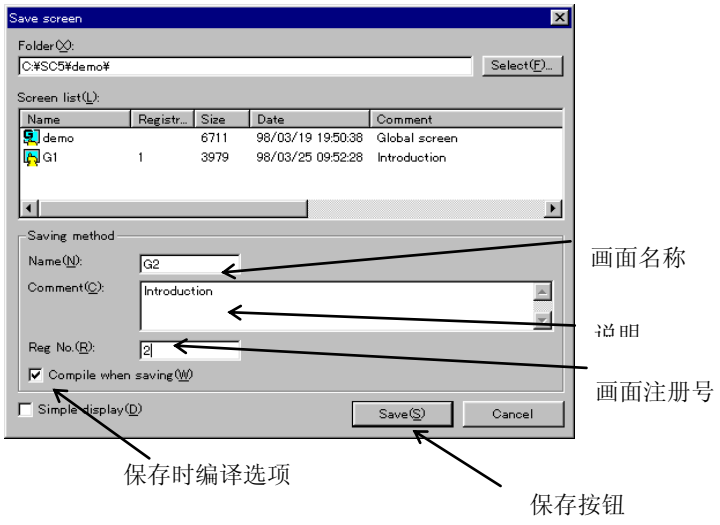


将部品放置在画面上恰当地位置。

(2) 保存局部画面

保存以上所创建的画面

选择“Screen(S)” 菜单下的 "Save(S)" 选项。



g. 输入画面名称（注意：必须开头第一个字符必须为字母）这里为 G2。

h. 输入 comment（注释）,这里以 "Introduction" 为例！

i. 输入画面注册号，用于画面的打开。这里为 2，该注册号在同一个工程里必须唯一。

j. 保存画面。如果编译选项有效，画面在保存的同时还将进行编译。（也就是将其转换成可下载画面）。如果出现错误，保存时会有错误提示框。

k. 点击保存（save）按钮。

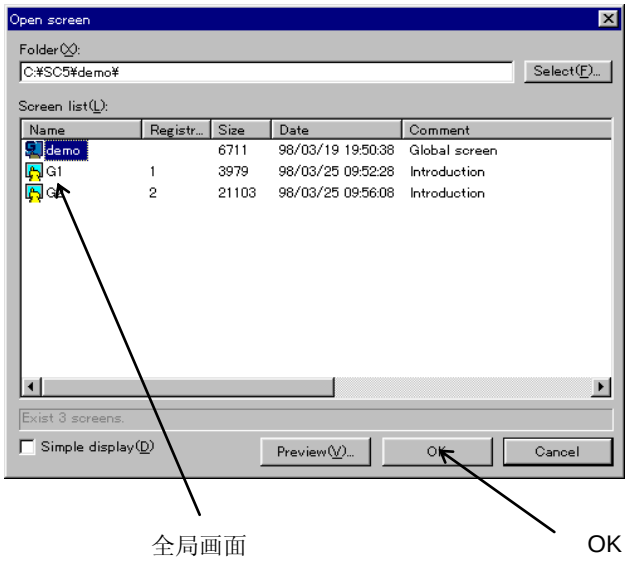
创建的画面以文件名 G2 保存。

(3) 将画面选择控制部品放在全局画面上。

将画面选择控制部品放在全局画面上。画面将随着所连单元里的数据作相应变换。

首先打开全局画面

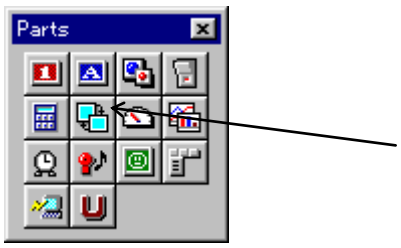
选择 “Screen(S)”菜单下的 "Open(O)" 选项。



l. 选择 全局画面"DEMO"

m. 点击 OK 键

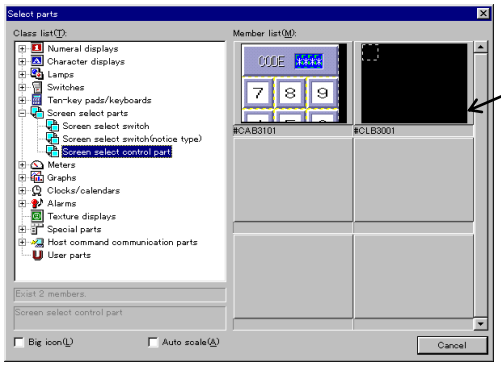
然后将画面选择控制部品放在全局画面上。该部品将画面切换到寄存器指定的画面。



在部品工具箱里选择画面选择部件

n. 在部件工具箱里选择画面选择部件！

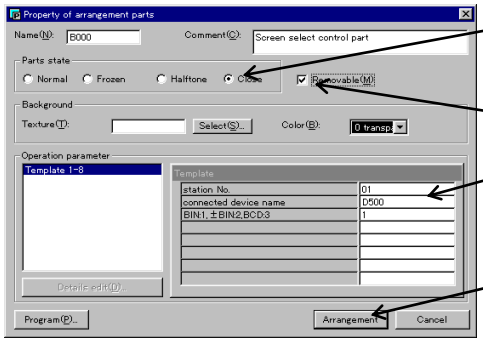
点击上述选项，会出现如下部品选择窗口
在窗口里选择一个功能部件，该部件没有外形。



选择所需部品

o. 点击 "#CLB301".

出项属性对话窗口



p. 选择“closed”状态

q. 置 "Movable"项有效 （注意：非 movable 部品不能以 closed 状态放置）。

r. 输入所连接的寄存器名。 （这里为 R2500）

s. 将其放在适当的位置。

放置部件。



这样画面的切换既可以通过 PLC 寄存器里的值来实现，又可以通过画面上的屏幕选择控制部件和屏幕选择开关来改变！

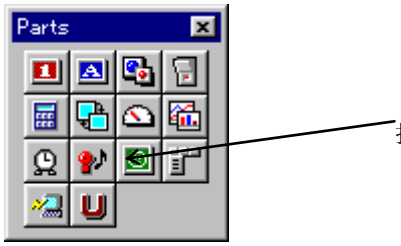
7.3 报警画面的设置

本部分介绍报警画面的设置 。报警功能反复地从规定的起始位置（PLC 内部的某位）检查设定的位数。当发现错误位置位时，画面就会显示与该位相对应的报警条文。当错误位变 off 后，报警画面将自动关闭。注意，报警画面必须放在全局画面上！

(1) 报警画面的安排

报警画面应放置在全局画面上。

选择报警控制选项



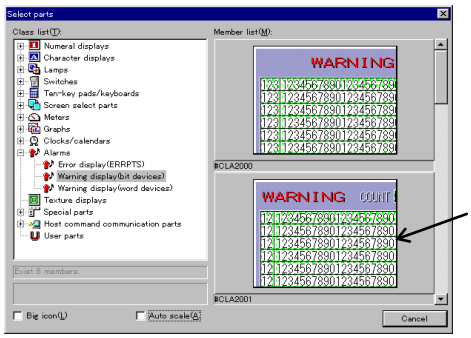
报警部品

a. 点击报警部品符号选项

选择报警部品选项，出现报警部品选择窗口。

在报警部品选择窗口里选择适当的部品

选择合适的功能和模式。



表型报警显示画面（位报警）

b. 点击清单上的部品（bit 设备）"#CLA2001" 。

出现部品属性窗口

参数设置

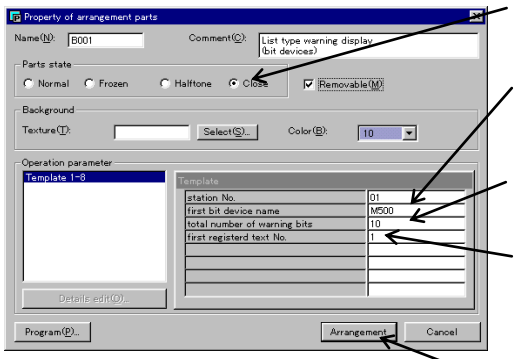
c. 选择 "Closed" 和 "Movable" 。

d. 输入所连接的设备(位)名。（这里为 M500）

e. 输入位报警位数，这里设为 10 位（bit）. 即 M500～M511.

f. 输入首条报警条文的登记号！

g. 点击放置按钮！



关闭状态

输入连接的设备名（位或单元等）

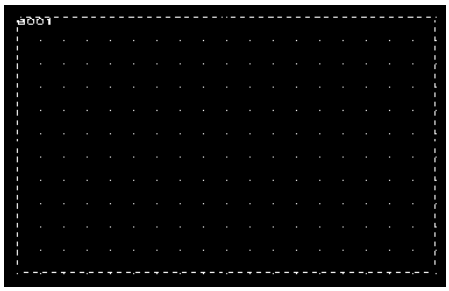
输入总的报警位数

输入第一个报警条文登记号

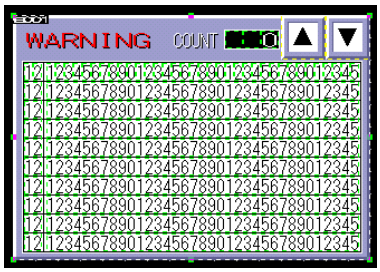
放置键

将表格式报警显示窗口放在希望的位置(bit 单元). 放下后，因为其状态为 closed，所以，只能看见虚框。

方法同上

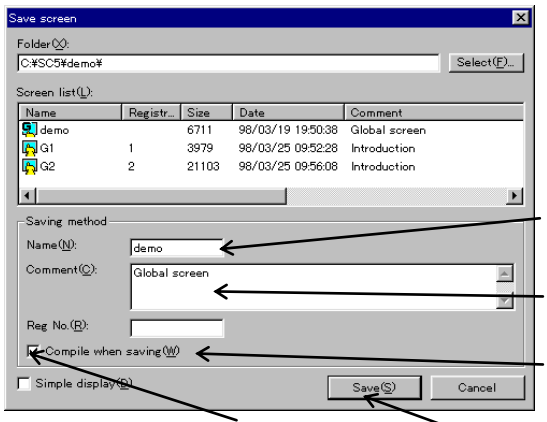


例如, M500 由 OFF 变 ON，这时在画面上会出现报警表格 ，并且登记号为 1 的字符串将显示在表的首行。报警条文的设置和登记在下面介绍。



(2) 全局画面的保存

保存全局画面



名称

说明

画面注册号

注意：只能为“空”

编译选项

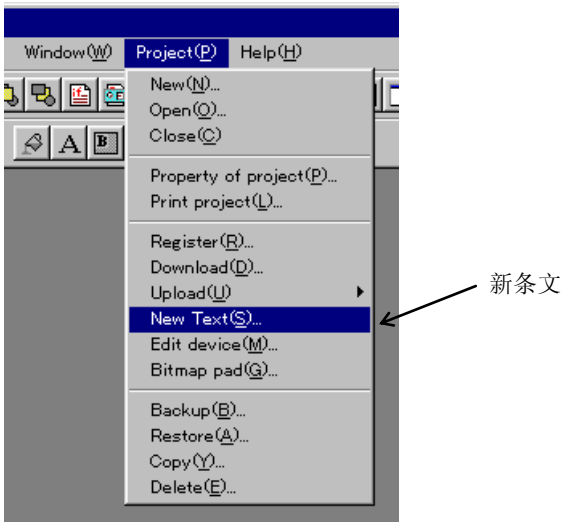
保存

方法同上

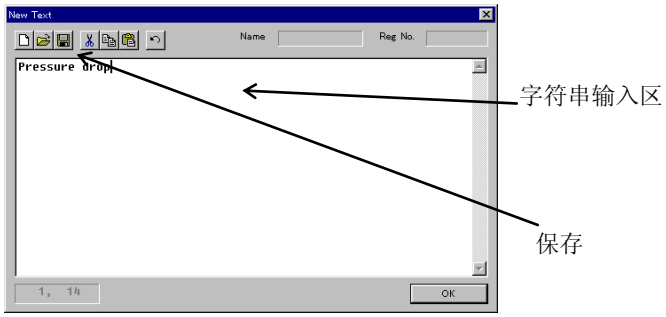
全局画面保存为 demo

(3) 创建报警字符串条文

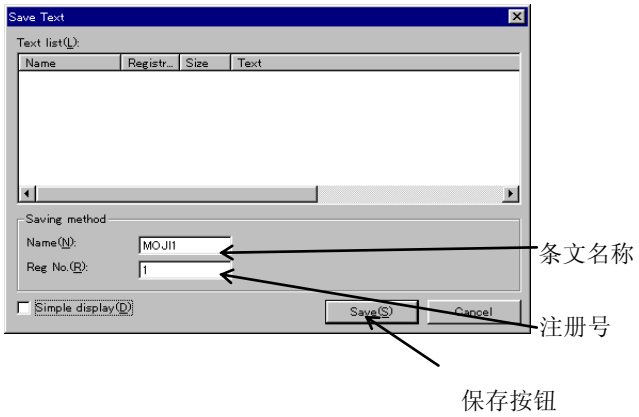
创建并注册报警时希望显示的报警条文



出现报警条文编辑窗口，你可以在这里编辑你希望的报警条文！



保存编辑的内容



这样就将报警条文保存好！

i. 选择 “Project(P)” 菜单下面的“New Text(S)”

j. 输入报警条文字符串，如"Pressure drop" 。

k. 点击”save” (保存)！

l. 输入字符串名称，这里为 "MOJI1"

m. 输入报警条文登记号！

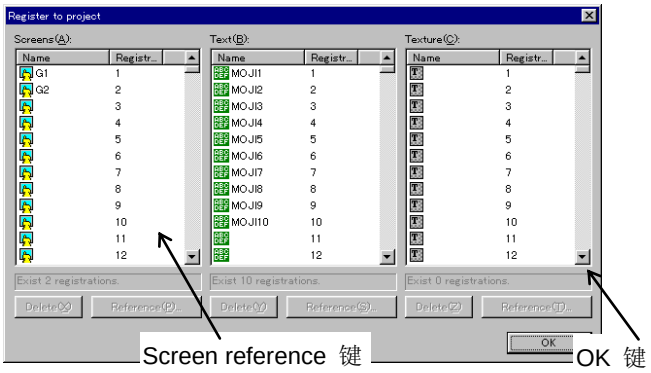
n. 保存

这样就编辑好了一个报警条文。

第八章. 画面数据的下载

(1) 登记需要下载的画面

给上述已经创建的画面注册，以供下面下载。同时也要将下面需要下载的报警条文注册！
（在上面存画面文件时的登记号是同样的功能！）



选择“Project(P)”菜单下的"Register(R)"，如果画面已经登记，在这里就不必重新登记！

a. 点击画面注册表的空白地方。

b. 点击 “Screen reference”键 选择需要登记的画面

c. 点击“OK”按钮

d. 如果需要登记或使用一个控件，可以采用同样的方法！

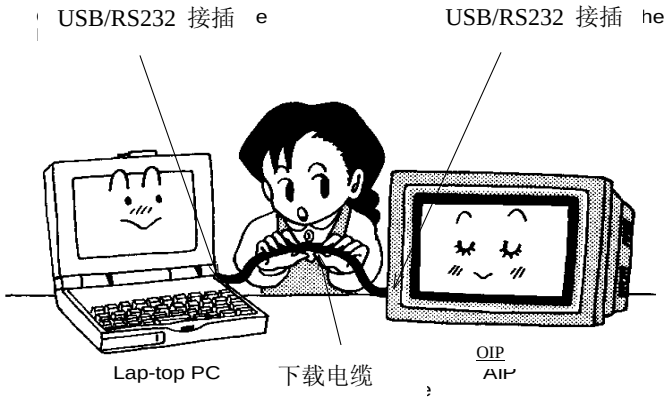
注意：

- 确保登记一幅登记号为 1 的画面，该画面在上电时或被切换到用户模式时即被打开。对于 2 及以后的编号可以随便登记，但值得注意的是如果不按顺序注册，将会稍微增加其占用的内存空间！

- 全局画面不能有注册号，每个工程只能有一个全局画面，该画面是由新建工程时定义的工程属性决定的！

(2) OIP 模式的改变

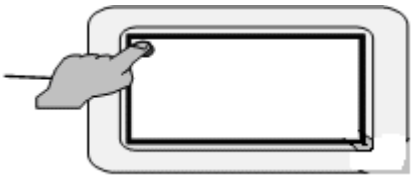
在下载画面数据之前，使用规定的通信电缆将 OIP 的 SUB 接插口与计算机的 USB 口相连。



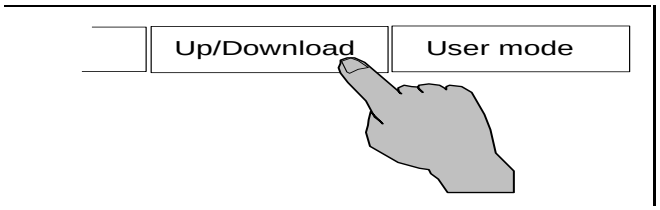
将电缆插紧

在 OIP 上选择 Up/download 菜单

a. 按住左上角 3 秒
（GC 型号为左上加右下角 3 秒）

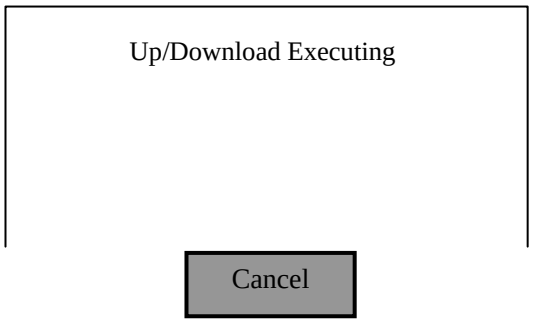


<<OIP>>



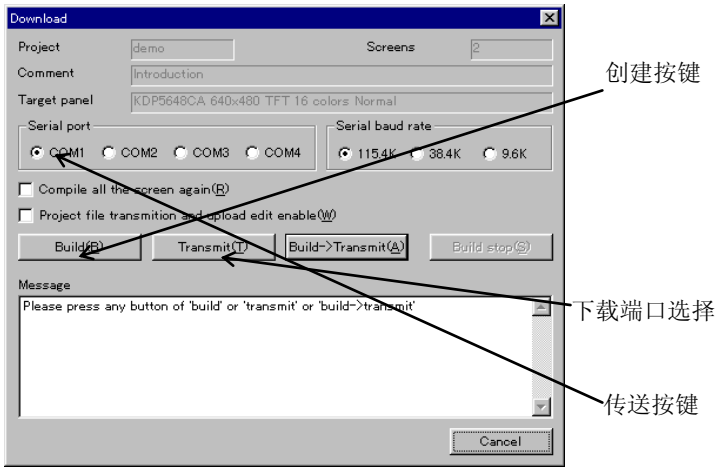
b. 屏幕上面的"Up/Download"将屏幕切换到下载状态。

<<OIP>>



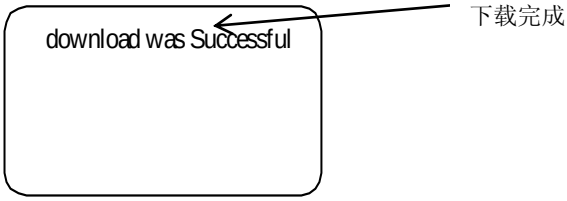
c. 则 EA7EAIPC 已处于等待下载状态！

(3) 下载执行



<<Screen Creator 5>>

▼



<<OIP>>

选择“Project(P)” 菜单下的 "Download(D)" 选项。

a. 选择计算机计算机的设置，选择 RS-232C 端口作为下载端口。这里为 COM1 口。

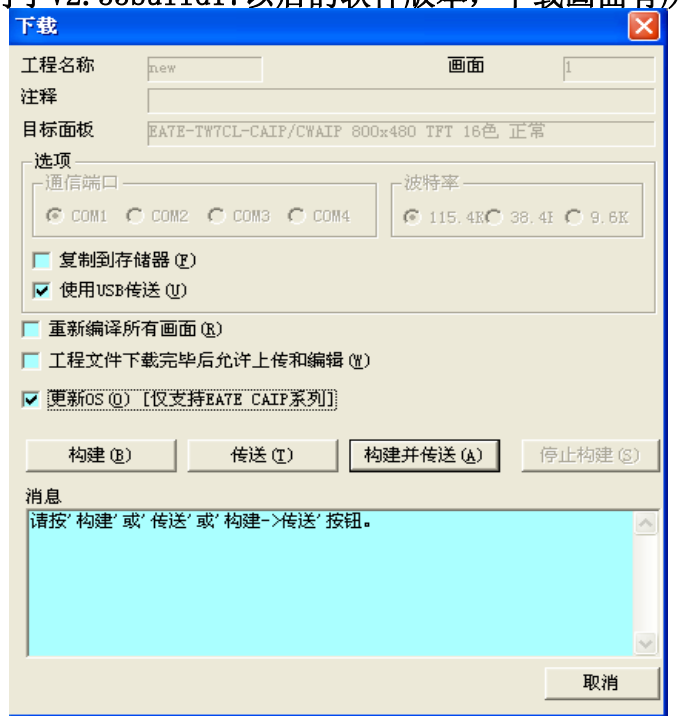
b. 点击 “Build”. 如果需要下载的画面上的部件参数包含错误，对话框将会告诉你在什么地方出项了语法错误！

c. 在 Build 结束后，点击"Transmit"进行画面传送！

点击“Build->Transmit” 可以让 Build 和 Transmit 连续执行。

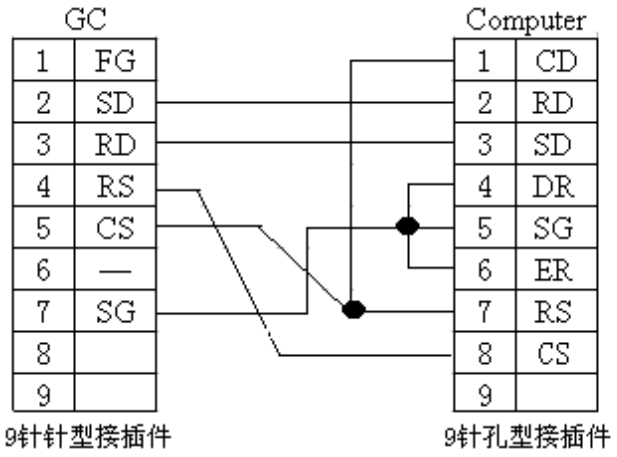
d. 当下载完毕，屏幕上会显示 "Download was successful"表示下载成功！

对于V2.55build17以后的软件版本，下载画面有所不同



针对 EA7E_AIP 系列增加了 USB 传送选项（EA7E_AIP 系列仅支持 USB 通讯）
增加更新 OS（底层系统程序）功能（仅支持 EA7E_AIP 系列）

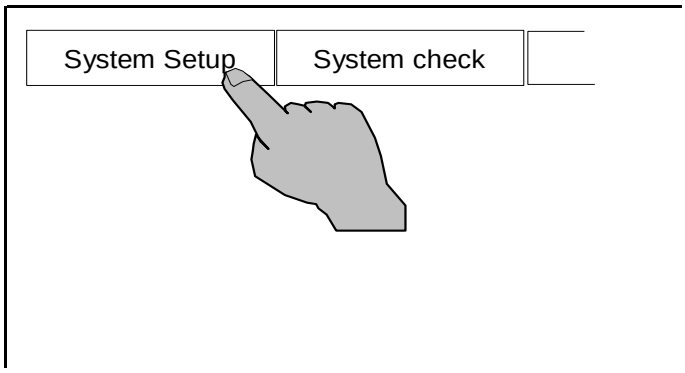
(4) 附GC型号下载电缆接线图



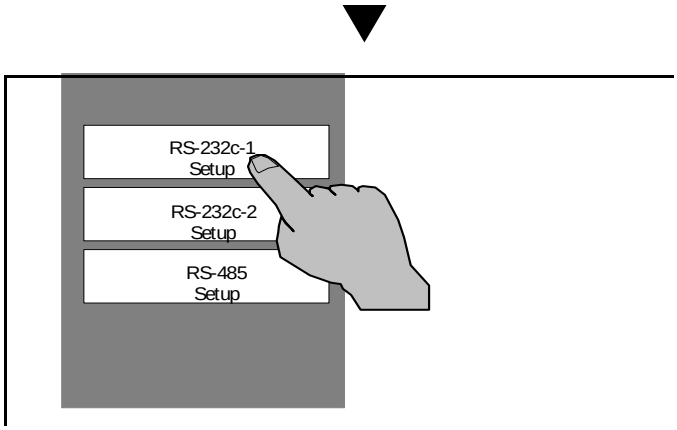
第九章. 画面测试

(1) 设置 OIP 通信口

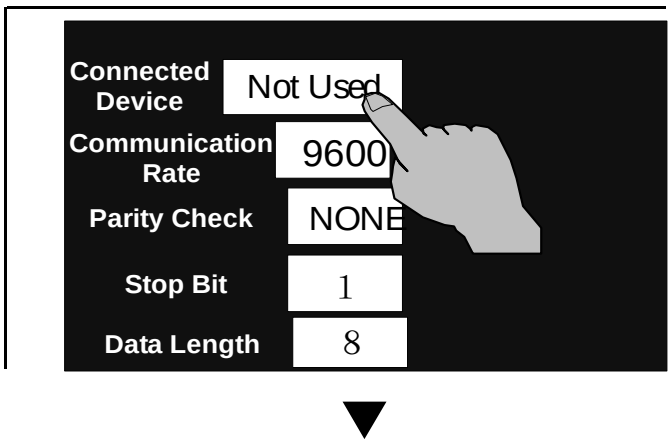
将其设置为 PLC 与 CH1 通信



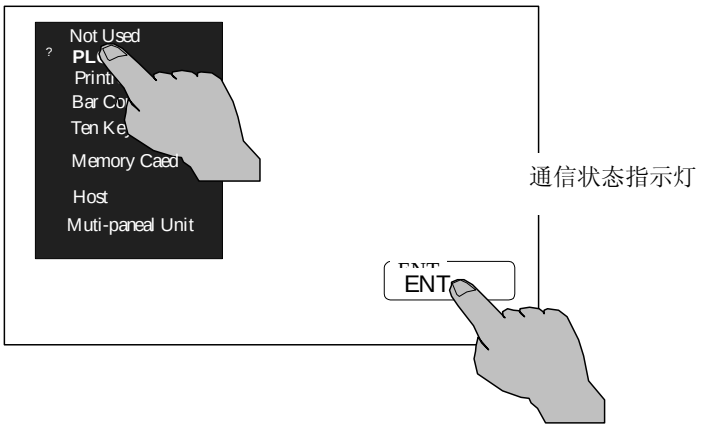
a. 按屏幕左上角的 "System setup"



b. 按 "RS-232C-1 Setup".

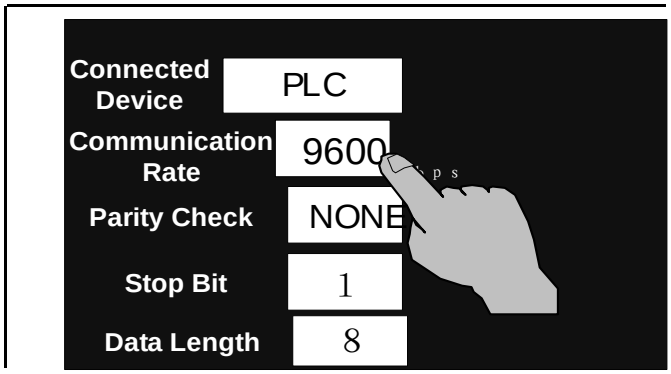


c. 触摸“Connected Device”，会弹出一个连接设备选项，红灯指向所选设备。



d. 选择“ PLC”.
选中标记★指向 PLC.

e. 按“ENT”键

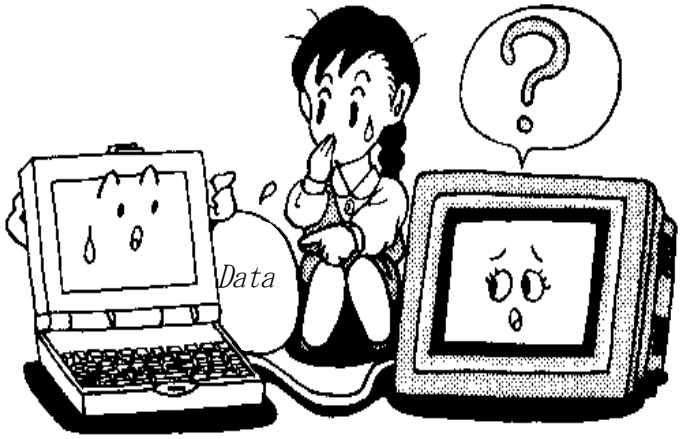


f. 设置通信速率及其它通信参数。

- 通信速率的速度变化是
  9600 → 19200 → 38400 →
  1200 → 2400 →4800 → 9600.
- 奇偶校验变化
  EVEN → NONE → ODD.
- 停止位变化
  1 → 2 → 1.
- 数据长度变化
  8 → 7 → 8.

注意点：

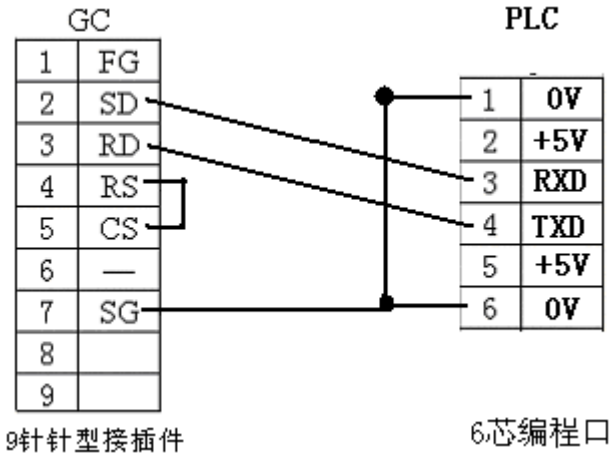
- 只有当画面数据已经下载到 OIP 后，才能设置通信端口。 所以如果不能设置通信端口，请检查画面数据是否已下载！
- 通信端口一旦设定，通信端口设置就存储起来，没有必要在每次下载画面时都进行通信端口设置！



（2） 与 PLC 的连接

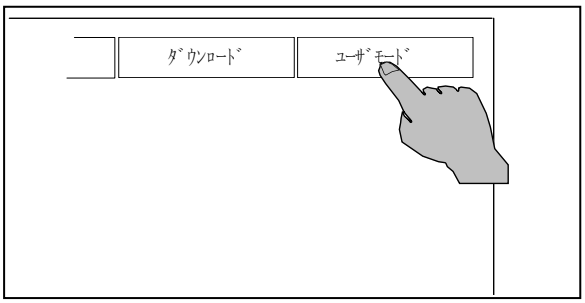
下载好画面，设置好 EA7EAIPC 的通讯口参数后，就可以把 EA7EAIPC 与 PLC 连接起来，进行实际的画面操作。这时，首先要做好两件工作：

1. 设置 PLC 相应通讯口的通讯参数，使之与 EA7EAIPC 的一致；
2. 制作 EA7EAIPC 与 PLC 通讯的通讯电缆，制作时请参见各厂家 PLC 串行通讯口相关资料。
3. GC 型号与 PLC 的连接电缆图如下



(3) 选择用户模式

既然已经将画面数据成功地传送到了 OIP，就可以将其与 PLC 连接起来进行试运行。



点击 "UserMode" 进入用户模式，OIP 便开始运行！

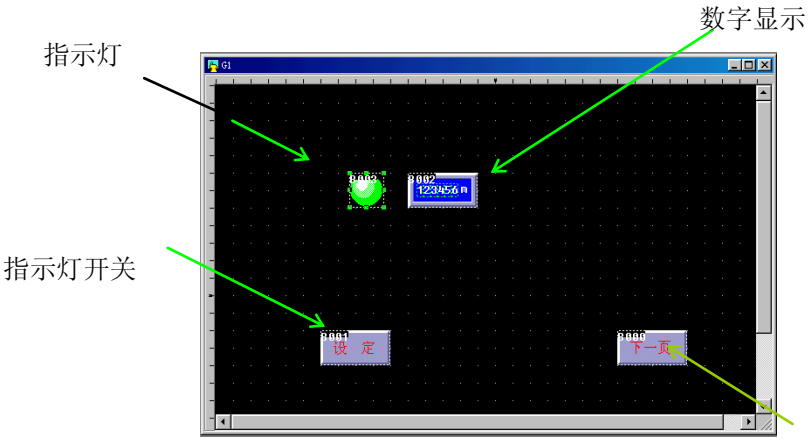
注意：

- 如果在选择用户模式之后要过很长时间才出现第一幅画面，这可能是 OIP 与 PLC 通信不正常。 请检查通信参数是否设置正确！

第十章. 画面测试

（1）G 1 画面测试

测试以上创建的画面如下。



注意点

检查画面功能是否正确。

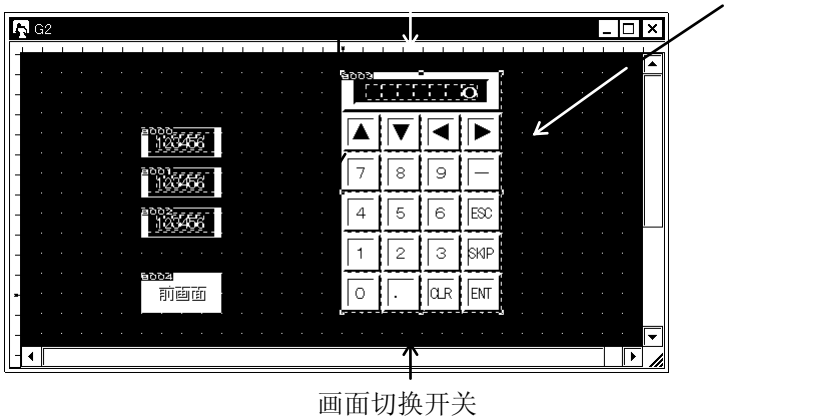
如果数字显示器上没有数字，可能是通信出错。如果 PLC 上的 SD 和 RD 灯快速闪烁，说明通信是正常的。但如果只是偶尔闪烁，则可能是某些通信参数设置错误。

（2）G 2 画面的测试

接下来对画面 G2 进行测试

输入数字显示器

数字键盘



当 OIP 进入用户模式后，首先显示 G1 画面。

a. 使用监视灯来对开关进行测试。触摸开关，看开关指示灯是否为“ON”。另外，可使用编程设备检查 M100 是否只有在开关被按下时才变 ON。

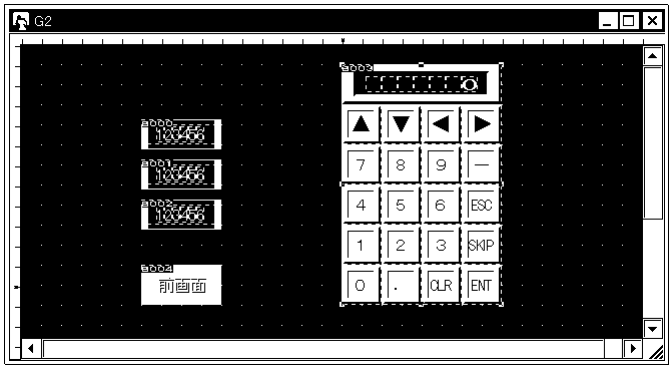
b. 指示灯测试。使用编程设备将 M200 置 ON。注意指示灯是否在 M200 为 ON 后变亮。

c. 检查数字显示器。首先确定当前显示为“0”。使用编程设备往 R2000 里输入一个值，检查显示器显示的是否是输入值！

d. 触摸画面切换开关，看其是否跳至 G2 画面。

e. 触摸最上面的输入数据显示器，看数字键盘是否打开。

显示数字键盘。



f. 通过数字键盘输入一个值，然后按“ENT”键。
检查最上面的数字显示器是否显示刚输入的数值。同时，检查光标是否跳到下一个显示器。

g. 使用同样的方法输入第二、第三个数字，并让显示。当第三个数字输入完之后数字键盘自动关闭。
使用编程设备监视 R2100、R2101、R2102 里面是否有输入的数字。

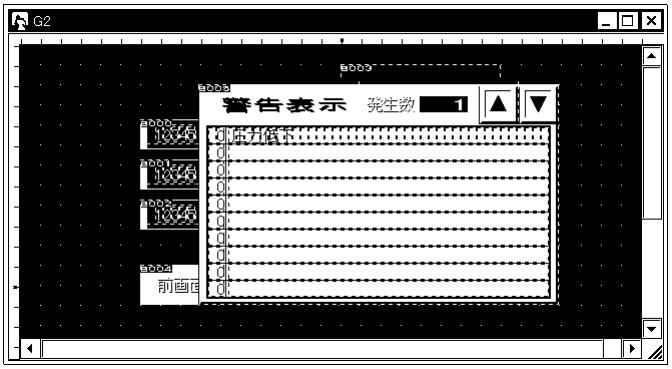
注意点

如果输入数据显示器不显示数据，可能是由与（1）相同的通信原因造成的。这种情况下可以采取与（1）相同的措施。如果通信执行不正确，那么就不能通过画面切换开关进行画面切换。

h. 按下画面切换开关，屏幕应该切换到画面 G1，这时使用编程设备可以在 R2500 里面监视到画面编号 1。

（3）全局画面测试

测试全局画面上的报警显示



i. 使用编程设备将与报警画面链接的继电器 M500 置 ON，观察报警画面是否打开，并在里面显示与 M500 相对应的报警条文“压力下降”!
将其它位置“ON”，看是否显示相应的报警条文!

所制作的画面已经测试完成，它们的功能是否正确呢？如果运行不正确，请阅读相应的错误描述并检查运行参数。

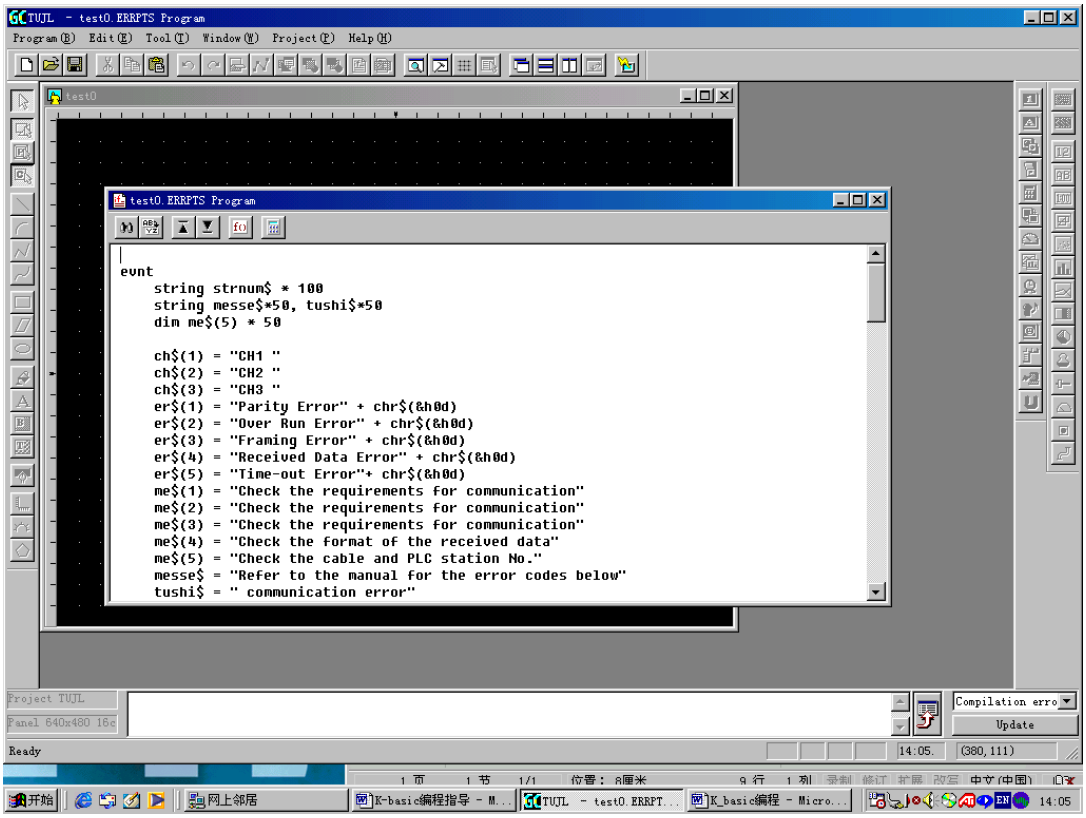
第十一章 K-basic 控制程序基本

11．1 简 介

1、什么是控制程序

当你用手指按操作面板的时候，你希望某部品显示数值、字符、或使某个开关运作时，你需要使用K-basic语言编制程序。

我们使用K-basic程序语言来为部品编制程序，使其完成期望的功能。



2、K-basic 语言描述的对象

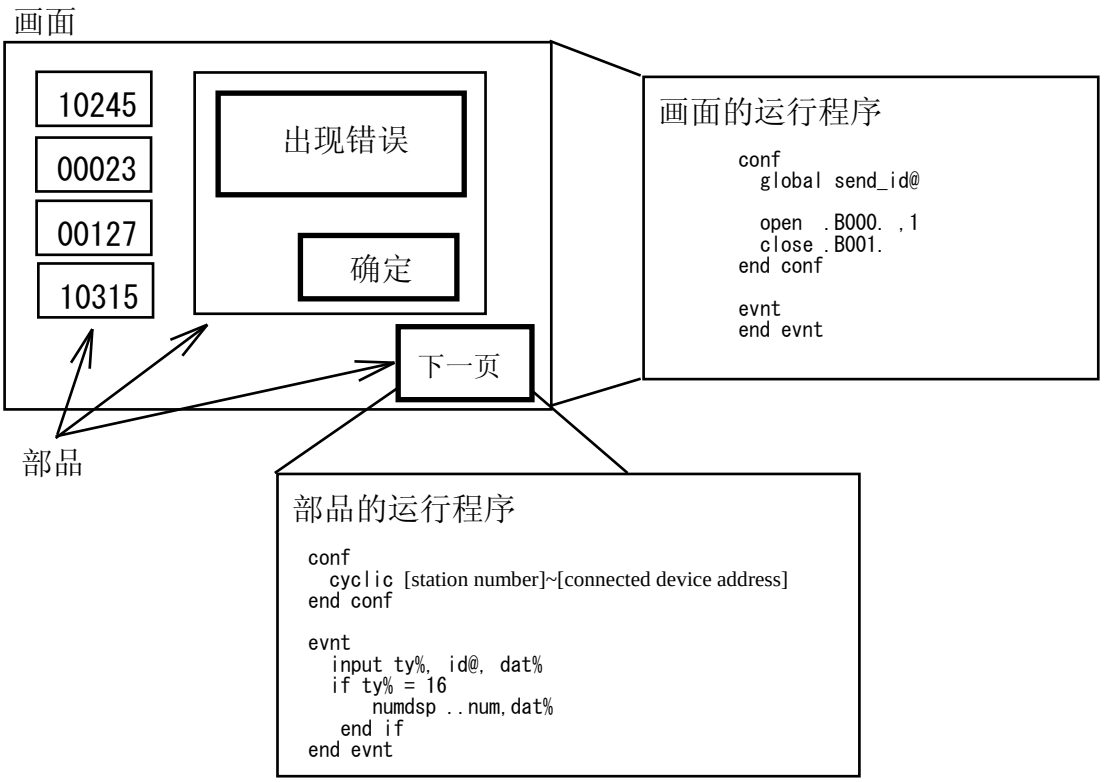
控制程序分部品的控制程序和画面的控制程序 2 种。

（1）部品的控制程序

我们可以为每个部品单独编制控制程序。

（2）画面的控制程序

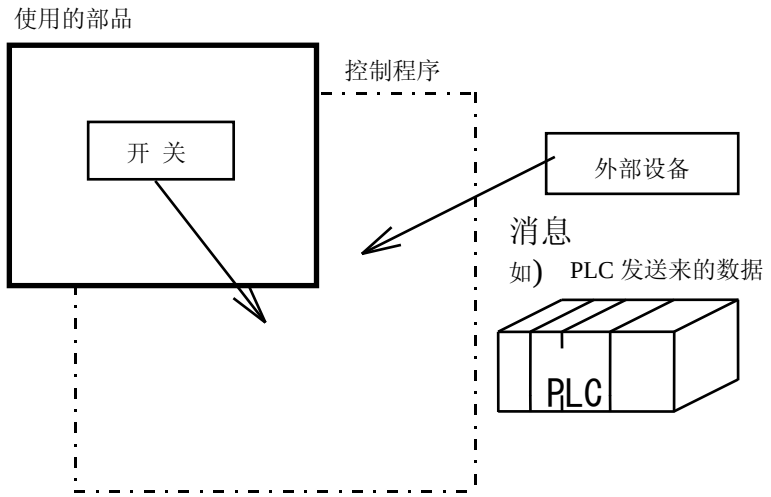
同部品一样，你可以为每幅画面编制控制程序。



（3）控制程序的触发执行

消息是程序执行的触发信号。部品在接收到发给它的消息之后才开始控制程序的执行。开关、定时器、PLC等外部设备都能发出消息。

每条消息包括发送设备的 ID (PLC 设备名、部品名称等)、数据类型、数值等。



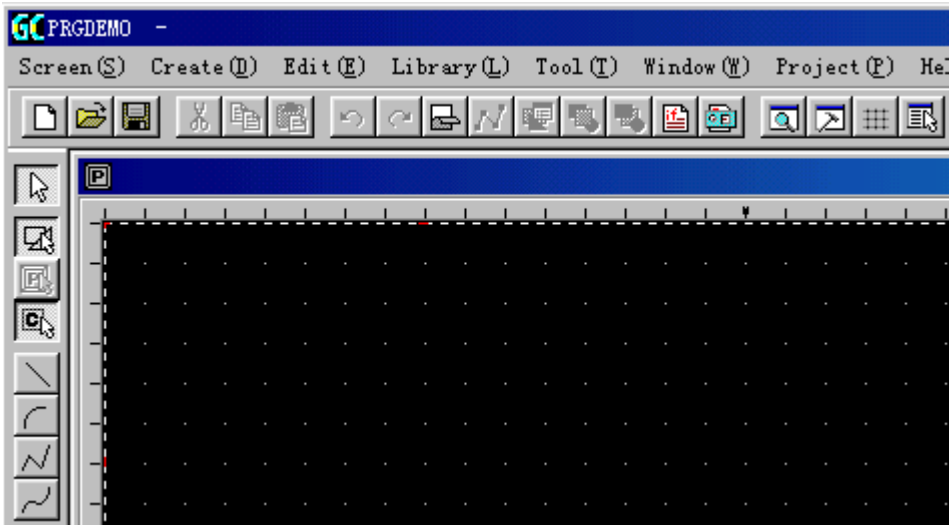
11．2 编程实例

1、创建一个显示数据的部品

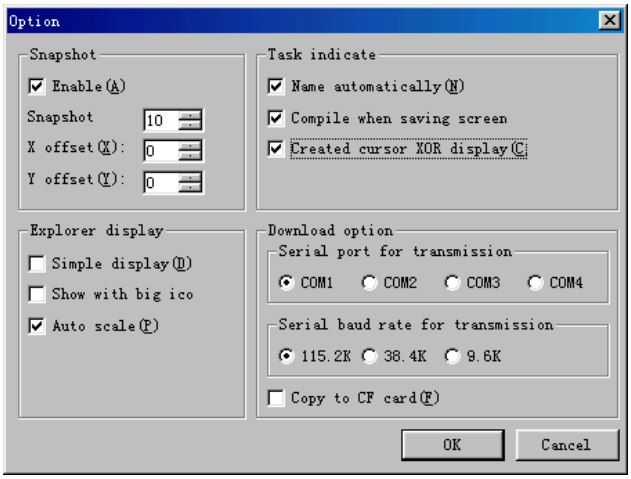


首先创建（或打开）一个工程，如前面章节所述。

然后，点击菜单“Library(L)-New(N)-Parts(P)”，会自动打开一个创建部品的窗口。左上角显示“P”，表示当前窗口是在创建部品（Part）。



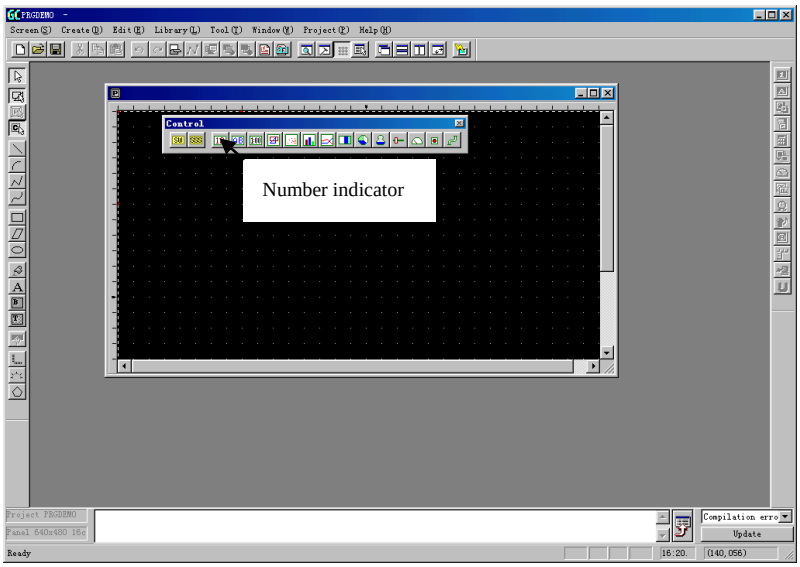
然后点击菜单“Tool－option(O)”，打开选项对话框。



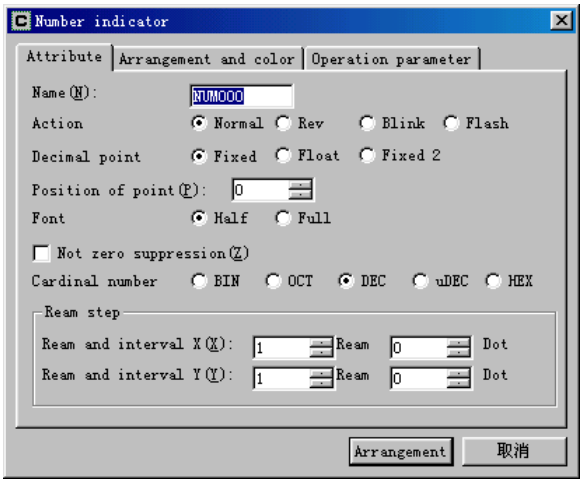
作如上选择，然后点击“OK”。

（1） 控件的放置

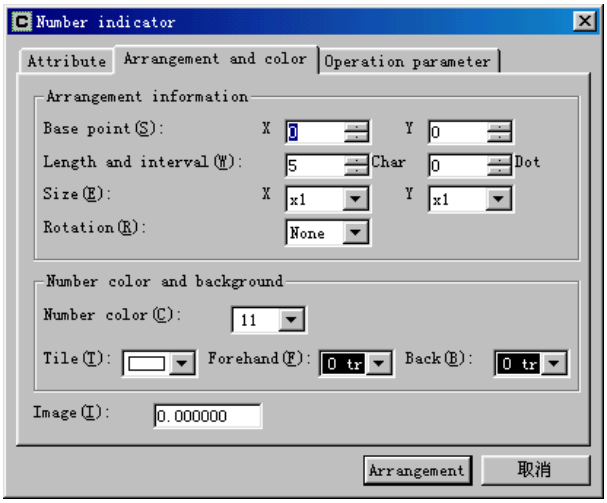
首先选择数字显示控件



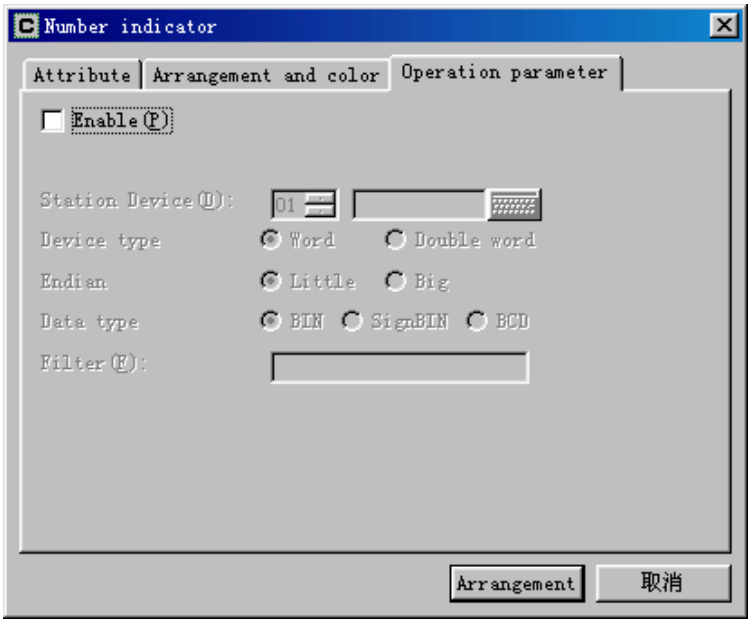
数据显示器属性设置对话框显示如下：



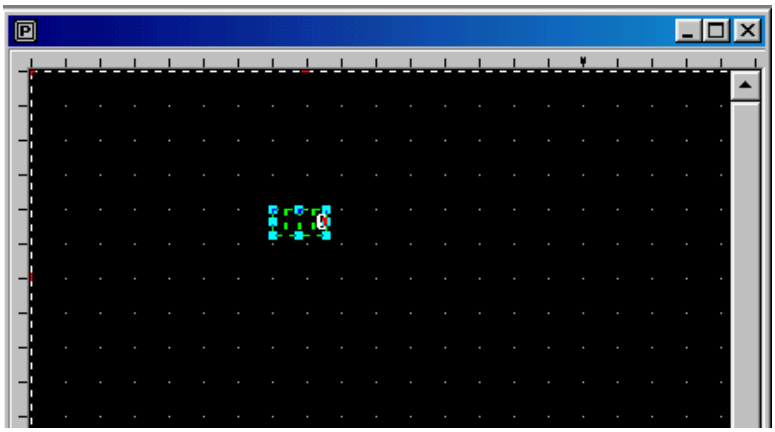
点击“Arrange and color”，显示本控件物理属性：



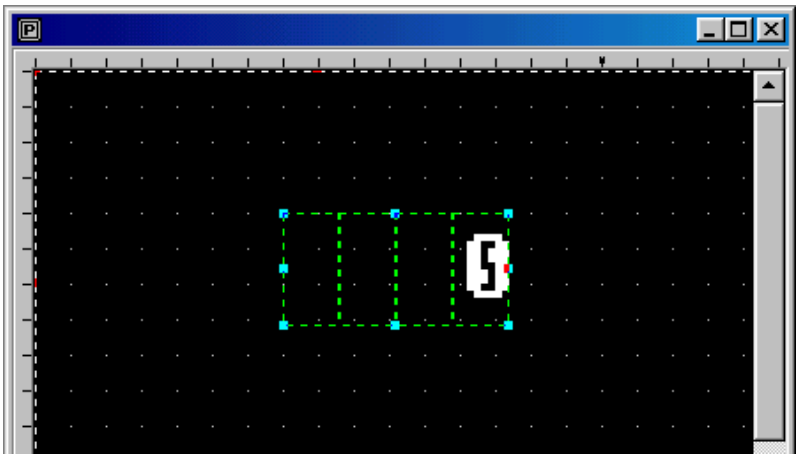
点击“Operation parameter”，不要选定“Enable”，因为其运行要通过程序实现.



点击“Arrange”，将控件放在窗口的某地方

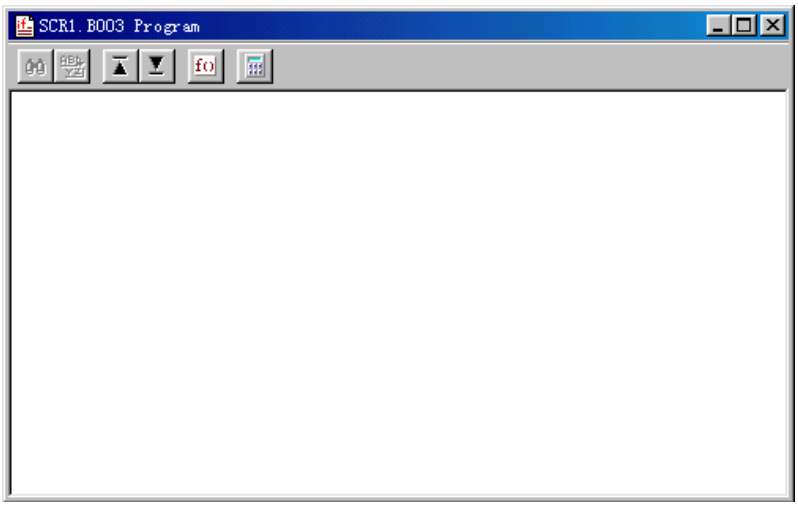


用鼠标拉住控件的右下角，将其拉到适当大小。



（2）编程

要为部品编制程序，可以选择菜单中的“Edit”，然后选择“Edit Part Programs”，或者直接点击编程的快捷方式，会弹出编程窗口如下：



根据要求我们编制程序如下：

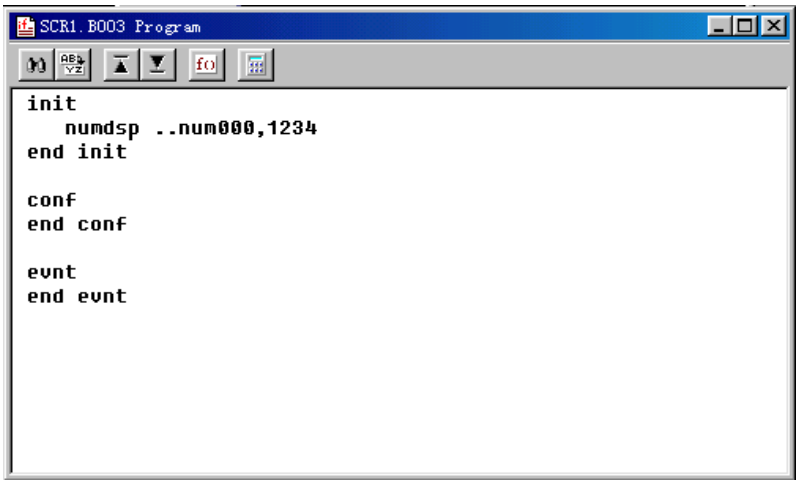
```
init
   numdsp ..NUM000,1234
end init

conf
end conf

evnt

end evnt
```

程序的内容将在后续部分介绍！首先，输入以上程序，程序编辑窗口变得如下：

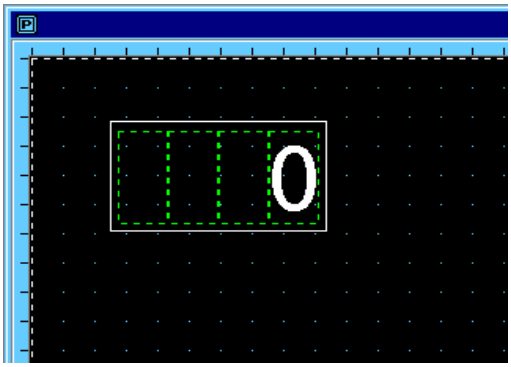


选择“Program”菜单中的“Save”或直接点击，保存上述程序。关闭编程窗口，这时部品编辑窗口重新出现在我们面前。

（3）在部品中绘图

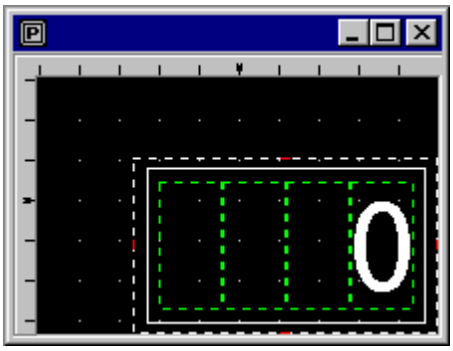
为了美观或需要，我们给数据显示器控价画一个长方形的外框。

选择“Create” 菜单,下拉选择“Rectancle”，然后用鼠标左键点击数据显示控件的左上角，同时按住左键不放，沿着显示控件的对角线拖动鼠标，同时会出现一个矩形框，该框的大小随着鼠标拖动的位置改变。当大小合适时放开鼠标左键。当然，在画框之前你可以选择其属性，诸如颜色、线型等。在画完后，鼠标还处于画框模式状态，这时只要点击鼠标右键即可取消。

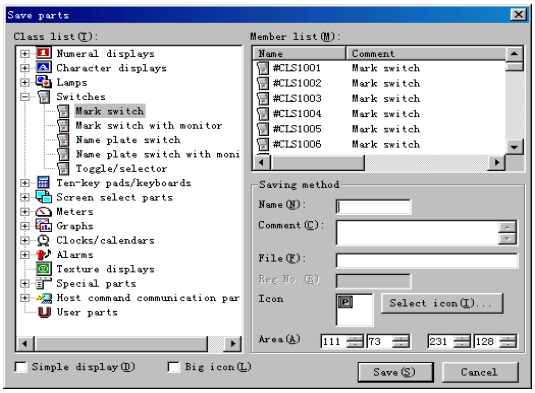


（4）部品的保存

部品的物理大小是由红白相间虚框的大小决定的，所以为了制作大小适当的部品，应将紧贴窗口边框的虚框缩小，缩小的方法是将鼠标放在外框上有红色标记的地方，然后按照鼠标光标箭头方向拖动。注意，虚框不能过小，即必须罩住所有的内容，否则不能正确保存！



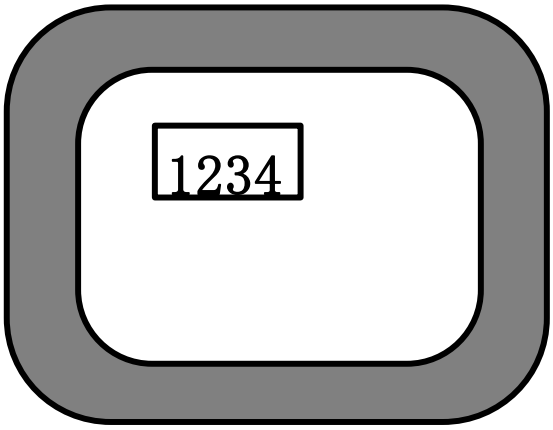
然后，保存部品，这时你既可以通过选择“Library”菜单中的“Save”来保存，也可以通过直接点击快捷方式“”来保存。这时出现如下对话窗口：



我们将部品保存为用户部品“User parts”，部品名称保存为“test”，在“Comment（注释）”中输入“test part”，点击“save”保存部品。
保存后，点击部品窗口右上角关闭按钮，将部品窗口关闭。
这样，一个部品就做好了！

（5）创建部品的调用

同其它部品的使用一样，在画面上直接调用就行。
将其放在画面上下传之后，画面显示如下：



（6）程序解释

程序：

```
init
     numdsp ..NUM000,1234
init init

conf
end conf

evnt
end evnt
```

程序解释如下：

① init～end init

这部分称为初始块（initialization block），这部分程序在程序运行的开始执行，用于对变量的声明和初始化，对于处于 CLOSE 状态的部品，也执行。

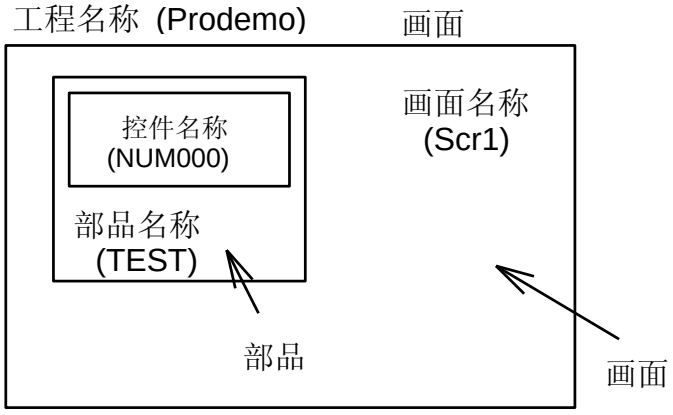
② numdsp ..NUM000,1234

上述语句的功能是将指定的数据显示到指定的数据显示控件。numdsp 命令，是在某个数据显示控件里显示某个数值。..NUM000 用于指出数据显示控件的名称，命名规则如下：

- 控件名及命名规则

Screen（画面）	Scr1..
Scr1 上的部品:	Scr1.Test
画面 Scr1 部品 Test 里的控件：	Scr1.Test.NUM000
当前画面上的当前部品：	..（其它省略）
当前画面上的当前部品里的当前控件：	..NUM000

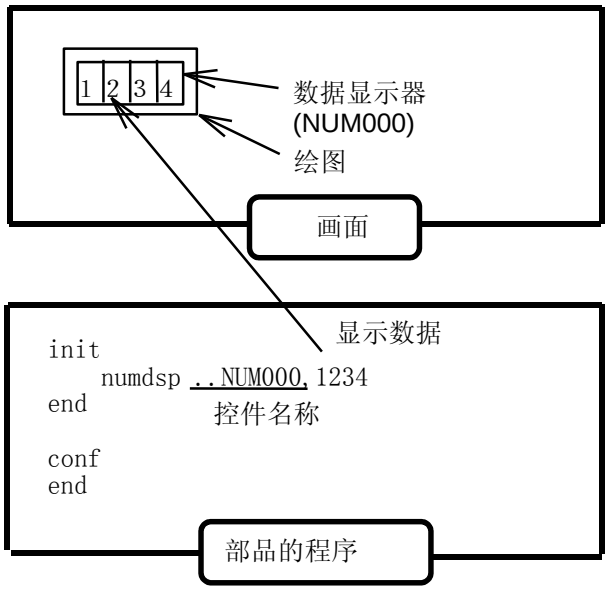
务必注意：部品名称只能是字母和数字，并且只能以字母开头！
控件名中的 2 个“.”不能省略。



程序中用于指定控件 NUM000 的是.

Scr1.TEST.NUM000
or ..NUM000

还有一个参数就是需要显示的数据，这里为“1234”，通过改变它，可以变更显示的数据。



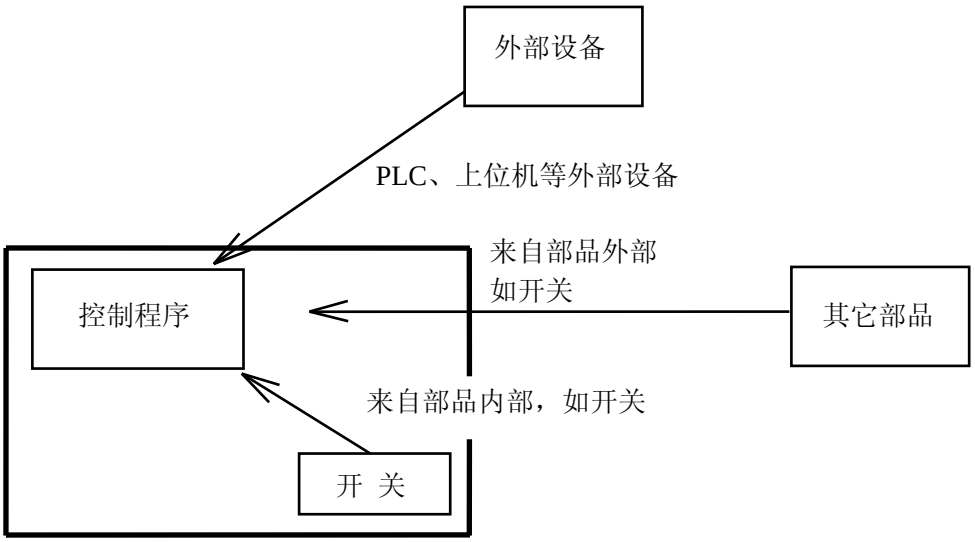
部品中的一些相互关系

NUM000 控件是本部品里的唯一控件。在触摸屏上显示“1234”

③ conf～end conf
本部分称为 Configuration Block(conf 块)。只有当部品已经在画面上显示出来（也就是当前画面的本部品已经打开），本部分程序才执行。本程序中，本部分为空。

④ Evnt～end evnt
本部分称为 evnt block(事件块)，这部分程序只有在收到相关消息之后才执行。

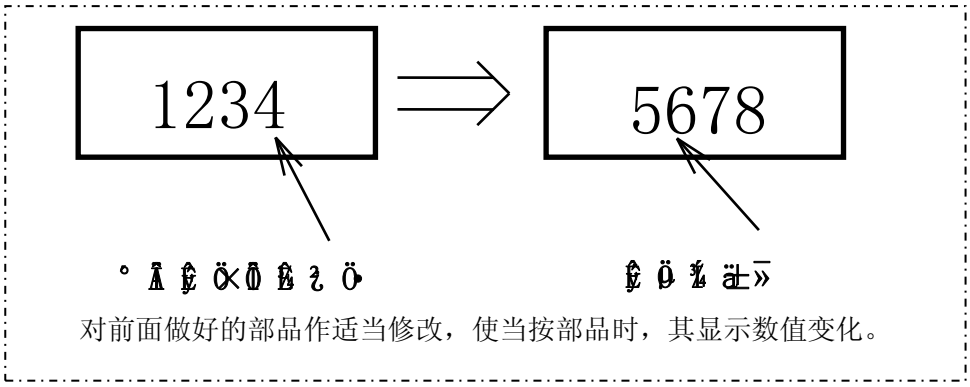
部品的消息来自所连接的 PLC 功能存储器或部品等等，如下图所示：



接受消息的部品

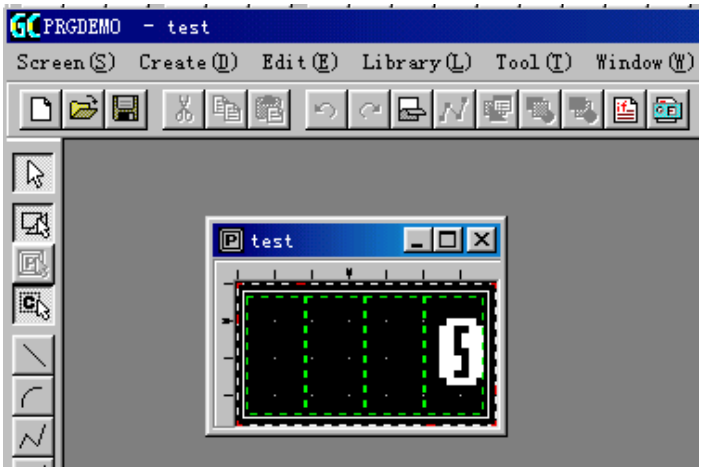
消息的发送

（7）部品的变更



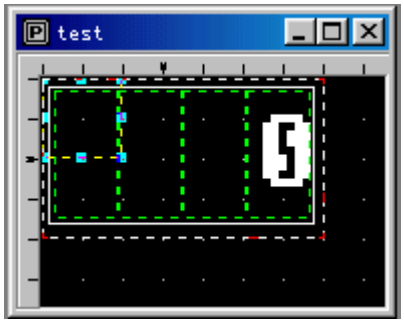
对前面做好的部品作适当修改，使当按部品时，其显示数值变化。

首先，要打开前面做好的部品“Test”。方法是：选择“Library”，选择 Open – Parts，然后打开部品对话框就打开了，选择“User Parts”并打开部品“Test”。



本例中，为了使用触摸面板，要在部品里添加一个开关。通过菜单或通过快捷方式选择“Control”-“Switch”(“控件”－“开关”)。弹出开关控件属性对话框，点击“Arrange”，光标变成小鼠形状。点击部品编辑窗口，光标变成方形开关控件形状，将其放于部品的左上角。然后用鼠标点击本控件，并拖住控件右下角，将其拉至适当大小。注意，因为是开关控件，所以拉大或缩小时只能与 20dots ×20dots 为单位。

注意：因为是默认设置，所以该开关是点动开关（即按下时为“ON”，放手时为“OFF”)。



当将开关控件拉到适当大小之后，接下来是添加部品程序。用同上方法打开程序编辑窗口，输入下述程序：

```
init
    local type%, id@, data%
    numdsp ..NUM000,1234
end init

conf
end conf

evnt
    input type%,id@,data%
    if type%=3 and id@=..SWT000 and data%=1 then
        numdsp ..NUM000,5678
    end if
end evnt
```

通过比较可以看出，

Init block 里对在 Evnt Block 里使用的变量作了声明。

我们主要在 Evnt block 里添加了一段程序。其中 input 是标准函数，用来接受来自开关的消息；消息包括消息发送者类型（type)，消息发送者（id)，及接受到的数据（data)。如果条件满足“if～end if”之间提出的条件，则显示数据变化。

首先，介绍“input”指令：

使用 Input 指令从消息里获得各种信息。“input”指令的标准使用方法如下：

input type%,id@,data%

本例中，使用本指令可以获得信息解释如下：

type%：表示消息发送者类型，3 表示消息来自开关。如果消息来自 PLC 则为 16。目前系统共可提供 1~22 共 22 种消息，更详细的情况，请参考《SC5 用户手册》。

id@：消息发送者身份（ID），如果消息来自开关，则它就是开关控件的名称。ID 的格式是：画面名称（Screen name）、部品名称（Part name）、控件名称（Control name），中间分别用句号（.）隔开。

如：Scr1.test.NUM000

这个 ID 称为 ID 型常量，它是 K-basic 独有的，在处理 ID 时，你可以在其后面加“@”符号，如“ID @”。

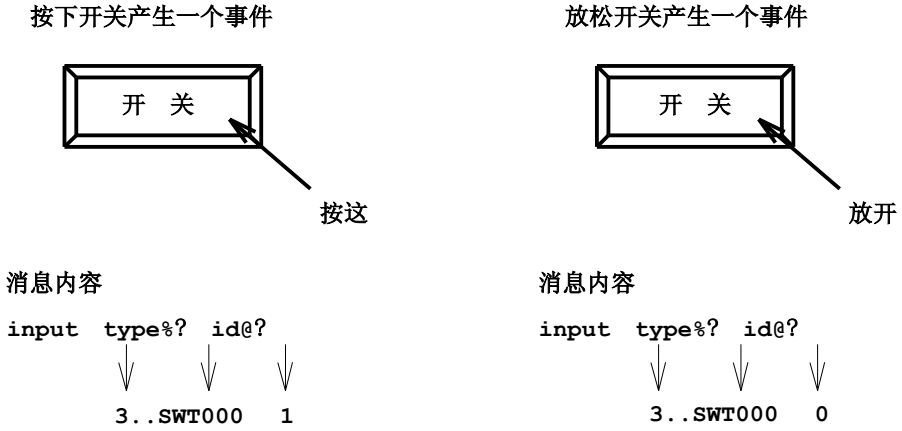
注意：消息包括类型（Type），身份代号（ID），和数据（Data）三部分。

Data%：消息发送者送过来的数据。如开关 ON 时，data%=1，开关 OFF 时，data%=0。

接下来，就是条件判断：

```
if type%=3 and id@=..SWT000 and data%=1 then
        : : : : : : : : : : : : : : : : :
end if
```

表示如果消息来自开关（type%=3），并且消息由SWT000发出（id@=..SWT000），且开关为ON（data%=1）。条件之间用“and”连接，表示条件要同时满足。如果只有type%=3 和id@=SWT000，则条件将满足两次，即手指按下时和手指放开时。本例中，只有在手指按下时程序执行。



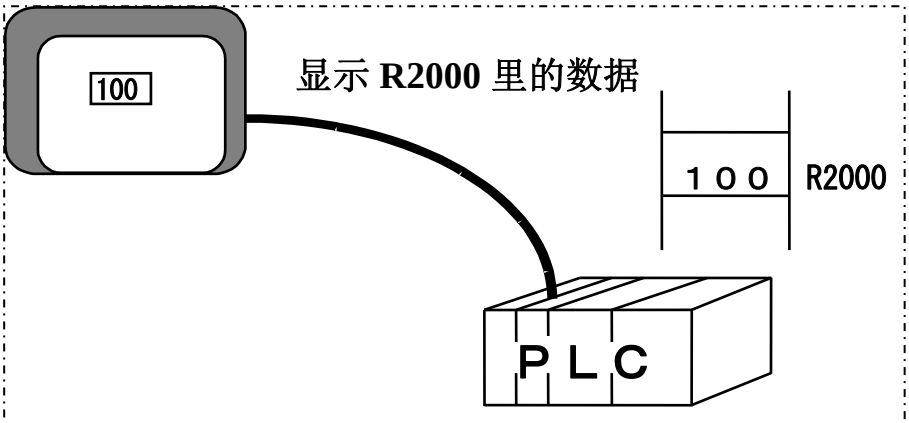
注意：如果开关设置为“momentary”，则开关按下一次将产生两条消息，即按下和放开。

采用同样的方法保存程序，然后调用、下载，就可以看到预期效果！

2．创建一个与 PLC 设备连接的部品

本资料的程序以 Koyo 公司 PLC 为例，如果你使用的是其它公司的产品，应该对局号、PLC 功能存储器名称、还有 PLC 类型进行修改，不过，原理上是完全一致的。

（1）数据显示

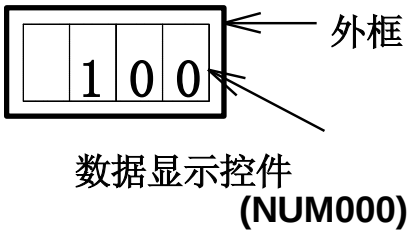


创建一个部品，使其在屏幕上显示对应寄存器里的值。

使用控件：

一个数据显示控件（NUM000）

外观：



程序例如下：

```
init
    local type%, id@, data%

    cyclic 01~R2000

end init

conf
end conf

evnt
    input type%,id@,data%

    if type%=16 and id@=01~R2000 then

       numdsp ..NUM000,data%
    end if
end evnt
```

● Init Block

cyclic 01~R2000　使用"cyclic"命令从"01"号局PLC里的R2000寄存器里读出数据。

"cyclic"指令用来始终监控 PLC 设备的值。

"cyclic"指令循环读取 PLC 设备的值。当 PLC 设备的值发生变化，"cyclic"指令将消息传送给 Evnt Block。Cyclic 后面依次接局号、设备名。中间使用"~"连接。

该指令即使在画面已经发生变化时，也进行数据传送。

● Conf Block

这里不作任何处理。

● Evnt Block

```
input type%,id@,data%
```

"input"指令用来读取传送给部品的消息。消息的按如下顺序格式："type%"（16）、id@（01~R2000）、data%(PLC 里的值：01~R2000)。

```
if type%=16 and id@=01~r2000 then
   : : : : : : : : : : : : : : : : :
end if
```

条件 type%=16，表示消息来自 PLC，id@=01~r2000 表示发送消息者是 01~r2000。中间用 and 连接表示两个条件同时满足时，后续程序才能执行。

```
numdsp ..NUM000,data%
```

"numdsp"指令的功能是在显示控件"num000"里显示变量"data%"的值。其值就是使用 input 指令从消息里读出的"data%"值。

程序的流程是：在初始块（init block）里使用"cyclic"指令始终监控着 01 号 PLC 里寄存器 R2000 里的值。当里面的值发生变化时，使用 input 指令来接收消息，并将最新的值送到数据显示控件 NUM000 里。

试着使 PLC 里 R2000 的值发生变化，你将看到屏幕上的值也将同时发生变化。

● 如何使部品更容易使用

如果你希望监控 PLC 多个寄存器里的值，你必须重新制作多个部品并分别编程，因为每个寄存器的名称都不一样。你是否觉得这很浪费时间呢？其实，若使用"参数"功能便可以使这个问题变得非常简单。

Cyclic [局号]~[寄存器号]

用中括号括起来的字符串称为 template（模板），一个模板里最多可写 32 个半角字符。因为写在括号里的字符串能在相应部品的模板（template）里显示，因此，这些部品也可以像标准部品一样使用。这样，对于不同的 PLC 内部寄存器，你只要在外部改变寄存器号即可，同标准部品一样方便。

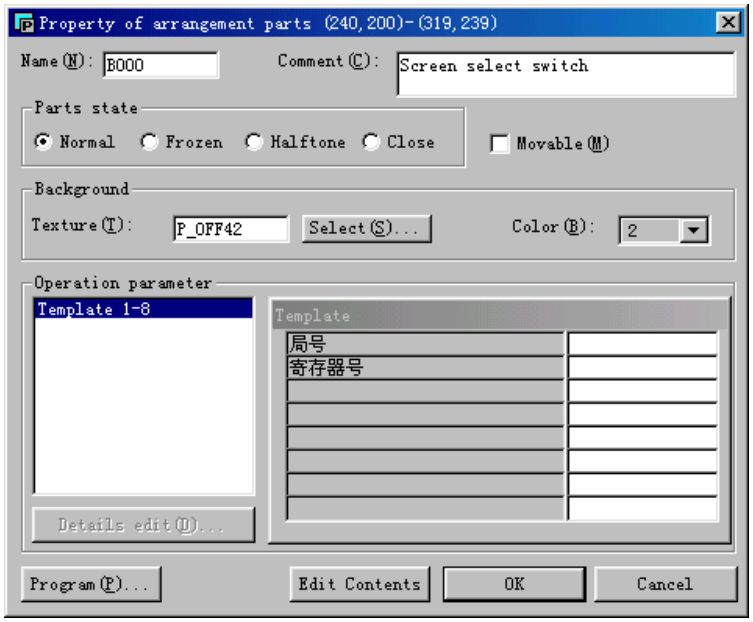
使用模板（template）的程序实例如下：

```
init
    local type%, id@, data%
    cyclic [局号]~[寄存器号]
end init

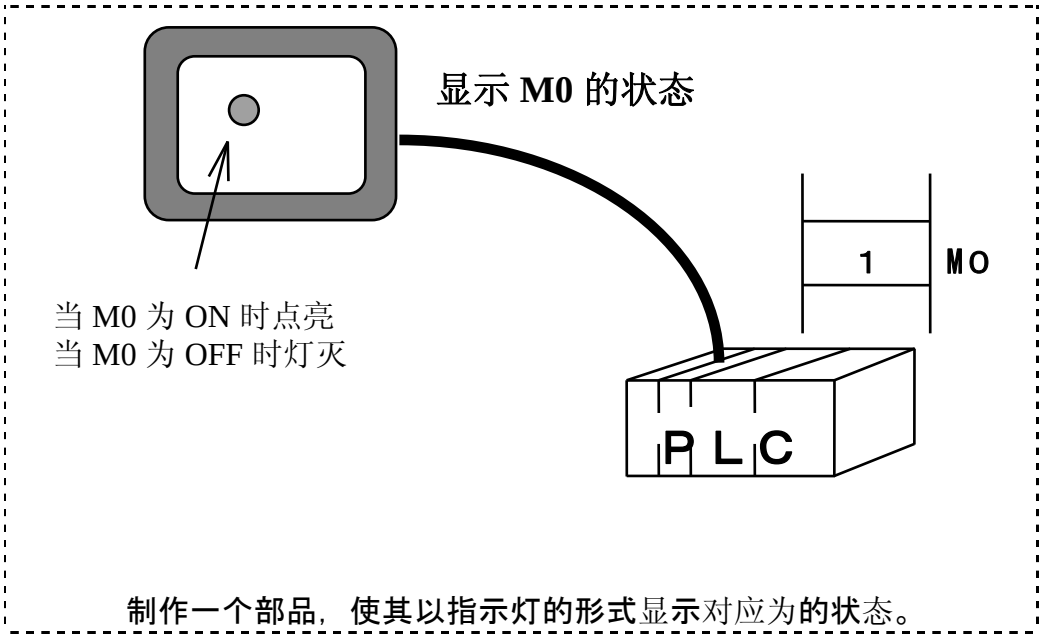
conf
end conf

evnt
    input type%,id@,data%
    if type%=16 and id@=[局号]~[寄存器号] then
      numdsp ..NUM000,data%
    end if
end evnt
```

“局号”和“寄存器号”作为部品的属性（property of part）显示在部品属性对话框里，在使用时可以为每个放置的部品输入不同的值（寄存器号）。



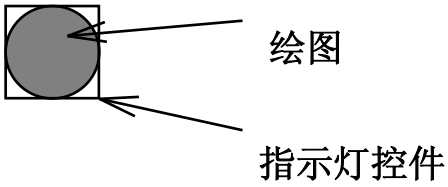
（2）指示灯



使用的控件：

指示灯控件（LAM000）

外形：



注意：指示灯由 OFF 变 ON 时，其颜色也相应地由“OFF 颜色”变成“ON 颜色”。所以给绘图填充颜色时必须用“OFF 颜色”。

与 PLC 相连的指示灯程序如下：

```
init
    local type%, id@, data%
    cyclic [局号]~[与指示灯相连的设备地址]
end init

conf
end conf

evnt
    input type%,id@,data%
    if type%=16 and id@= [局号]~[与指示灯相连的设备地址] then
      lampdsp ..LAM000,data%
    end if
end evnt
```

● Initialization Block

同前面一样，在初始块里使用Cyclic指令。

● Configuration Block

这里什么也不处理!

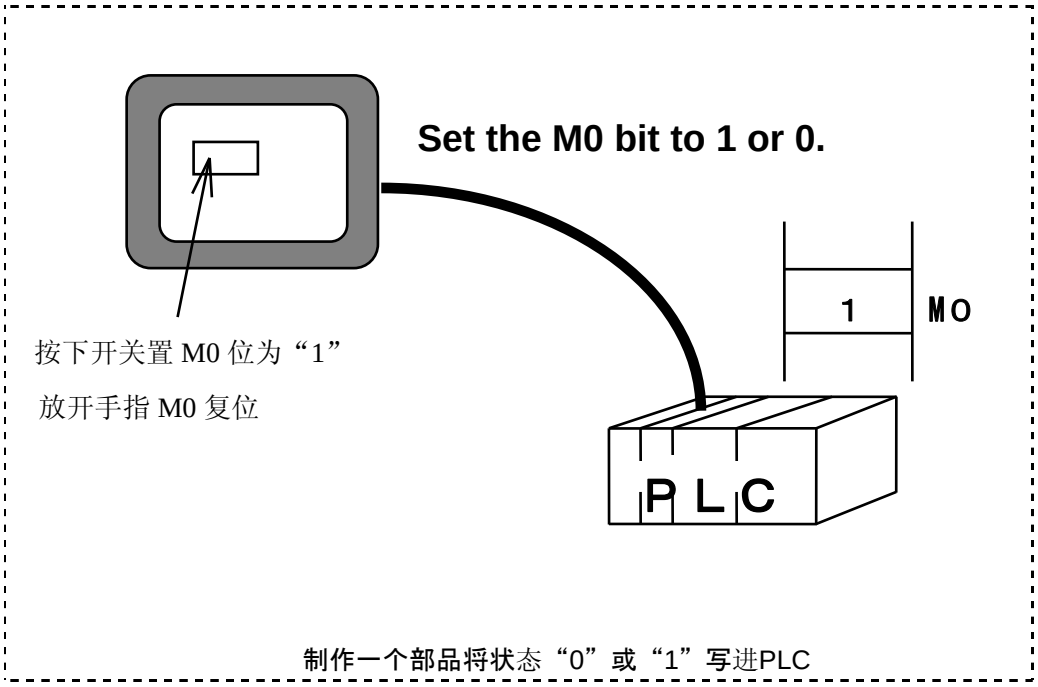
● Event Block

lampdsp ..LAM000,data%

“Lampdsp”指令使用指示灯显示 ON/OFF 状态，当数据是“0”时，指示灯显示“OFF”颜色，当数据为“1”时，指示灯显示“ON”颜色。

试图改变相应内部继电器的状态，观察指示灯颜色的改变。

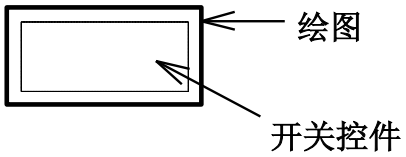
（3）开关



使用控件：

一个开关控件SWT000

外观：



程序如下：

```
init
    local type%, id@, data%
end init

conf
end conf

evnt
   input type%,id@,data%
   if type%=3 and id@=..SWT000 and data%=1 then
      [局号]~[连接设备]=1
   else if type%=3 and id@=..SWT000 and data%=0 then
      [局号]~[连接设备]=0
    end if
end evnt
```

- 初始块（Initialization Block）

  因为是将值写入PLC，所以没有必要使用“Cyclic”指令。

- Configuration Block

  没有处理任务。

- Event Block

```
input type%, id@, data%
```

input指令读取来自开关的消息，读取顺序为“type%=3”、 “id@=...SWT000.”、data%” =ON/OFF (1 or 0)

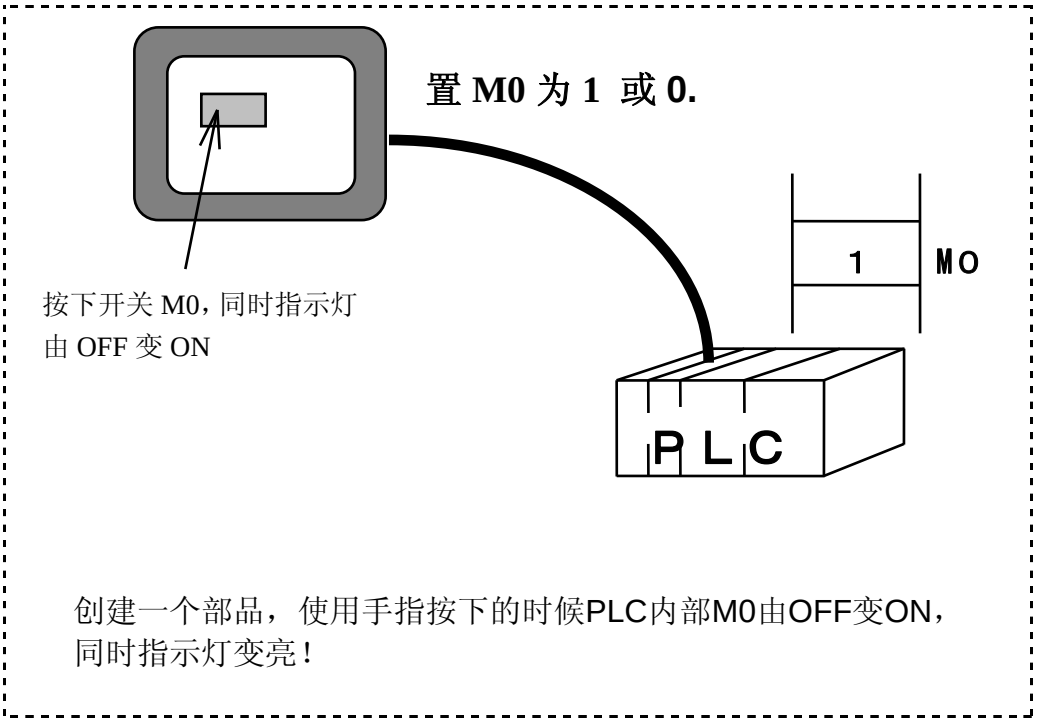
```
if type%=3 and id@=..SWT000 and data%=1 then
     : : : : : : : : : : : : : : :
else if type%=3 and id@=..SWT00 and data%=0 then
     : : : : : : : : : : : : : : :
end if
```

解释同前面所述。

[局号]~[连接设备]=1，[局号]~[连接设备]=0

表示将1或0写进“PLC”。

（4）带指示灯的开关

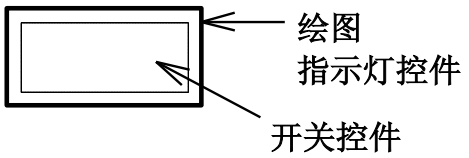


创建一个部品，使用手指按下的时候PLC内部M0由OFF变ON，同时指示灯变亮！

使用控件：

开关控件SWT000和指示灯控件LAM000

外观：



指示灯控件和开关控件相互重叠。

程序如下：

```
init
    local type%, id@, data%

    cyclic[局号] ~ [连接设备地址]

end init

conf

end conf
```

```
evnt
    input type%,id@,data%
    if type%=3 and id@=..SWT000 and data%=1 then
        [局号]~[连接设备地址] = 1
    else if type%=3 and id@=..SWT000 and data%=0 then
       [局号]~[连接设备地址]=1
    else if type%=16 and id@=[局号]~[连接设备地址] then
        lampdsp ..NUM000,data%
    end if
end evnt
```

- Initialization Block

本程序使用cyclic指令监控PLC内部继电器的状态，内部继电器的状态将决定指示灯的状态是ON还是OFF。

- Configuration Block

没有处理内容。

- Event Block

```
    input type@,id@,data%
```

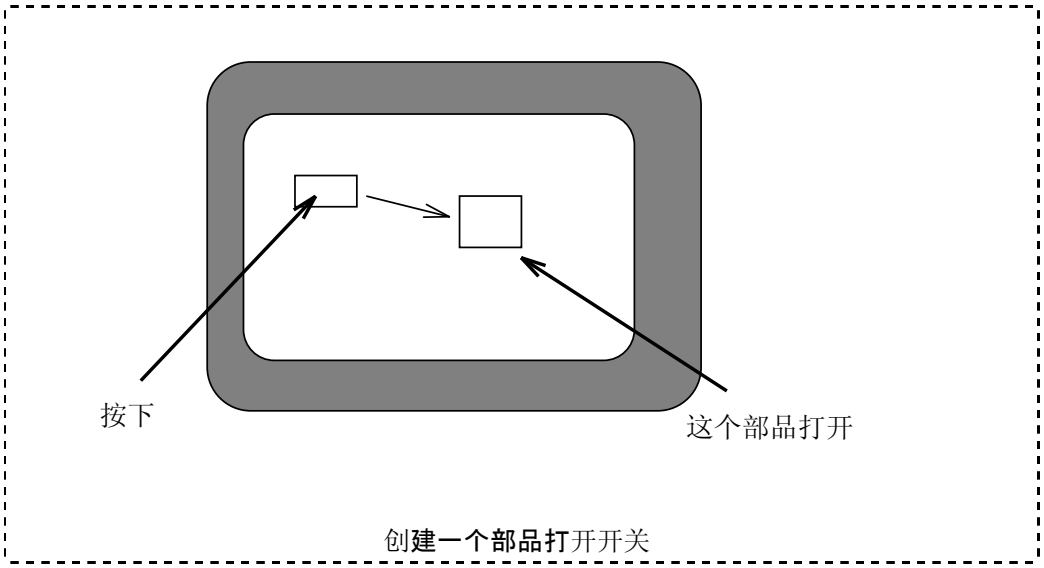
同前面讲述的一样。

```
    if type%=3 and id@=..SWT00 and data%=1 then
         : : : : : : : : : : : : : :
    else if type%=3 and id@=..SWT000 and data%=0 then
         : : : : : : : : : : : : : :
    else if type%=16 and id@=[station-number]~[connected-device-address]
         : : : : : : : : : : : : : :
    end if
```

在本部分，来自开关的消息写进PLC设备，同时来自PLC的消息将决定指示灯的ON/OFF状态。

3. 制作一个部品来控制其它的部品

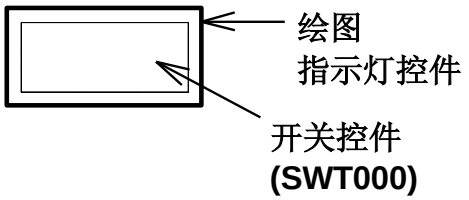
（1） 从画面上调用其它部品



使用的控件：

一个开关控件SWT000

外观：



从触摸屏上调用其它的部品的程序如下：

```
init
    local type%, id@, data%
end init

conf
end conf

evnt
   input type%,id@,data%
   if type%=3 and id@=..SWT000 and data%=1 then
      open .[要打开的部品名称].,1
   end if
end evnt
```

● Initialization Block

除了声明局部变量之外，没作任何处理。

● Configuration Block

```
input type%,id@,data%
```

“input”指令从开关控件读取信息。

```
if type%=3 and id@=..SWT000 and data%=1 then
     : : : : : : : : : : : : :
end if
```

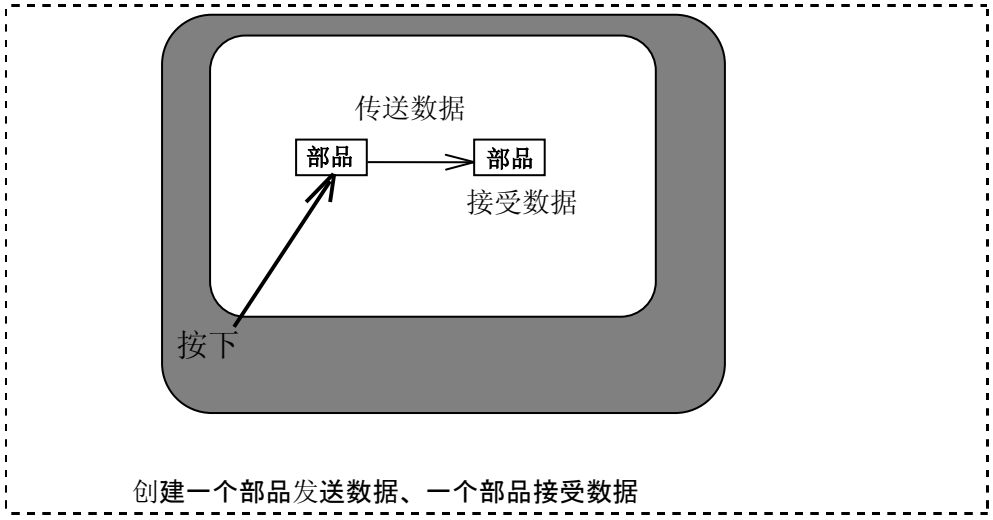
“if”与“end if”之间的内容，只用在开关被按下时才执行。

```
open .[要打开的部品名称].,1
```

“open”指令将由“ID”指出的部品由“Close”状态打开。

如果部品名称后面是“1”，则被打开部品的Configuration Block在部品被打开的时候被执行；反之不执行。被打开的部品名称采用参数形式使部品调用时更加方便。

（2）从（向）其它部品接收（发送）数据



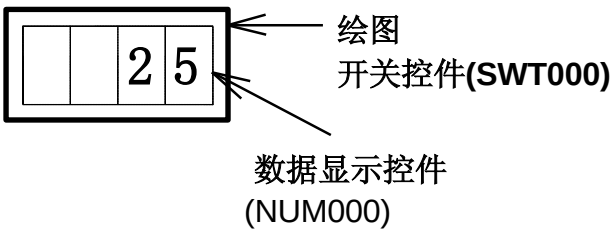
创建一个部品发送数据、一个部品接受数据

首先创建一个发送数据的部品：

使用的控件：

一个数据显示控件（NUM000）和一个开关控件（SWT000）

外观：



部品用于数据发送的程序如下：

```
init
    local type%, id@, data%
conf
   numdsp ..NUM000,[要显示的数据]
end init

conf
end conf

evnt
   input type%,id@,data%
   if type%=3 and id@=..SWT000 and data%=1 then
      print [要显示的数据]
      send .[目标部品名称].
   end if
end evnt
```

- Initialization Block

  numdsp ..NUM000,[要显示的数据]

  “numdsp”为数据显示命令。

- Configuration Block

  没有处理内容！

- Event Block

```
   input type%,id@,data%
```

“input”指令前面已讲过，是从开关控件里读取消息。

```
   if type%=3 and id@=..SWT000 and data%=1 then
      print [要显示的数据]
      send .[目标部品名称].
   End if
```

“print”和“send”指令在开关被按下时执行。

print [要显示的数据]

“print”指令用于将消息传送给其它部品，消息包括“type”、“ID”、“显示的数值”。如果需要传送两个或两个以上的数据，可以将它们连写并用逗号隔开，如：

例：print 123,234,345

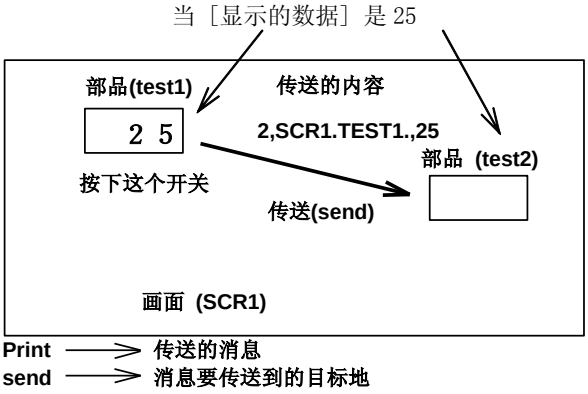
这时，“input”指令要读取3条数据消息，如下：

input type%,id@,data1%,data2%,data3%

其中，data1%读取“123”，data2%读取“234”，data3%读取“345”。

send .[目标部品名称].

Send指令将print指令传送来的数据送给指定的部品（[目标部品名称]），记住：“print”和“send”指令要同时使用。



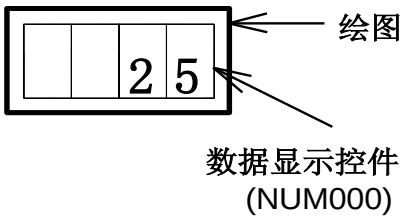
程序到此结束。当开关按下时，程序将包括参数[要显示地数据]、[目标部品名称]在内的消息发送出去。

然后，创建一个接收数据的部品：

使用的控件：

一个数据显示控件（NUM000）

外观：



部品用于数据接收的程序如下：

```
init
    local type%, id@, data%
end init

conf
end conf

evnt
   input type%,id@,data%
   if type%=2 then
      numdsp ..NUM000,data%
   end if
end evnt
```

- Initialization Block
  定义一个局部变量
- Configuration Block
  没有处理内容！
- Event Block

```
input type%,id@,data%
```

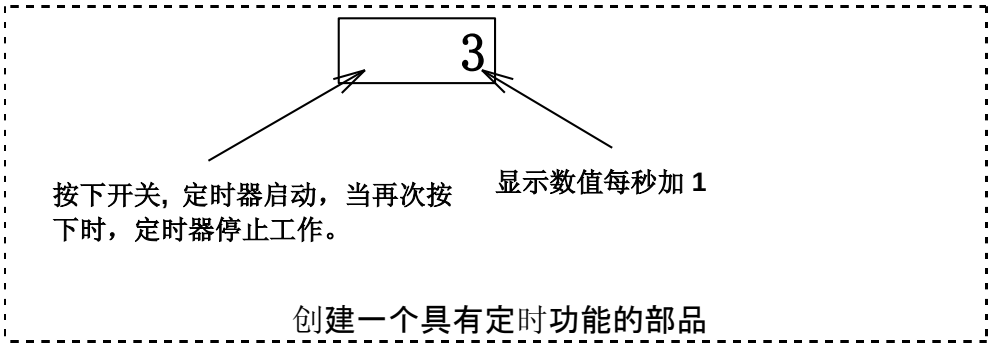
“input”指令前面已讲过，是从开关控件里读取消息。

```
if type%=2 then
   numdsp ..NUM000,data%
end if
```

“type%=2”的意思是从部品接收消息，显示的数据由“data%”给出。
程序结束。当test1将数据送过来时，本部品显示test1送来的数据。

同时使用这两个部品，并恰当填写参数，按下开关运行时，两者显示相同的数据！

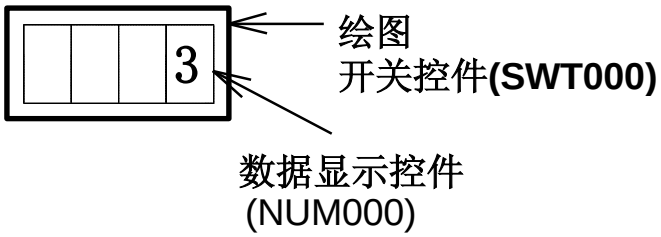
4. 创建一个使用定时器的部品



使用的控件：

一个数据显示控件（NUM000）和一个开关控件（SWT000）

外观：



使部品显示的数据自动加1的程序如下：

```
init
    local type%, id@, data%
    static timeid@
    static flag%
    static number%
    flag%=0
    numdsp ..NUM000,0
end init

conf
end conf
```

```
evnt
    input type%,id@,data%
    if type%=3 and id@=..SWT000 and data%=1 then
        if flag%=0 then
            timeid@=opentim()
            settim timeid@,10,1
            starttim timeid@
            flag%=1
        else if flag%=1 then
            stoptim timeid@
            closetim timeid@
            flag%=0
        end if
    else if type%=4 then
        number%=number%+1
        numdsp ..NUM000,number%
    end if
end evnt
```

● Initialization Block
在本程序块中定义了局部变量和静态变量。

```
static timeid@
static flag%
static number%
```

使用“static”指令使在程序执行的过程中变量的内容得以保留。本例中，本指令保留定时器的ID、定时器的ON/OFF标志、显示的数值。也可以将许多“static”指令合并书写。即将各参数连写，中间用逗号隔开。如下：

```
例：static timeid@,flag%,number%
```

flag%=0
“flag%=0”是将定时器ON/OFF标志初始化。

```
numdsp ..NUM000,0
```
开始时，在数据显示器里显示“0”。

● Configuration Block
没有处理内容!

● Event Block

```
input type%,id@,data%
```

“input”指令从开关和定时器读取消息。

```
if type%=3 and id@=..SWT000 and data%=1 then
       : : : : : : : :   ← 当开关按下的时候执行
else if type%=4 then
       : : : : : : : :   ← 当读到来自定时器的消息时执行
end if
```

当开关按下或当接收到来自定时器的消息分别执行“then”后面的操作。每接到一次来自定时器的消息，显示器计数一次。（消息每秒钟传送一次。）

当按下开关时执行如下程序：

```
if type%=3 and id@=..SWT000 and data%=1 then
   if flag%=0 then
        timeid@=opentime()
        settim timeid@,10,1
        starttim timeid@
        flag%=1
   else if flag%=1 then
        stoptim timeid@
        closetim timeid@
        flag%=0
   end if
```

当定时器停止（flag%=0）时，紧接“if flag%=0 fthen”后的操作执行。

```
timeid@=opentim()
```

本函数是获得定时器的ID，定时器的ID用“timeid@”标示。

```
settim timeid@,10,1
```

设置定时时间。定时时间的最小单位为100ms，10表示定时时间为100ms×10＝1s。其后面的“1”表示定时器能反复触发时间。

```
starttim timeid@
```

启动定时器。

以上三条指令需一块联合使用。

```
flag%=1
```

Flag用来标识定时器的状态，并保持其状态。flag%=1表示定时器正在运行。

“else if”之后的程序用于停止定时器的运行。

```
stoptim timeid@
```

使定时器停止向上计数。

```
closetim timeid@
```

关闭“opentim”打开的定时器，并将其返回给系统。

注意：一个工程中最多可以同时使用16个定时器，其编号为0~15。不使用的定时器应及时返回给系统。

```
flag%=0
```

因为定时器已经停止计时，所以将标志定时器状态的标识置“0”.

当收到来自定时器的消息时，执行如下程序：

```
else if type%=4 then
    number%=number%+1
    numdsp ..NUM000,number%
end if
```

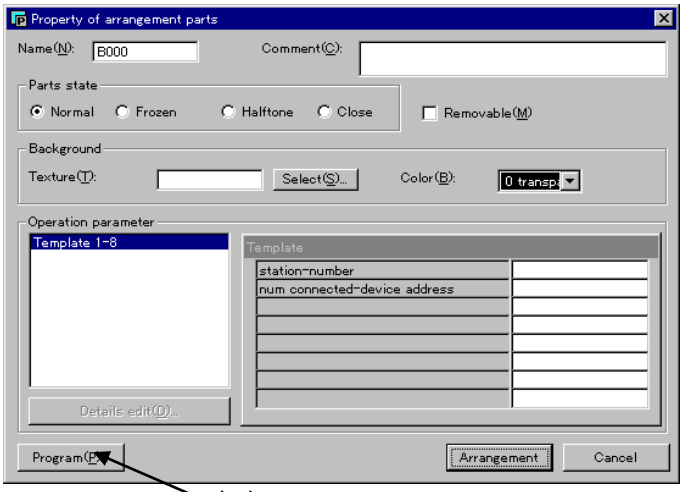
每当定时器发送一次消息，数据显示器单元显示的数值加“1”。

程序到此结束!

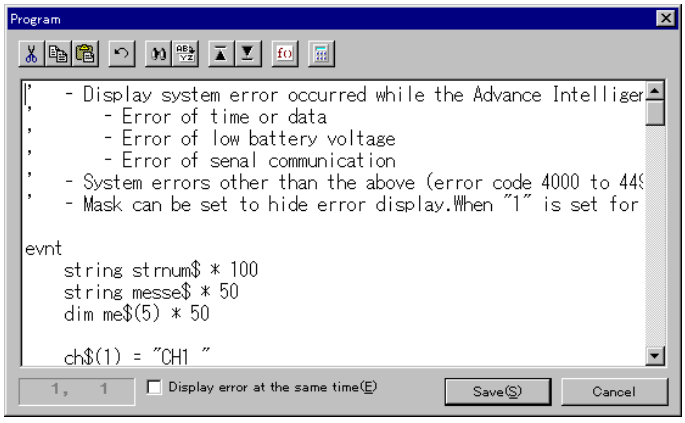
5．为画面上的部品编程

如果有需要，你可以给任何部品添加控制程序，或修改其原有的控制程序。

进入画面上的部品程序编辑状态的方法是：双击部品 → 弹出部品属性窗口 → 选择“program”进入编程窗口。



打开程序编辑窗口



此时，你可以添加或修改控制程序了。

附录：EA7E-CAIP 硬件规格相关

1.. 性能仕様

項目	仕様	
型号	EA7E-TW7CL-C	EA7E-TW10CL-C
CPU	SAMSUNG S3C2416 16/32-bit RISC CPU (400MHz)	
电池	使用RB-10(CR2354)	
电池寿命	約5年(25℃)	
停电保持	时间·日历、SRAM	
系统存储	DDR2 64MB	
系统闪存	256MB	
备用存储	256KB(控制数据保存用、SRAM)	
最大画面数	999	
时间功能	内蔵(PLC的时间功能也可以利用)、月差±60秒以内	
日历功能	有、至2099年有效	
节电功能	背光灯关闭(根据画面工具的设定)	
串行接口	RS-232/422/485、D-sub15针接插件	
USB端口A	USB设备用端口(可连接 参照自选清单) 无效	
USB端口B	与PC之间的程序数据传送用端口	
蜂鸣器输出	有 発振周波数2300±300Hz(但是，根据使用环境的不同，音质会发生变化。)	

2.. 显示性能仕様

項目	仕様	
型号	EA7E-C-AIP	EA7E-TW10CL-C
液晶尺寸	154.08× 85.92	235.0X145.8
液晶类型	TFT	
显示色	64K 彩色	
显示区域	154.08 × 85.92	**222.0X132.48**
分辨率	800×480 点阵(WVGA)	
亮度	170 cd/m2	
点距	0.0642×0.1790	0.0925X0.276
背光灯	LED 平均寿命:連続点灯 20,000 小时以上／25℃ ※※	
背光灯交换	不可	
触摸屏	模拟型(分辨率1024×1024)	

※※：背光灯的平均寿命是指在 25℃开始连续点灯的时间到亮度衰减为 50%的经过时间。

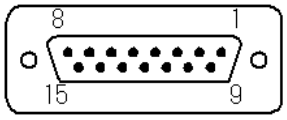
3.. 硬件構成

3-1. D-Sub通信端口

下述通信設定依据程序软件的设定方式。

項目	仕様
通信方式	RS-232/422/485、非绝缘
传送比率	2400,4800,9600,19200,38400bps,57600bps,115200bps,187500bps 数据长度:8/7 位、停止位:2/1 位、奇偶校验:無/偶数/奇数
端子形状	D-SUB15 针(母)
适合接插件	D-SUB15 针 公

端口定义



针 号	信号名	针 号	信号名
1	Frame Ground	9	RXD+ (422/485)
2	TXD (232)	10	RXD- (422/485)
3	RXD (232)	11	TXD+ (422/485)
4	Vcc (DH-485)	12	TXD- (422/485)
5	Logic GND	13	终端电阻 (连接 9 号针
6	LE (DH-485)		
7	CTS (232)	14	空
8	RTS (232)	15	空
		SHELL	Frame Ground

※4号端子的Vcc为DH-485通信专用，不可作为其他的用途。

※空端子不可做任何连接。

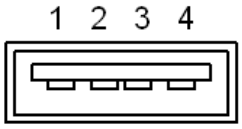
适应PLC通信电缆(另售)

型号	接插件形状	对応PLC
EA-2CBL	模块 6 極(RJ12)	Direct Logic PLC DL05、DL105、DL205、DL350、DL450 用(RS-232)
EA-2CBL-1	高密度 D-Sub 15 针(公)	Direct Logic PLC DL250 用(RS-232)
EA-4CBL-1	D-Sub15 针(公)	Direct Logic PLC DL405 用(RS-232)
EA-4CBL-2	D-Sub25 针(公)	Direct Logic PLC DL405、DL350、DL305 DCU、全部 DCM 用(RS-232)
EA-MITSU-CBL	D-Sub25 针(公)	三菱 FX 系列用(RS-422)
EA-MITSU-CBL-1	小型 DIN8 针	三菱 FX 系列用(RS-422)

3-2. USB A端口

项目	规格
规格	依据 Universal Serial Bus ver1.1
工作电压	5V、最大 500mA
关于插头形状的注意	根据不同形状的插头的物理干涉，邻接的端口可能会不能使用。请根据外形图确认。

端口定义

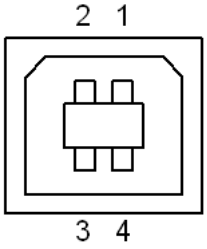


针 号	信号名
1	Vbus
2	D-
3	D+
4	GND
SHELL	Shield

3-3. USB B端口

項目	仕様
規格	依据 Universal Serial Bus ver2.0
关于插头形状的注意	根据不同形状的插头的物理干涉，邻接的端口可能会不能使用。请根据外形图确认。

端口定义



针 号	信号名
1	Vbus
2	D-
3	D+
4	GND
SHELL	Shield

3-4. LED灯

(1)电源灯(绿)

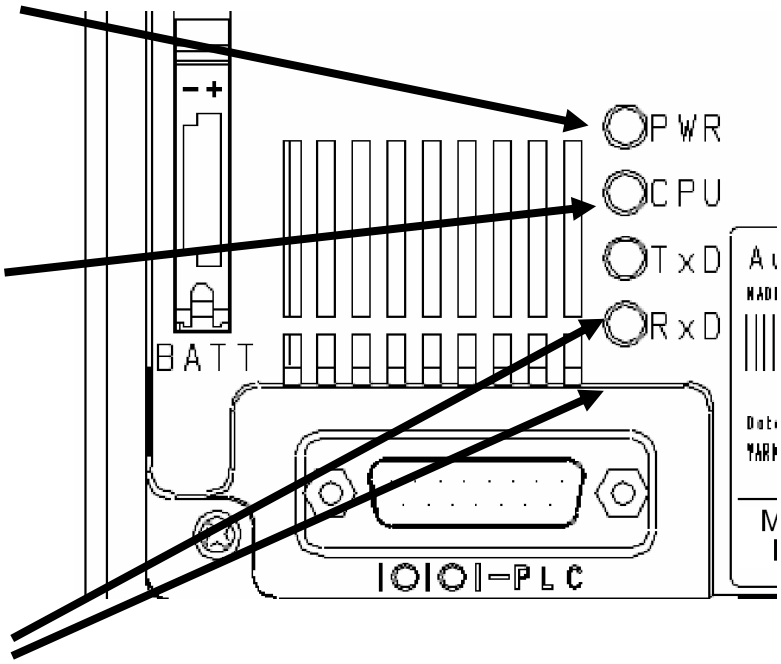
点灯色	説明
消灯	无电源
绿色灯	有点怨

(2)状态灯(绿、橙、赤)

点灯色	説明 （*）
消灯	无电源
绿灯	正常工作
红灯	没有操作系统
黄灯	电源电压低下

(3)RXD·TXD灯(绿)

点灯色	説明
消灯	未通信
绿点灯	数据通信中



（*） 系统画面→Information→Boot Loader 版本号2001（含）以上

并且系统画面→Information→OS 的时间显示Jul 1 2011 at 12:14:03(含) 以后

3-5. 蜂鸣器

EA7本体搭载了单一的蜂鸣器，可以在系统安装工具里进行控制。

3-6. DC电源插头

項目	仕様
使用插头	EA-DC-CON (市販品:フェニックスコンタクト社製 MSTB2,5/5-STF-5,08 ※)
电缆要求	単線:0.2mm^2～2.5mm^2、撚線:0.2mm^2～2.5mm^2、AWG24～12 線長:7mm
端子的安装扭矩	接続用螺钉:0.5～0.6Nm 法兰盘用螺钉:0.4～0.5Nm

※使用市场销售的物品时请注意：

插头的1・2号端子上不能有任何连接，可能会引起故障。

端口定义



针 号	信号名
1	禁止连接
2	禁止连接
3	+24V
4	GND
5	Frame Ground

光洋电子(无锡)有限公司
Koyo ELECTRONICS （WUXI） CO., LTD.
地址：江苏省无锡市滨湖区建筑西路 599 号 1 栋 21 层
邮编：214072
电话：0510-85167888 传真：0510-85161393
http：//www.koyoele.com.cn

KEW-M9018B
2015 年 8 月