



DUP-B 系列人机界面使用手册

DUP-B 系列人机界面使用手册



中达电通股份有限公司
地址：上海市浦东新区民夏路238号
邮编：201209
电话：(021) 5863-5678
传真：(021) 5863-0003
网址：<http://www.deltagreentech.com.cn>
<http://中达电通>
服务热线：(021) 5863-9595

第一章 台达人机简介

■ DOP 系列人机界面产品

欢迎您使用台达人机界面产品。本产品拥有简单易学的编辑软件及高速的硬件架构，提供给您一个功能强大及品质稳定的人机产品。同时也期望您提供宝贵的意见，让 DOP-B 系列产品能够有更强大的功能。

■ 功能介绍

支持多种厂牌的控制器

台达人机界面产品支持 Delta、Omron、Siemens、Mitsubishi 等超过二十种不同厂牌的 PLC(控制器)通讯协议。同时也会随时在网站上更新最新的 PLC(控制器)通讯协议（本手册内所提到的各厂商的品名与商标为各厂商所有）。

支持 Windows®字体的画面编辑器

画面编辑器支持使用 Windows®系统的字型来编辑画面。

便利的宏指令

利用宏功能可以帮助 PLC(控制器)处理复杂的运算功能，有效分担 PLC(控制器)的工作量。用户也可配合通讯宏指令自行撰写通讯协议，并通过串口与特定系统或控制器连接。

使用 USB 快速上下载程序

台达人机界面除可利用传统 RS-232 上/下载人机画面程序外，也可利用 USB 界面上/下载人机画面程序，将大大缩短传输资料的时间。部份机种也支持 Ethernet 网络上/下载。

便利的配方功能

提供方便好用的配方编辑器，类似 Excel 的编辑界面让用户可以轻松的编辑配方；并且可以同时输入多组的配方。

可同时支持多台不同厂牌的控制器

可同时支持多个通讯端口连接多台不同或是相同的控制器。DOP-B 系列机种最多可同时连接支持 3 种不同通讯协议的设备。

一台人机对多台 PLC 连线功能

使用 COM 2/COM 3 的 RS-485 界面，可串接多台的控制器的^{※注一}。

模拟功能^{※注二}

人机编辑软件 Screen Editor 提供方便好用的模拟功能^{※注三}。用户可于人机程序的设计过程中，先行在电脑端进行程序模拟动作与除错。而在人机软件里提供两种模拟功能，分别为离线模拟与线上模拟。

离线模拟功能：当用户编辑与编译完人机的程序后，直接使用电脑（不连接控制器）进行模拟人机画面动作是否正确。

线上模拟功能：当用户编辑与编译完人机的程序后，直接使用电脑连接控制器，先行模拟人机动作是否正确。

支持 USB Host 功能

DOP-B 系列机种支持 USB Host 功能，可连接 U 盘、鼠标及 USB 界面的打印机等周边设备，大大加强了人机的功能。

U 盘资料备份功能

DOP-B 系列机种支持 U 盘。不需电脑即可通过 U 盘来更新韧体、下载画面程序以及备份资料等。

打印功能

DOP-B 系列机种可接上 COM Port 界面打印机或 USB 界面打印机作 Hard-Copy 或排版打印。

多重保密功能

提供密码保护功能，保障程序设计者的设计，同时提供元件的权限功能，只有符合元件设定的权限的用户，才可以使用此元件。

多国语系

支持多国语言显示，最多可支持 16 种不同语言。

注一：控制器必须为支持 RS-485 界面。

注二：离线/线上模拟仅支持部份控制器，且线上模拟功能可连续执行时间约莫半个钟头，待模拟时间已至，人机线上模拟视窗会自动跳回 Screen Editor 主程序画面。

注三：执行模拟功能时，在电脑端萤幕的分辨率须设为 24 位元以上，方能顺利运行模拟功能。

3.7 功能菜单—查看



3.7.1 工具栏

标准工具栏



| 图形 | 名称 | 功能 |
|----|-------|----------------------------------|
| | 新建 | 开启新的编辑文件。 |
| | 打开旧档 | 开启已有的编辑文件。 |
| | 保存文件 | 保存正在编辑的文件。 |
| | 汇出 | 将目前编辑区的画面数据以 BMP 图档的格式保存。 |
| | 撤消 | 撤消上一个动作（有些动作无法撤消）。 |
| | 重复 | 重复上一个动作。 |
| | 剪切 | 将元件剪下。 |
| | 复制 | 复制所选取的元件。 |
| | 粘贴 | 粘贴所复制或剪切的元件。 |
| | 查找 | 查找特定的文字、写入地址以及读取地址。 |
| | 建立新画面 | 开启一个新画面。 |
| | 打开旧画面 | 打开旧画面。 |
| | 打印 | 打印编辑内容。 |
| | 关于 | 版本说明。 |

表格 3-7-1 标准工具栏

状态栏

此栏会显示目前编辑的状态（图 3-7-1）。

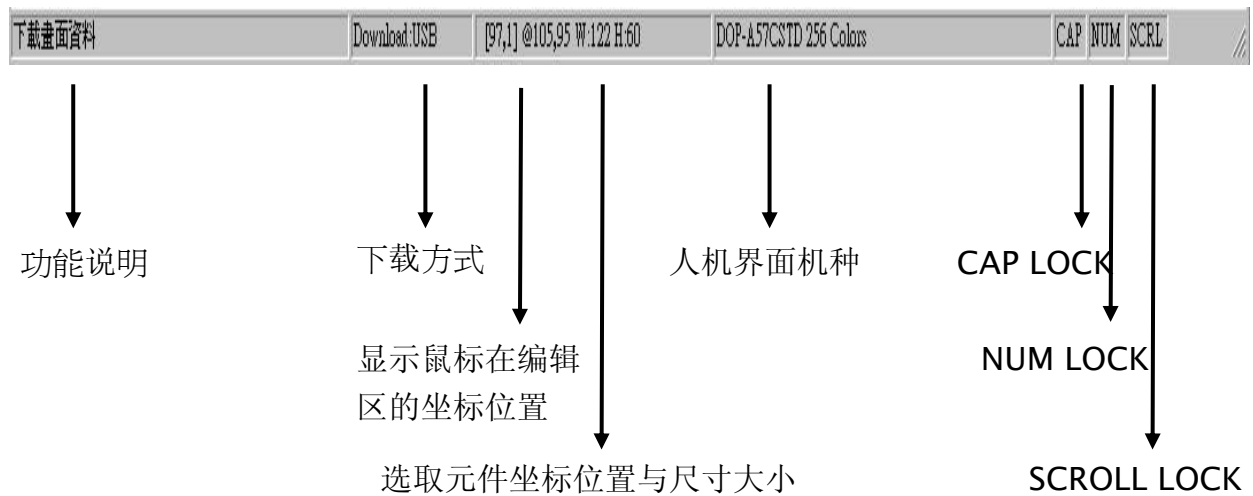
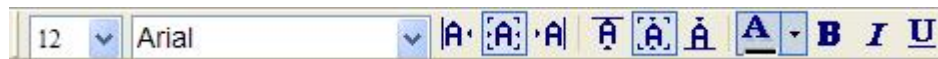


图 3-7-1 状态栏

文字工具栏





| 图形 | 名称 | 功能 |
|----|--------|-----------|
| | 字体大小 | 显示并改变文字大小 |
| | 字体 | 所选择的字体 |
| | 文字靠左对齐 | 文字靠左 |
| | 文字水平置中 | 文字置中 |
| | 文字靠右对齐 | 文字靠右 |
| | 文字靠上对齐 | 文字靠上 |
| | 文字垂直置中 | 文字垂直置中 |
| | 文字靠下对齐 | 文字靠下 |
| | 字体颜色 | 改变字体颜色 |
| | 文字粗体 | 文字粗体 |
| | 文字斜体 | 文字斜体 |
| | 文字加底线 | 文字加底线 |

表格 3-7-2 文字工具栏

图形工具栏



| 图形 | 名称 | 功能 |
|----|------------------|--|
| | 选取图形透明色 | 以吸管的方式选择图形上的哪个颜色要变成透明 |
| | 处理所有状态 图形模式切换 | 若此功能被启动，执行元件上图形的缩放功能并不仅仅只有目前的状态，元件的所有状态的汇入图形者皆会完成指 |

| 图形 | 名称 | 功能 |
|---|----------|------------------|
| | | 定的缩放功能 |
|  | 延展图形 | 将所选图形延展至该元件范围 |
|  | 保持比例延展图形 | 按照原图形的长宽比作等比例的缩放 |
|  | 实际图形大小 | 将所选图形恢复原实际尺寸大小 |
|  | 图形靠左对齐 | 将所选图形靠左 |
|  | 水平置中 | 将所选图形置中 |
|  | 图形靠右对齐 | 将所选图形靠右 |
|  | 图形靠上对齐 | 将所选图形靠上 |
|  | 图形垂直置中 | 将所选图形垂直置中 |
|  | 图形靠下对齐 | 将所选图形靠下对齐 |

表格 3-7-3 图形工具栏

元件工具栏



| 图形 | 名称 | 展开项目 |
|---|----|--|
|  | 按钮 | <ul style="list-style-type: none">  设On  设Off  保持型  交替型  复状态  设值  设常数值  加值  减值  换画面  回前页  系统时间日期  设置密码表  密码输入  调整对比亮度  设为最低权限  系统目录  输出报表  撷取画面  移除USB  汇入/汇出配方  触碰校正 |
|  | 仪表 | <ul style="list-style-type: none">  仪表(1)  仪表(2)  仪表(3) |

| 图形 | 名称 | 展开项目 |
|---|------|--|
|  | 柱状图 | <ul style="list-style-type: none">  一般型  差量型 |
|  | 管状图 | <ul style="list-style-type: none">  管状图(1)  管状图(2)  管状图(3)  管状图(4)  管状图(5)  管状图(6)  管状图(7) |
|  | 扇形图 | <ul style="list-style-type: none">  扇形图(1)  扇形图(2)  扇形图(3)  扇形图(4) |
|  | 指示灯 | <ul style="list-style-type: none">  状态指示灯  数值范围指示灯  简易指示灯 |
|  | 数据显示 | <ul style="list-style-type: none">  数值显示  文数值显示  日期显示  时间显示  星期显示  一般型信息显示  走马灯信息显示 |
|  | 图形显示 | <ul style="list-style-type: none">  状态图显示  动画  动态线条  动态矩形  动态椭圆形  即时图显示 |
|  | 输入 | <ul style="list-style-type: none">  数值输入  文数字输入  Barcode输入 |
|  | 曲线图 | <ul style="list-style-type: none">  一般曲线图  X-Y曲线图  X-Y即时分布图 |
|  | 取样功能 | <ul style="list-style-type: none">  历史趋势图  历史数值数据表  历史信息表 |

| 图形 | 名称 | 展开项目 |
|---|------|--|
|  | 警报显示 | <ul style="list-style-type: none">  历史报警表  当前报警表  报警频次表  报警信息走马灯 |
|  | 键盘 | <ul style="list-style-type: none">  键盘 (1)  键盘 (2)  键盘 (3) |

表格 3-7-4 元件工具栏所有元件

绘图工具栏



| 图形 | 名称 | 功能 |
|---|-----|-------|
|  | 线 | 线绘图 |
|  | 矩形 | 矩形绘图 |
|  | 圆 | 圆绘图 |
|  | 多边形 | 多边形绘图 |
|  | 弧 | 弧绘图 |
|  | 文字 | 文字方块 |
|  | 刻度 | 绘制刻度 |
|  | 表格 | 绘制表格 |

表格 3-7-5 缩放工具栏

元件规划工具栏



| 图形 | 名称 | 功能 |
|---|------------|---------------|
|  | 目前状态的文字 | 显示目前编辑元件状态的文字 |
|  | 查看状态 0/OFF | 切换并查看状态 0/OFF |
|  | 查看状态 1/ON | 切换并查看状态 1/ON |
|  | 查看所有元件读写地址 | 查看所有元件读写地址 |

| 图形 | 名称 | 功能 |
|---|-----------|--------------------------|
|  | 上一个视窗 | 选择上一个视窗 |
|  | 下一个视窗 | 选择下一个视窗 |
|  | 编译 | 编译所编辑的元件与画面 |
|  | 下载画面数据与配方 | 下载画面数据与配方 |
|  | 下载画面数据 | 下载画面数据 |
|  | 线上模拟 | 在 PC 端测试元件编辑后的档案，必须连接控制器 |
|  | 离线模拟 | 在 PC 端测试元件编辑后的档案，不必连接控制器 |

表格 3-7-6 上层规划工具栏



| 图形 | 名称 | 功能 |
|---|--------|--------------------|
|  | 组合 | 将所圈选的元件设成组合 |
|  | 取消组合 | 将组合取消 |
|  | 上推至最顶层 | 将所选元件上推至最顶层 |
|  | 下推至最底层 | 将所选元件下推至最底层 |
|  | 上推一层 | 将所选元件上推一层 |
|  | 下推一层 | 将所选元件下推一层 |
|  | 靠左对齐 | 所选元件靠左对齐 |
|  | 靠右对齐 | 所选元件靠右对齐 |
|  | 靠上对齐 | 所选元件靠上对齐 |
|  | 靠下对齐 | 所选元件靠下对齐 |
|  | 垂直置中 | 所选元件垂直置中 |
|  | 水平置中 | 所选元件水平置中 |
|  | 横向等间距 | 所选元件横向间距设为相等 |
|  | 纵向等间距 | 所选元件纵向间距设为相等 |
|  | 造成相同宽度 | 将所选元件的宽度设成和基准元件一样 |
|  | 造成相同高度 | 将所选元件的高度设成和基准元件一样 |
|  | 造成相同尺寸 | 将所选元件的长跟宽设成和基准元件一样 |

表格 3-7-7 下层规划工具栏

缩放工具栏



| 图形 | 名称 | 功能 |
|---|------|--|
|  | 显示比例 | 显示缩小放大比例，有 25%、50%、75%、100%、150%、200%、300% |

| 图形 | 名称 | 功能 |
|----|-----|---------------------------|
| | 放大 | 放大显示的比例, 有 150%、200%、300% |
| | 缩小 | 缩小显示的比例, 有 25%、50%、75% |
| | 1:1 | 恢复原始大小比例 100% |

表格 3-7-8 缩放工具栏

3.7.2 属性表

提供各种元件列表的属性设定与编辑画面预览 (图 3-7-2)。

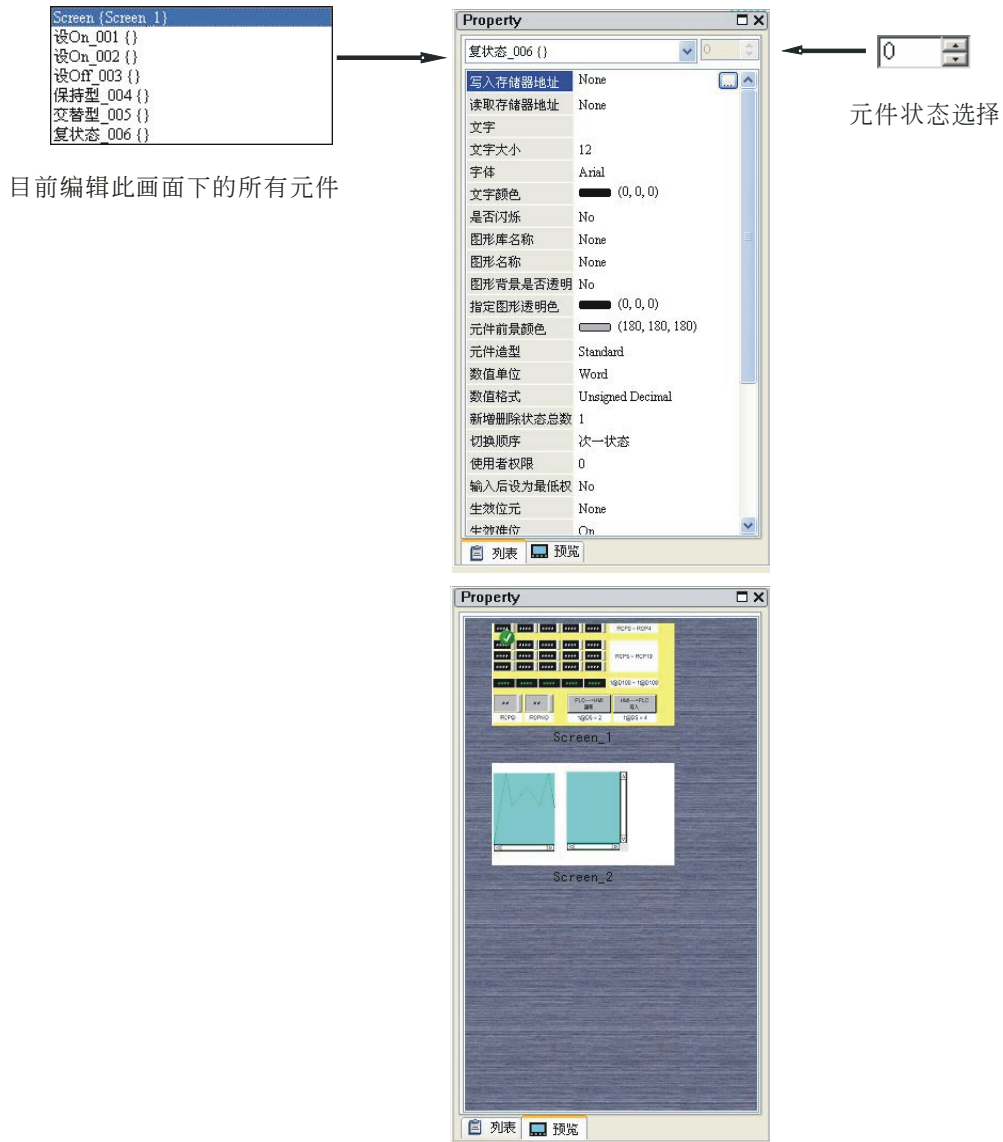


图 3-7-2 属性表：元件列表的属性设定与编辑画面预览

3.7.3 输出

此项功能会将编译时的过程显示在此栏位, 以让用户了解程序编译的结果, 便于追查编辑时发生错误的问题 (图 3-7-3、图 3-7-4、图 3-7-5、图 3-7-6)。



图 3-7-2 记录栏



图 3-7-4 输出栏



图 3-7-5 编译结果输出栏



图 3-7-6 集中错误输出栏

3.7.4 拉近

将整个编辑画面放大，以方便用户编辑。请参考（图 3-7-7、图 3-7-8）。

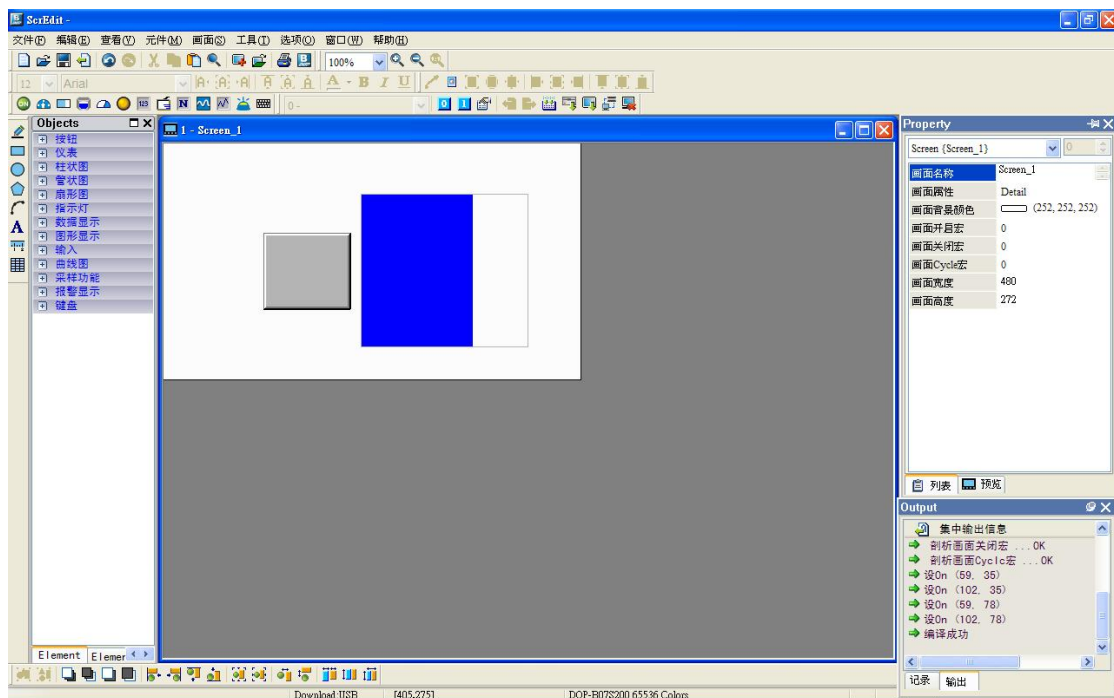


图 3-7-7 未拉近前编辑视窗大小 100%

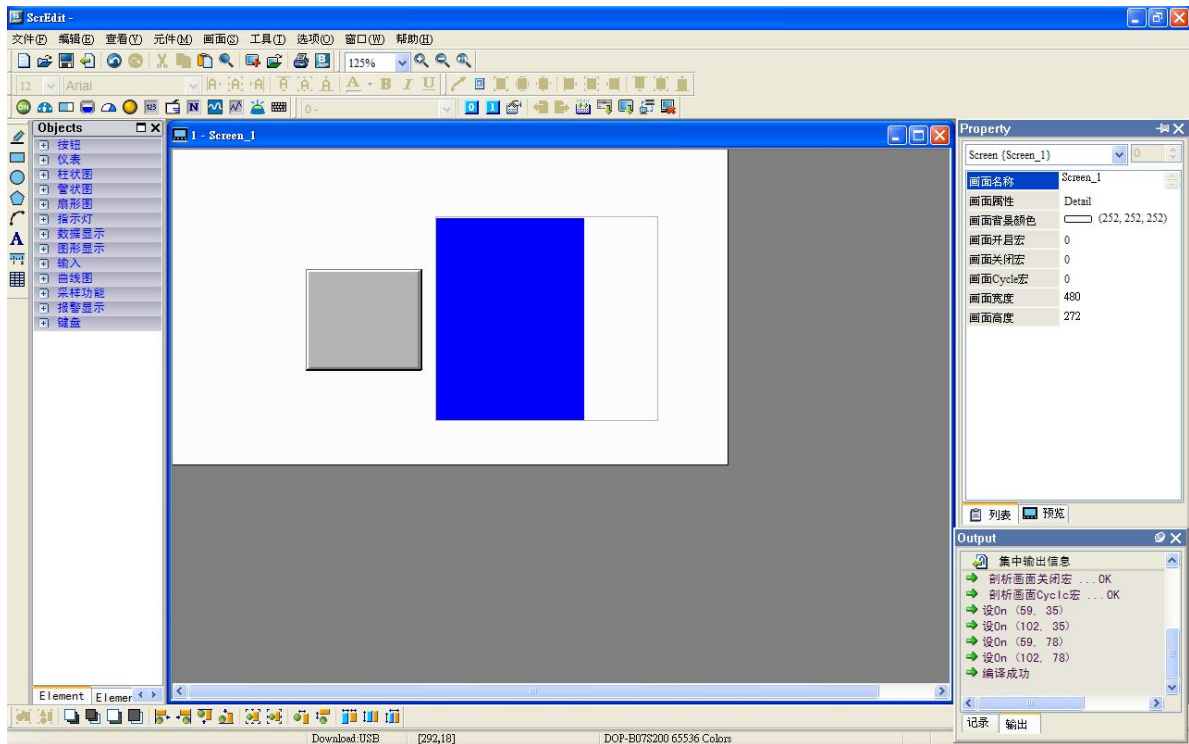


图 3-7-8 拉近后编辑视窗大小 125%

3.7.5 拉远

将整个编辑画面缩小，请参考（图 3-7-9）。

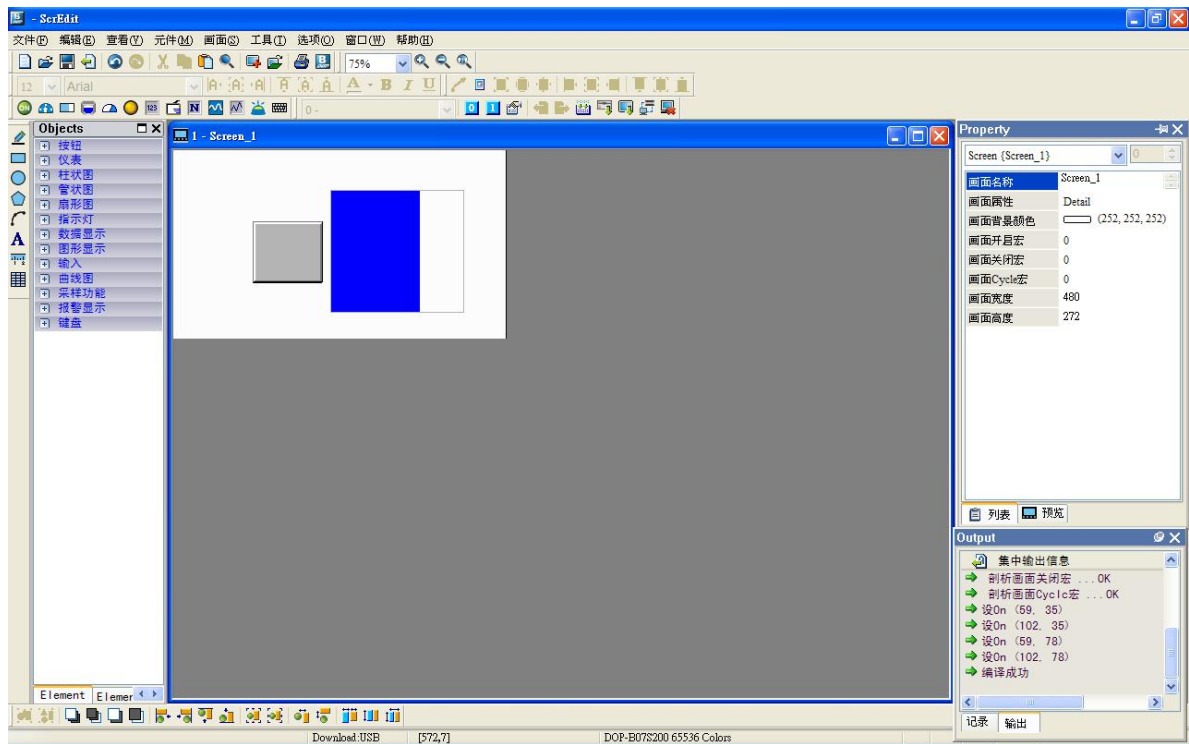




图 3-7-9 拉远后编辑视窗大小 75%

3.7.6 实际大小

立即恢复 100%的比例，此比例是相对于人机屏幕的大小。

不管是拉近或是拉远，其缩放比例从 25%、50%、75%、100%、150%、200%、300%，当然您也可以从查看工具栏上直接按下  拉近或按下  拉远查看或直接选择其缩放比例（图 3-7-10）。

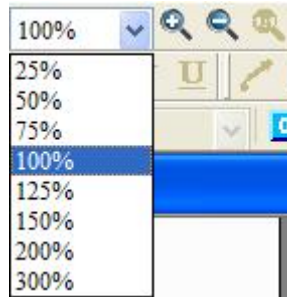


图 3-7-10 选择缩放比例

3.7.7 全屏幕

如图 3-7-11。将编辑的画面以全屏幕显示，并显示此画面有使用的宏

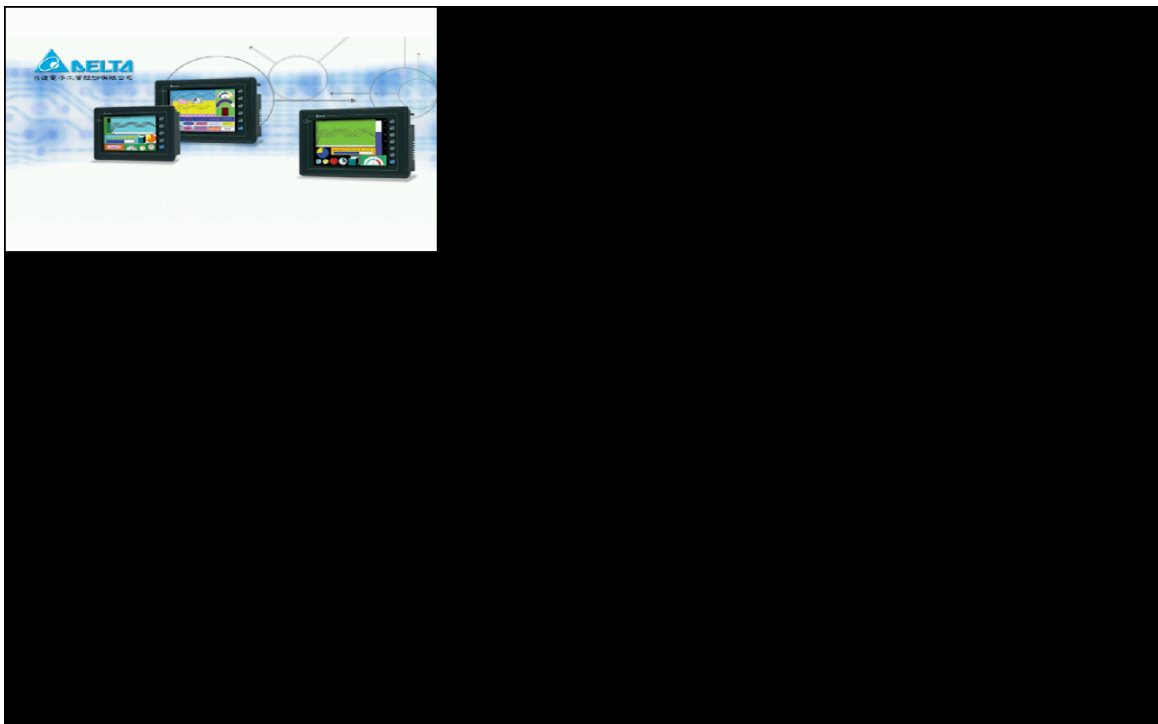


图 3-7-11 选取全屏幕后编辑视窗画面，按 Esc 键或鼠标左键离开

3.7.8 全屏幕及输出点

如图 3-7-12。此与全屏幕功能差别在于此全屏幕及输出点功能会将元件的读写地址一并显示出来，同样地也会将此画面有设定的宏行数显示出来。

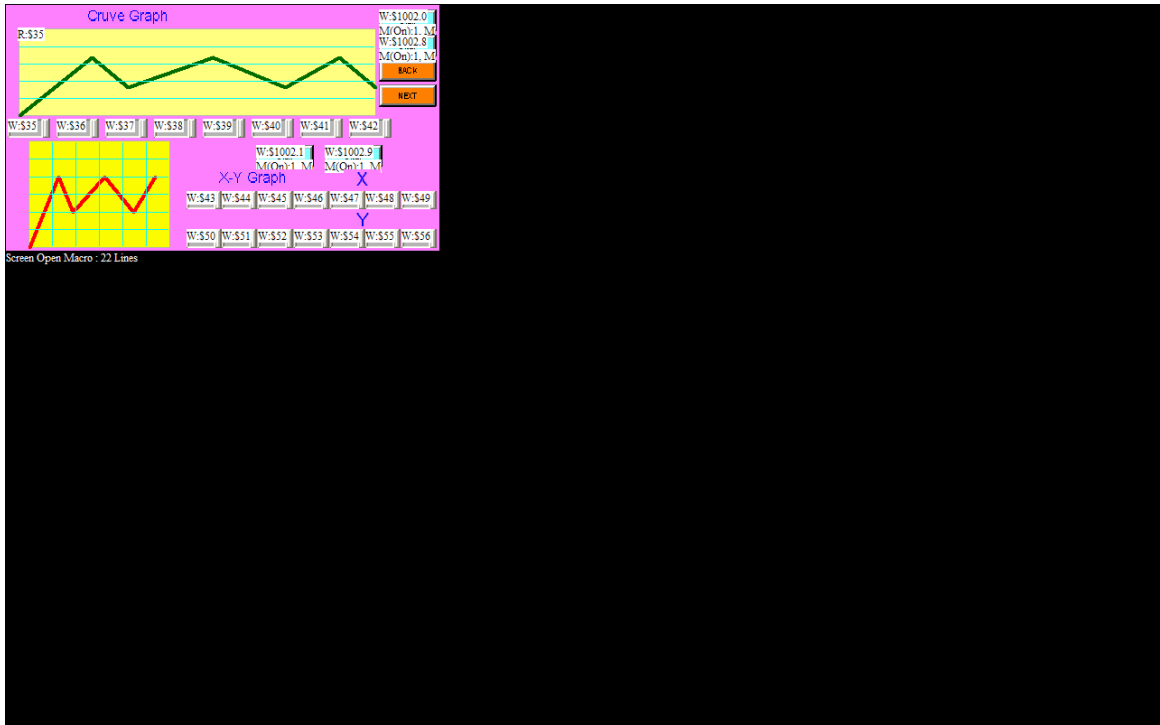


图 3-7-12 选取全屏幕及输出点后编辑视窗画面，按 Esc 键或鼠标左键离开

3.7.9 格子设置

其主要用途是帮助用户在编辑时能够很容易的将各元件对齐，而对齐格点的间距可由用户自由设定（图 3-7-13、图 3-7-14）。



图 3-7-13 格子设置对话框



图 3-7-14 编辑元件时启动显示格点的状况

3.7.10 地址映射表

在规划众多读写地址元件时，往往会在规划地址过程中忘记或重覆使用到相同的地址。所以在 Screen Editor 中提供了地址映射表的功能，能查看出目前被选择到的元件其读写地址与哪些元件或者宏或者控制区等等地址有相互关联，如图 3-7-15，程序中警报地址与宏皆使用到内部存储器\$3。

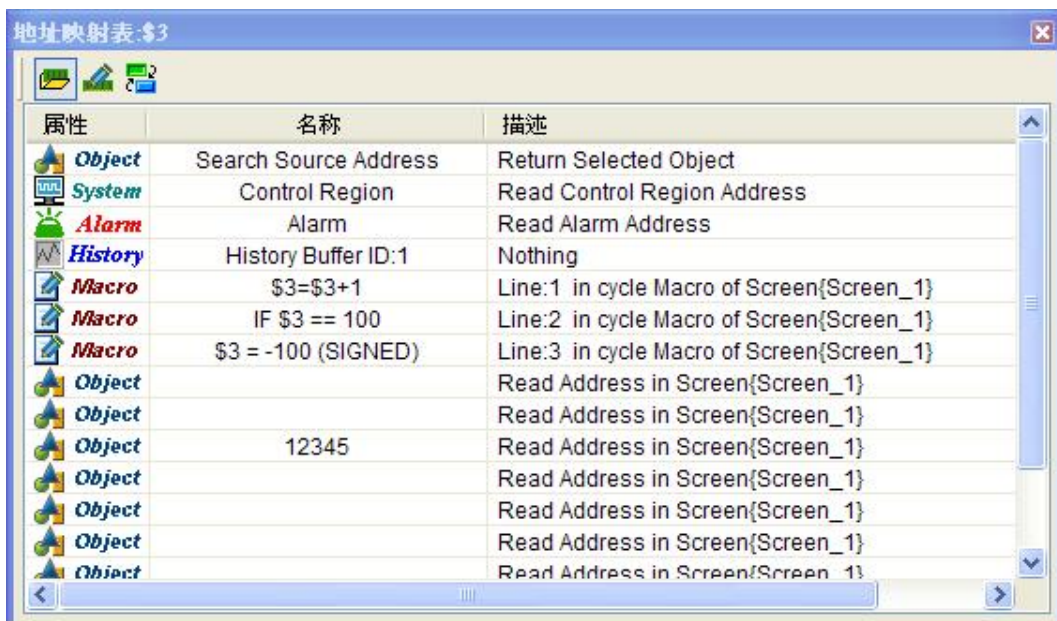


图 3-7-15 地址映射表

3.7.11 元件地址清单

开启元件地址清单，程序会将目前编辑画面上的所有元件作分类，您可以用点击页签的方式切换您想查看的分类。每一分类里会列出所属该分类里元件的相关地址属性（图 3-7-16）（以按钮类别来看分别有写入地址、读取地址、触发地址、生效地址），其中也包括了一些较重要的属性，譬如：触发方式、生效准位等等。这些所列出的栏位您可以直接更改其属性同时您也可以鼠标双击(Double-Click)该列，让程序自动将此元件圈选起来并于属性表编辑该元件详细的属性。

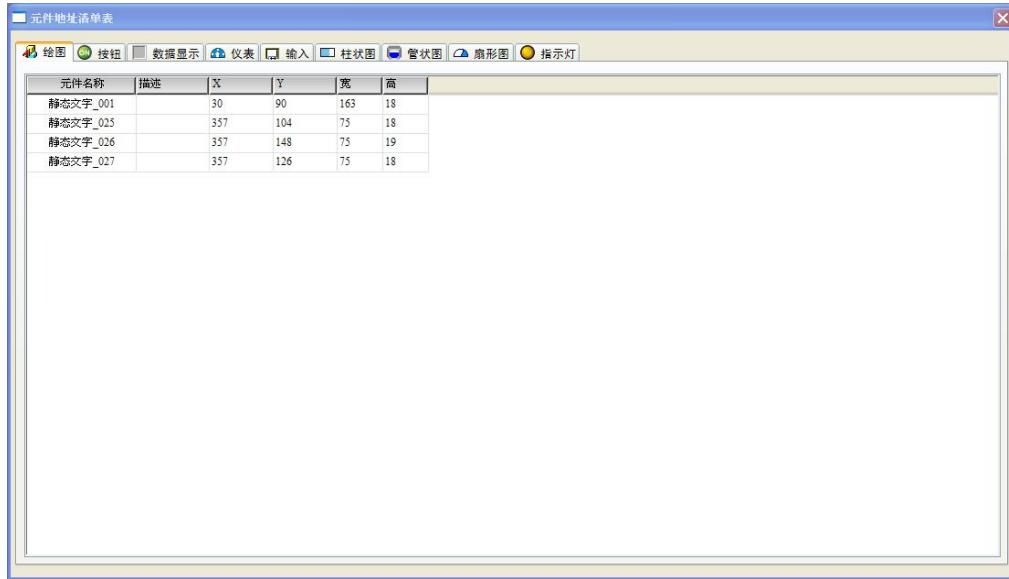


图 3-7-16 元件地址清单对话框

3.7.12 存储器使用清单

存储器使用清单可分为以下四个部分：

1. ROM：实际下载到人机所占的文件空间(包含配方，画面资料等)。
2. SRAM：断电保持资料存放位置预设为存放在 SRAM 静态存储器。当用户如果项目中有规划到历史、警报等资料，则可于此区域查看空间的运用状态。
3. SDARM：人机实际执行项目的地方，用户必须先于项目建立后并进行编译才会得到此信息。
4. External Storage：指外部存储器使用空间。当断电保持资料存放位置设为外部存储设备如 U 盘，此时 U 盘就会将原本存在静态存储器 SRAM 的资料区块搬移到此区域。

| 名稱 | 斷電保持 | 儲存位置 | 檔案名稱 |
|----|------|-----------------------|------|
| 警報 | No | HMI | |
| 配方 | Yes | HMI USB Disk SD | |

以下举几个例子来说明此一部份。

- (1). 画面汇入图形所占用空间
- (2). 建立历史资料所占用空间

(1). 画面汇入图形

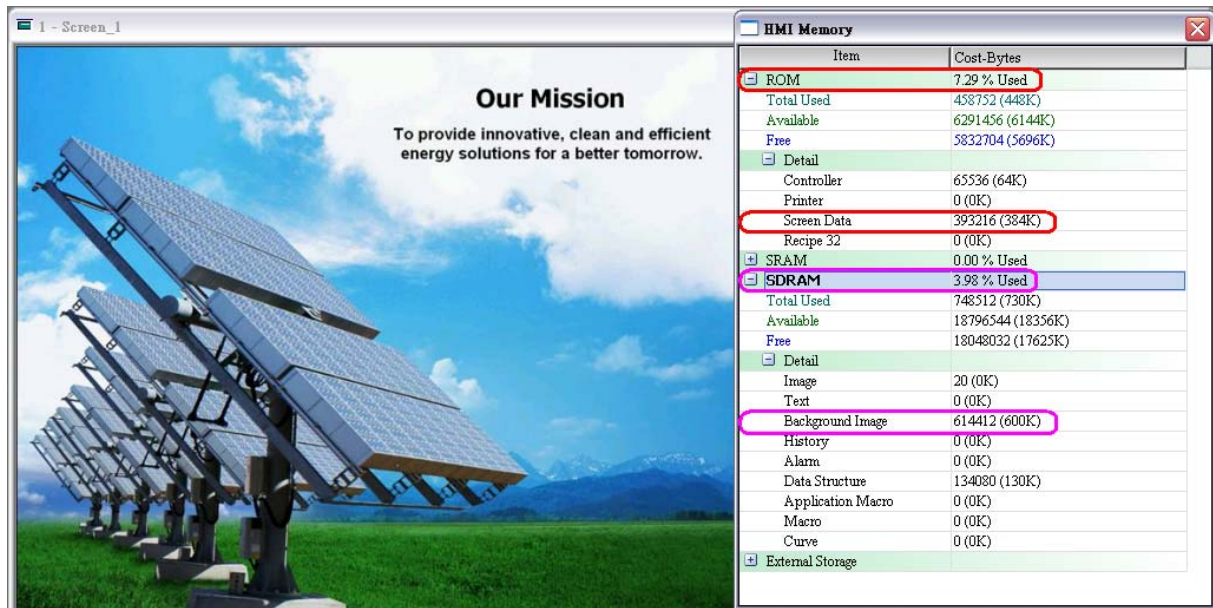
此部分会使用到 ROM 与 SDRAM 的存储器。

当开启一个新专案时，系统计算出来会使用的存储器空间如上图所示

ROM=3.13%

SDAM=0.71%

当汇入一张图形的时候，存储器的配置就会出现变化，如下图所示。



此时可以注意当汇入图形后存储器空间的变化，

ROM=7.29%

SDAM=3.98%

ROM 的部分多出了 256K (384K-128K)，SDRAM 的部分多出了 600K (600K-0K)。

(2). 建立历史数据

我们使用刚刚的专案继续来规划。

单独看建立历史资料的部分会使用到 SRAM 的存储器。

开启上述专案时，显示的存储器空间为

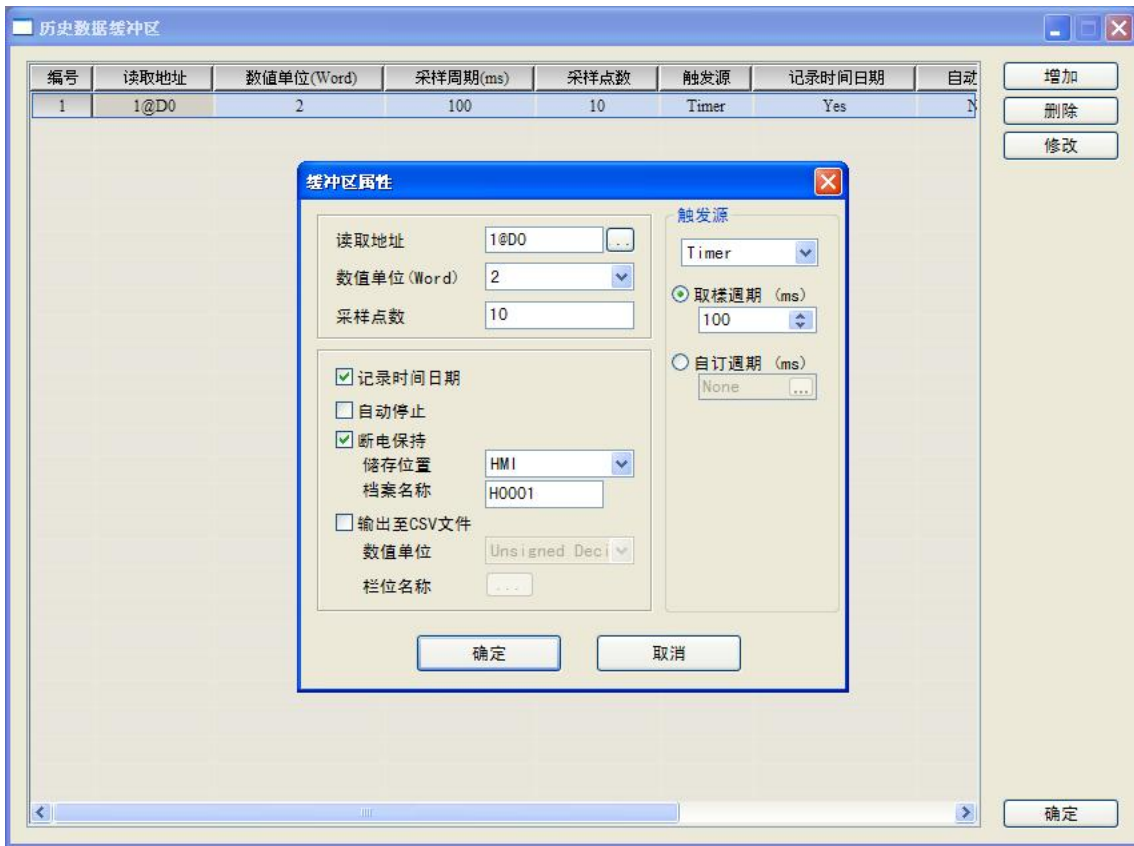
ROM=7.29%

SDAM=3.98%

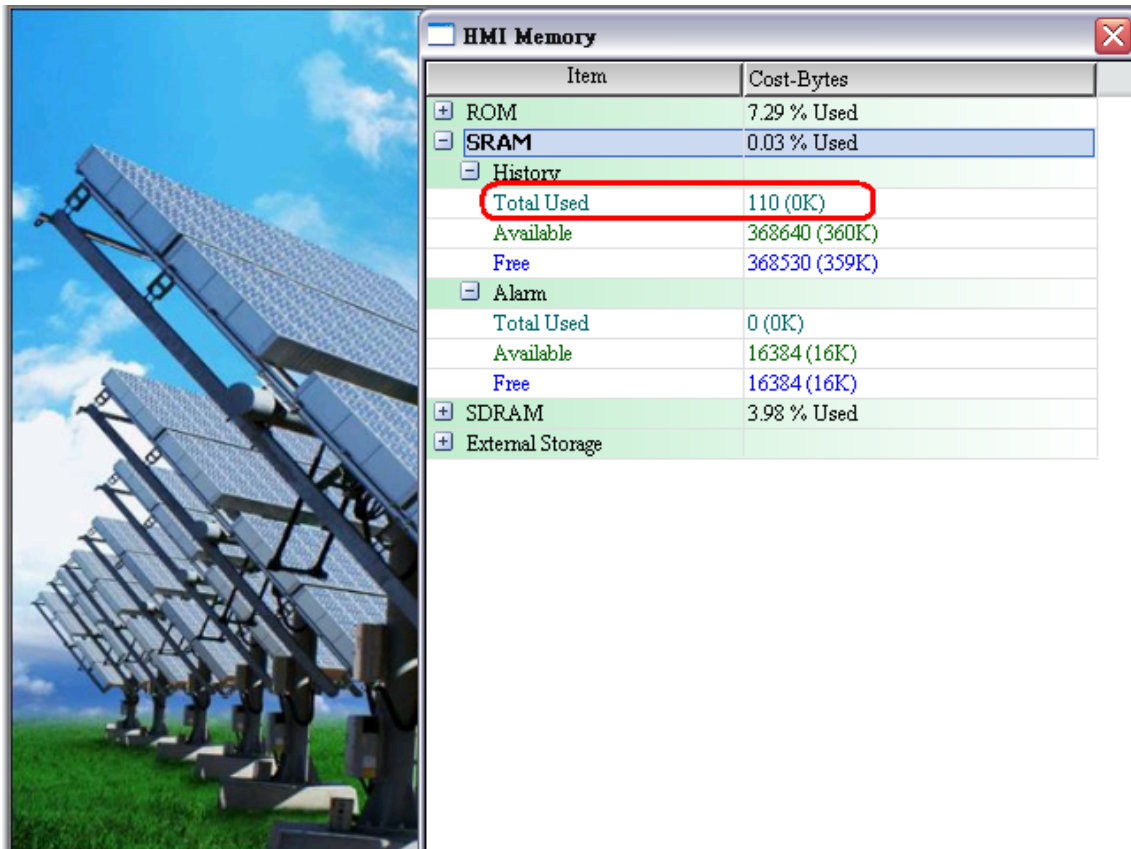
SRAM=0%

External Storage=0%

当建立一个历史资料缓冲区条件如下，



长度为 2word、取样点数 10 笔、记录时间日期并勾选将历史资料存放在断电保持区。断电保持资料存放位置目前设定在 SRAM 静态存储器。设定完成后，内存的配置就会出现以下变化。



每储存一笔皆须记录时间日期需占用 6Byres

每储存一笔会占用 2word (4Byres)

SRAM= (6+4) x 10 笔+10 Byres (系统使用) =110 Byres

此时可以看到存储器空间为

ROM=7.29%

SDAM=3.98%

SRAM=0.03%

External Storage=0%

当将断电保持资料存放位置设为 U 盘时

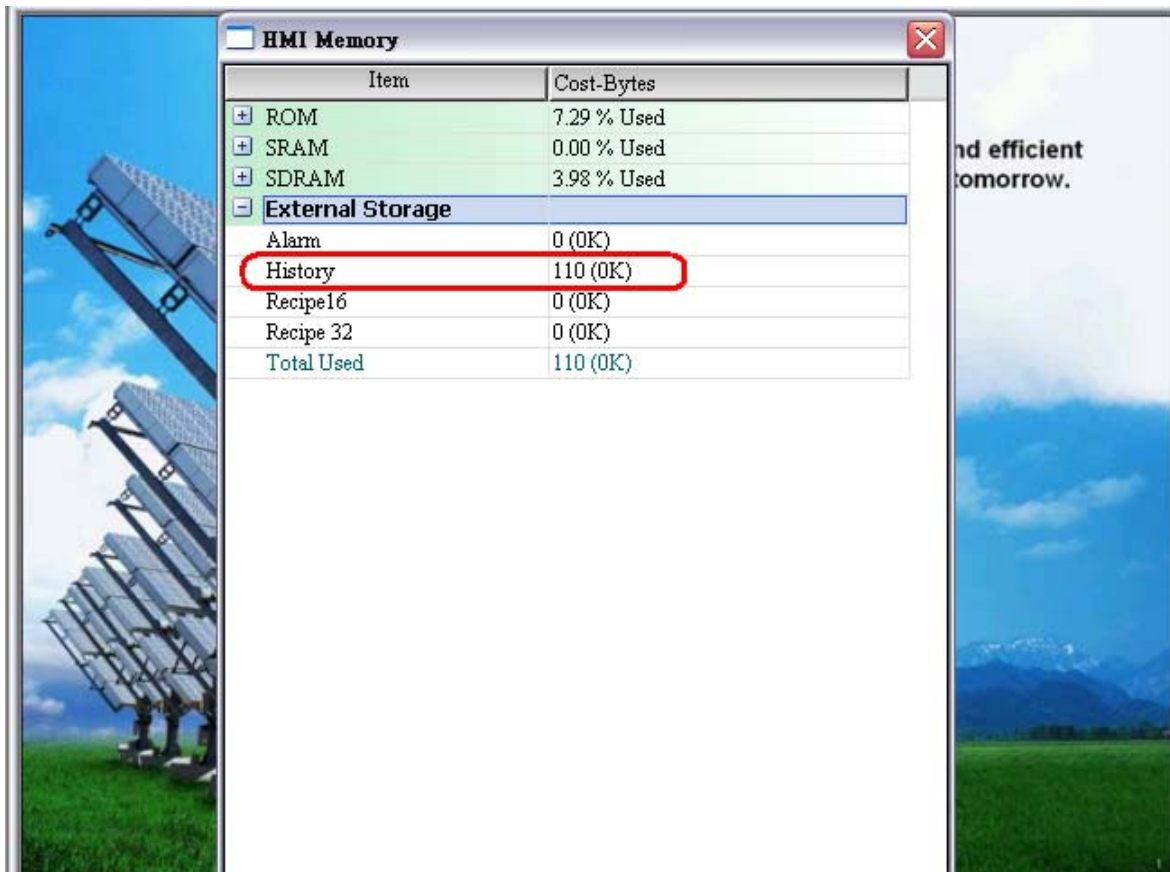
存储器的变化如下

ROM=7.29%

SDAM=3.98%

SRAM=0%

External Storage=110Bytes



详细关于警报与历史区域空间使用的计算，请参阅本手册第 3.11.3 节中有详述。

第三章 Screen Editor 画面编辑软件

操作说明

本章将详细介绍 Screen Editor 画面编辑软件的安装方式以及各项功能，用户可以通过它简单的操作界面以及强大的功能，来设计所需的工作画面。

3.1 安装与升级 Screen Editor

3.1.1 操作环境需求

安装 Screen Editor 编辑软件的基本电脑操作环境需求如下表所示：

| 硬件/软件 | 规格 |
|-------|---|
| 个人电脑 | Pentium 4.1Hz 以上 |
| 内存 | 1G MB 以上 |
| 硬盘 | 400 MB 以上 |
| 显示器 | 支持分辨率 1024x768 以上全彩显示器 |
| 打印机 | Windows 2000 / Windows XP 相兼容的打印机 |
| 操作系统 | Windows 2000 / Windows XP / Windows Vista / Windows 7 |

3.1.2 软件安装

Screen Editor 编辑软件可安装于 Windows 2000 / WindowsXP / Vista 操作系统，以下分别说明 WindowsXP 以及 Vista 的安装方式。

Screen Editor 安装程序可免费从台达网站上取得，下载网址为：

http://www.delta.com.tw/product/em/download/download_main.asp

【Windows 2000/WindowsXP 操作系统安装方式】

1. 安装 Screen Editor 前，请先启动您的电脑进入 Windows 2000 / WindowsXP 操作系统，在此以 WindowsXP 为操作系统说明如下（图 3-1-1）。



图 3-1-1 开启 Microsoft Windows 画面

2. 在 Windows 视窗下，点击安装程序执行文件（Screen Editor 2.00.XX.exe）后，程序便会开始进行安装程序，此时屏幕中间会显示语言选择信息对话框，用户可自行选择安装繁体中文、简体中文、英文三种不同语言的软件操作界面（图 3-1-2）。

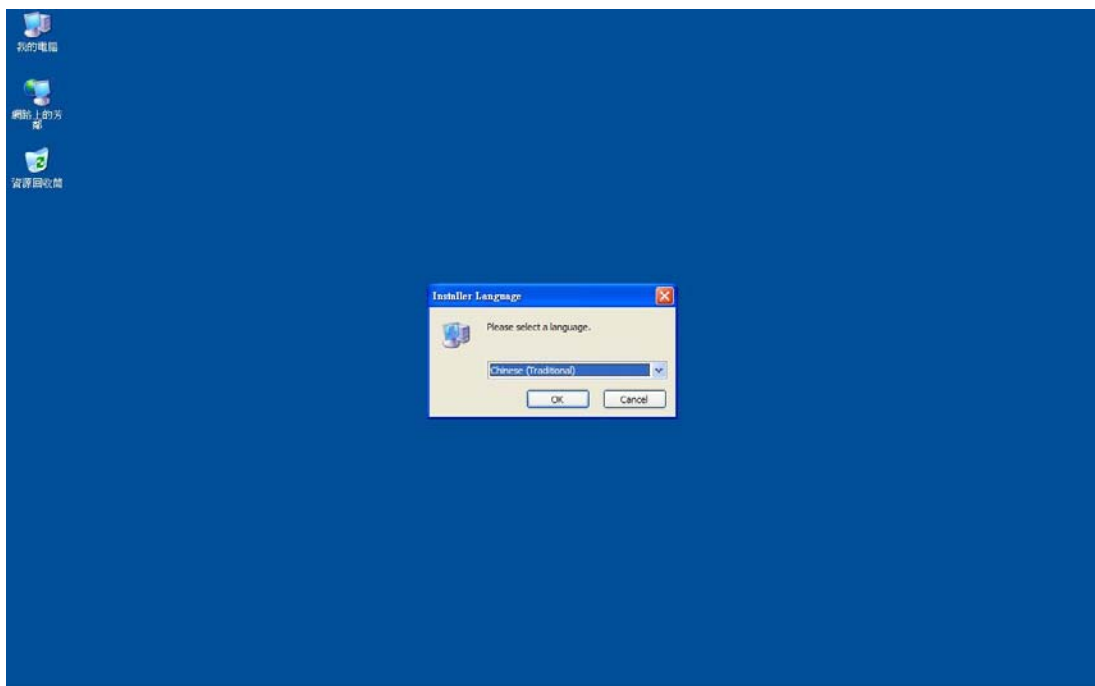


图 3-1-2 选择语言

语言选择完成并按下 OK 键后，信息对话框会显示软件即将安装的硬盘及目录名称，请确认软件即将安装的硬盘及目录名称（图 3-1-3）。本系统的默认路径为 C:\Program Files\Delta Industrial Automation\Screen Editor 2.00.18，您可以自行变更所要安装的硬盘位置以及目录名称。请注意，若用户自行改变安装位置，则未来使用软

件升级文件(Patch 文件)时, 务必注意其安装路径, 以避免软件无法更新升级成功。

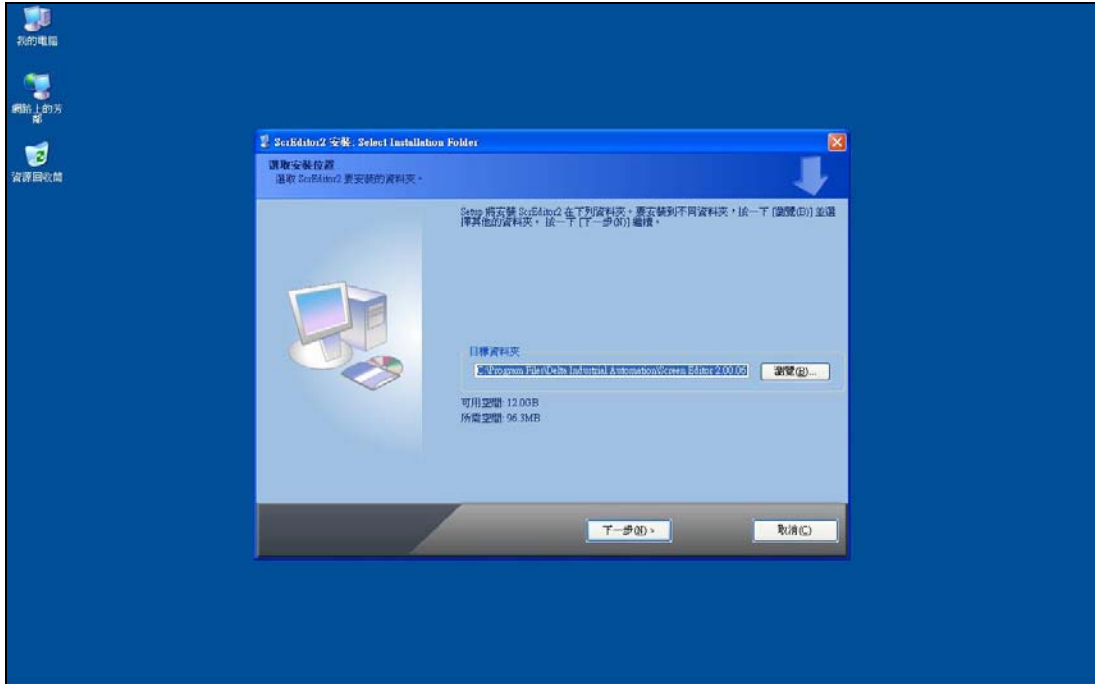


图 3-1-3 Screen Editor 系统安装的硬盘及目录名称

3. 按下【下一步】键后, 信息对话框内会显示可安装的元件项目(图 3-1-4)。目前软件并无其他可选择的元件, 因此直接按下【安装】键, Screen Editor 程序便开始执行安装。未来若有提供其他元件选择, 用户可自行选取所需安装的项目安装。

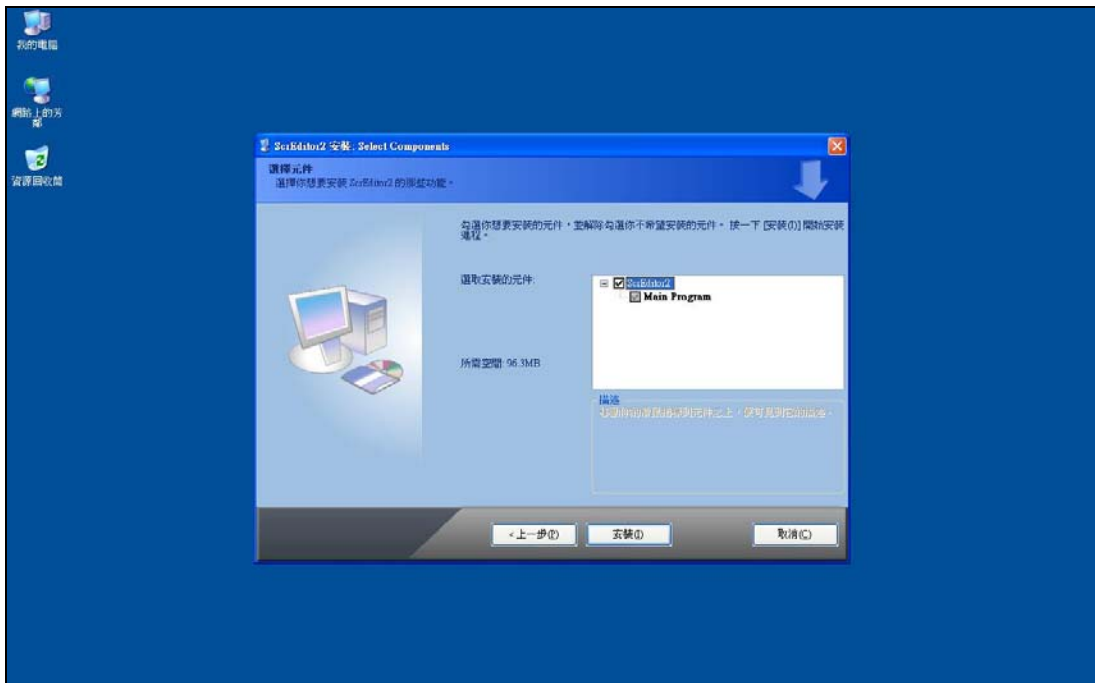


图 3-1-4 选择安装元件画面

4. 按下【安装】键后，Screen Editor 将自行安装完成（图 3-1-5、图 3-1-6）。

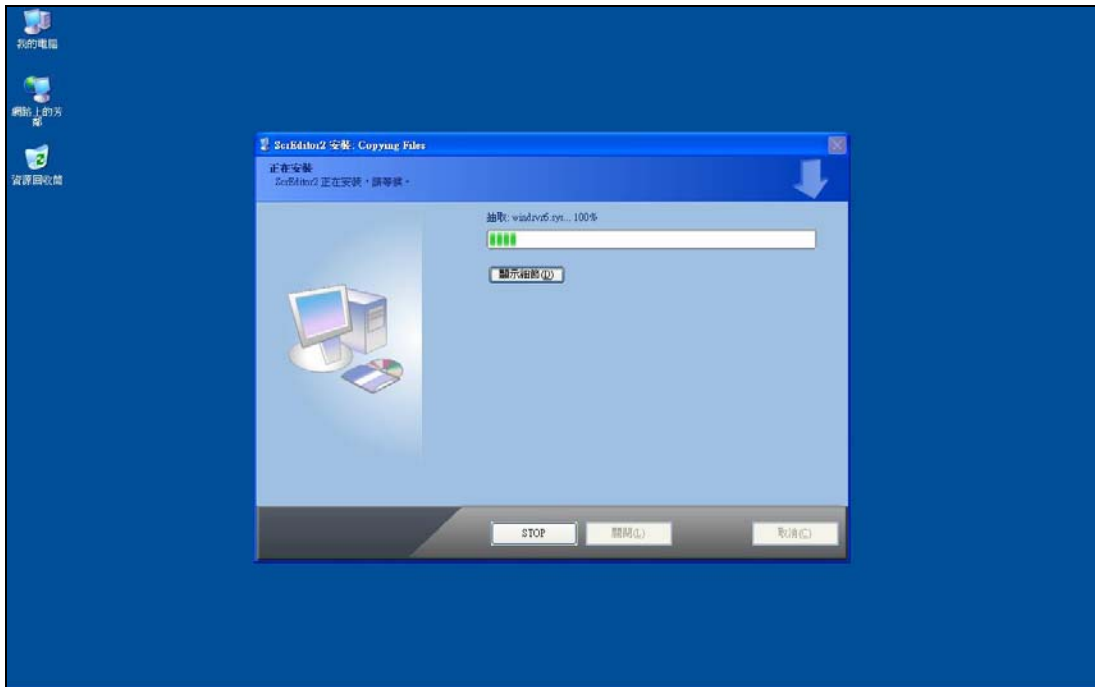


图 3-1-5 Screen Editor 安装画面

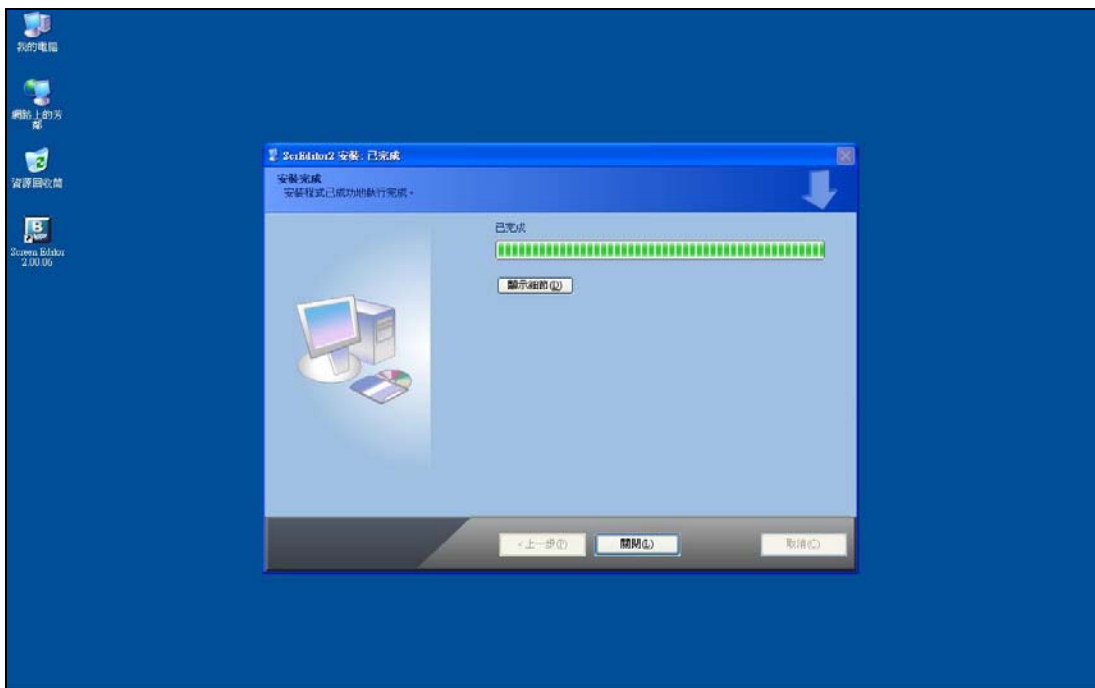


图 3-1-6 Screen Editor 安装完成画面

5. 软件安装完成后，系统会跳出视窗安装 HMI USB 驱动程序，此时点击继续安装即可（图 3-1-7）。如此即完成 Screen Editor 软件安装。
6. 完成后，使用 USB 线连接电脑和人机，等待几秒，开启 Screen Editor 软件，执行【工具】下拉式菜单【取得目前韧体序号】，若顺利取得韧体序号代表软件安装成功，且电脑可以透过 USB 与 HMI 通讯。

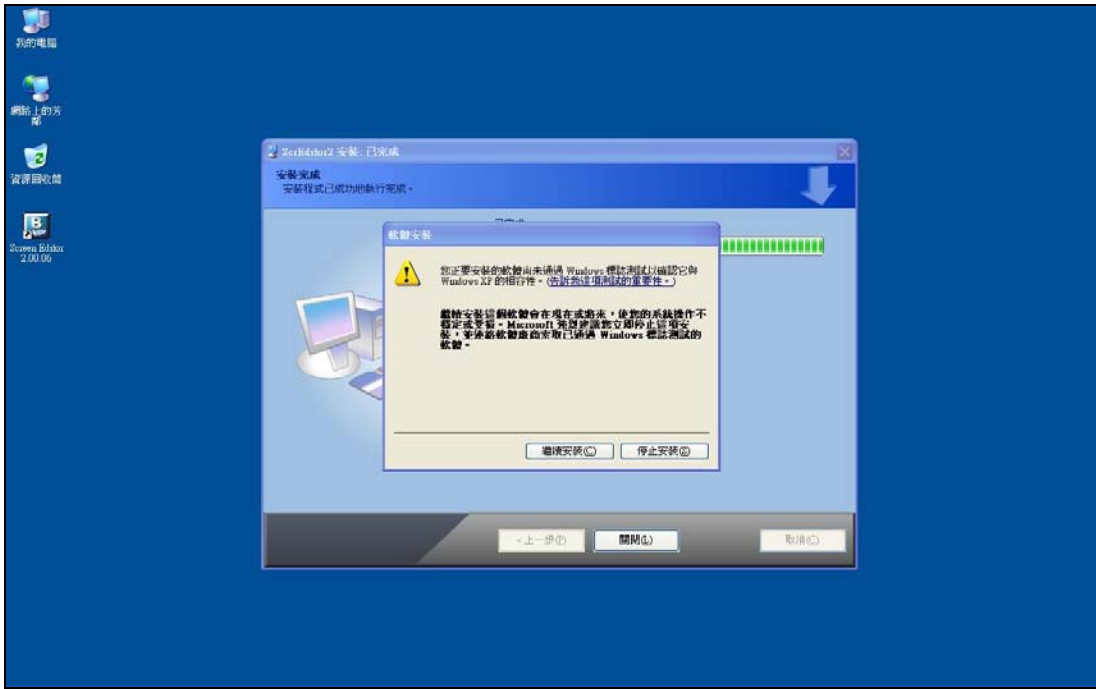


图 3-1-7 HMI USB 驱动程序安装画面

【Windows Vista 操作系统安装方式】

1. 安装 Screen Editor 前, 请先启动您的电脑进入 Windows Vista 操作系统。
2. 以管理者 (Administrator) 的身份登录。
3. 当 Vista 操作系统首次安装 Screen Editor 软件, 请先进入控制面板【关闭用户帐号】。控制面板→用户帐户和家庭安全→用户帐户→开启或关闭用户帐户控制 (图 3-1-8)。

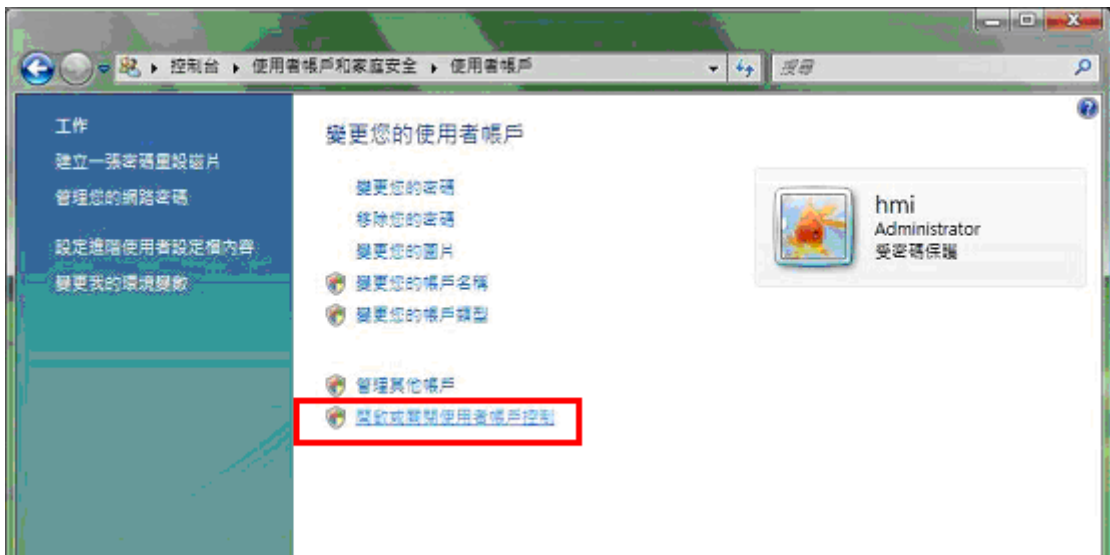


图 3-1-8 开启或关闭用户帐户控制画面

4. 取消勾选使用【用户帐户控制(UAC)】(图 3-1-9)。请注意此选项务必取消勾选, 否则软件安装将会无法成功。

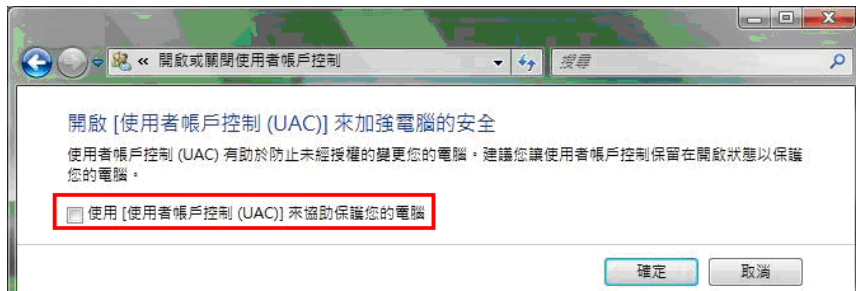


图 3-1-9 用户帐户控制(UAC)画面

- 完成上述设定后，在 Windows 视窗下，点击安装程序执行文件（Screen Editor 2.00.XX.exe）后，程序便会开始进行安装程序，此时屏幕中间会显示语言选择信息对话框，用户可自行选择安装繁体中文、简体中文、英文三种不同语言的软件操作界面（图 3-1-10）。

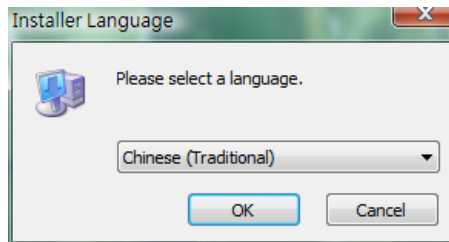


图 3-1-10 选择语言

- 语言选择完成并按下 OK 键后，信息对话框会显示软件即将安装的硬盘及目录名称，请确认软件即将安装的硬盘及目录名称（图 3-1-11）。本系统的默认路径为 C:\Program Files\Delta Industrial Automation\Screen Editor 2.00.18，您可以自行变更所要安装的硬盘位置以及目录名称。请注意，若用户自行改变安装位置，则未来使用软件升级文件(Patch 文件)时，务必注意其安装路径，以避免软件无法更新升级成功。

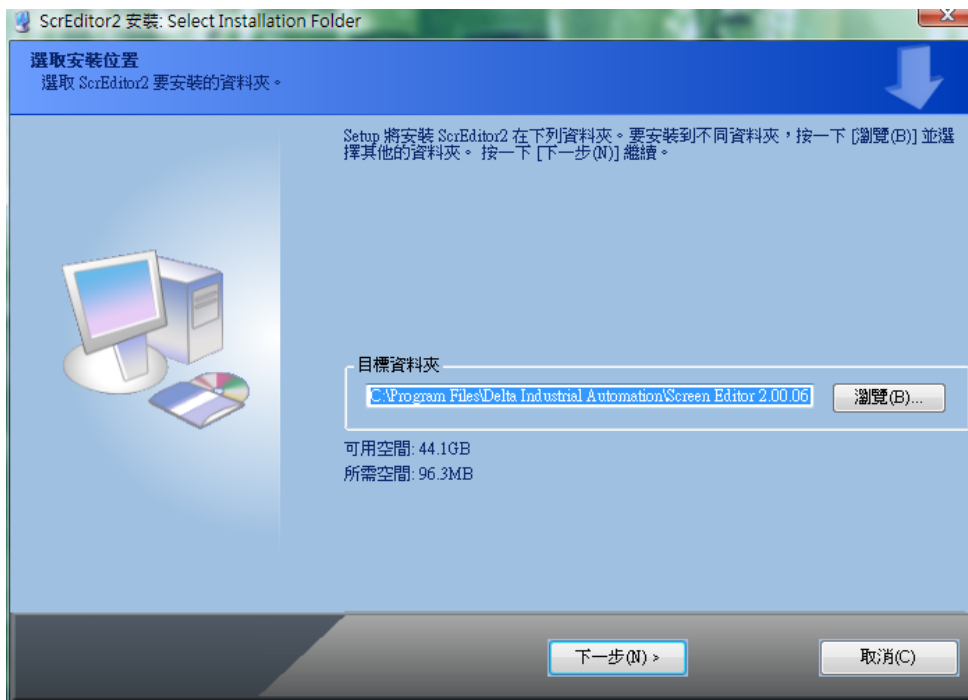


图 3-1-11 Screen Editor 系统安装的硬盘及目录名称

7. 按下【下一步】键后，信息对话框内会显示可安装的元件项目（图 3-1-12）。目前软件并无其他可选择的元件，因此直接按下【安装】键，Screen Editor 程序便开始执行安装。未来若有提供其他元件选择，用户可自行选取所需安装的项目安装。

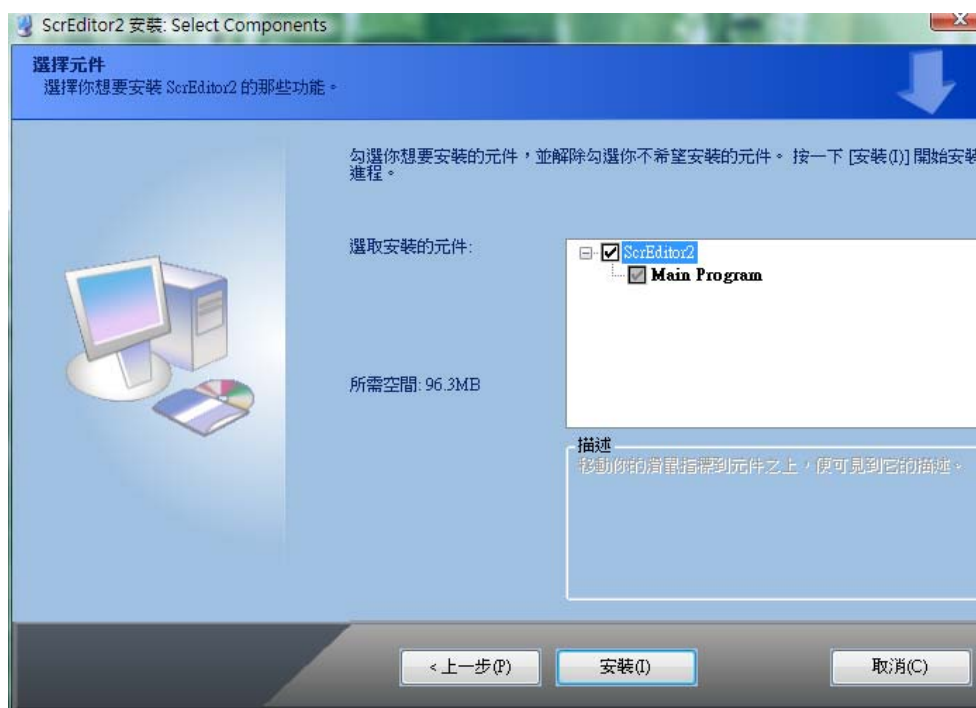


图 3-1-12 选择安装元件画面

8. 按下【安装】键后，Screen Editor 将自行安装完成（图 3-1-13、图 3-1-14）。

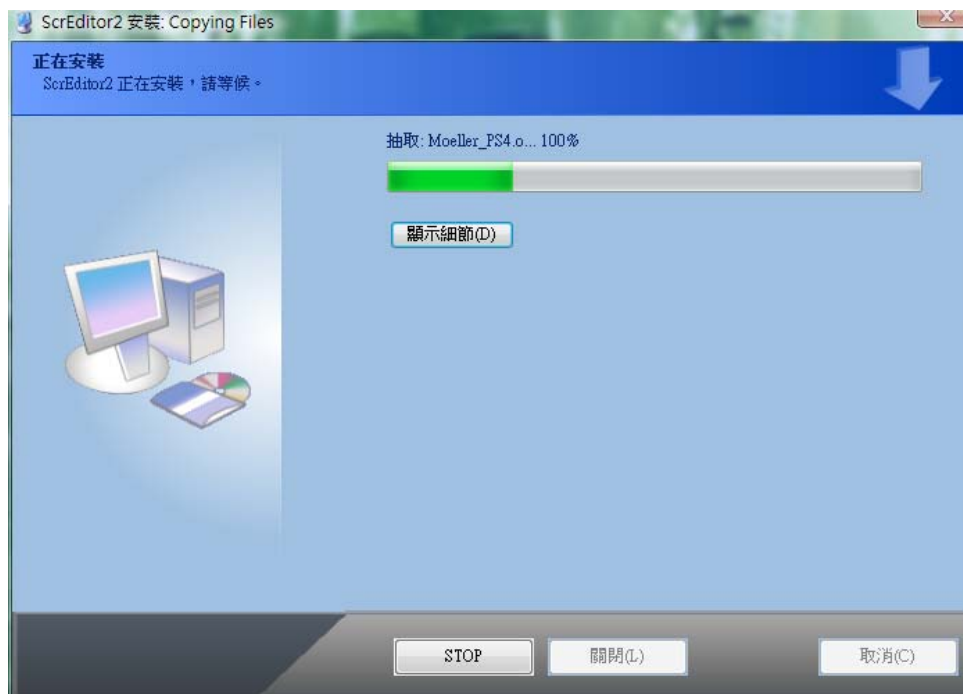


图 3-1-13 Screen Editor 安装画面

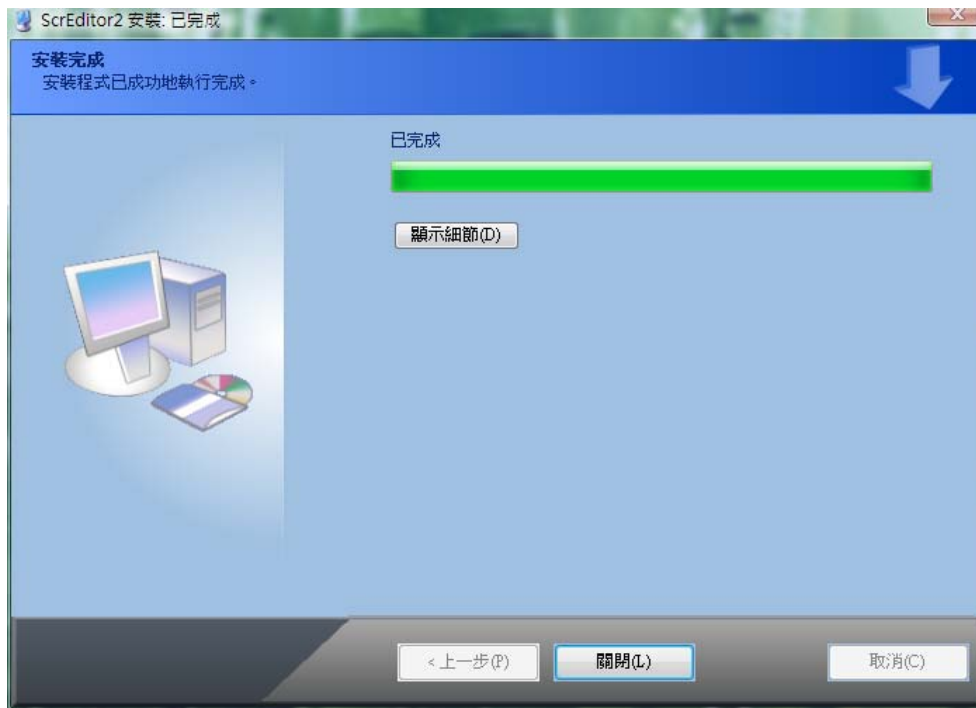


图 3-1-14 Screen Editor 安装完成画面

9. 首次安装后会出现如下 BDE 资料库视窗（图 3-1-15），按下 ok 键后继续安装。若是按下 Cancel 键，则 Screen Editor 程序将无法编译梯形图(Ladder)。

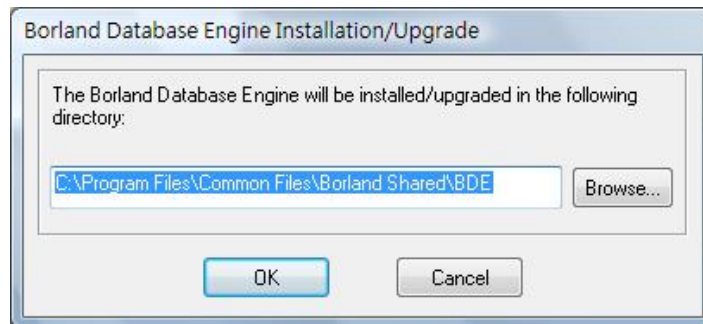


图 3-1-15 BDE 资料库视窗画面

10. 安装时出现验证视窗（图 3-1-16），请选择”仍安装此驱动程序软件”继续安装。如此即完成 Screen Editor 软件安装。

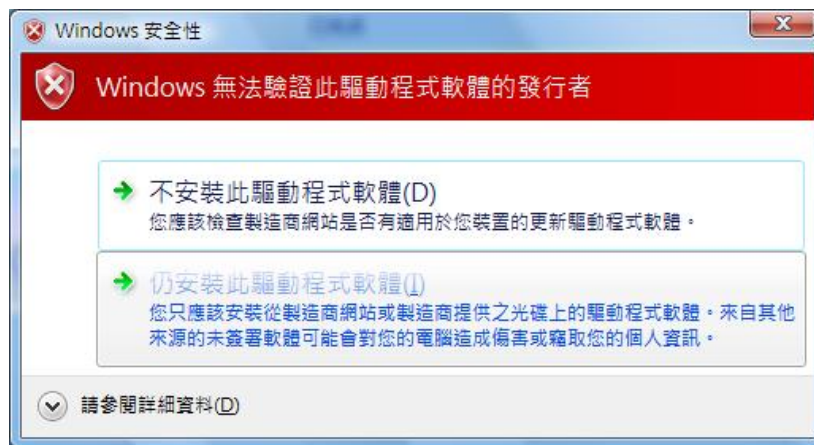


图 3-1-16 验证驱动程序画面

11. 完成后，使用 USB 线连接电脑和人机，等待几秒，开启 Screen Editor 软件，执行【工具】下拉式菜单【取得目前韧体序号】，若顺利取得韧体序号代表软件安装成功，且电脑可以透过 USB 与 HMI 通讯。

3.1.3 升级 Screen Editor

用户在收到 Screen Editor 软件更新通报后，可到台达网页下载软件更新程序(或称之为升级文件)Patch_DOPB_yyyymmdd，升级文件包含了韧体以及功能上的更新与加强。

进行升级动作前务必确认旧版 Screen Editor 程序路径是否是预设的 C:\Program Files\Delta Industrial Automation\Screen Editor 2.00.XX，否则升级动作将无法成功。

【升级文件 Patch_DOPB_yyyymmdd 安装方式】

在 Windows 视窗下，点击升级文件 (Patch_DOPB_yyyymmdd) 后，程序便会开始进行安装程序 (图 3-1-17)。信息对话框内会显示可安装的元件项目。目前软件并无其他可选择的元件，因此直接按下【Install】键，升级文件便开始执行安装 (图 3-1-18)。未来若有提供其他元件选择，用户可自行选取所需安装的项目安装。而安装过程进行中可按下【Show Details】键，显示安装项目 (图 3-1-19)。安装完成如图 3-1-20、图 3-1-21，按下确定后即完成软件升级动作。

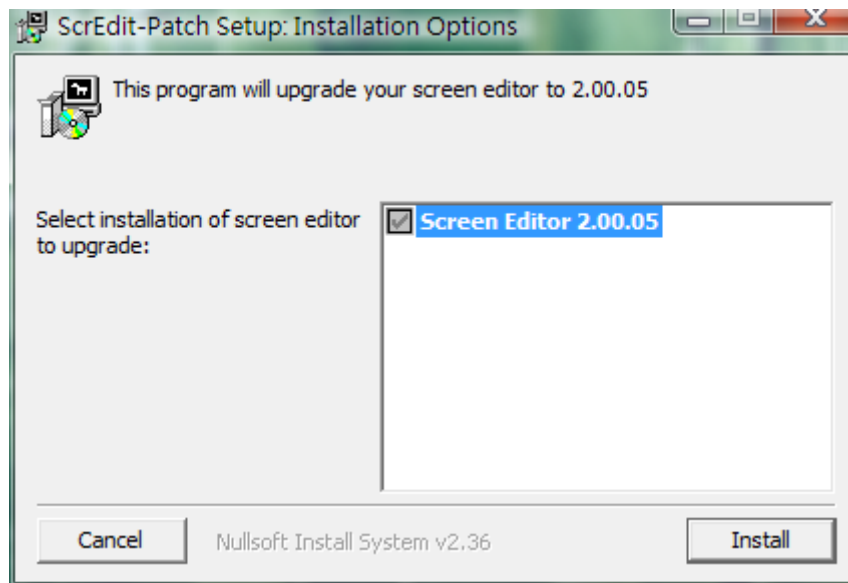


图 3-1-17 升级文件 Patch_DOPB_yyyymmdd 开启画面

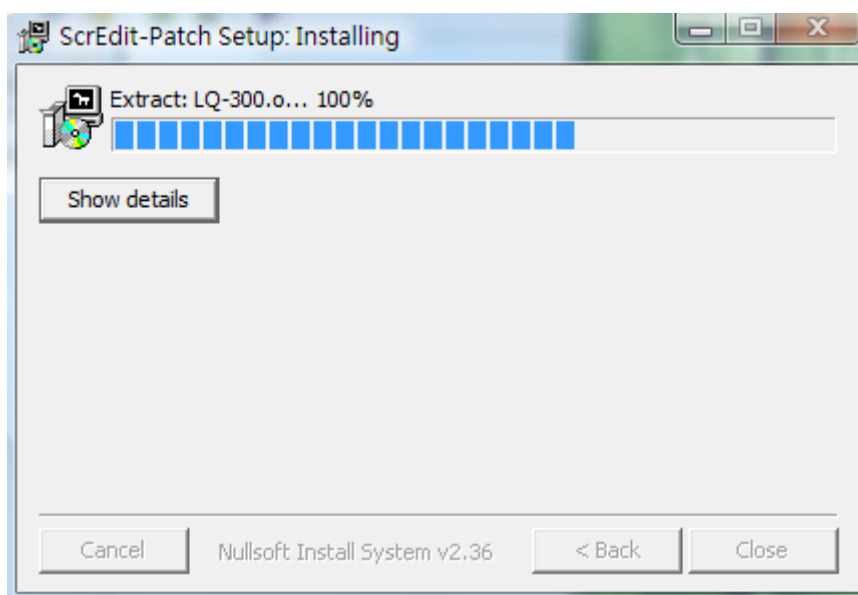


图 3-1-18 安装过程

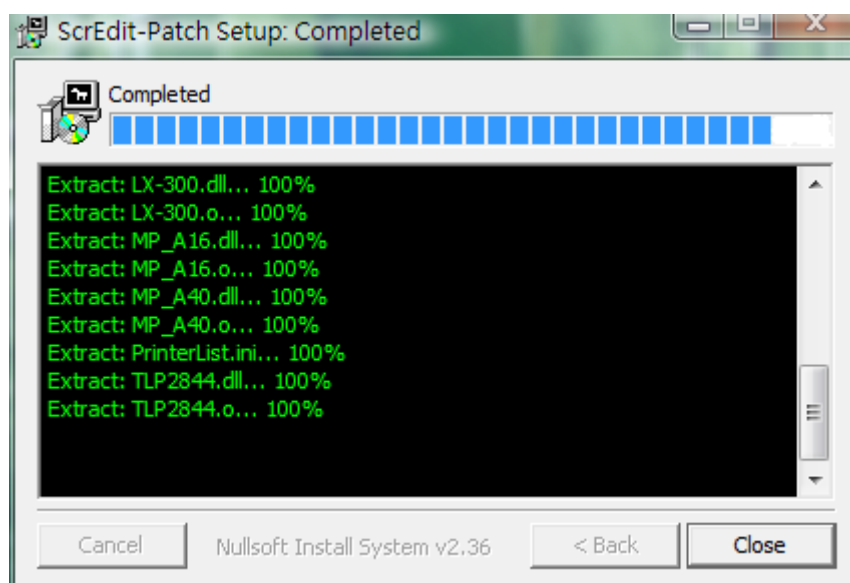


图 3-1-19 Show Details 画面

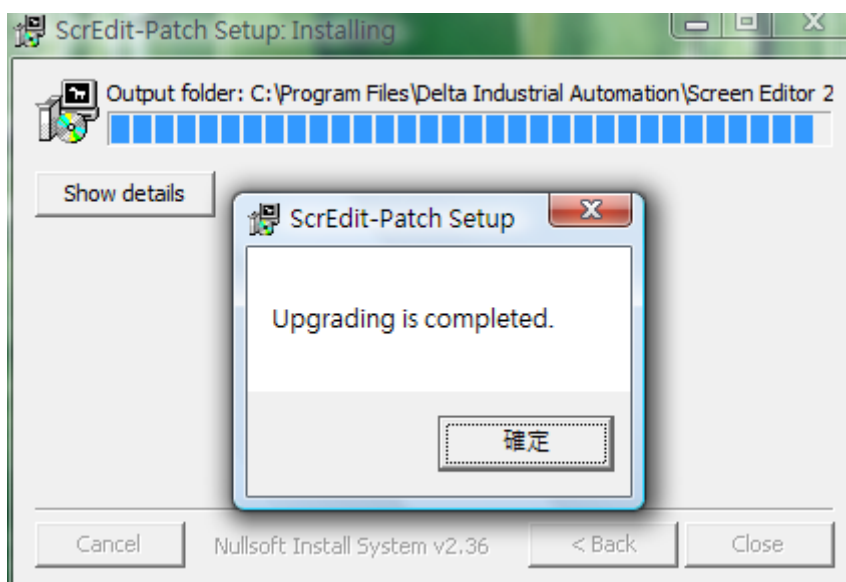


图 3-1-20 安装完成画面

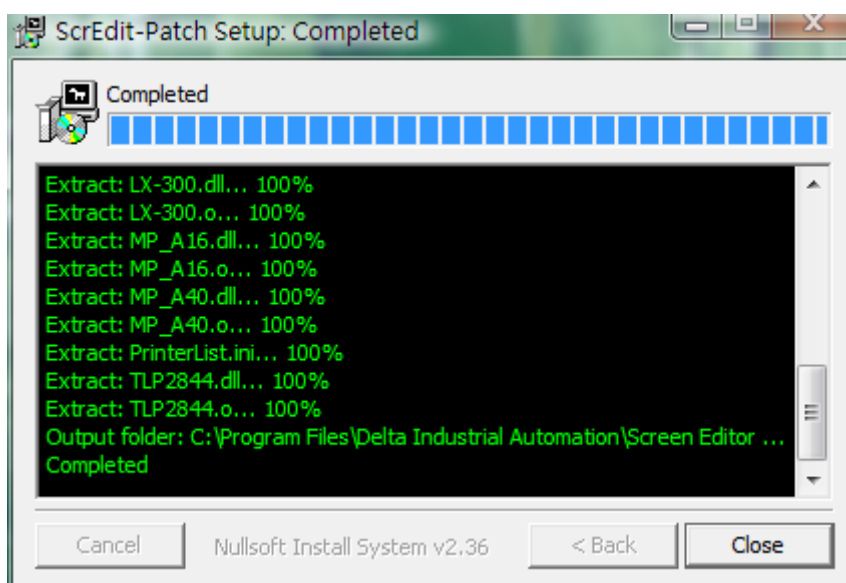


图 3-1-21 安装完成画面

3.2 执行 Screen Editor 与视窗功能说明

3.2.1 执行 Screen Editor 应用程序

当安装好了 Screen Editor 后, 请依照下列步骤来执行 Screen Editor:

1. 点击 Window 视窗中的 Screen Editor 2.00.XX 图示(红色框框所示), 即可执行 Screen Editor (图 3-2-1、图 3-2-2)。



图 3-2-1

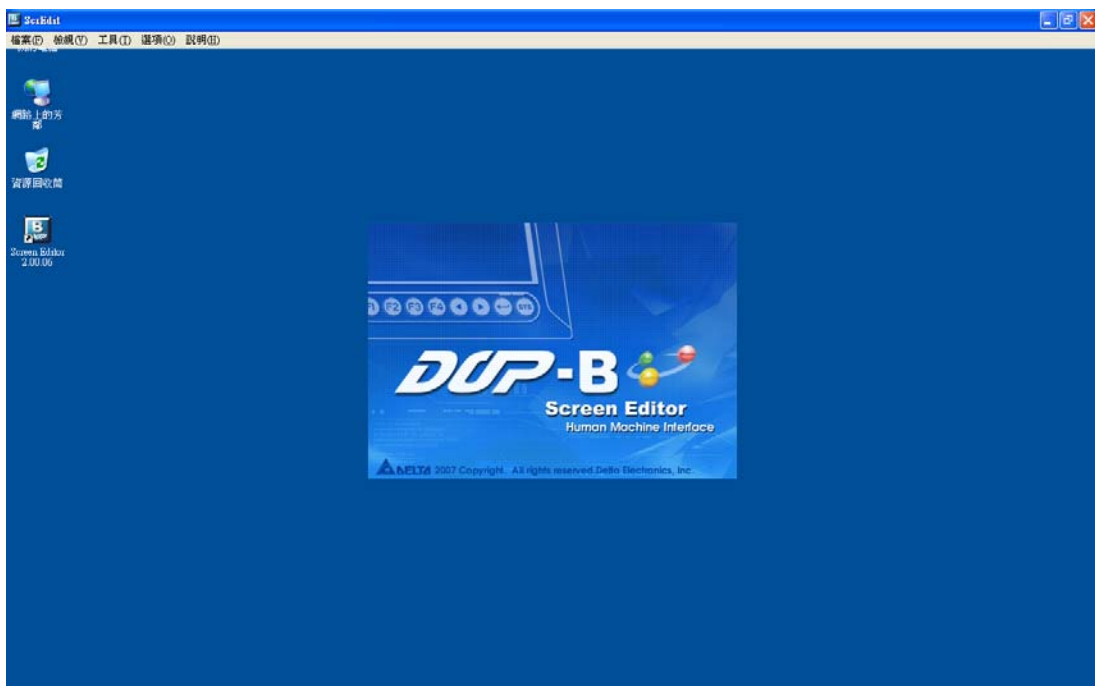


图 3-2-2 程序开始执行时的画面

当您第一次开启 Screen Editor 应用程序时，Screen Editor 应用程序将会呈现无编辑档案时的画面（图 3-2-3）。

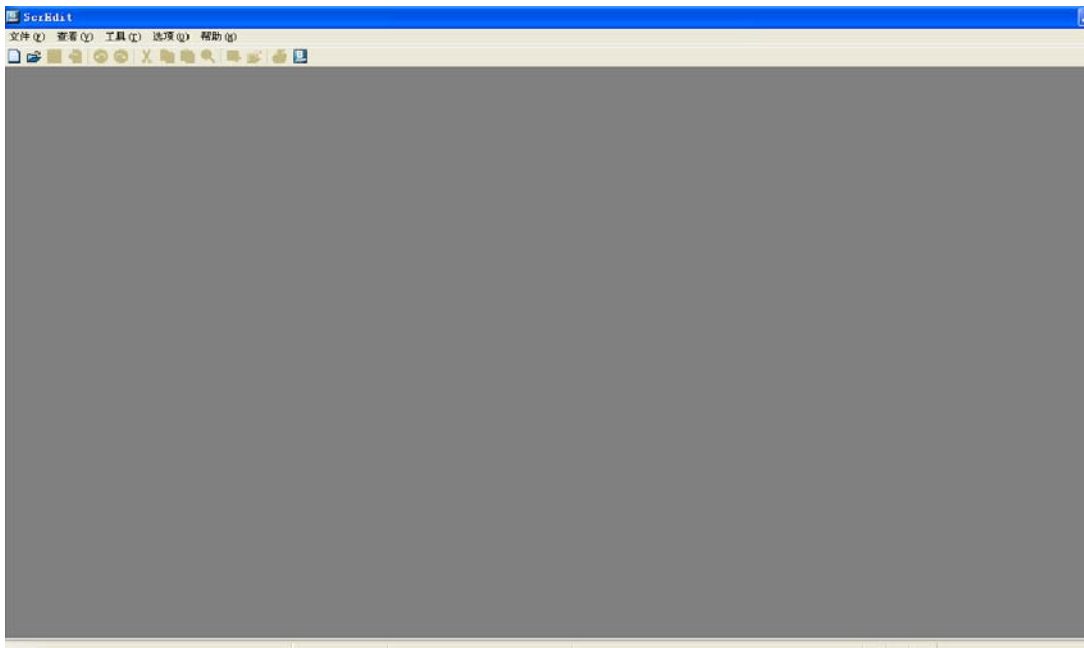


图 3-2-3 第一次开启 Screen Editor 时的画面


2. 按下  新增一个编辑专案，Screen Editor 应用程序会弹出一对话框，如(图 3-2-4)所示。其中专案名称预设为 HMI，画面编号预设为 Screen_1，画面编号预设为 1，以上三个参数，用户皆可自行输入更改；人机界面种类、Base Port 控制器及打印机，用户也可依照自己的设备型号自行选择机型。



图 3-2-4 Screen Editor 新增专案编辑对话框

3. 各选项皆确认选择无误后，按下确定按钮，直接执行下一步，Screen Editor 应用程序会建立一新编辑画面如(图 3-2-5)。

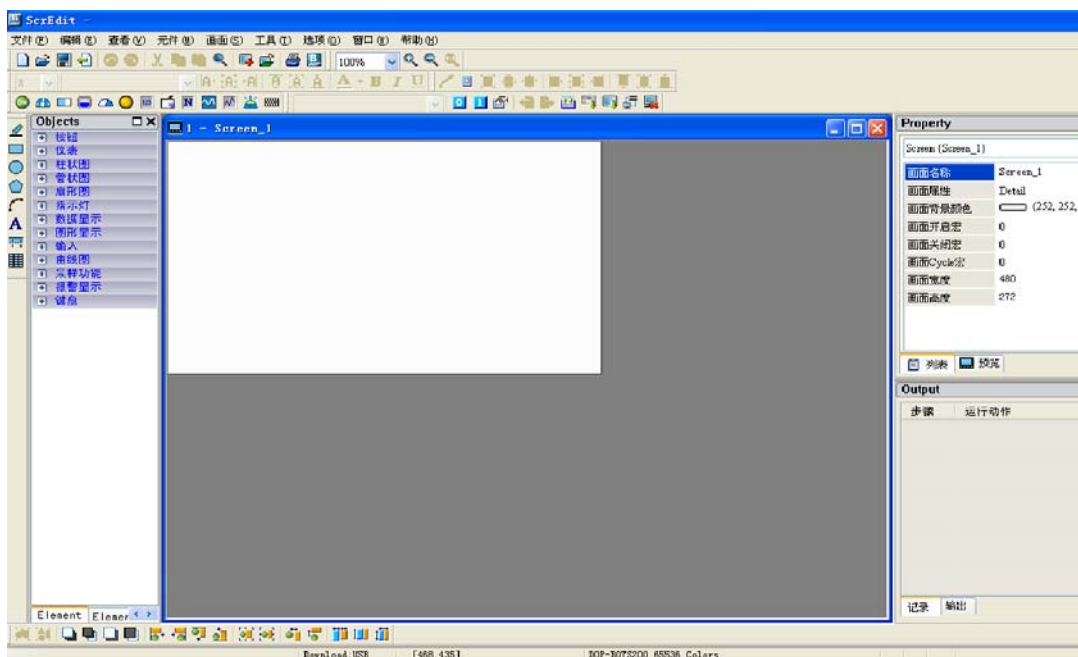


图 3-2-5 Screen Editor 编辑画面

3.2.2 视窗功能说明

在 Screen Editor 编辑视窗中被分为功能菜单、工具栏、元件工具箱与元件库、元件属性表、输出栏、画面编辑区及状态栏七个区域，七个区域预设版面配置如图 3-2-5，而各区域说明如下所述。

1. 功能菜单(Menu)

在 Screen Editor 应用程序界面下，提供九项功能菜单。

文件(F) 编辑(E) 查看(V) 元件(M) 画面(S) 工具(T) 选项(O) 窗口(W) 帮助(H)

2. 工具栏(Toolbar)

工具栏(图 3-2-6)都是标准的 Windows® 视窗程序，因此就像 Windows®里面的工具栏一样，可以随意安排它的位置。例如把元件工具栏移到屏幕左侧，用户可以依照自己的使用习惯，来放置工具栏(图 3-2-7)所示。以下为 Screen Editor 所提供的工具栏的种类。

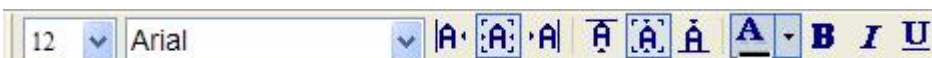
标准工具栏



缩放工具栏



文字工具栏



图形工具栏



元件工具栏



元件规划工具栏



绘图工具栏

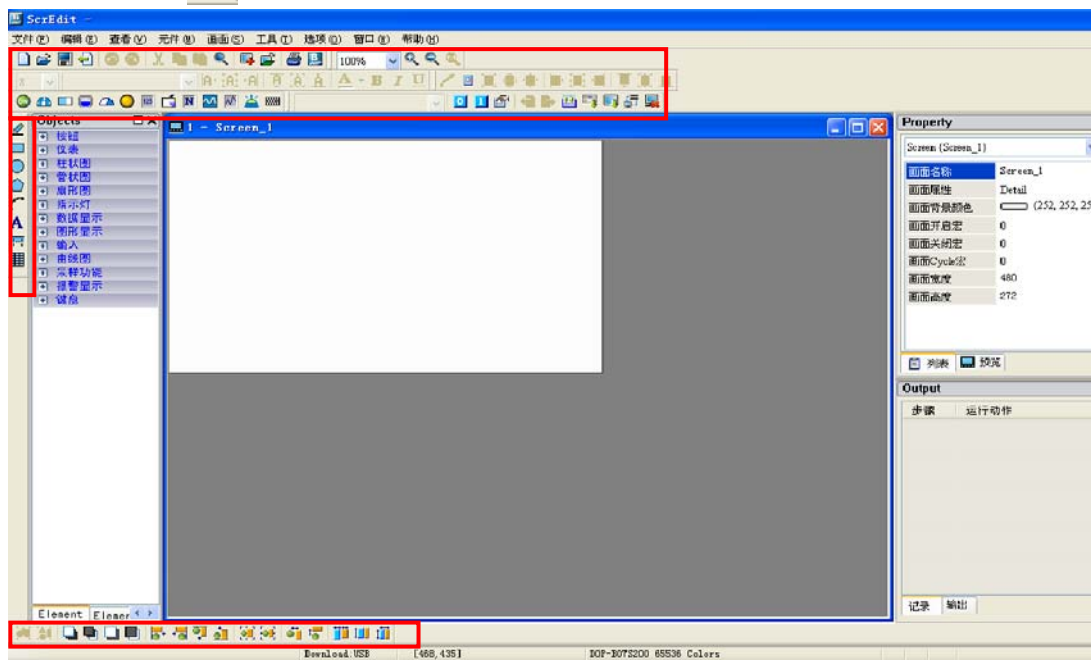


图 3-2-6 Screen Editor 工具栏

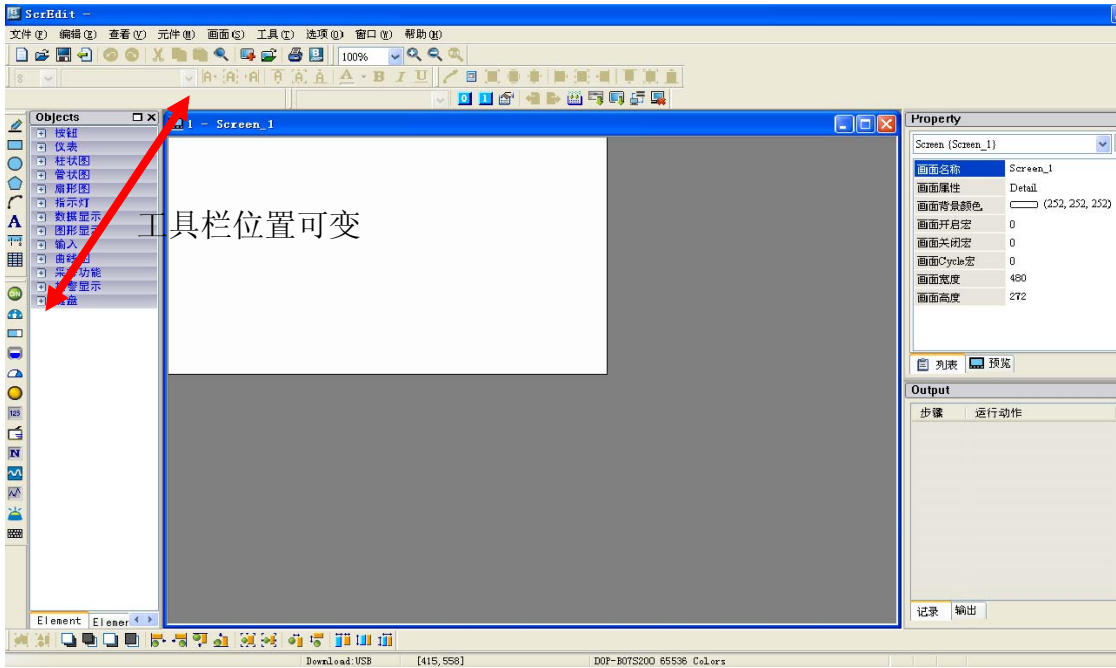


图 3-2-7 元件工具栏移到屏幕左侧

3. 元件工具箱与元件库

提供元件与已编辑完成的元件。用户编辑完成的元件可存放至元件库 (Element bank) 中，下次要重复使用时拖曳拉出到视窗即可使用，如下图 3-2-9 所示。

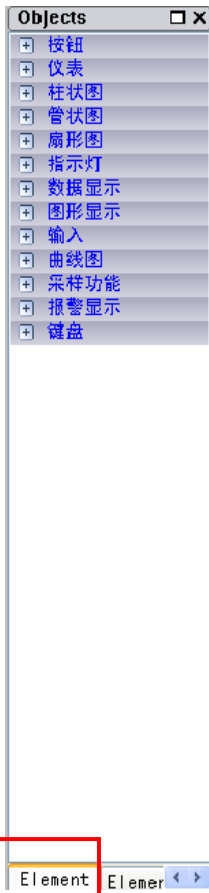


图 3-2-8 元件工具箱

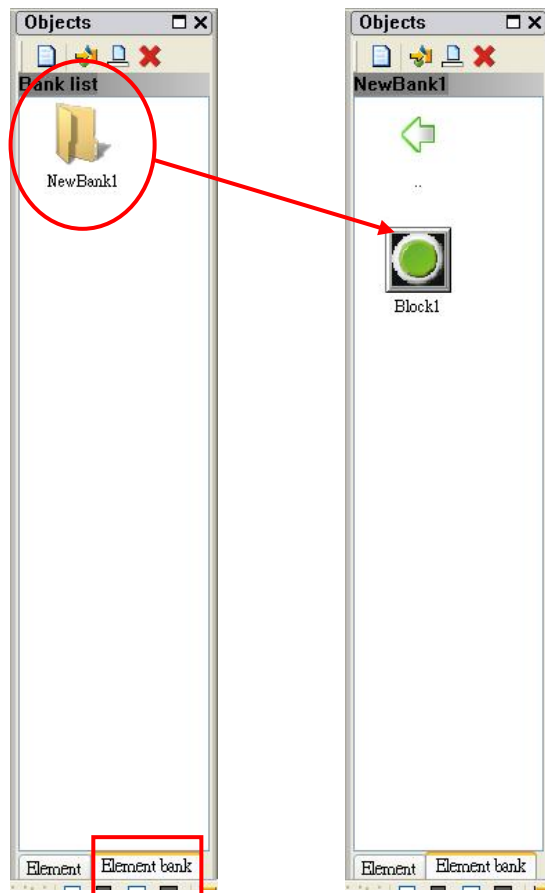


图 3-2-9 元件库

4. 元件属性表

提供各种元件列表的属性设定与编辑画面预览（图 3-2-10）。



图 3-2-10 属性表：元件列表的属性设定与编辑画面预览

5. 记录栏与输出栏

记录用户编辑的动作及画面编译时的输出信息（图 3-2-11）。当执行编译功能时，Screen Editor 会进行程序编译，如果有错误，输出栏会产生对应信息。用户点击错误信息后，则自动跳至错误元件所在的画面以方便除错。



图 3-2-11 记录栏与输出栏内容

6. 画面编辑区

依照用户所选定的 DOP 系列人机界面种类，给予适当的编辑范围（图 3-2-12）。下图为一个编辑画面的范例。

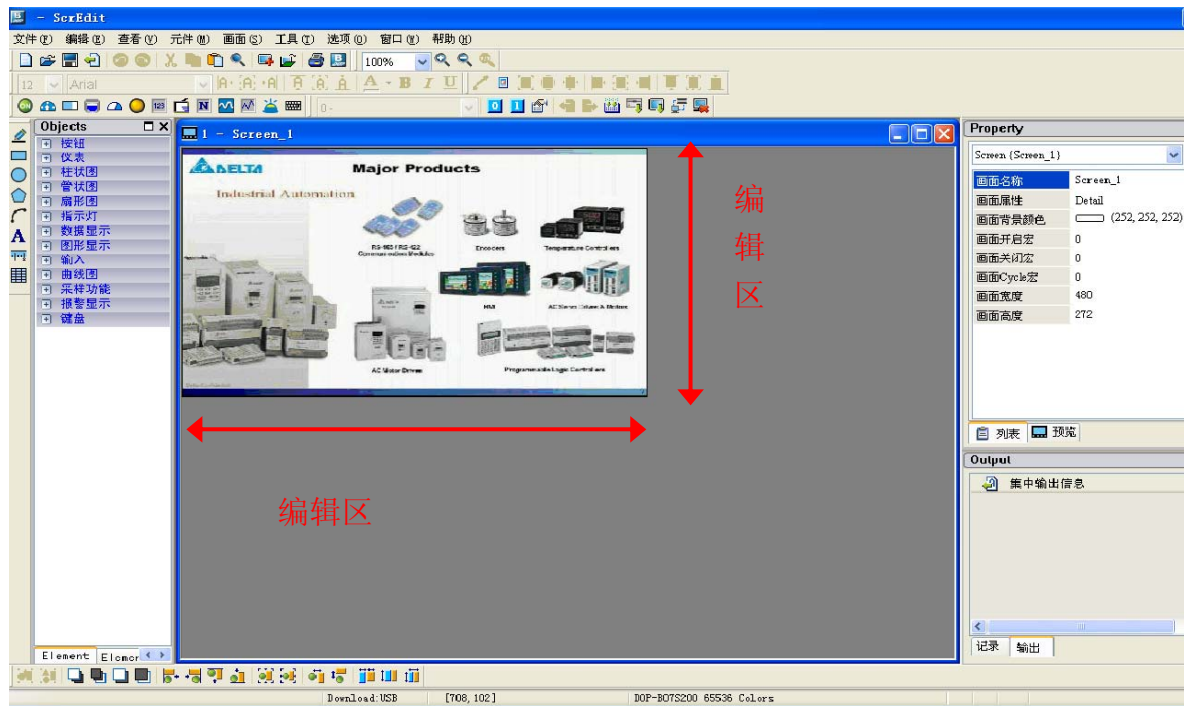


图 3-2-12 Screen Editor 画面编辑区

7. 状态列

此栏会显示目前编辑的状态（图 3-2-13）。

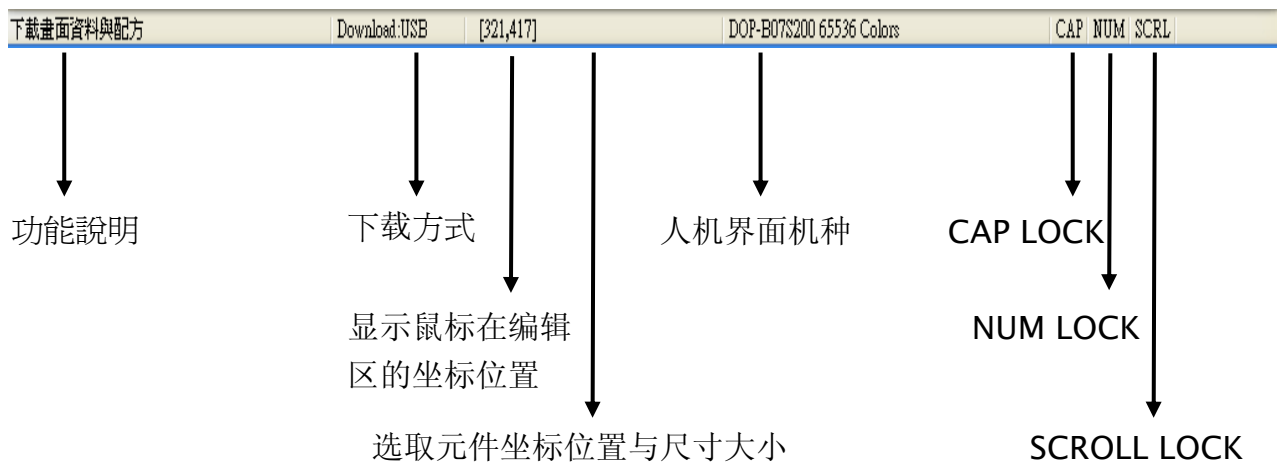



图 3-2-13 状态栏

以上七点为各个区域的功能说明，详细功能菜单与工具栏使用细项将于 3.5 节开始一一详述。

3.10 功能菜单—工具

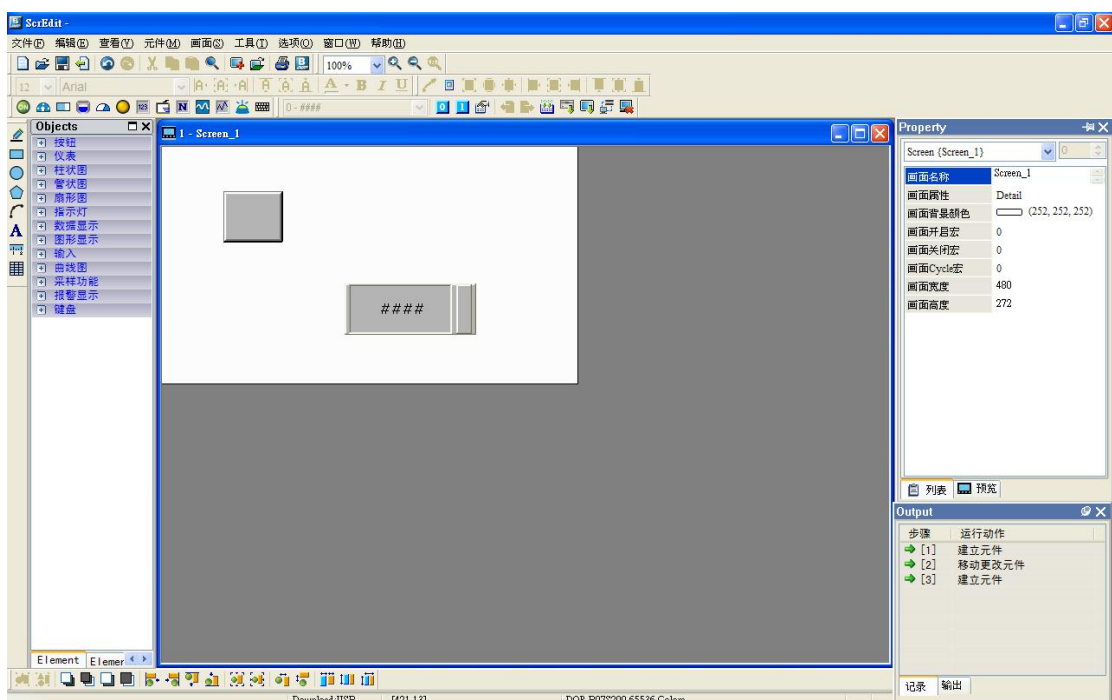
| | | |
|---|-----------|---------|
|  | 编译 | Ctrl+F7 |
|  | 下载画面数据与配方 | Ctrl+F8 |
|  | 上传画面数据与配方 | |
|  | 下载画面数据 | Ctrl+F9 |
| | 上传配方 | |
| | 下载配方 | |
|  | 在线模拟 | Ctrl+F4 |
|  | 离线模拟 | Ctrl+F5 |
| | 更新砵体 | |
| | 取得目前砵体序号 | |


3.10.1 编译

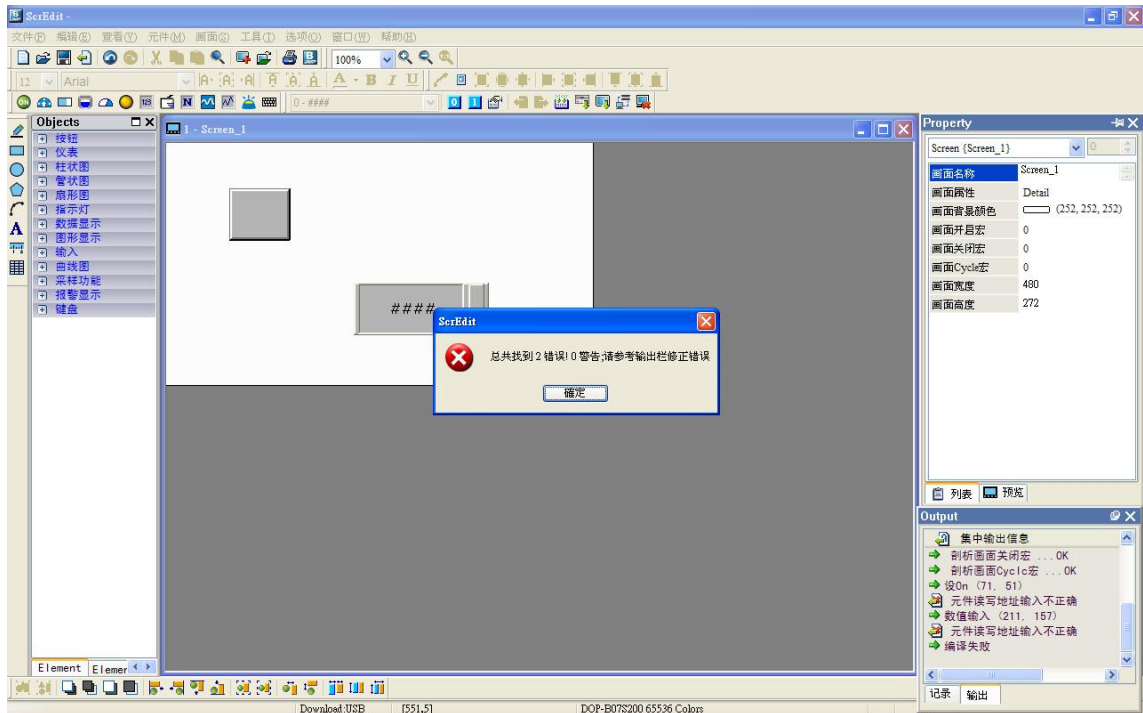
选取此项目可以把编辑好的画面文件，编译成人机可以接受的格式并同时检查画面编辑是否有错误。编译过程中会将信息输出至输出栏。如果有错误，也会一并列出提醒用户，如果有错误发生时，则不会产生目的文件。用户可以直接选取工具中的编译，也可以点选工具栏上的图示或是使用系统内建的热键 Ctrl+F7。如果此一专案为新开启的专案时，则在编译前会要求用户先储存文件之后，才开始编译；如果此专案已有存档或是旧文件，则直接编译。而编译完成后，Screen Editor 会自动作一次保存的动作。

编译错误时除错流程

1. 建立一专案。
2. 再画面上建立一个按钮元件、一个数值输入元件，元件属性保持预设不变更，如下图所示：




- 按下  执行编译，会有错误信息对话框如下图所示。在错误信息对话框里，明白的指出有两项错误。我们可以从输出栏看到有这两项错误详尽的说明。



- 这时我们在输出栏以鼠标快速点击最后一项错误，Screen Editor 会自动切换到该错误项目的元件上，并将此元件圈选起来。

3.10.2 下载画面数据与配方

下载画面数据跟配方到 DOP 人机端。可点选工具里面的选项或是直接点选规划工具栏里面的图示，或是使用系统内定的热键 Ctrl+F8。如果 PC 跟人机连接的界面无法接通，那么会出现错误信息来警告用户（图 3-10-1、图 3-10-2、图 3-10-3、图 3-10-4）。您可以在两个地方设置下载传输界面：

1. 菜单里的「选项」→「设置模块参数」
2. 菜单里的「选项」→「环境设置」，传输界面可以是 USB 或是 RS232。

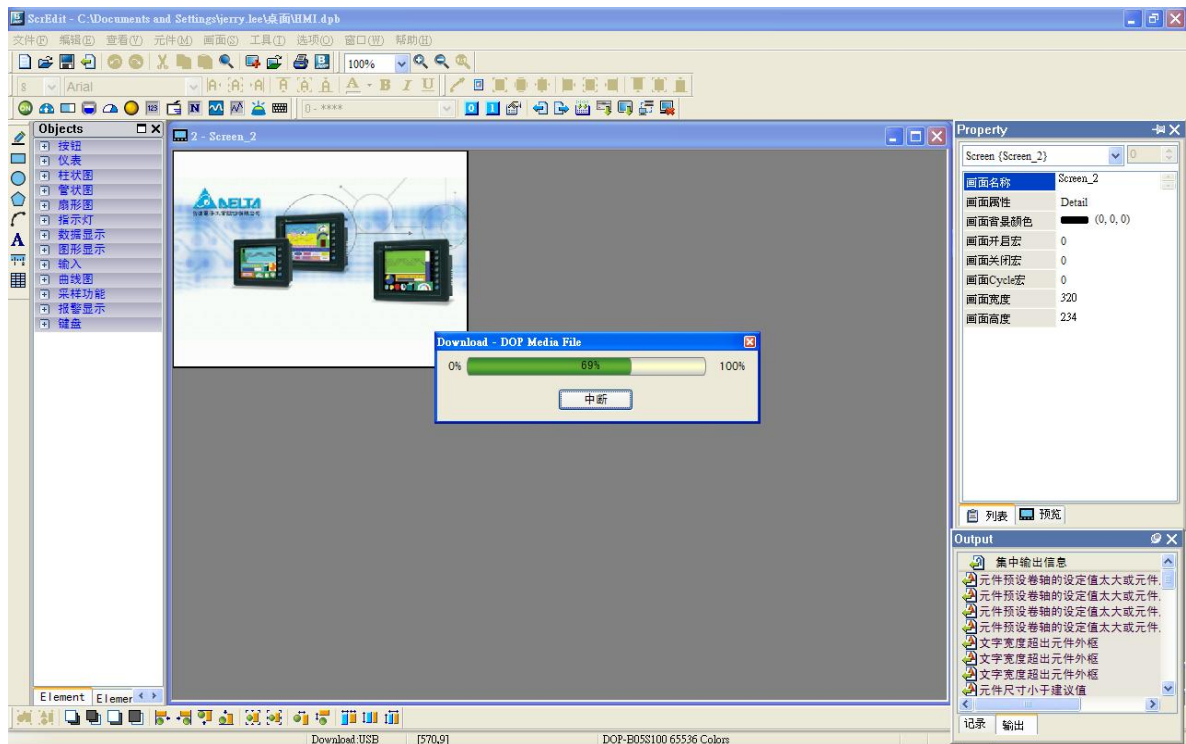


图 3-10-1 开始下载由 0%开始到 100%完成

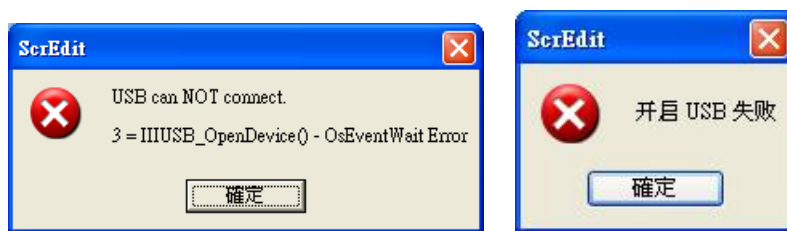


图 3-10-2 USB 开启错误时的错误信息



图 3-10-3 下载过程中将传输线拔除或是通讯中断的错误信息



图 3-10-4 机型设置错误

3.10.3 上载画面数据与配方

选择此功能后，软件会先要求输入密码（图 3-10-5）（此密码为「设置模块参数」中设置的最高权限密码），输入正确密码后，系统会出现一个另存对话框（图 3-10-6）。输入要储存的文件名称后按下保存后将出现画面（图 3-10-7），系统将弹出一上载的进度栏。当进度表达达到 100%后则代表上载结束，用户可以按下【中断】按钮终止此上载程序，此选项会读取人机中的画面数据并上载回电脑。而在「环境设置」中可设置「上载包括图形资料」（图 3-10-8），除了上载画面数据外也可以将人机中的图形库一并上传。以方便用户于原始文件消失时，仍然可以从人机得到原来的画面数据来加以修改。

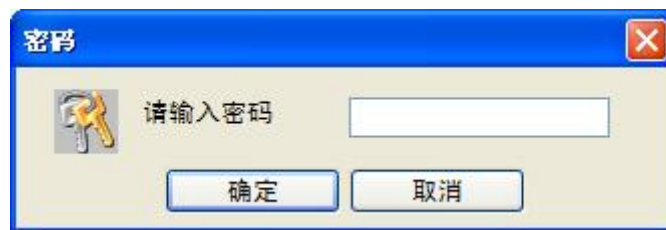


图 3-10-5 上载时输入密码对话框

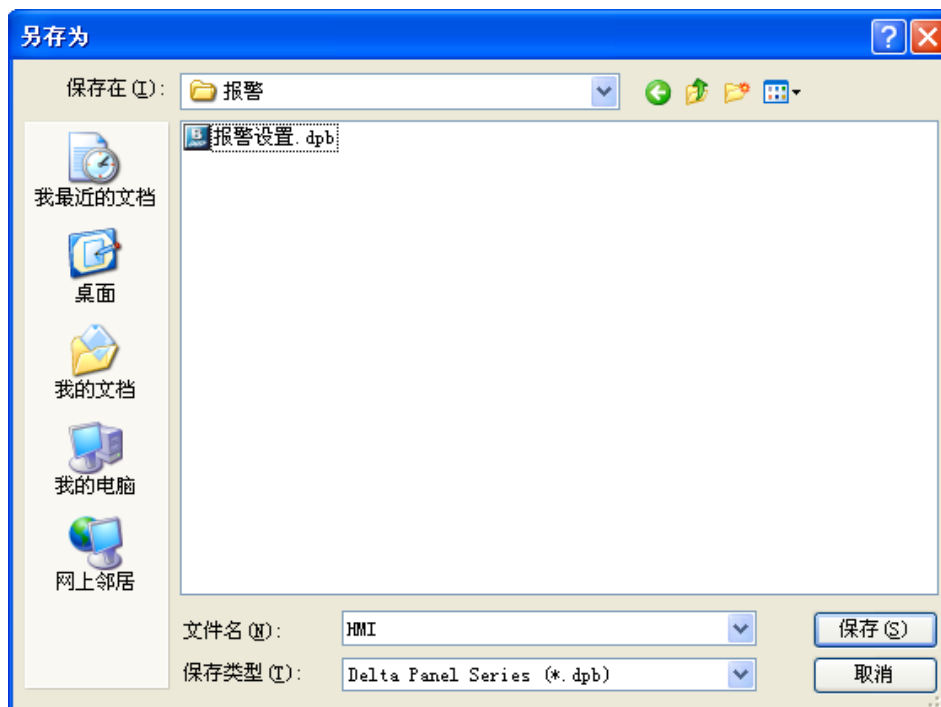


图 3-10-6 另存对话框

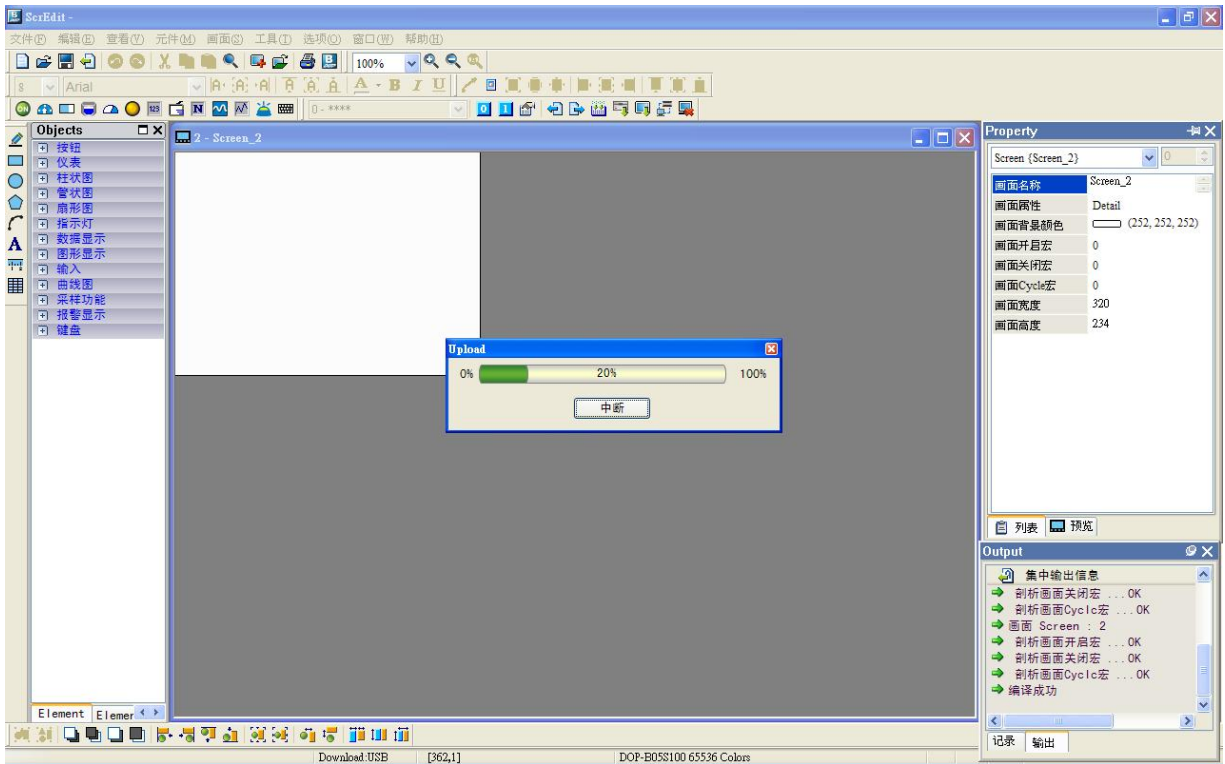



图 3-10-7 上载画面数据与配方时的画面



圖 3-10-8 环境设置

3.10.4 下载画面数据

下载画面数据到 DOP 人机端，过程与下载画面数据与配方一样，只是没有下载配方而已。

可点选工具里面的选项或是直接点选规划工具栏里面的图示，或是使用系统内定的热键 Ctrl+ F9。

3.10.5 上载配方

选取好上载配方的选项后，与上载画面数据与配方功能类似。但是只上载配方资料，而且一样要先输入密码（此密码为「设置模块参数」中设置的最高权限密码），输入正确密码后才能得到资料。

3.10.6 下载配方

只下载配方到人机端。如果只是修改配方，其他的画面编辑资料并无修改，则只要执行此一选项即可，可以节省下载的时间。点选所要下载的配方文件之后，再下载到人机端便可以了（图 3-10-9）。

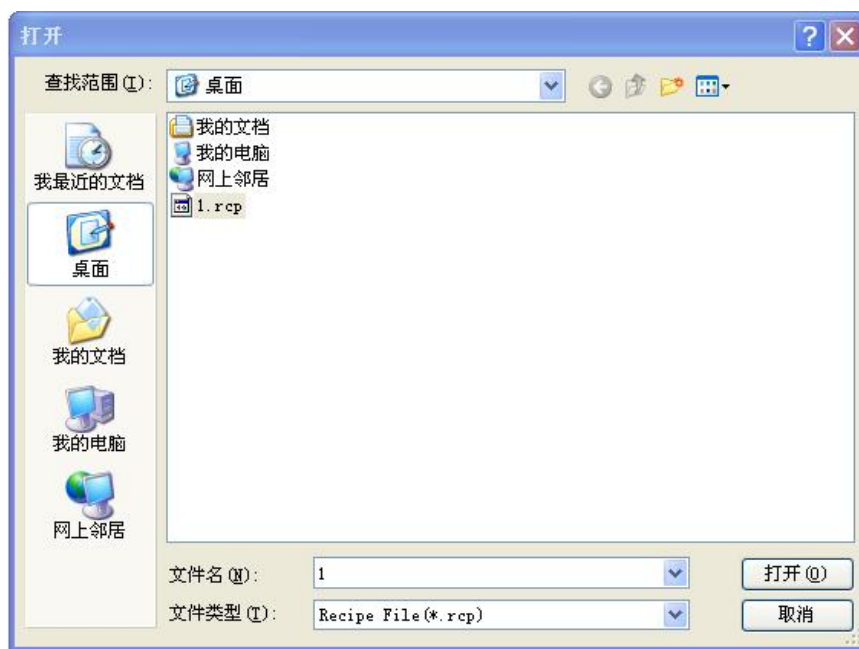



图 3-10-9 下载配方时需指定配方文件的打开对话框

3.10.7 在线模拟

用电脑完整模拟 DOP 人机的程序执行动作，并通过电脑的通讯口去与 PLC 通讯，可模拟人机实际使用时动作，因此没有接 PLC 的话将无法执行程序。可点选工具里面的在线模拟选项或是点选规划工具栏的图示，或是使用系统内定的热键 Ctrl+F4。而再执行之前，模拟器会先要求设置对应的通讯口供通讯用（图 3-10-10），在线模拟执行时的情况请参考（图 3-10-11）。

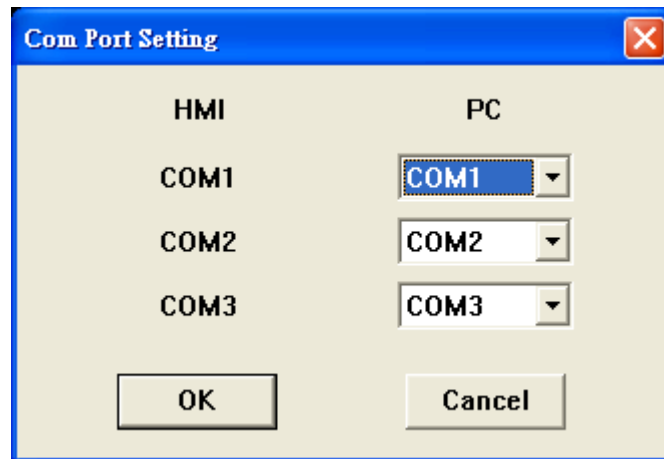


图 3-10-10 在线模拟 COM 设置



图 3-10-11 在线模拟画面

3.10.8 离线模拟

离线模拟可用于测试所写的画面、读写位址以及宏等等是否正确。并不需要真的接上 PLC 来执行，可直接点选工具选项里面的离线模拟选项，也可以点选规划工具栏里面的图标



或是使用系统内定的热键 Ctrl+F5。执行时画面请参考（图 3-10-11）。

3.10.9 更新韧体

此选项是为了能够更新人机界面里面的韧体。更新韧体的目的是为了让人机界面运行达到最佳状态，因此使用前请确认您所使用的软件版本与人机的韧体版本是否一致。可使用「工具」→「取得目前韧体序号」来得到人机的韧体版本（图 3-10-12），而软件所搭配的韧体版本请看「说明」→「关于 Screen Editor」（图 3-10-13）。韧体更新画面请参考（图 3-10-14）。



图 3-10-12 取得目前韧体序号



图 3-10-13 关于 Screen Editor

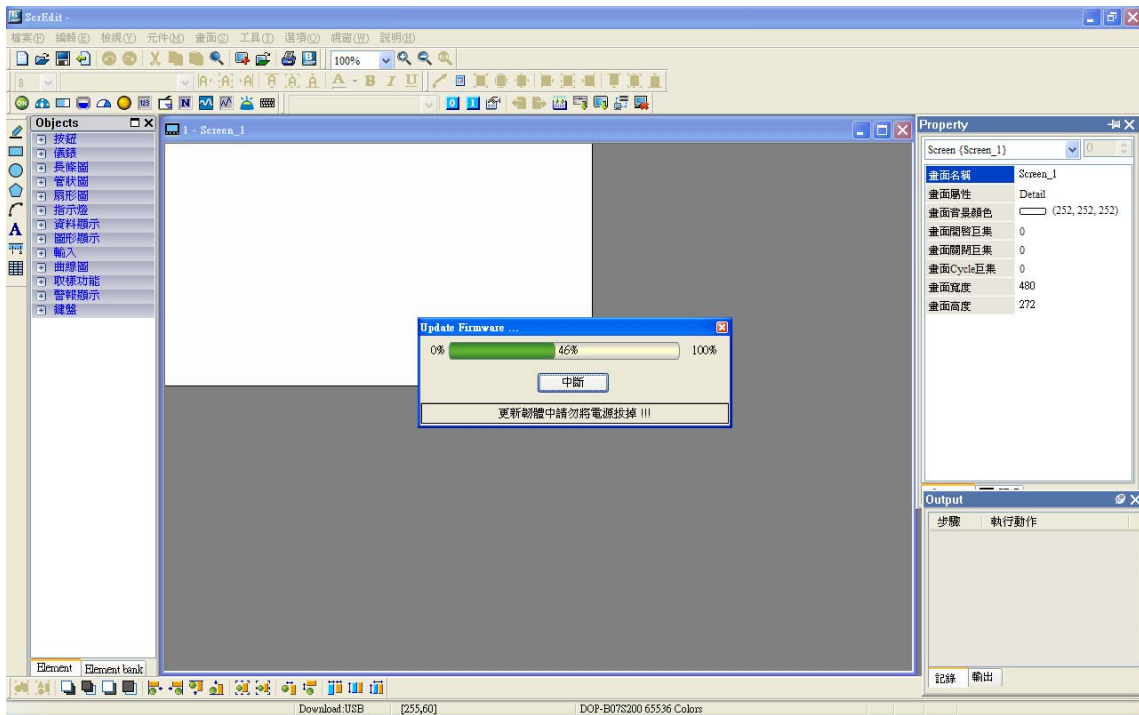


图 3-10-14 更新韧体画面

3.10.10 取得目前韧体序号

由电脑与人机连线（经由 USB Cable、COM Port）而取得目前所连线人机界面的相关信息。



图 3-10-15 取得目前韧体序号画面

3.11 功能菜单—选项

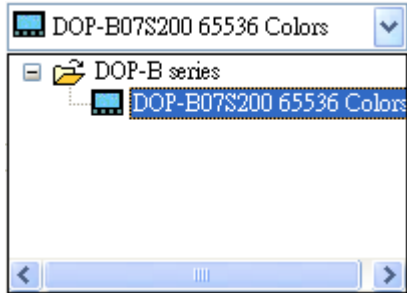
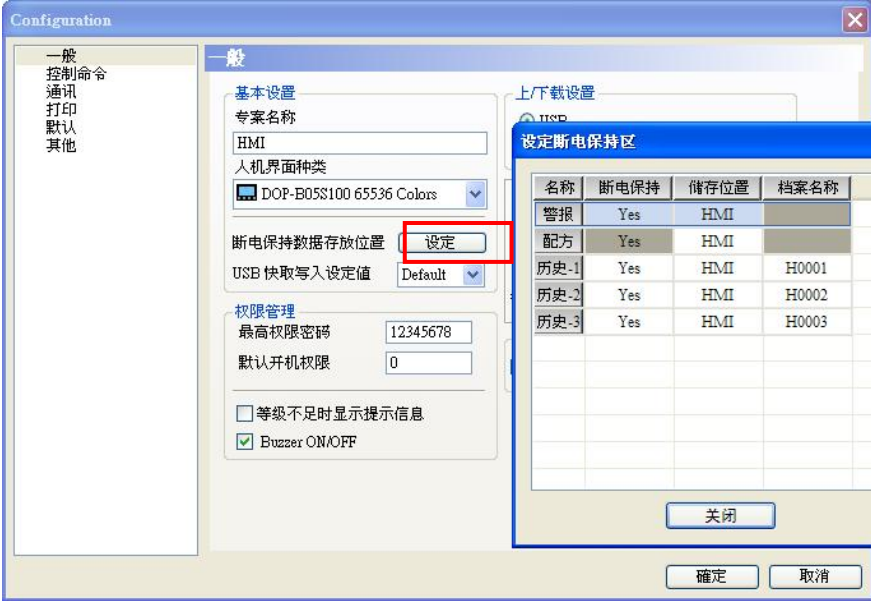



3.11.1 设置模块参数


此项只能经由点击选项里面的设置模块参数来启动。其中分为【一般】、【控制命令】、【通讯】、【打印】、【默认】、【其它】六个页面，各页面详细说明请参照模块参数的设置。

模块参数设置说明（一般页面设置）

| | |
|-------------|----------------------|
| 专案名称 HMI | 专案的名称也是存档时，文件的默认的名称。 |
|-------------|----------------------|

| 模块参数设置说明（一般页面设置） | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|---|------|-------|------|------|----|-----|-----|--|----|-----|-----|--|------|-----|-----|-------|------|-----|-----|-------|------|-----|-----|-------|
| 人机界面种类 | <p>选择 DOP 人机界面的种类, 针对不同的人机编辑所需要的功能。</p>  | | | | | | | | | | | | | | | | | | | | | | | | |
| 断电保持数据存放位置 | <p>人机界面所提供的断电保持数据存放位置包含有 HMI、U 盘, 但仍要看您所选择的人机界面种类与功能而定。断电保持区的设置如下图操作, 按下设置按钮后出现设置断电保持区对话框, 此对话框可以设置包括警报, 配方, 历史资料的断电保持区。</p>  <table border="1" data-bbox="1066 891 1426 1205"> <caption>设定断电保持区</caption> <thead> <tr> <th>名称</th> <th>断电保持</th> <th>储存位置</th> <th>档案名称</th> </tr> </thead> <tbody> <tr> <td>警报</td> <td>Yes</td> <td>HMI</td> <td></td> </tr> <tr> <td>配方</td> <td>Yes</td> <td>HMI</td> <td></td> </tr> <tr> <td>历史-1</td> <td>Yes</td> <td>HMI</td> <td>H0001</td> </tr> <tr> <td>历史-2</td> <td>Yes</td> <td>HMI</td> <td>H0002</td> </tr> <tr> <td>历史-3</td> <td>Yes</td> <td>HMI</td> <td>H0003</td> </tr> </tbody> </table> | 名称 | 断电保持 | 储存位置 | 档案名称 | 警报 | Yes | HMI | | 配方 | Yes | HMI | | 历史-1 | Yes | HMI | H0001 | 历史-2 | Yes | HMI | H0002 | 历史-3 | Yes | HMI | H0003 |
| 名称 | 断电保持 | 储存位置 | 档案名称 | | | | | | | | | | | | | | | | | | | | | | |
| 警报 | Yes | HMI | | | | | | | | | | | | | | | | | | | | | | | |
| 配方 | Yes | HMI | | | | | | | | | | | | | | | | | | | | | | | |
| 历史-1 | Yes | HMI | H0001 | | | | | | | | | | | | | | | | | | | | | | |
| 历史-2 | Yes | HMI | H0002 | | | | | | | | | | | | | | | | | | | | | | |
| 历史-3 | Yes | HMI | H0003 | | | | | | | | | | | | | | | | | | | | | | |
| 最高权限密码 | 设置最高权限等级 8 的密码, 同时此密码也是项目储存后的保护密码, 密码文字格式为 0~F 的 16 进制单位。 | | | | | | | | | | | | | | | | | | | | | | | | |
| 默认开机权限 | 设置开机时的使用权限, 等级最高为 7 最小为 0。 | | | | | | | | | | | | | | | | | | | | | | | | |
| 等级不足时显示提示信息 | 当画面元件有设置用户权限, 而用户的等级不足时, 该元件则会显示  图示, 提示用户权限不足。 | | | | | | | | | | | | | | | | | | | | | | | | |
| Buzzer ON/OFF | Buzzer 是否要启动。勾选表示 ON | | | | | | | | | | | | | | | | | | | | | | | | |
| 通讯最佳化模式 | <p>动态最佳化</p> <p>每次与控制器通讯时, 只读取当时显示画面上元件的数值, 并依照画面元件多少, 动态调整资料读取长度, 可有效节省与控制器的通讯时间。若无特殊的需要, 建议使用此最佳化方式。</p> <p>动态最佳化的启动, 取决于通讯页面的设置选项【读取最佳化】是否有勾选。若无勾选, 虽在此处有设置动态最佳化的模式, 系统仍然不会启动。</p> | | | | | | | | | | | | | | | | | | | | | | | | |

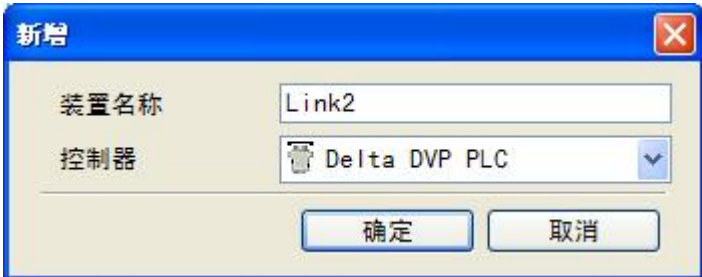
| 模块参数设置说明（一般页面设置） | |
|------------------|---|
| 通讯最佳化模式 | <p>每次与控制器通讯时，读取所有画面上元件的数值，但是不会依照元件多少，调整资料读取长度。</p> <p>静态最佳化的启动，取决于通讯页面的设置选项【读取最佳化】是否有勾选，若使用静态最佳化但发觉通讯反应偏慢可将「长度限制」启动，避免读取过长的连续地址而造成画面更新变慢。若无勾选【读取最佳化】，虽在此处有设置静态最佳化的模式，系统仍然不会启动。</p> |
| 上/下载设置 | 可以选择 USB, PC 通讯端口（即：RS232），或是以太网网络作为上/下载的传输界面。 |
| 开机延迟时间 | 设置开机延迟时间以等待控制器（如：PLC）的启动，从 0 到 255 秒。 |
| Clock 宏延迟时间 | 设置每次执行 Clock 宏时的间隔时间，值从 100~65535ms。 |
| Clock 宏优先权 | 设置 Clock 宏执行优先权，可确保较准确的 Clock 宏延迟时间。 |
| 背景宏的更新周期 | 设置背景宏一次周期执行的行数。设置的行数为 1~512 行 |
| 显示磁盘存取失败警告信息 | 设置当警报，历史缓冲区，配方设在断电保持区时，如 U 盘，若是存取失败，是否要显示错误信息。 |

| 模块参数设置说明（控制命令设置） | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---------------------|----------------------|---------------------|--------|------|-------|--------|--|-------|-----------|--|-------|-----------|--|-------|----------|--|-------|-------|--|----------|------|------|-------|------|------|--|------|-------|--|------|-------|--|-------|-------|--|
|  <p>The screenshot shows the 'Configuration' dialog box with the 'Control Command' tab selected. It includes input fields for 'Command Area Address' (1@D0), 'Command Sampling Period' (300 ms), 'Status Area Address' (1@D10), and 'Length' (8). There is also a checkbox for 'Action Completion Clear Flag'. Below these is a table with columns for 'Control State', 'Command Area Address', and 'Status Area Address'. The table lists various control states and their corresponding addresses.</p> <table border="1" data-bbox="555 1361 1265 1783"> <thead> <tr> <th>Control State</th> <th>Command Area Address</th> <th>Status Area Address</th> </tr> </thead> <tbody> <tr> <td>一般控制状态</td> <td>1@D0</td> <td>1@D10</td> </tr> <tr> <td>画面切换状态</td> <td></td> <td>BIT 0</td> </tr> <tr> <td>警报缓冲区清除状态</td> <td></td> <td>BIT 3</td> </tr> <tr> <td>警报计数器清除状态</td> <td></td> <td>BIT 4</td> </tr> <tr> <td>USB盘快取写入</td> <td></td> <td>BIT 5</td> </tr> <tr> <td>使用者等级</td> <td></td> <td>BIT 8-10</td> </tr> <tr> <td>画面编号</td> <td>1@D0</td> <td>1@D11</td> </tr> <tr> <td>控制标志</td> <td>1@D1</td> <td></td> </tr> <tr> <td>通讯开关</td> <td>BIT 0</td> <td></td> </tr> <tr> <td>背灯开关</td> <td>BIT 1</td> <td></td> </tr> <tr> <td>蜂鸣器开关</td> <td>BIT 2</td> <td></td> </tr> </tbody> </table> | | Control State | Command Area Address | Status Area Address | 一般控制状态 | 1@D0 | 1@D10 | 画面切换状态 | | BIT 0 | 警报缓冲区清除状态 | | BIT 3 | 警报计数器清除状态 | | BIT 4 | USB盘快取写入 | | BIT 5 | 使用者等级 | | BIT 8-10 | 画面编号 | 1@D0 | 1@D11 | 控制标志 | 1@D1 | | 通讯开关 | BIT 0 | | 背灯开关 | BIT 1 | | 蜂鸣器开关 | BIT 2 | |
| Control State | Command Area Address | Status Area Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 一般控制状态 | 1@D0 | 1@D10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 画面切换状态 | | BIT 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 警报缓冲区清除状态 | | BIT 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 警报计数器清除状态 | | BIT 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| USB盘快取写入 | | BIT 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 使用者等级 | | BIT 8-10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 画面编号 | 1@D0 | 1@D11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 控制标志 | 1@D1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 通讯开关 | BIT 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 背灯开关 | BIT 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 蜂鸣器开关 | BIT 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 命令区起始地址 | 设置系统命令区起始地址。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 命令区长度 | 系统命令区地址长度会随着您所需要的功能增长或减少（例如当您使用 多国语言功能 时，长度设置至少需要 8 个 Words）。可从视窗下方的表格中各个地址所代表的意义， 请注意当系统 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

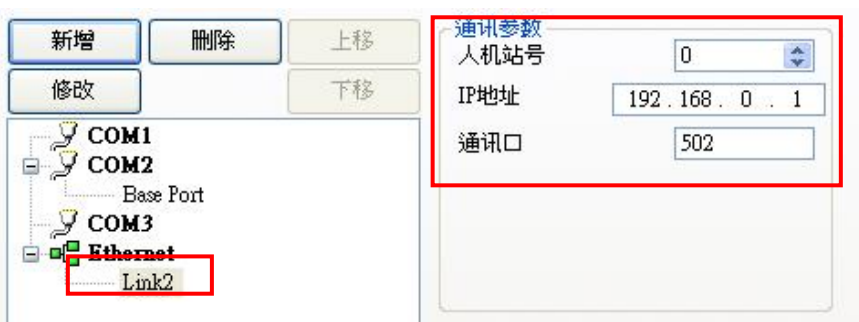
| 模块参数设置说明（控制命令设置） | |
|------------------|--|
| | 命令区长度被设为 0 时，系统命令区是无任何作用的。 |
| 动作完成清除标志 | 在命令区有任一动作结束后就将该控制寄存器清除为 0。 |
| 系统状态区 起始地址 | 设置系统状态区起始地址。长度固定为 8 个 Words，每个 Word 皆指出系统会有不同的状态值。 |

模块参数设置说明（通讯页面设置）



| | | |
|------------------------|------------|---|
| 新增/ 删除 控制器 连线 | 新增 | <p>按下新增按钮决定连线装置名称与控制器，请参阅下图：</p>  <p>依照不同的人机界面型号，可支持连接的控制器个数皆不同（DOP-B 系列人机界面可同时支持 3 种不同的控制器）。</p> |
| | 删除 | 将存在的连线装置删除。（一个项目至少要存在一个连线装置） |
| | 修改 | 改变连线的控制器或者改变装置名称。 |
| 控制器 连线设 置 | 通讯端口 | 设置与人机通讯的通讯端口。 |
| | 密码 | 有些控制器需在通讯前传入一组密码方能作通讯。 |
| | 通讯 延迟时间 | 每次下达通讯命令的间隔时间，范围从 0 ms 到 255 ms。 |
| | Timeout | 通讯过程中，控制器多久无回应，才算是 Timeout，范围从 10 ms ~ 65535 ms。 |

| 模块参数设置说明（通讯页面设置） | | |
|------------------|--|---|
| 控制器 连线设置 | Retry 次数 | 通过程中，控制器无回应，人机端尝试再送通讯命令的次数，若达到设置的 Retry 次数，人机端才会弹出通讯异常对话框。范围从 0 ~ 255 次。 |
| | 读取最佳化 | 启动或关闭最佳化处理模式。设置有参考到此连线之所有元件读取地址，是否要作最佳化的计算。 |
| | 长度限制 | 只有您在【一般】的页面选择 静态最佳化 才可勾选是否要长度限制。勾选长度限制用途在于避免读取过长的连续地址而造成画面更新变慢。 |
| | 通讯中断 n 次后，取消连线 | 主要用途是避免当通讯异常发生，并 Retry 过后通讯异常对话框一直显示于人机界面的画面上。设置此值可在上述情况发生后就不再与控制器连线。范围从 1 ~ 255 次。当人机与控制器因通讯中断次数达到而停止通讯，此时，若要想恢复人机与此控制器通讯，可利用与命令区 D1（假设命令区的起始位置为 D0）的 Bit 0。在无通讯状态下，此 Bit 0 为 ON；当恢复通讯后，此 Bit 0 为 OFF。 |
| | 人机站号 | 设置人机站号。范围从 0 ~ 255。 |
| | PLC 默认站号 | 元件读写 PLC 地址的默认站号值，将采用此设置值。范围从 0 ~ 255。 |
| | 通讯界面 | 选定使用 RS232、RS422、RS485。所提供的通讯界面会依照您选的人机界面种类以及选择的通讯端口而提供不同的通讯界面选项。 |
| | 数据位元 | 选择 7 Bits 或是 8 Bits。 |
| | 停止位元 | 选择 1 Bits 或是 2 Bits。 |
| | 波特率 | 选择 4800、9600、19200、38400、57600 或是 115200。您也可以直接输入波特率，但输入的最大波特率不能超过 187500。 |
| 校验位 | 选择 None（无校验）、Odd（奇校验）或是 Even（偶校验）。 | |
| 以太网 网络设置 | <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;"> <p>人机</p> </div> <div style="flex: 2; padding-left: 10px;"> <ol style="list-style-type: none"> 选择「Ethernet」可以设置人机的以太网地址如上图。 启动画面写入 IP 地址功能，下载画面之后会顺便将人机上网络设置更新。 自动取得 IP 地址功能是由人机自动取得 IP 地址，此功能必须确认网络上有 DHCP 服务器。 </div> </div> | |

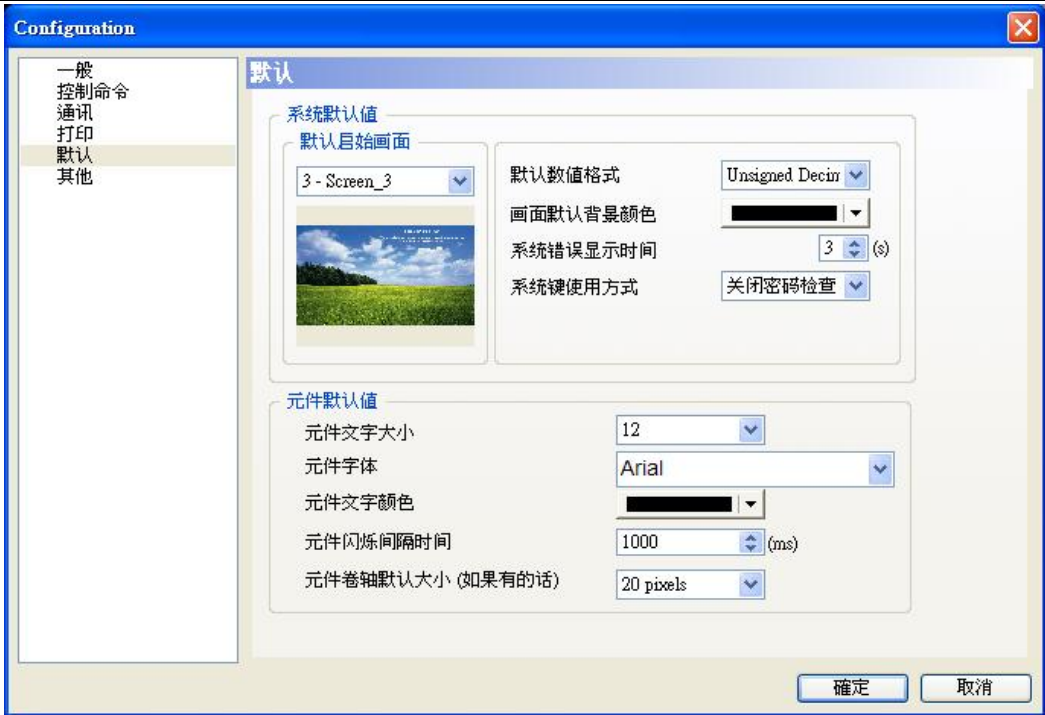
| 模块参数设置说明（通讯页面设置） | | |
|------------------|-----|--|
| 以太网网络设置 | 人机 | 当自动取得 IP 地址功能取消，用户必须手动设置人机网络，如 IP 地址，子网掩码，默认网关。 |
| | 控制器 |  <p>点选「Ethernet」底下的节点，如上图，每个控制器会有属于自己的通讯参数，如 IP 地址与通讯端口。</p> |

| 模块参数设置说明（打印页面设置） | | |
|---|--|--|
|  | | |

打印目前提供两种模式，一种是 Hard Copy，只是单纯将人机当下执行的画面撷取并打印出来，一次只能打印一页。另外一种则是排版打印，将所需要打印的画面透过打印排版设置好，除了打印画面外，还可将历史资料一并列出。详细打印设置方式，请见【打印排版管理】。


| | | |
|------|---|---|
| 基本设置 | 打印机 | 设置所使用的打印机。Screen Editor 2.0 提供树状界面以打印机的制造商作分类，方便用户选用。 |
| | 纸张 | 目前提供 A4, Letter 与 CUSTOM 三种纸张大小供用户选择。 |
| | 品质 | 目前仅只提供 72DPI。 |
| 边界设置 | 可以由边界设置于打印时保留部份区域为空白不作打印处理。您可以设置纸张的上、下、左、右的边界范围，单位为 mm。 | |


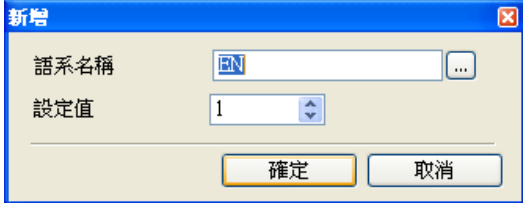
| 模块参数设置说明（打印页面设置） | |
|------------------|--|
| 打印大小 | 当用户纸张选择 CUSTOM，用户可以设置纸张打印范围，单位为 mm |
| 打印机界面 | 您可以指定打印机的传输端口，仅有 USB 可供选择。 |
| 自动换页 | 勾选后打印资料完成后打印机会强制退纸。无勾选，若纸张仍有空间可打印会将下次要打印的资料接续下去。 |
| 打印方向 | 可以选择要竖印还是横印。 |


| 模块参数设置说明（默认页面设置） | | |
|---|----------|---|
|  | | |
| 系统默认 | 默认起始画面 | 设置在人机端上电启动后第一个开启的画面。 |
| | 默认数值格式 | 建立元件时，默认的数值格式。 |
| | 画面默认背景颜色 | 新增一个编辑画面时，默认的画面背景颜色。 |
| | 系统错误显示时间 | 错误信息对话框显示停留时间，范围为 0 秒~5 秒。 请注意若值设为 0 秒时，若有系统错误发生，错误信息对话框是不会显示在屏幕上。 |
| | 系统键使用模式 | 设置当用户按下人机端的系统按键时需作的反应，系统提供 3 种选项，分别为 系统键无效 、 密码检查 、 关闭密码检查 。 |
| 元件默认 | 元件文字大小 | 建立元件时，默认的元件文字大小。 |
| | 元件字体 | 建立元件时，默认的元件字体。 |
| | 元件文字颜色 | 建立元件时，默认的元件文字颜色。 |
| | 元件闪烁间隔时间 | 建立元件时，默认的元件闪烁间隔时间。设置值为 500~5000ms。 |
| | 元件卷轴大小 | 用于元件有卷轴功能。例如：警报表等元件。默认卷轴大小 20 pixels，可调整的大小有 20、30、40 pixels |

模块参数设置说明（其它页面设置）



| | | |
|------|---|---|
| 一般设置 | 启动触碰鼠标 | 人机提供鼠标功能，当启动时人机画面被触碰的位置会出现鼠标图示。 |
| | 开机时显示初始画面 | 此选项可以决定人机开机时是否要显示出厂默认的开机画面。 |
| | 状态灯显示模式 |  <p>此选项功能可以设置人机上蓝色灯号要在何种状态显示，状态可以设置为「不显示」，「Com 通讯」，「磁盘存取」，「以太网路」。</p> |
| 屏幕保护 | 在【画面管理】的菜单里有介绍如何以拖曳的方式决定展示的屏幕保护画面，接这是如何启动屏幕保护的功能。 | |
| | 启动屏幕保护 | 请注意若有在【画面管理】的菜单里编辑屏幕保护画面，但未勾选此选项，则屏幕保护是不会被启动的。 |

| 模块参数设置说明（其它页面设置） | |
|------------------|--|
| 屏幕保护 | <p>读取控制器地址</p> <p>1. 有设置读取地址是由读取控制器地址是否为 0，来控制是否开始启动屏幕保护（为 0：关闭；非 0：启动）。用户触碰人机时也视为关闭屏幕保护。</p> <p>2. 无设置读取地址则是由人机界面里系统目录内的 Screen Saver Time（如下图所示）来判断启动屏幕保护。而当屏幕保护已开始启动时，用户触碰人机即为结束屏幕保护模式。若是人机中有设置 Screen Saver Time 但是用户并未设置屏幕保护画面，则当屏幕保护启动时，屏幕则会显示全黑的画面。</p>  |
| | <p>画面切换延迟时间</p> <p>屏幕保护启动时，彼此屏幕保护画面切换的间隔时间。范围从 1s ~ 255s。</p> |
| | <p>屏幕保护结束后</p> <p>1. 跳至原画面：切换至当启动屏幕保护时当时所在画面。</p> <p>2. 指定画面：指定于当屏幕保护结束后所需跳至的画面。</p> |
| 多国语言 | <p>新增</p> <p>按下新增按钮您可以增加一个语言。</p>  <p>如上图所示，您必须要输入设置值，此设置值会被在设置切换语言命令区里所参考到，范围从 0~255。设置语言名称可便于在编辑文字时的操作，您也可以按下 ... 决定书本图案颜色。</p> |
| | <p>启用/停用</p> <p>此选项主要用途是给用户能在编辑多国语言的画面时，虽然已输入各国语言内容，但希望下载至人机端时能规划该人机只支持哪几种语言。</p> |
| | <p>删除</p> <p>将已存在的语言移除。一个编辑项目至少要存在一个语言。</p> |
| | <p>修改</p> <p>修改已存在的语言名称与设置值。</p> |

| 模块参数设置说明（其它页面设置） | | |
|------------------|-----------|--|
| 多国语言 | 默认语言 | 设置默认语言。 |
| | 启动多语言编辑界面 | 提供页签控制的编辑界面。以编辑元件文字属性来看，其编辑界面将如下图所示：  |

3.11.2 警报设置

此项可由选项里面的警报设置来启动。警报设置功能需配合元件选项中警报显示功能使用，一旦设置好了后，在人机便会自动执行。每当符合警报设置条件的时候，人机便会自动出现警告的视窗。点击警报设置选项后会出现图 3-11-1 的对话框，里面有删除、修改、汇入、汇出、确定等选项。

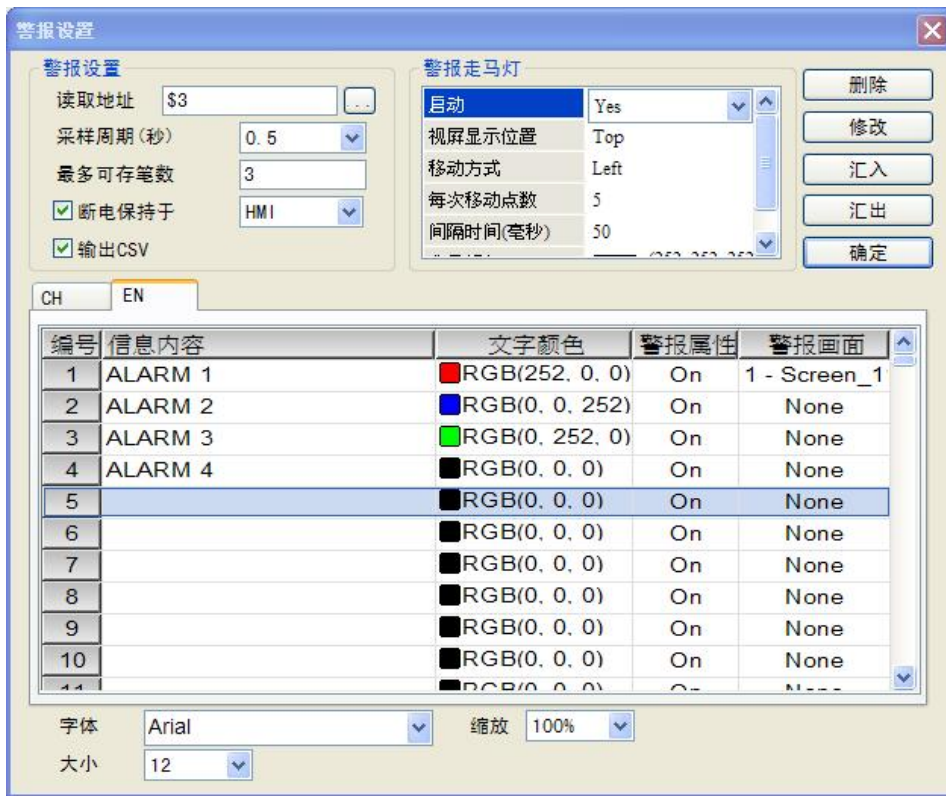
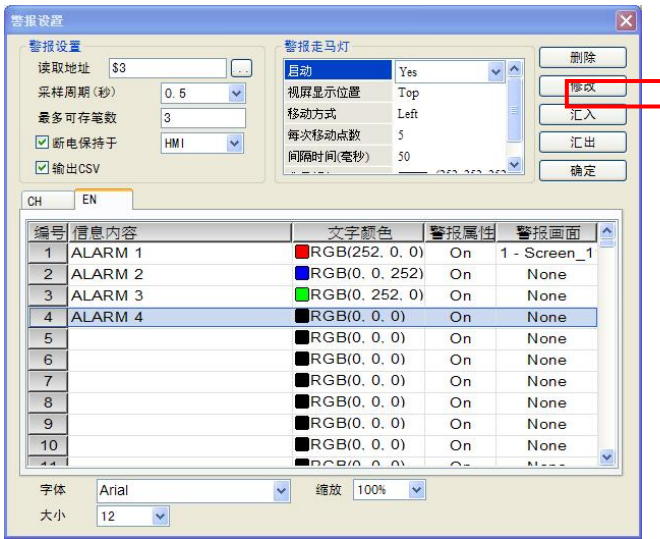



图 3-11-1 警报设置视窗

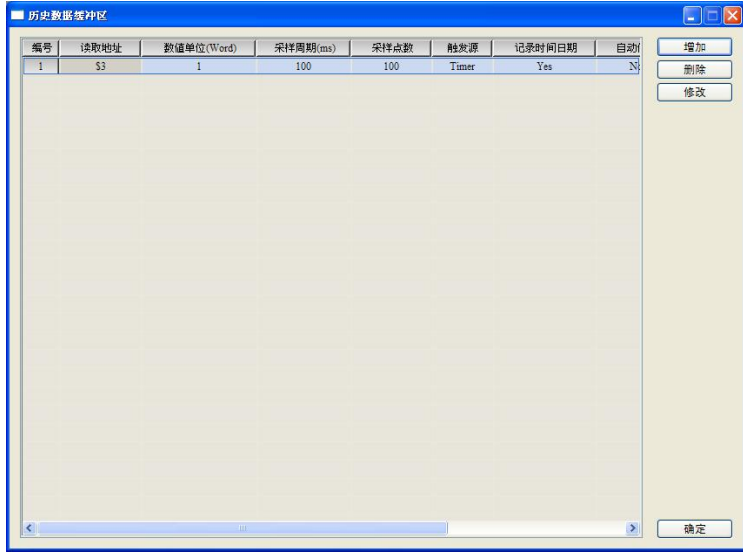

| 警报设置说明 | | |
|--------|------|--|
| 警报设置 | 读取地址 | 设置警报起始地址，共提供 512 笔（点）警报，32 Words。设置此读取位置须注意是以 word 为单位，以台达 PLC 控制器来说，若输入位置为 M、S 等元件种类，须为 16 的倍数方能合法正确的输入，如位置 M0、M16。倘若读取位置为 M1，系统即判定不合法。 |


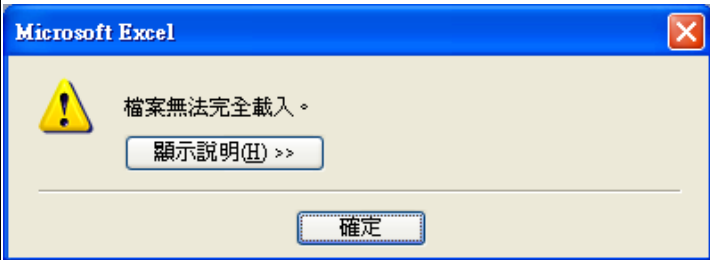
| 警报设置说明 | | | | |
|----------|---|------|---------------|------|
| 取样周期 (秒) | 设置特定时间内 (秒) 取样一次。 | | | |
| 最多可存笔数 | 依序储存数据, 当记录到所设置的取样点数后, 将会将第一笔资料覆盖掉从头开始记录。 | | | |
| 断电保持 | 选择如果失去电源时, 是否把资料记录于 HMI SRAM 中, B 系列机种可储存容量为 4 Kbytes、(断电时, HMI SRAM 的电源为电池)。(在某些人机界面机种里, 您可以设置断电保持区位于 U 盘或 SD 卡, 所能储存的容量就端看 U 盘或 SD 卡容量决定)。 | | | |
| 输出 CSV | 警报资料可以利用此功能输出成 CSV 文件格式。 | | | |
| 警报设置 | <p>您可以在警报内容列表里直接输入警报内容, 并按修改按钮如下图所示:</p> <p>(请注意: 警报内容也支持多语言的设置, 您可以点选页签控制项来切换您要输入的警报语言内容, 与设置字体内容)</p>  | | | |
| | <p>警报内容</p> <p>接下来会出现如下图的对话框:</p>  <p>警报内容不接受空白字元, 若未输入警报内容, 则在画面上看到的警报信息会是” ? ”。</p> <table border="1"> <tr> <td>信息内容</td> <td>当警报发生时要显示的信息。</td> </tr> <tr> <td>显示颜色</td> <td>当警报发生时信息要显示的颜色。</td> </tr> </table> | 信息内容 | 当警报发生时要显示的信息。 | 显示颜色 |
| 信息内容 | 当警报发生时要显示的信息。 | | | |
| 显示颜色 | 当警报发生时信息要显示的颜色。 | | | |

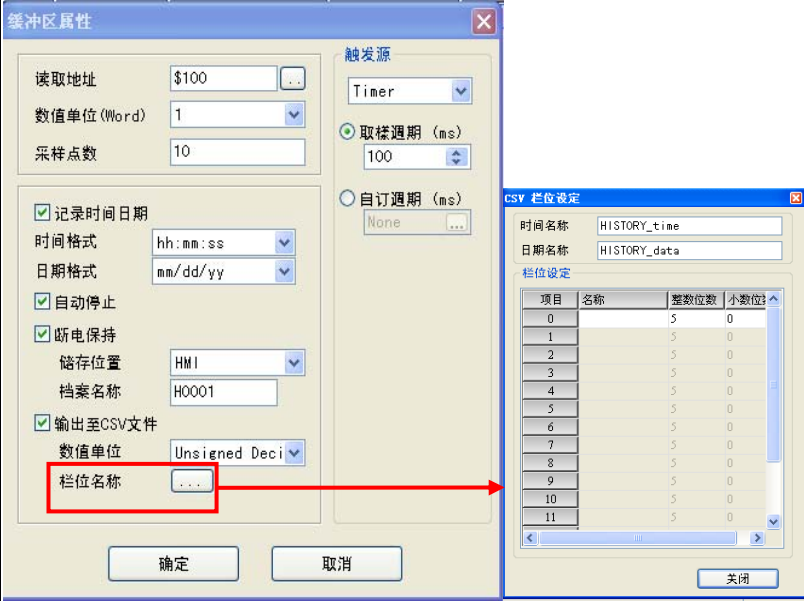
| 警报设置说明 | | | | | | | | | | | | |
|--|--|--|-------------------------------------|---|--------|--------|------|------|---|------|---|---------|
| | | 警报属性 | 设置判别地址 (Bit) 是 On 时发出警报或是 Off 发出警报。 | | | | | | | | | |
| 警报设置 | 警报内容 | 警报画面 | 当警报发生时指定显示画面，一般常配合警报显示用。 | | | | | | | | | |
| | 删除 | 将指定的警报内容资料清空。 | | | | | | | | | | |
| | 修改 | 在警报内容列表里用点击已输入的一列作警报内容的修改。 | | | | | | | | | | |
| | 汇入 | 将已存在的警报描述文件 (Alarm Describe File) 从文件汇入至警报列表里。 | | | | | | | | | | |
| | 汇出 | 将警报列表里的内容输入至警报描述文件。 | | | | | | | | | | |
| | 确定 | 结束警报的编辑。 | | | | | | | | | | |
| | 警报记录文件大小计算方式： $\{[6\text{Bytes}(a) + 2\text{Bytes}(b)] \times N(c)\} + 6\text{Bytes}(d) = \text{实际存档大小 Bytes}$ | | | | | | | | | | | |
| | <table border="1"> <tr> <td>a</td> <td>时间日期资料</td> </tr> <tr> <td>b</td> <td>警报资料</td> </tr> <tr> <td>c</td> <td>取样点数</td> </tr> <tr> <td>d</td> <td>数据文件文件名</td> </tr> </table> | | | | a | 时间日期资料 | b | 警报资料 | c | 取样点数 | d | 数据文件文件名 |
| | a | 时间日期资料 | | | | | | | | | | |
| | b | 警报资料 | | | | | | | | | | |
| c | 取样点数 | | | | | | | | | | | |
| d | 数据文件文件名 | | | | | | | | | | | |
| EX: 若【最多可存笔数】设为 100 笔，并记录时间日期，需要的容量空间为 $[(6\text{ Bytes} + 2\text{ Bytes}) \times 100] + 6\text{ Bytes} = 806\text{ Bytes}$ 若【最多可存笔数】设为 100 笔，没有记录时间日期，需要的容量空间为 $(2\text{ Bytes} \times 100) + 6\text{ Bytes} = 206\text{ Bytes}$ | | | | | | | | | | | | |
| 警报频次表大小计算方式： $2\text{Bytes}(a) \times N(b) = \text{实际存档大小 Bytes}$ | | | | | | | | | | | | |
| <table border="1"> <tr> <td>a</td> <td>警报频次资料</td> </tr> <tr> <td>b</td> <td>警报种类</td> </tr> </table> | | | | a | 警报频次资料 | b | 警报种类 | | | | | |
| a | 警报频次资料 | | | | | | | | | | | |
| b | 警报种类 | | | | | | | | | | | |
| EX: 若设置警报种类为五种，需要的容量空间为： $2\text{ Bytes} \times 5 = 10\text{ Bytes}$ | | | | | | | | | | | | |
| 警报走马灯 | 警报走跑灯是不论用户在人机那个页面操作，只要人机侦测到有警报发生，就会产生一行走马灯在屏幕上。显示的选项可由下列参数设置。 | | | | | | | | | | | |
| | 启动 | 决定是否启动警报走马灯。 | | | | | | | | | | |
| | 视屏显示位置 | 决定警报走马灯显示位置，可以是 Top(上)或 Bottom(下)。 | | | | | | | | | | |
| | 移动方式 | Left | 警报信息的移动将由右往左方向移动。 | | | | | | | | | |
| | | Right | 警报信息的移动将由左往右方向移动。 | | | | | | | | | |
| Up | | 警报信息的移动将由下往上方向移动。 | | | | | | | | | | |

| 警报设置说明 | | |
|--------|-----------|--|
| | | Down 警报信息的移动将由上往下方向移动。 |
| 警报走马灯 | 每次移动点数 | 警报信息移动点数，单位为 Pixel，范围为 1~50 点。 |
| | 间隔时间 (ms) | 警报信息每次移动时的间隔时间，单位为 ms，范围在 50 ms~3000 ms。 |
| | 背景颜色 | 设置警报走马灯的背景颜色。 |

3.11.3 历史缓冲区设置

| 历史缓冲区设置说明 | | |
|---|---|--|
|  | | |
| 增加 | <p>按下【增加】按钮会出现地址输入视窗来增加一笔历史资料缓冲区，最多只能增加到 12 组。如下图所示：而所设这 12 组则分别对应到命令区的历史取样标志或清除标志，以作为取样或清除历史缓冲区的依据。</p>  | |
| 历史缓冲区设置说明 | 读取地址 | 设置该历史缓冲区取样资料的起始地址。 |
| | 数值单位 | 设置要取样多少个 Word，可以设置取样 1~16 个连续 Word。 |
| | 取样周期 | 设置每隔多少时间读取地址一次。如果触发源是指定控制器，则取样动作是由命令区中的历史缓冲区取样标志控制。取样周期范围从 100ms 到 86400000 ms (一天)。 |

| 历史缓冲区设置说明 | |
|-----------|---|
| 取样点数 | 此功能需配合自动停止使用。如果有设自动停止，自动记录到所设置的取样点数后，便不会再储存数据。若未设置自动停止，当记录到所设置的取样点数后，将会将第一笔数据覆盖掉从头开始记录。所记录的文件（History.dat）需要使用人机软件中所附的 FlashTransfer 开启，而 FlashTransfer 则会自动将历史数据作排序。 |
| 记录时间日期 | 设置是否在取样过程中记录取样的时间日期。 |
| 自动停止 | 设置当取样达到数据额满时是否停止不记录。 |
| 断电保持 | <p>取样的资料是储存于 SRAM 中 B 系列机种可储存容量为 240 Kbytes。（在某些人机界面机种里，您可以设置断电保持区位于 U 盘或 SD 卡，所能储存的大小端看 U 盘或 SD 卡容量决定）。每个历史数据可以自行决定断电保持位置设置方式如下图-</p> <div style="text-align: center;">  </div> <p>断电保持可以设置断电保持文件名称。</p> |
| 输出至 CSV 档 | <p>将缓冲区所产生资料输出成 csv 文件，以方便用户使用 Windows® Excel 或是其他文字编辑软件对这些数据资料方便管理或进一步进行编辑。此外由于 Windows® Excel 最多此提供到 65,536 列及 256 栏的限制的数据浏览，所以若设置的取样点数超过 65536，使用 Windows® Excel 开启 csv 记录文件时，则会出现文件无法完全载入的信息，此时可通过其他文字编辑软件开启，如记事本等。</p> <div style="text-align: center;">  </div> |

| 历史缓冲区设置说明 | | |
|-----------|--|---|
| 历史缓冲区设置说明 | 输出至 CSV 文件 | <p>如果设自动停止，人机记录到所设置的取样点数后，便不会再储存数据。若未设置自动停止，当记录到所设置的取样点数后，将会将第一笔数据覆盖掉从头开始记录，输出的 csv 文件并不会依照时间先后作排序的动作。</p> <p>输出 CSV 文件，可以决定每个数值的栏位名称，设置方法如下，包括 Time, Date, 数值 0~15</p>  |
| | 触发源 | <p>选定 Timer 触发或是控制器触发。如果触发源是指定控制器，此选项将无法设置，取样动作是由命令区中的历史缓冲区取样标志控制</p> |
| | 确定/取消 | <p>储存/不储存设置并且离开。</p> |
| 删除 | <p>按下【删除】按钮来删除一笔历史数据缓冲区设置。</p> | |
| 历史缓冲区设置说明 | <p>历史缓冲区记录文件大小计算方式： 历史缓冲区记录文件分成两种格式一种为 dat，另一种为 csv 格式两种格式的计算公式如下：</p> <p>1. dat 每个历史数据会存成 Hxxxx.dat，代表第 xxxx 个历史数据记录。每个 dat 会使用底下的文件容量。</p> $\frac{[6\text{Bytes}(a) + 2\text{Bytes}(b)] \times N(c)}{1024 \times 1024} = \text{实际存档大小 MBytes}$ | |

| 历史缓冲区设置说明 | | | | | | | |
|---|---|------------|------------|---|------|---|------|
| 历史缓冲区设置说明 | <table border="1"> <tr> <td>a</td> <td>时间日期资料</td> </tr> <tr> <td>b</td> <td>数值单位</td> </tr> <tr> <td>c</td> <td>取样点数</td> </tr> </table> | a | 时间日期资料 | b | 数值单位 | c | 取样点数 |
| | a | 时间日期资料 | | | | | |
| | b | 数值单位 | | | | | |
| | c | 取样点数 | | | | | |
| | 额外历史数据会另外占用文件标记文件 $\{[8\text{Bytes}(a)] \times N(b)\} = \text{实际存档大小 Bytes}$ | | | | | | |
| | <table border="1"> <tr> <td>a</td> <td>每个历史数据文件标记</td> </tr> <tr> <td>b</td> <td>取样点数</td> </tr> </table> | a | 每个历史数据文件标记 | b | 取样点数 | | |
| | a | 每个历史数据文件标记 | | | | | |
| | b | 取样点数 | | | | | |
| | EX: | | | | | | |
| | 若新增两组历史缓冲区, 其中一组【数值单位】设为 1、【取样点数】设为 10000, 另一组历史缓冲区【数值单位】设为 2、【取样点数】设为 20000, 两组都记录时间与日期, 其所需要的 dat 文件容量空间为: | | | | | | |
| 第一组历史缓冲区 = $[(6\text{Bytes} + 2\text{Bytes}) \times 10000] = 80000\text{Bytes}$ 。 | | | | | | | |
| 第二组历史缓冲区 = $[(6\text{Bytes} + 2 \times 2\text{Bytes}) \times 20000] = 200000\text{Bytes}$ 。 | | | | | | | |
| 文件标记容量 $8\text{Bytes} \times 2 = 16 \text{ Bytes}$ | | | | | | | |
| 总共需要: $80000\text{Bytes} + 200000\text{Bytes} + 16\text{Bytes} = 280016\text{Bytes} = \text{约 } 0.268\text{MBytes}$ 。 | | | | | | | |
| 2. csv | | | | | | | |
| $\frac{N(a) \times 240\text{Bytes}}{1024 \times 1024} = \text{实际存档大小 MBytes}$ | | | | | | | |
| <table border="1"> <tr> <td>a</td> <td>取样点数</td> </tr> </table> | a | 取样点数 | | | | | |
| a | 取样点数 | | | | | | |
| EX: | | | | | | | |
| 若新增两组历史缓冲区, 其中一组【数值单位】设为 1、【取样点数】设为 10000, 另一组历史缓冲区【数值单位】设为 2、【取样点数】设为 20000, 两组都记录时间与日期, 其所需要的 csv 文件容量空间为: | | | | | | | |
| 第一组历史缓冲区 = $10000 \times 240\text{Bytes} = 2400000\text{Bytes}$ 。 | | | | | | | |
| 第二组历史缓冲区 = $20000 \times 240\text{Bytes} = 4800000\text{Bytes}$ 。 | | | | | | | |
| 总共需要: $2400000\text{Bytes} + 4800000\text{Bytes} = 7200000\text{Bytes} = \text{约 } 6.8\text{MBytes}$ 。 | | | | | | | |

3.11.4 密码表设置

人机界面提供 7 个权限, 分别为权限 1 到 7。如下方密码表所示

权限 0 为无权限, 无保护功能, 任何人均可操作。

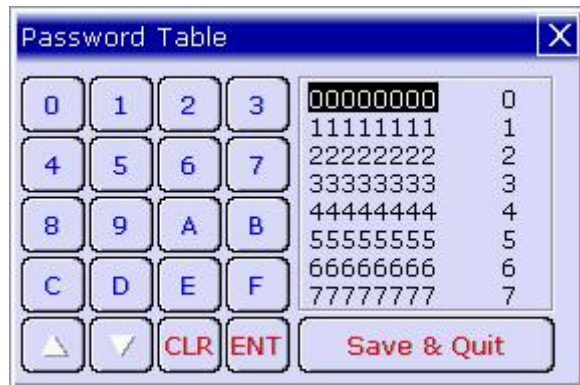
权限 1 等级比较低, 需输入相对应的密码或权限等级较高, 方能进行操作。

权限 4 等级居中, 需输入相对应的密码或权限等级较高, 方能进行操作。

权限 7 等级比较高, 需输入相对应的密码或最更等级密码, 方能进行操作。

最高权限密码是权限等级最高密码，凌驾于权限 1 到权限 7 密码，同时此密码也是项目储存后的保护密码。

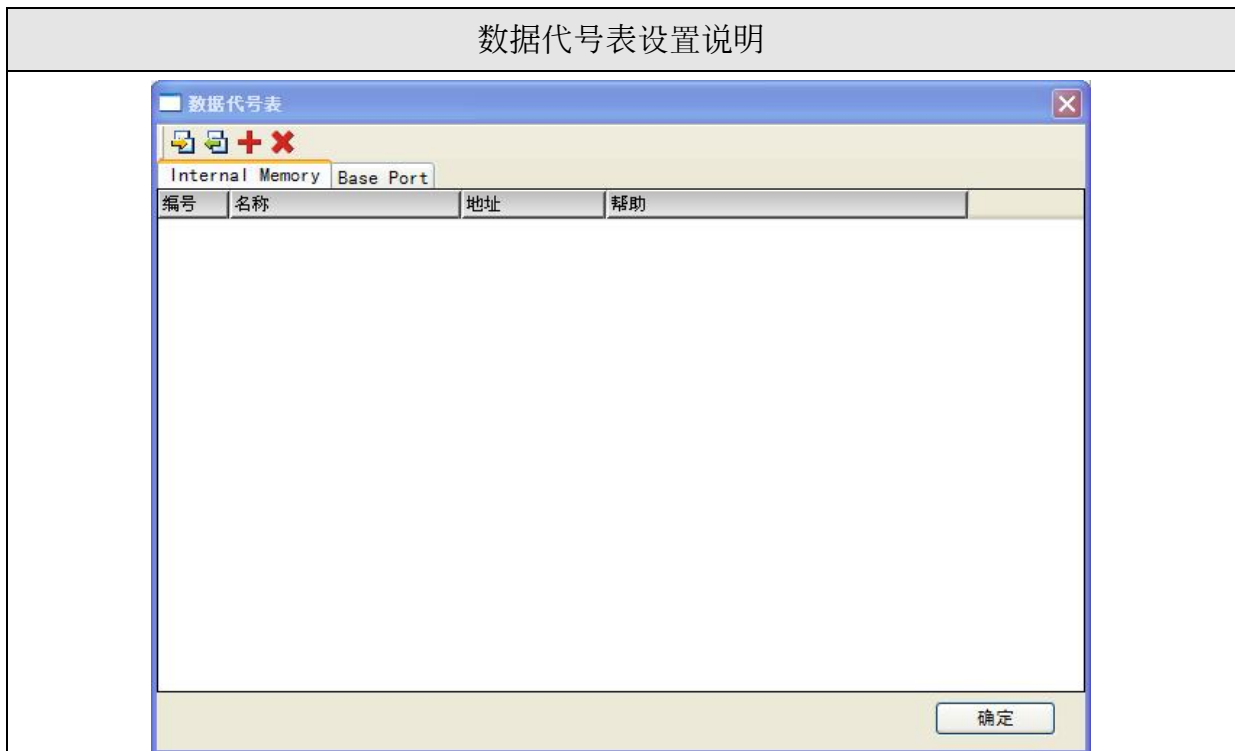
密码表的设置可由软件上去定义，也可以由在人机上制作一个【设置密码表】按钮来变更。





上述密码最多可输入八位数由 0~F 英文字元不分大小写所组成，用户可弹性变化密码位数或密码字元。

3.11.5 数据代号表

将特定的地址以自行设置的代号来代替。例如 PLC 的地址 1@Y0 选用 OS 代替，那么在以后需要输入 1@Y0 的地方，只要输入 OS 即可。



| | |
|-----|--|
| 打开 | 按下 来将已存在磁盘里的数据代号文件 (Tag File) 汇入数据代号表里。 |
| 另存为 | 按下 来将数据代号表里所定义的数据与资料储存到磁盘里。 |

| 数据代号表设置说明 | | | | | | | | | | | | | |
|-----------------|--|-----------------|----|-----------|--|----|----|----|----|---|----|------|--|
| 新增 | <p>请先选择您要新增数据代码的型式标签，有内部存储器 (Internal Memory) 与 Base Port (若您有指定使用 3 个以上的通讯端口，您会在看到 Link2、Link3 ... 的标签) 按下 ，您就可以如下图般开始定义您的数据资料：</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">Internal Memory</th> <th colspan="2">Base Port</th> </tr> <tr> <th>編號</th> <th>名稱</th> <th>位址</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>OS</td> <td>0@Y0</td> <td></td> </tr> </tbody> </table> | Internal Memory | | Base Port | | 編號 | 名稱 | 位址 | 說明 | 1 | OS | 0@Y0 | |
| Internal Memory | | Base Port | | | | | | | | | | | |
| 編號 | 名稱 | 位址 | 說明 | | | | | | | | | | |
| 1 | OS | 0@Y0 | | | | | | | | | | | |
| 删除 | 点选数据代号列表里其中一列，按下  来移除定义的数据资料。 | | | | | | | | | | | | |
| 确定 | 按下【确定】按钮来储存输入的数据资料。 | | | | | | | | | | | | |

3.11.6 打印排版管理

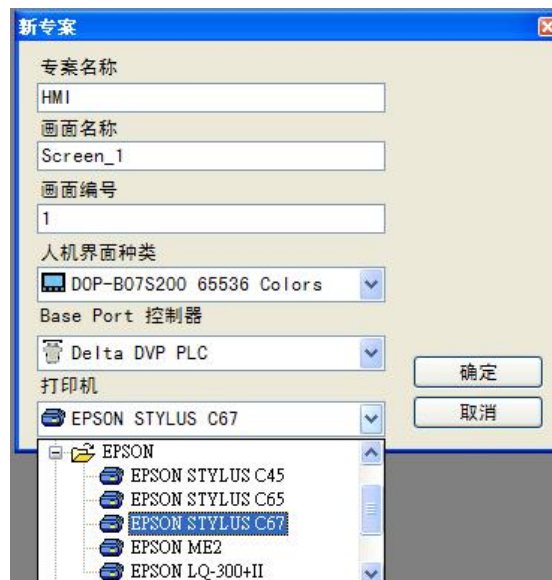
提供制作有效的打印编排管理，请参阅以下说明的范例：

打印功能

■ 打印机设置

1. B 系列只支持 USB 打印机打印功能
2. 选择打印机

可以在【建立新专案】或【设置模块参数】选择打印机



3. 打印机版面配置

选择功能表【选项】>>【设置模块参数】，点击【打印】标签，可以设置打印机的打印参数，如纸张大小，品质，边界等各项设置。



排版打印

排版打印一次可以打印多页，并可以透过【打印排版管理】规划欲打印的版面，并可打印历史数据。但有几点是需要注意的：

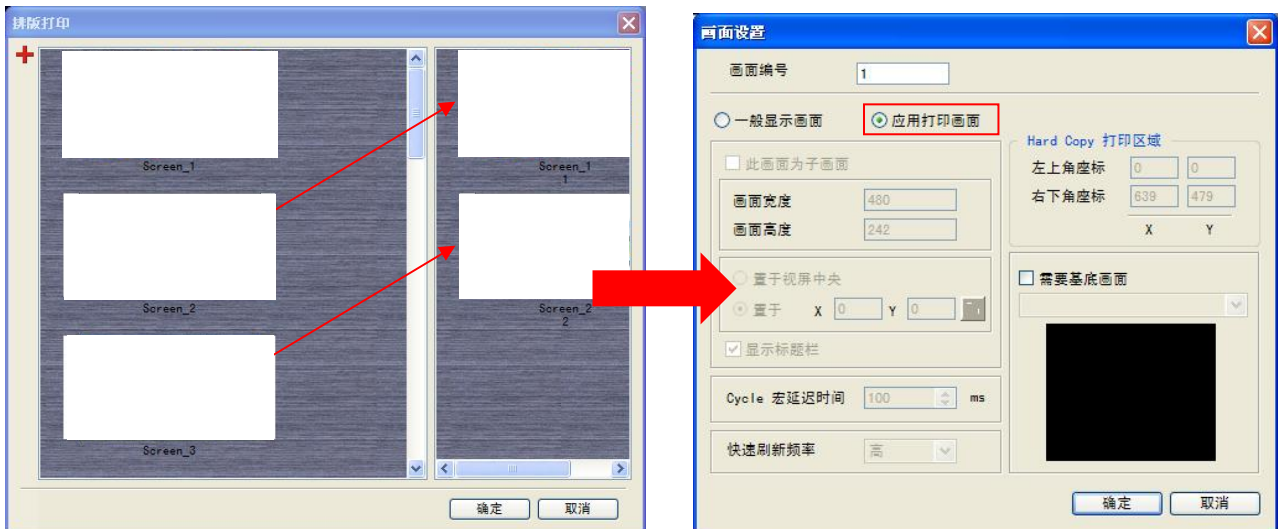
1. 换画面不能指定到排版打印画面。

2. 排版打印画面不能为默认画面。
3. 排版打印画面不能为基底画面。
4. 排版打印画面不能为屏幕保护画面。
5. 排版打印画面不能为子画面。

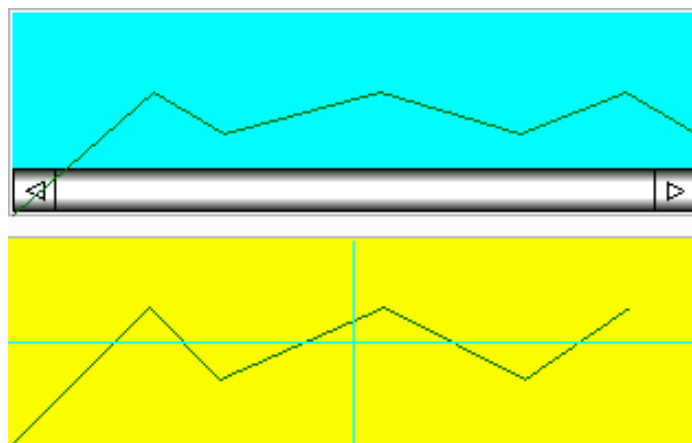
■ 建立打印报表画面

打印报表画面编辑

Step 1. 开启一个新专案，新增一个画面后，执行【选项】>>【打印排版管理】，以拖曳的方式决定哪一页面需作排版打印，左边为被选择画面，右边为选中的画面，如此画面属性会自动变成打印画面。

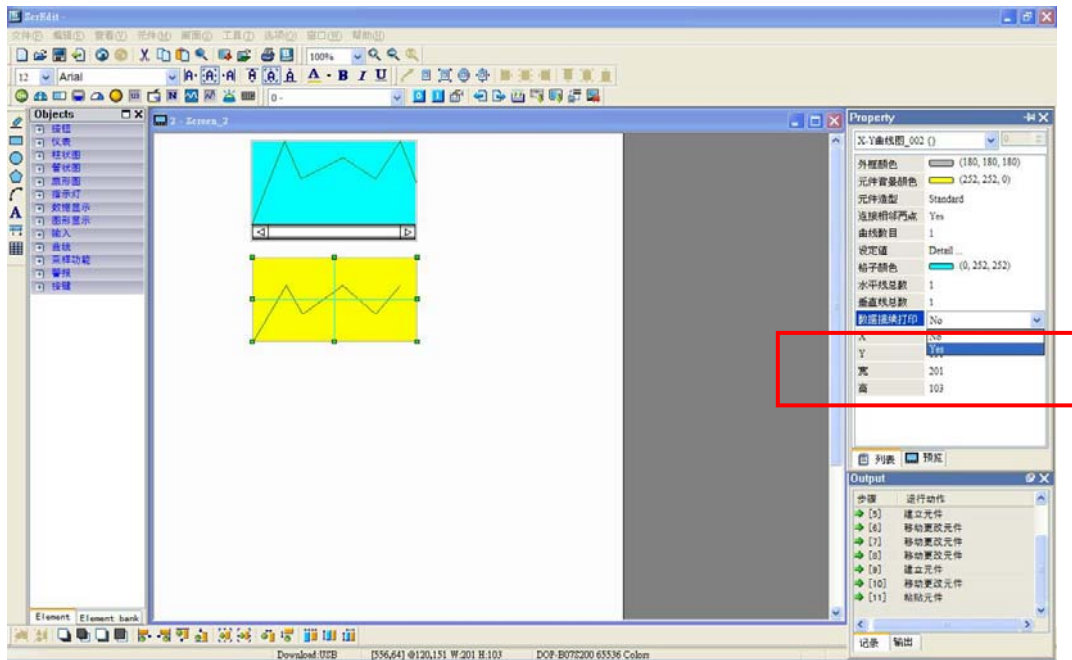


Step 2. 建立打印元件，依照画面编辑动作，建立各种打印元件，例如要打印历史曲线与 X-Y 曲线，就在打印画面建立一个历史曲线图与 X-Y 曲线图，并设置属性，相关设置请参考各元件属性设置。



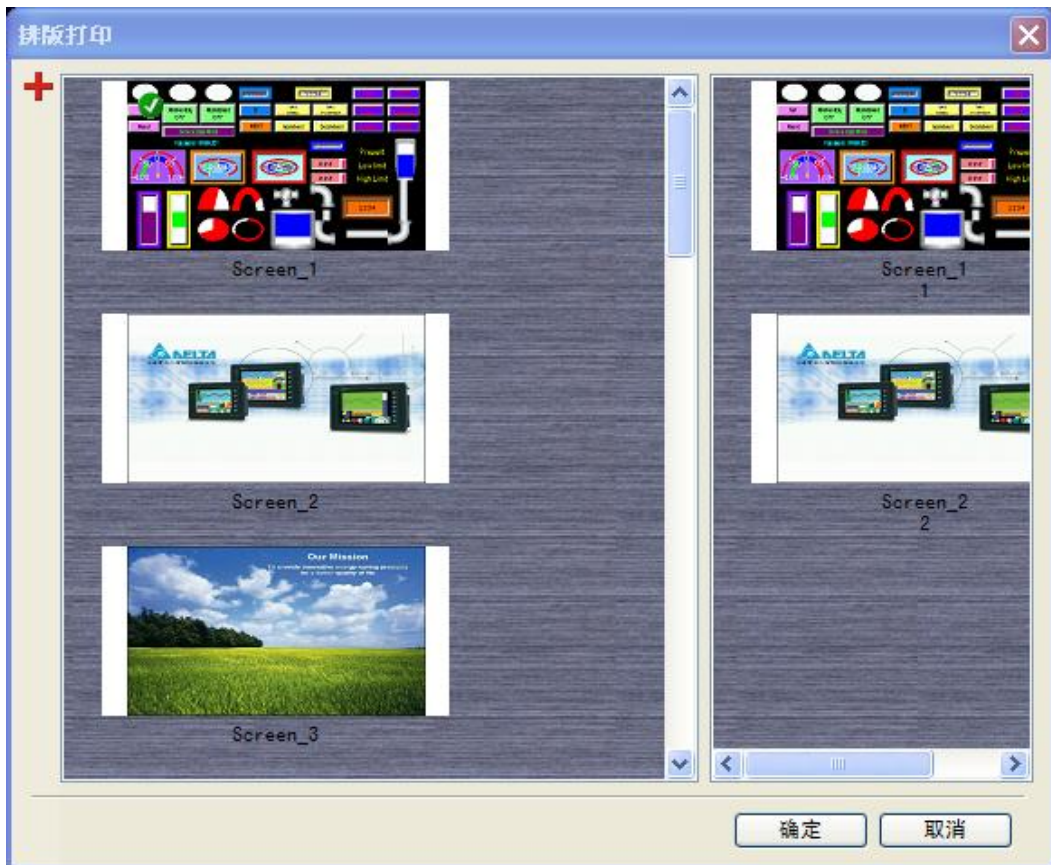
Step 3. 决定打印元件是否要【资料接续打印】，连续打印的意义在于当打印完此页后，但若此画面上所编辑元件所记录的资料并未全部采样完成，可利用此设

置，将剩余资料接续打印完。选择打印元件，并且将【资料接续打印】的属性设为【Yes】。



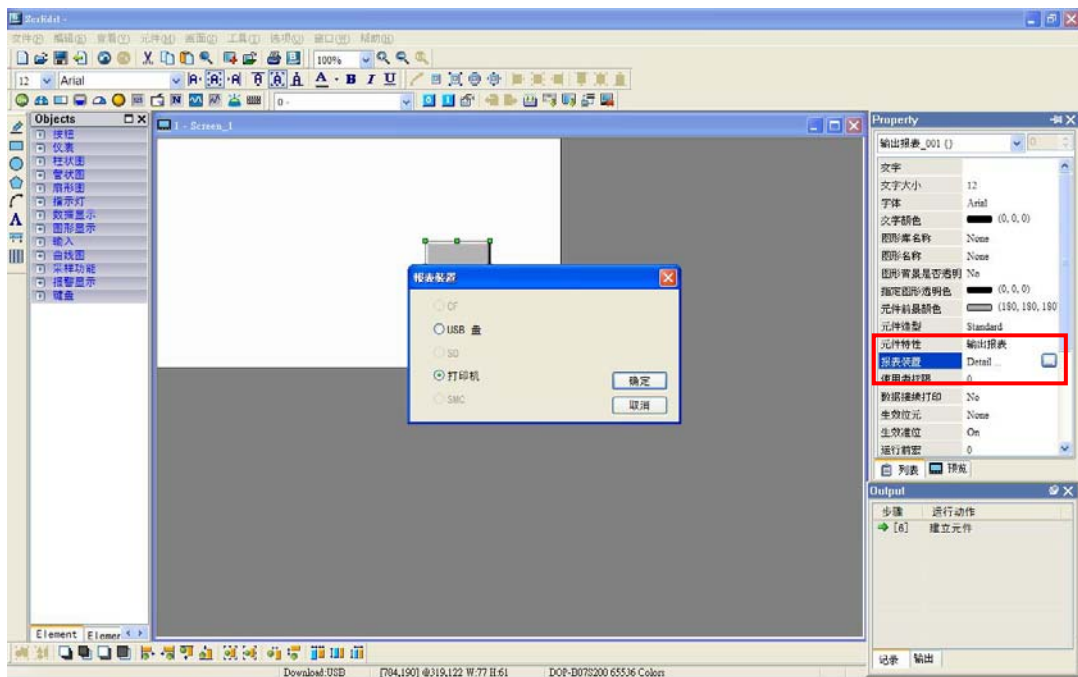
■ 打印画面编排与输出

Step 1. 执行【选项】>>【打印排版管理】，以拖曳的方式决定哪一页面需作排版打印，左边为被选择画面，右边为选中的画面，注意若是拖曳一般显示画面进入排版打印，此画面属性会自动变成打印画面。



Step 2. 在一般显示画面建立【输出报表】按钮，以便于触发启动打印功能。

Step 3. 设置输出报表按钮的【报表装置】属性，可分为U盘、Printer，注意若是选择U盘只会将资料输出到U盘，而不会执行打印动作。

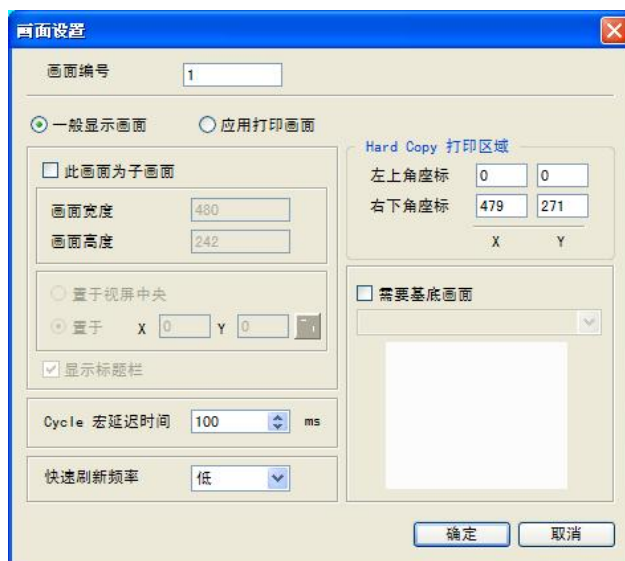


Step 4. 接上打印机，启动人机，按下输出报表按键，即会开始打印。当打印画面较多，或是设置打印的元件取样资料较多时，人机处理时间都会相对较久。

Hard Copy

此功能设置必须为一般显示画面，不能为打印排版（套用打印）画面，若系统发现所编辑的资料有排版打印，会以【排版打印】优先处理，Hard Copy 功能将失效。

Step 1. 在欲 Hard Copy 的画面属性中，设置 Hard Copy 的打印区域。



Step 2. 开启打印功能

在一般显示画面建立【输出报表】按钮，以便于触发启动打印功能，如同排版打印一样。

Step 3. 接上打印机，启动人机，按下输出报表按键，即会开始打印。


3.11.7 配方

配方的功能，最主要是提供用户有大批数值参数的需求而存放在人机内部存储器区。例如：面包业者对于每个不同款式的面包，需要有不同烘焙时间，而这些众多时间的变量就可以利用人机所提供的配方功能来达到时间变量的要求。其目的是为了减少控制器的负担，使控制器的寄存器有更大弹性的运行空间。


配方功能提供搭配的系统一套便捷的参数输入的方式。当用户完成配方编辑时，利用系统控制区的配方选定参数，就可将指定对应的参数传递至所搭配系统的存储器位置。利用配方编辑的对话框（图 3-11-2），可以设置或是修改配方内容，并且可独立存取不用跟着专案一同存取，除了台达人机自订的配方文件格式（rcp）也可将配方储存成 Windows® Excel 支持的 csv 文件。详细配方设置方法及配方编辑对话框的使用方法详细解说于后续的配方编辑项目中。




图 3-11-2 配方设置对话框

| 配方编辑说明 | |
|--------|--|
| 启动配方功能 | 如果要使用配方功能，一定要勾选此功能。 |
| 断电保持区 | 配方会有自己的断电保持区，必须启动配方功能，配方的断电保持区设置才可以使用。 |
| 地址 | 您可以在此处直接输入配方的起始地址（配方地址若选择 D4095（含）之前的地址，注意配方地址加上 配方长度不可超过 D4095），此处支持控制器地址的输入格式与内部存储器地址的输入格式。您也可以按下  开启地址输入对话框，如下图所示： |

配方编辑说明

| | |
|-----------|--|
| <p>地址</p> |  |
| <p>长度</p> | <p>此处设置配方的长度为多少，单位为 Word。其值不可设置为 0，如果设 0 会出现警告，如下图所示：</p>  |
| <p>组数</p> | <p>配方输入可以设多组的配方，用来切换，以方便用户使用。其组数不可设置为 0，如果设 0 会出现警告，如下图所示：</p>  |
| <p>输入</p> | <p>设置完长度与组数后，点选此输入按钮便可以开始编辑配方数据。由于受限于人机内部存储器的大小，所以配方的数量不可以超过 64K 个（长度 × 组数不可大于 65536）。如果配方的数量超过 64K 便会出现如下图的警告：</p>  |

| 配方编辑说明 | |
|--------|--|
| 输入 | 上述的人机内部存储器是指人机端的断电保持区 SRAM，其中规划 64K 给配方使用的。而部份人机界面机型有支持 USB Host 的功能，即支持随插即用的 U 盘，因而您可以使用较多配方，但不可超过 4096K 个配方（长度 × 组数不可大于 4194304）。如何变更配方储存的位置，也就是改变断电保持区的位置。 |
| 开启 | 点选此项后，可以开启储存于磁盘里的配方文件，来载入编辑。而载入的配方文件并不包含起配方的起始地址，因此用户能够将相同的配方文件案使用于不同的厂商的控制器。开启的配方文件的格式有支持 Windows® Excel CSV 以及 RCP。特别要注意的是当使用开启配方功能时，要注意所以选择的配方文件是 16 位元的配方文件还是 32 位元的配方文件，如果选择错误，会造成程序的不正常动作。 |
| 储存 | 将所编辑的配方数据以文件的方式储存起来，储存时并不会记录配方的起始地址。这是为了方便用户能够将相同的配方文件案使用于不同的厂商的控制器。储存配方文件的格式有支持 Windows® Excel CSV 以及 RCP |
| 重置 | 删除所有的配方设置与输入的配方数据。删除后，除了从储存的配方文件中汇入外，没有其他方法可以回覆设置。 |
| 清除 | <p>将所有输入的配方数据值设置为 0，如下图所示：</p>  |
| 打印 | 将所有输入的配方数据值输出至打印机。 |
| 确定 | 当配方数据输入完毕后，当您点选确定按钮时配方编辑对话框会依序检查所有输入的合法性，若有不合法的输入值是无法正常结束并离开此对话框的，请参照下图，因没有输入配方的起始地址，所检查到的错误： |

配方编辑说明

配方设置

启动配方功能 配方断电保持区: HMI

地址: 长度: 4 组数: 4

| | W1 | W2 | W3 | W4 |
|---|----|----|----|----|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |

ScrEdit

❌ 存储器地址输入失败

确定

取消 不管是否编辑完毕或是有无储存，点选后一律强迫离开。如果再次开启也一样不会留下任何资料，因此点选前，要确定是否真的要取消才点选。

3.11.8 32 位配方

32 位配方则可储存 32 位的资料，数据格式包括 Signed Decimal、Unsigned Decimal、以及 Floating（整数与小数位数总和不可超过 7 个）。

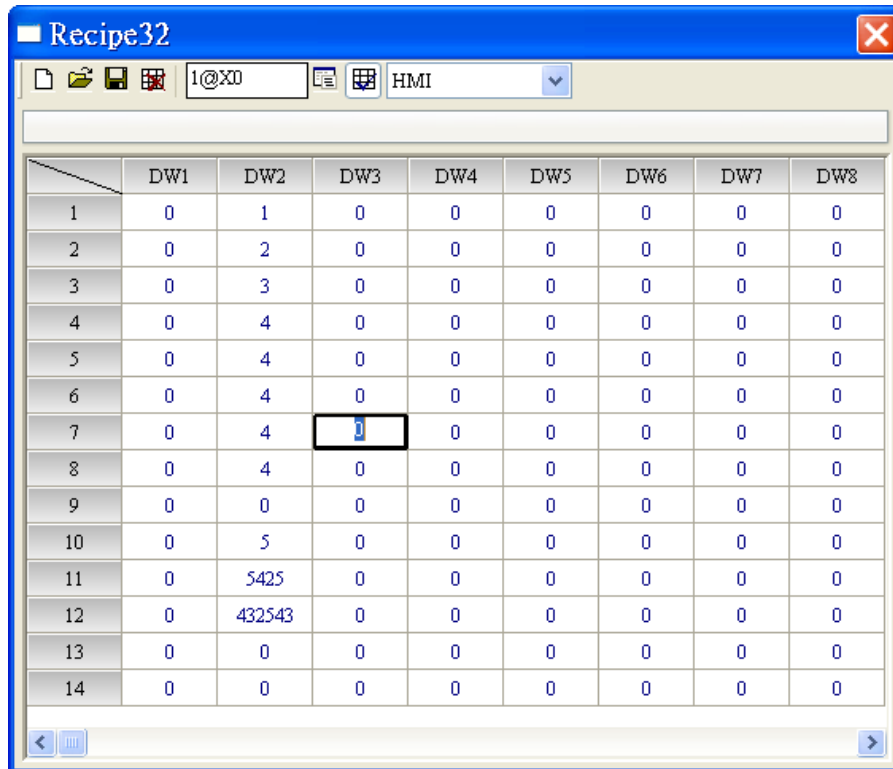





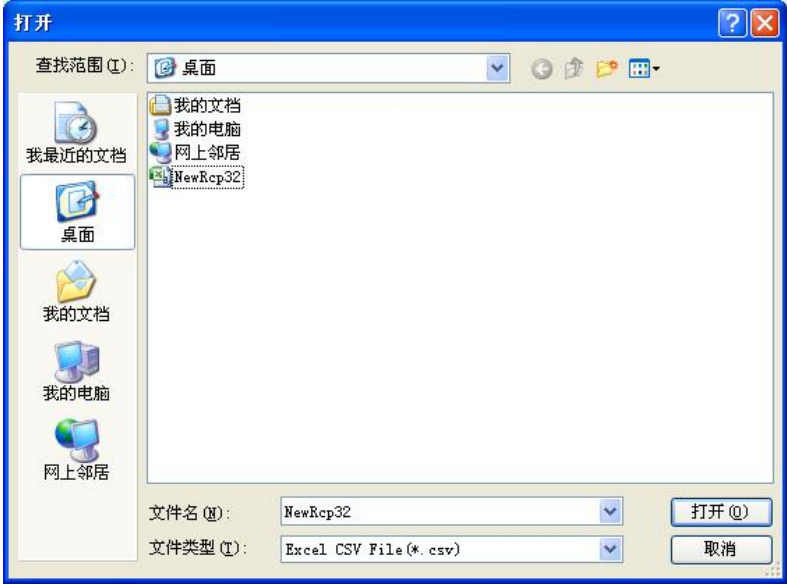


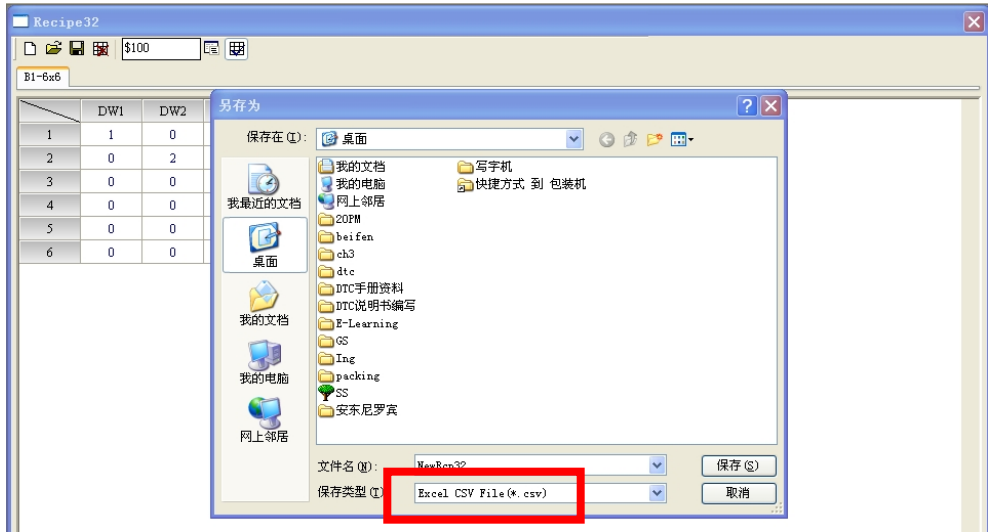


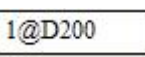




图 3-11-3 32 位配方设置对话框

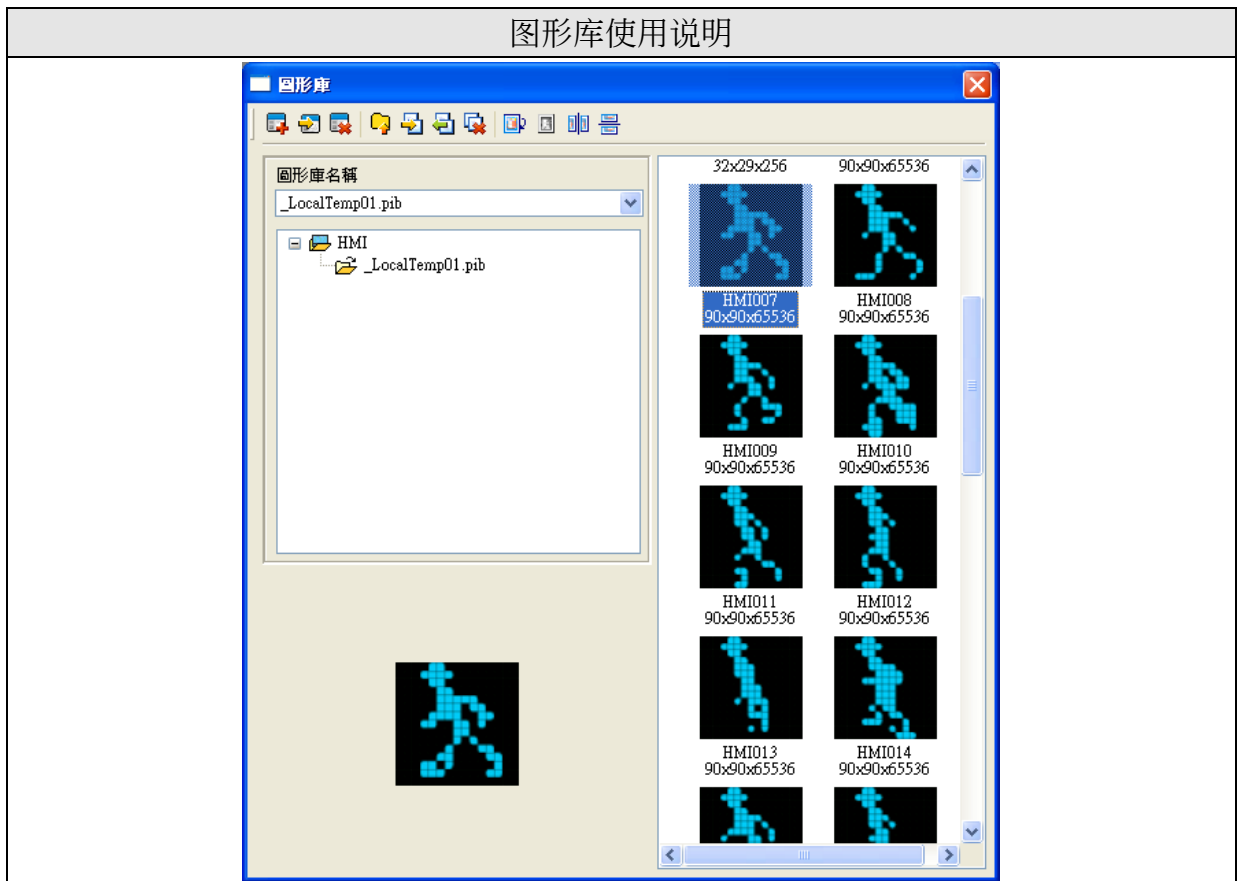
| 配方编辑说明 | |
|---|---|
|  | <p>新增配方组合：</p> <p>按下  新增一个配方组合，点选后会出现设置视窗，利用此视窗设置配方组合。可设置配方储存地址（配方地址若选择 D4095（含）之前的地址，注意配方地址加上 配方长度不可超过 D4095），配方长度、组数，数值格式，整数位数，小数位数。数值格式可设置三种格式，包括 Unsigned Decimal、Signed Decimal、Floating，而在 Floating 的部分，软件提供可设置整数及小数位数，整数位数加上小数位数不能超过 7 位数。配方组合最多可增加到 255 组。</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid blue; padding: 5px;"> <p>32 Bits配方设置</p> <p>地址: 16D400</p> <p>长度: 1</p> <p>组数: 1</p> <p>数值格式: Unsigned Dec</p> <p>整数位数: 3</p> <p>小数位数: 2</p> <p>确定 取消</p> </div> <div style="border: 1px solid blue; padding: 5px;"> <p>32 Bits配方设置</p> <p>地址: 16D400</p> <p>长度: 1</p> <p>组数: 1</p> <p>数值格式: Signed Decima</p> <p>整数位数: 3</p> <p>小数位数: 2</p> <p>确定 取消</p> </div> </div> |
| | <div style="border: 1px solid blue; padding: 5px; width: fit-content; margin: auto;"> <p>32 Bits配方设置</p> <p>地址: 16D400</p> <p>长度: 1</p> <p>组数: 1</p> <p>数值格式: Floating</p> <p>整数位数: 3</p> <p>小数位数: 2</p> <p>确定 取消</p> </div> |
|  | <p>开启配方文件：</p> <p>按下  可开启配方文件，32 位元配方文件格式为 Windows® Excel CSV 格式或是 rcp 格式。特别要注意的是当使用开启配方功能时，要注意所以选择的配方文件是 16 位元的配方文件还是 32 位元的配方文件，如果选择错误，会造成程序的不正常动作。</p> |


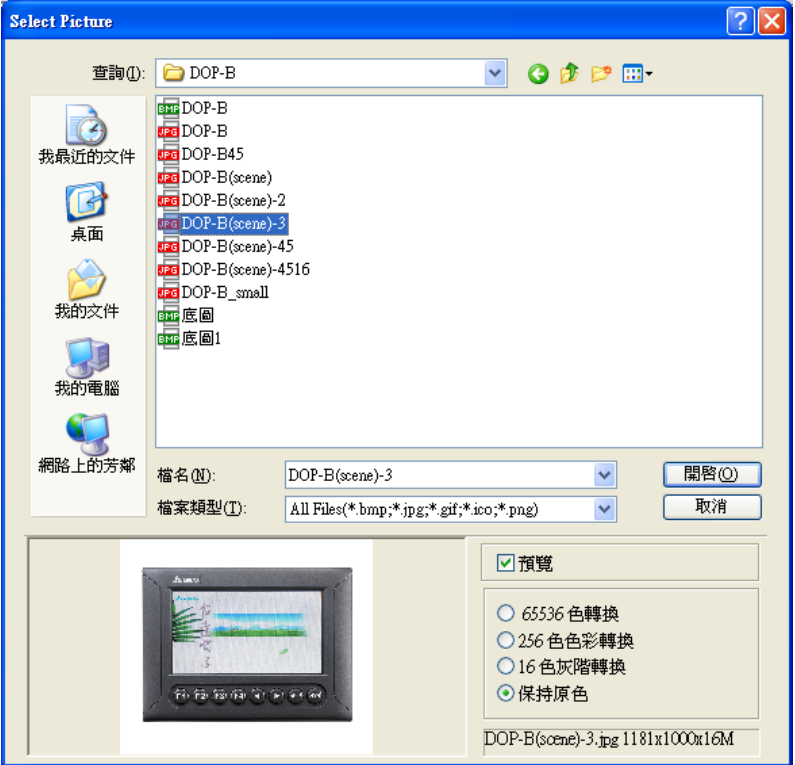

| 配方编辑说明 | |
|---|--|
|  |  |
|  | <p>储存配方文件： 按下  可储存配方文件，32 位元配方文件格式为 Windows® Excel CSV 格式。</p>  |
|  | <p>删除配方组合： 按下  可删除不需要的配方组合。</p> |
|  | <p>配方储存地址： 显示配方储存地址。</p> |
|  | <p>配方设置： 可变更配方设置</p> |
|  | <p>启动 32 元配方功能： 若要使用 32 元配方功能，此选项一定要点选。 32 元配方也提供断电保持区选择，断电保持区的位置与 16 位元配方相同。</p> |



3.11.9 图形库

用户可利用图形库里的功能汇入不同的图形以丰富图形库的内容。

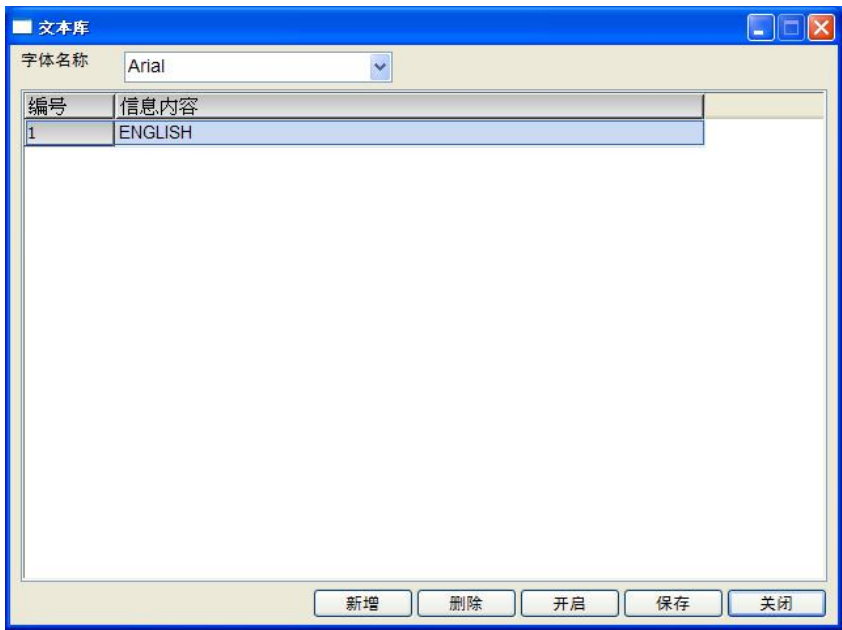


| 图形库使用说明 | |
|---|--|
| <p>点选图形库列表会重新显示图形库内容于图形列表，点选图形列表中的某一图形，会在预览视窗中看到图形样貌。在点选图形上鼠标左键按两下，可浏览原尺寸的图形。</p> | |
| 新增图形库 | <p>新增空白的图形库，点选后会出现如下图所示：</p>  |
| 开启安装图形库 | 从磁盘里汇入已存在的图形库。 |
| 移除图形库 | 移除您选择的图形库，被移除的图形库会置于回收站里。 |
| 将图形文件汇入至图库 | <p>从磁盘里将图形文件汇入到指定的图库里： 支持汇入的图形格式有 BMP、JPG、GIF、ICON 等，其中 GIF 图文件若是动画图文件会自动分解所有的动画，并依文件名依序编号加入图形库。其中我们在汇入图形的对话框里如上图所示，您可以指定汇入图形库的图形先行作颜色的转换，以方便加快编译时间。</p>  |
| 更新图库内容 | 每次编辑（诸如：色彩转换、水平镜向…）完图形必须更新图库内容，新的图形内容才会被储存至图形库里。 |
| 将图形库内容汇出至档案 | 将图形以 BMP 格式输出至磁盘文件。 |
| 色彩转换 | <p>色彩转换，负片效果。</p>  |

| 图形库使用说明 | |
|------------|---|
| 转成 256 色灰阶 | 将彩色图形转成 256 色灰阶。  |
| 水平镜向 | 将图形作左右镜射。  |
| 垂直镜向 | 将图形作上下镜射。  |
| 快显功能表 | <p>快显功能表将常用的图形工具放在功能表上，可以让用户更快速管理图形库的内容。在图形列表上按鼠标右键可叫出快速功能表，如下图所示：</p> <div style="border: 1px solid gray; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> <p>剪切 复制 粘贴 删除 更改名称</p> <hr/> <p>汇入... 汇出...</p> <hr/> <p>色彩对换 水平镜像 垂直镜像 转成256灰度</p> </div> |

3.11.10 文本库

将常用到的词句放入此文本库中，如此当元件上需要输入字串的地方，便可以直接从文本库中汇入先行编辑的字串。而不需要再重新键入。

| 文本库使用说明 | |
|--|--|
|  | |
| 新增 | 按下【新增】按钮来增加文本库的输入。文本库支持多国语言的编辑，您可以在不同的语言里输入不同的该国语意用词，同时设置该编辑语言的字体。 |

| 文本库使用说明 | |
|---------|----------------------------|
| 删除 | 按下【删除】按钮来将已输入的文本库内容移除。 |
| 开启 | 按下【开启】按钮从磁盘里汇入编辑的内容。 |
| 储存 | 按下【储存】按钮将输入的文本库列表输出至磁盘文件里。 |
| 关闭 | 离开并关闭文本库对话框。 |

3.11.11 子宏

子宏的介绍与宏的编写请参考 3.14 节。

3.11.12 Initial 宏

Initial 宏的介绍与宏的编写请参考 3.14 节。

3.11.13 Background 宏

Background 宏的介绍与宏的编写请参考 3.14 节。

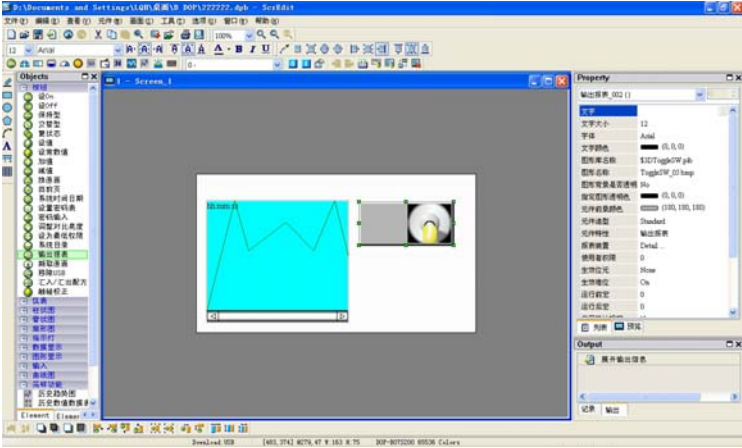
3.11.14 Clock 宏

Clock 宏的介绍与宏的编写请参考 3.14 节。

3.11.15 环境设置

设置 Screen Editor 2.00.18 操作环境：



| 环境设置说明 | | |
|----------------|---|---|
| 系统路径 | 执行文件所存放的路径，包含一些系统参考的资料与动态连结文件 (*.DLL) 会一并存放在这路径底下。此项目无法更改。 | |
| 输出路径 | 画面编译后的文件输出路径。一些功能诸如：在线模拟、离线模拟、文件上载、文件下载等等都会参考此路径下的资料。因此如果没有必要或是版本上没有更改，请不要修改此一路径，以免程序执行错误或是找不到文件。 | |
| 选项 | 工具列/视窗 | 设置 Screen Editor 2.0 视窗上的工具列使否要使用。 |
| | 语言 | 可以选择 繁体中文 简体中文 土耳其文 以及 英文 等四种语言环境。 |
| | 上/下载设置 | 决定程序上下载的通讯界面，可以是 USB 以太网或是 COM 端口。 |
| | 自动存档 | 设置于固定时间内 Screen Editor 2.0 会自动存档，单位为分钟 (M)。范围从 0M~120M。 |
| | 绘图区置中 | 画面编辑窗会以置中的方式提供给用户编辑，如下图：  |
| 上载包含图形资料 | 启动 Screen Editor 2.0 上载功能时，是否要包含图形资料。上载过后的图形资料会被汇总至一个取名为_LOCALTEMP01.PIB 的文件里，所有编辑元件有参考到图形资料者其【图形库名称】与【图形名称】两者属性皆会参考到此文件。当程序不正常结束时下次所上载建立的图库资料名称将会以后两数字码递增的方式来取名：_LOCALTEMP02.PIB、_LOCALTEMP03.PIB ...。 | |
| 自动将输入地址转换为数据代码 | 以数据代码表的例子来说当您输入 1@Y0 时，程序会自动将您输入的地址转换为 OS。 | |
| 驱动程序 | 重新安装人机 USB 驱动程序/解除安装人机 USB 驱动程序。 | |
| 确定 | 储存更改的设置并关闭对话框。 | |
| 取消 | 直接关闭对话框离开。 | |

3.12 功能菜单—窗口



3.12.1 关闭目前窗口

关闭目前正在显示的窗口。此窗口并不是被删除，而是被隐藏起来，如果要重新开启，就必须以开启旧画面的方式打开。

3.12.2 关闭所有窗口

关闭所有窗口，并不是被删除，而是被隐藏起来。如果要重新开启，就必须以开启旧画面的方式打开。

3.12.3 下一个窗口

切换正在显示的窗口的下一个窗口。如果此窗口已经是最后一个，则不会切换。

3.12.4 上一个窗口

切换正在显示的窗口的上一个窗口，如果此窗口已经是第一个窗口，则不会切换。

3.12.5 重叠显示

点选窗口的重叠显示选项来选取画面以重叠的方式显示，以一次显示多个画面切换后，所有画面会以重叠的方式显示出来（图 3-12-1）。

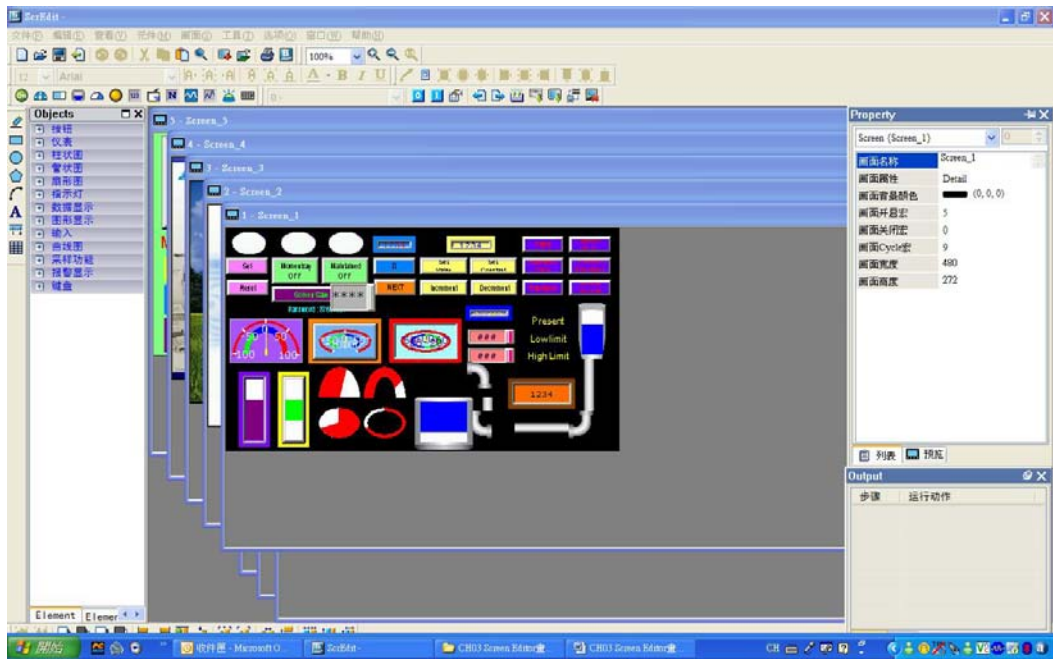


图 3-12-1 重叠显示选取后画面

3.12.6 水平并排显示

点选窗口的水平并排显示选项来选取，画面以水平并排的方式显示，画面会自动缩小高度让所有的画面显示出来，所以能一次显示多个画面（图 3-12-2）。

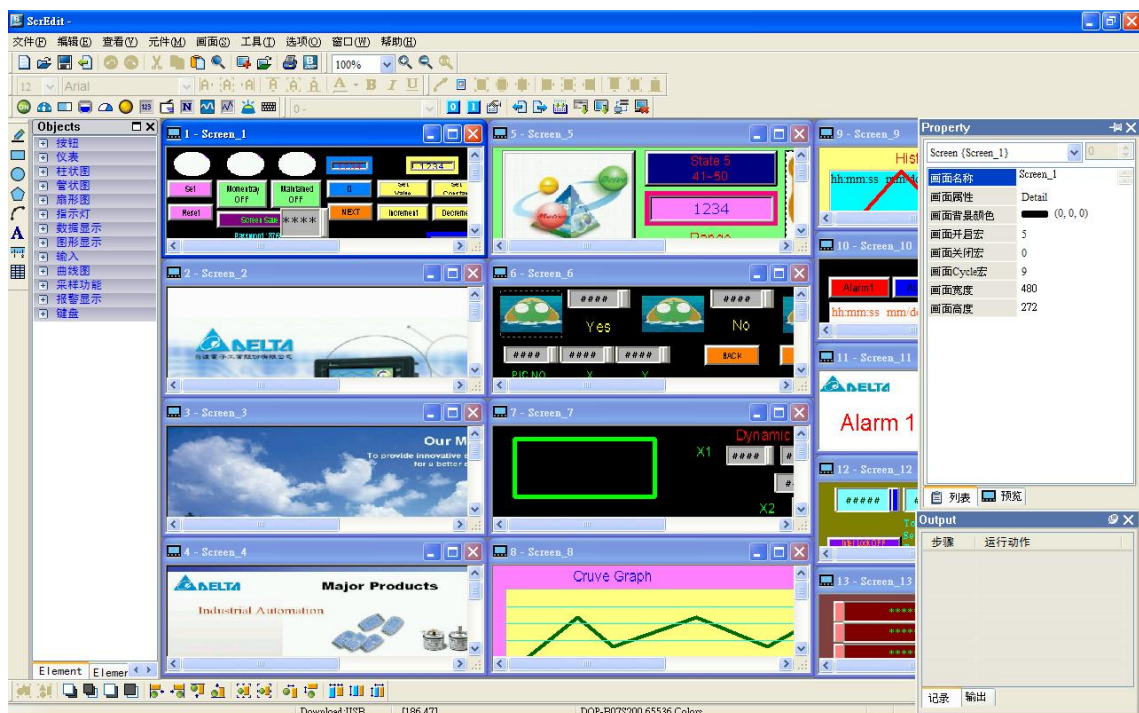


图 3-12-2 水平显示显示选取后画面

3.12.7 垂直并排显示

点选窗口的垂直并排显示选项来选取，画面以垂直并排的方式显示，画面会自动缩小宽度让所有的画面显示出来，所以能一次显示多个画面（图 3-12-3）。

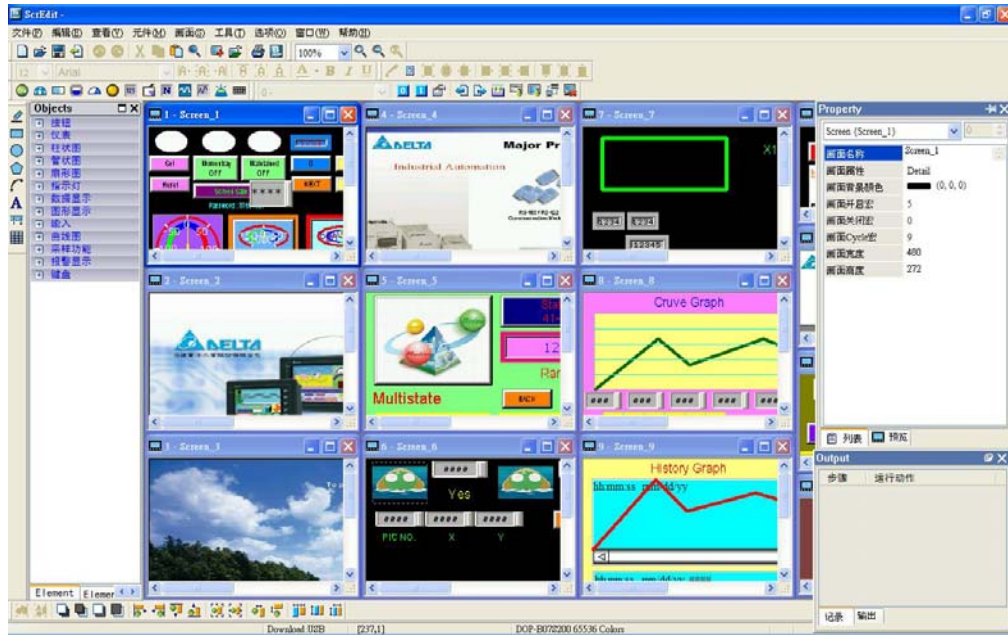


图 3-12-3 垂直并排选取后显示后画面

3.12.8 窗口预览

当有窗口时，会在窗口的选项里面的最下面显示所有的窗口。如果要切换，可以直接点选便可以切换了（图 3-12-4）。

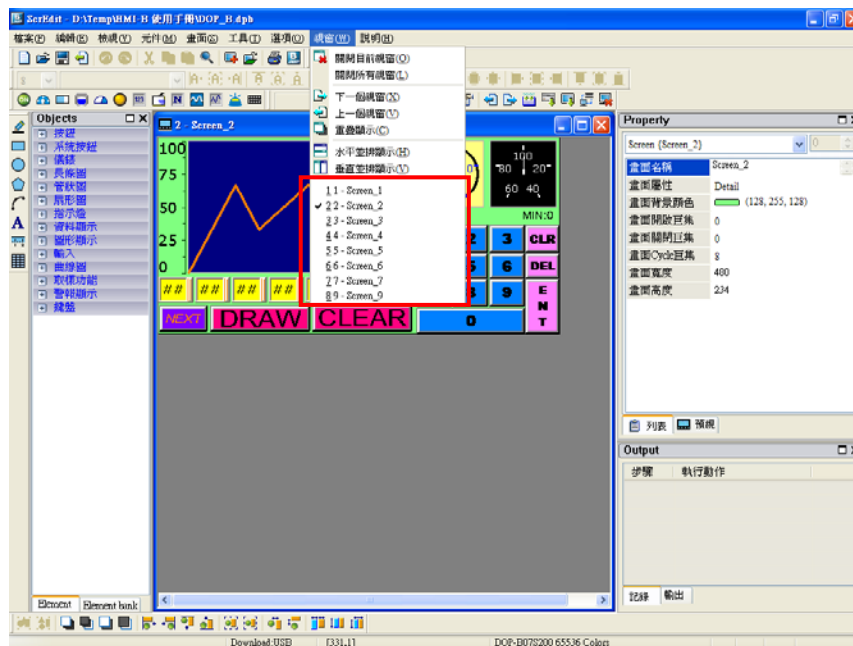
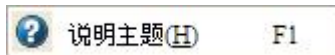


图 3-12-4 窗口预览

3.13 功能菜单—帮助



3.13.1 说明主题

显示 Screen Editor 的版本及包含的人机韧体版本（图 3-13-5）。



图 3-13-1 Screen Editor 的版本

3.14 宏功能说明

人机提供用户相当多样的宏指令，有算术运算宏、逻辑运算宏、数据搬移宏、数据转换宏、比较宏、流程控制宏、位设置宏、通讯宏及绘图宏等（图 3-14-1），



图 3-14-1 宏种类

善用宏指令的优点众多，例如：

1. 画面程序最佳化；
2. 有效分担 PLC 控制器的工作量，以减轻其负担；
3. 克服人机硬件与软件先天设计上的限制。

以下将详细介绍宏分类、编辑方法与语法撰写。

3.14.1 宏的种类与说明

人机宏的种类共有十一种，细分成四大类：

1. ON/OFF 宏：存在于所有 Bit 按钮元件，例如保持型按钮或交替型按钮等。
2. 执行前/执行后宏：存在于所有的按钮元件及（文）数值输入元件。
3. 画面开启/画面关闭/画面 Cycle 宏：以画面为单位，在每一个画面有各自独立的画面开启/画面关闭/画面 Cycle 宏。
4. Initial 宏/Background 宏/Clock 宏/子宏：以专案（整个人机程序）为单位，在每一个专案有各自独立的 Initial 宏/Background 宏/Clock 宏/子宏。

| 宏名称 | 附注 |
|------------|--|
| On 宏 | 存在于所有 Bit 按钮元件，1 个按钮存在 1 个 On 宏。 |
| Off 宏 | 存在于所有 Bit 按钮元件，1 个按钮存在 1 个 Off 宏。 |
| 执行前宏 | 存在于所有的按钮元件及（文）数值输入元件，1 个元件存在 1 个执行前宏。 |
| 执行后宏 | 存在于所有的按钮元件及（文）数值输入元件，1 个元件存在 1 个执行后宏。 |
| 画面开启宏 | 1 个画面有 1 个画面开启宏，n 个画面有 n 个画面开启宏。 |
| 画面关闭宏 | 1 个画面有 1 个画面关闭宏，n 个画面有 n 个画面关闭宏。 |
| 画面 Cycle 宏 | 1 个画面有 1 个画面 Cycle 宏，n 个画面有 n 个画面 Cycle 宏。 |

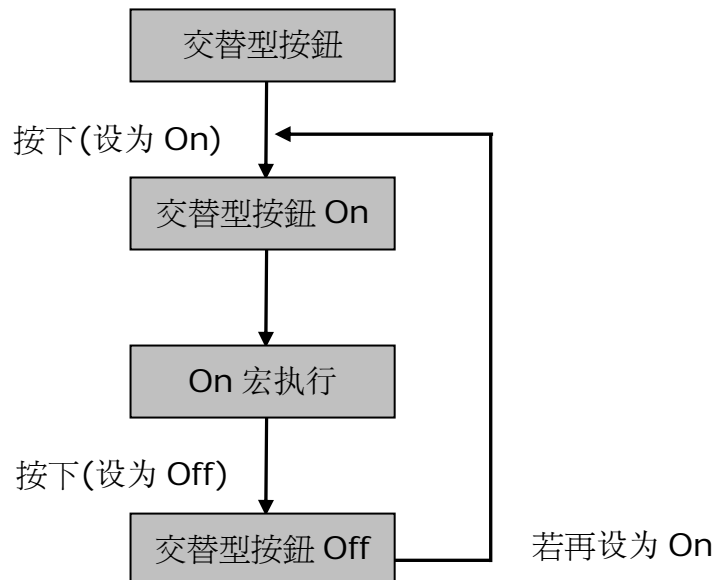
| 宏名称 | 附注 |
|--------------|-----------------------------------|
| Initial 宏 | 1 个专案（整个人机程序）中有 1 个 Initial 宏。 |
| Background 宏 | 1 个专案（整个人机程序）中有 1 个 Background 宏。 |
| Clock 宏 | 1 个专案（整个人机程序）中有 1 个 Clock 宏。 |
| 子宏 | 1 个专案（整个人机程序）中有 512 个子宏。 |

表 3-14-1 宏分类说明

3.14.1.1 On 宏

此宏存在于所有 Bit 按钮元件，当按钮被设为 On 时便会执行该宏，执行后需等到下次此按钮又由 Off 被设为 On 时才会再次执行。注意，若非透过按按钮动作设 On（例如使用触发设 On），则不会执行 On 宏。

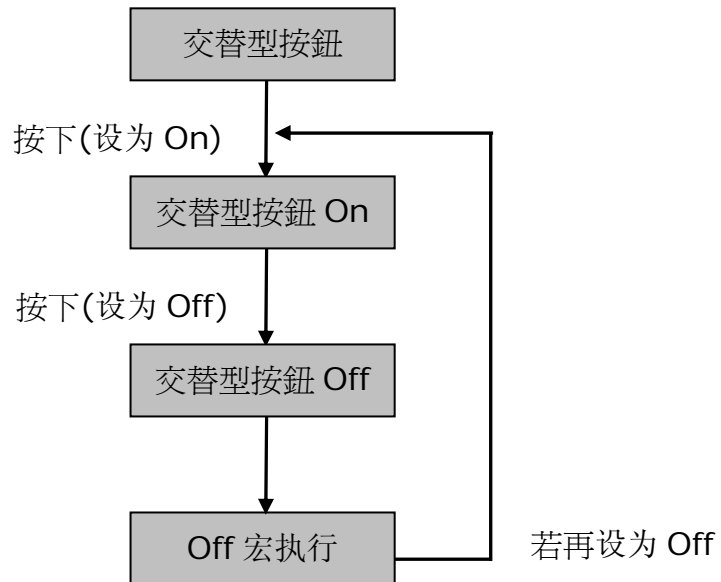
假设画面中有一交替型按钮，则按下之后执行程序如下图所示：



3.14.1.2 Off 宏

此宏存在于所有 Bit 按钮元件，当按钮被设为 Off 时便会执行该宏，执行后需等到下次此按钮又由 On 被设为 Off 时才会再次执行。注意，若非透过按按钮动作设 Off（例如使用触发设 Off），则不会执行 Off 宏。

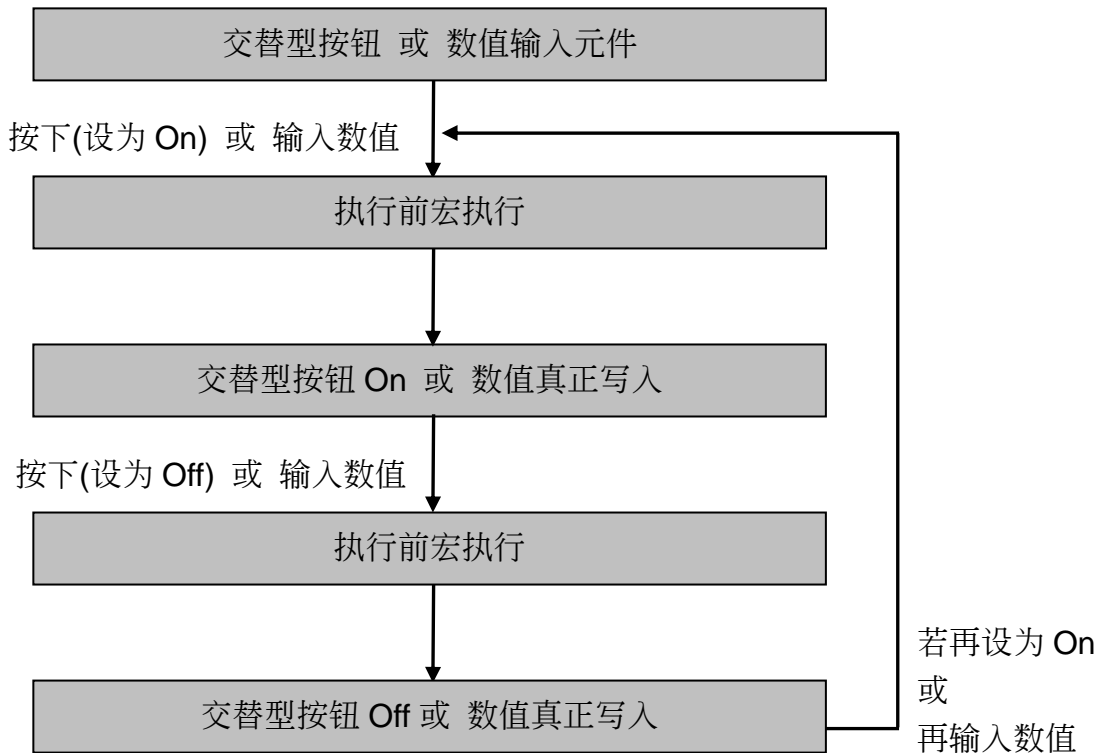
假设画面中有一交替型按钮，则按下之后执行程序如下图所示：



3.14.1.3 执行前宏

存在于所有的按钮元件及（文）数值输入元件，其动作顺序是先判断按钮元件或（文）数值输入元件的状态变化，若发现有变化，执行前宏会先执行，然后再执行按钮动作或（文）数值输入动作。注意，如果不是通过按按钮动作设 On/Off（例如使用触发设 On/Off），则执行前宏不会被执行。同理，如果不是通过数值输入元件键盘输入数值（例如使用宏给值），则执行前宏也不会被执行。

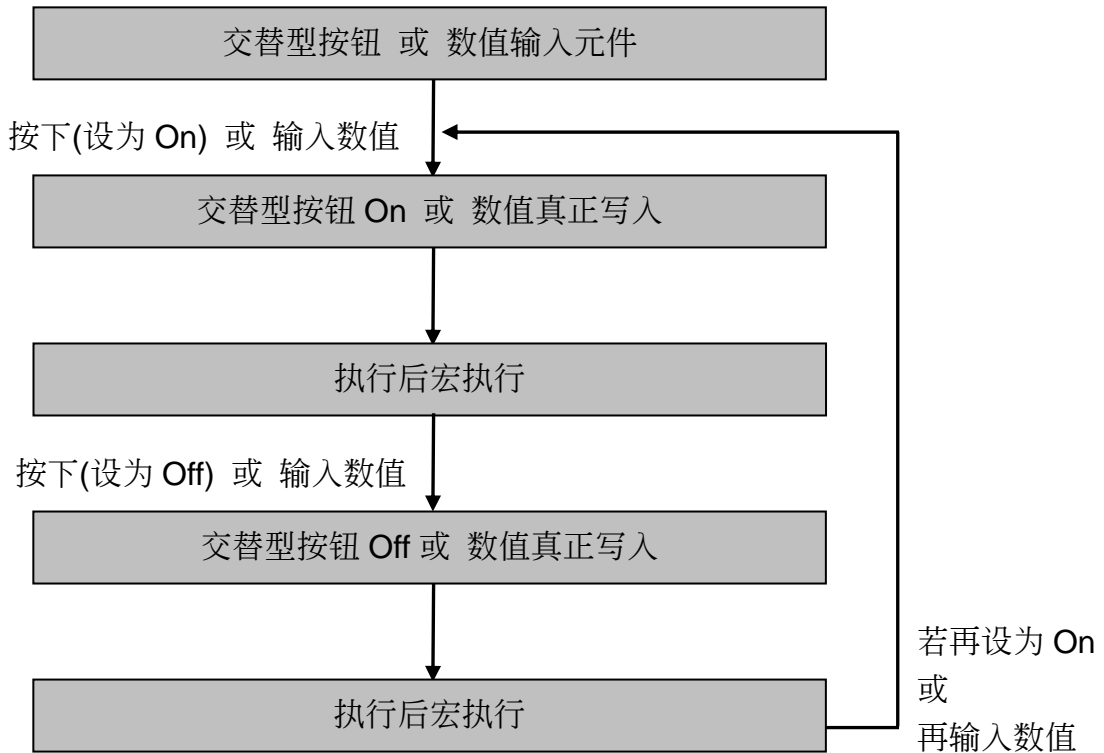
假设画面中有一交替型按钮或数值输入元件，则按下之后执行程序如下图所示：



3.14.1.4 执行后宏

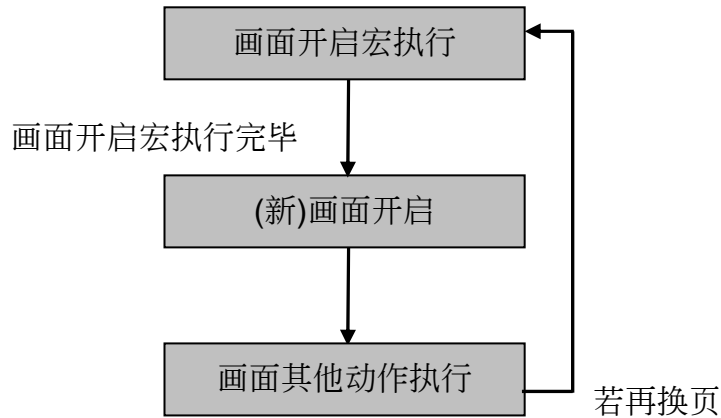
存在于所有的按钮元件及（文）数值输入元件，其动作顺序是先判断按钮元件或（文）数值输入元件的状态变化，若发现有变化，按钮动作或（文）数值输入动作会先执行，然后再执行执行后宏。注意，如果不是通过按按钮动作设 On/Off（例如使用触发设 On/Off），则执行后宏不会被执行。同理，如果不是通过数值输入元件键盘输入数值（例如使用宏给值），则执行后宏也不会被执行。

假设画面中有一交替型按钮或数值输入元件，则按下之后执行程序如下图所示：



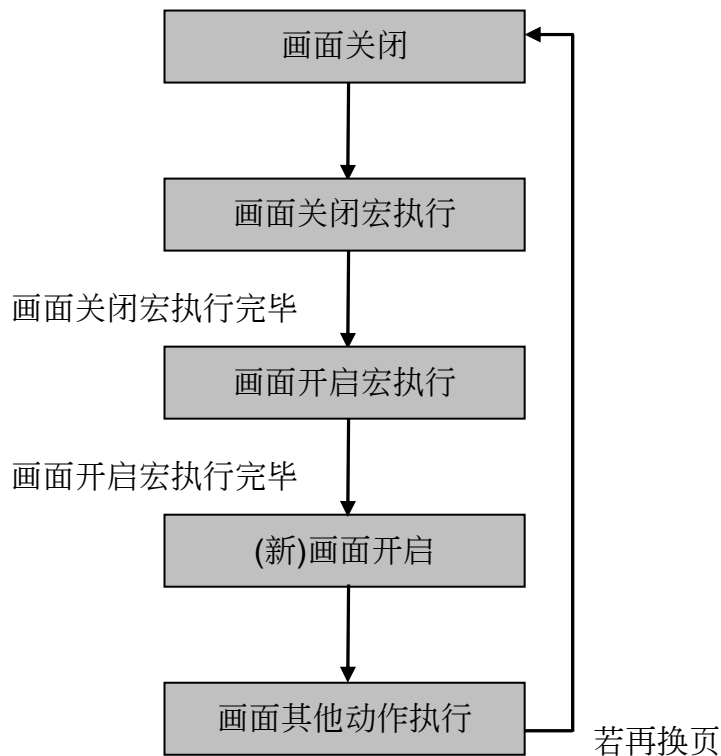
3.14.1.5 画面开启宏

每一个画面都有一个画面开启宏，当用户开启画面或是切换至新画面后便会执行画面开启宏。而整个画面的其他动作，要等画面开启宏执行完毕后才开始执行。
画面开启宏执行程序如下图所示：



3.14.1.6 画面关闭宏

每一个画面都有一个画面关闭宏，当用户关闭画面或是切换至新画面前便会执行画面关闭宏。此宏要先执行完毕，新画面的动作才会开始执行。
画面关闭宏执行程序如下图所示：



3.14.1.7 画面 Cycle 宏

每一个画面都有一个画面 Cycle 宏，当画面开启宏执行完毕后，便依照用户所设定的 Cycle 宏延迟时间来执行画面 Cycle 宏。用户可以在画面属性中设定 Cycle 宏延迟时间（图 3-14-2），也就是说每一次画面 Cycle 宏执行结束后，延迟多久时间再重新开始执行，系统预设时间为 100ms。

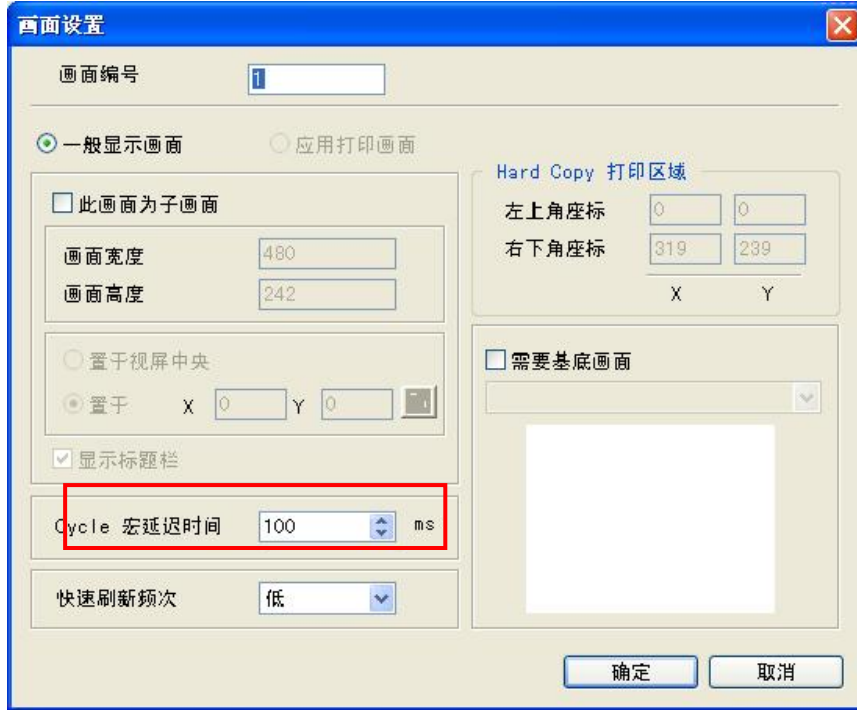
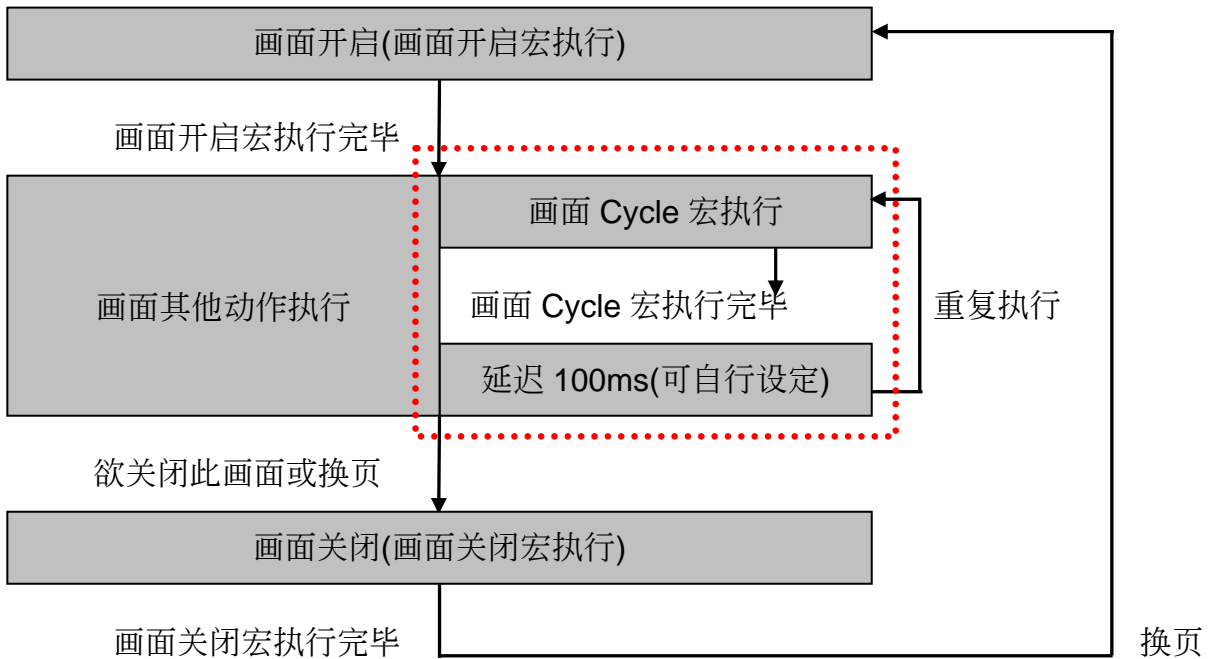


图 3-14-2 画面 Cycle 宏延迟时间设定

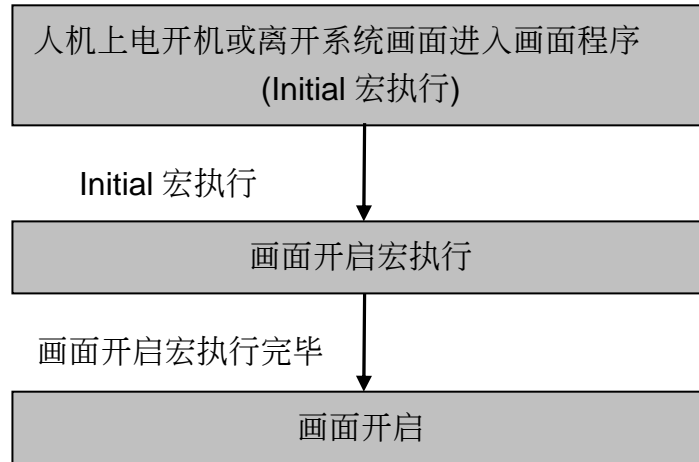
画面 Cycle 宏执行程序如下图所示：



3.14.1.8 Initial 宏

一个专案（整个人机程序）中有一个 Initial 宏，Initial 宏为人机启动时第一个执行的宏，因此用户可将整个人机程序中所需的一些初始设定写在 Initial 宏中。

Initial 宏执行程序如下图所示：



3.14.1.9 Background 宏

一个专案（整个人机程序）中有一个 Background 宏，Background 宏在人机运作过程中会一直重复执行，一次执行一行或是数行的程序（并非一次执行完毕），执行到最后一行程序后，会直接重头重新执行一次。而 Background 宏执行行数可由模块参数中背景宏的更新周期来设定，如图 3-14-3。

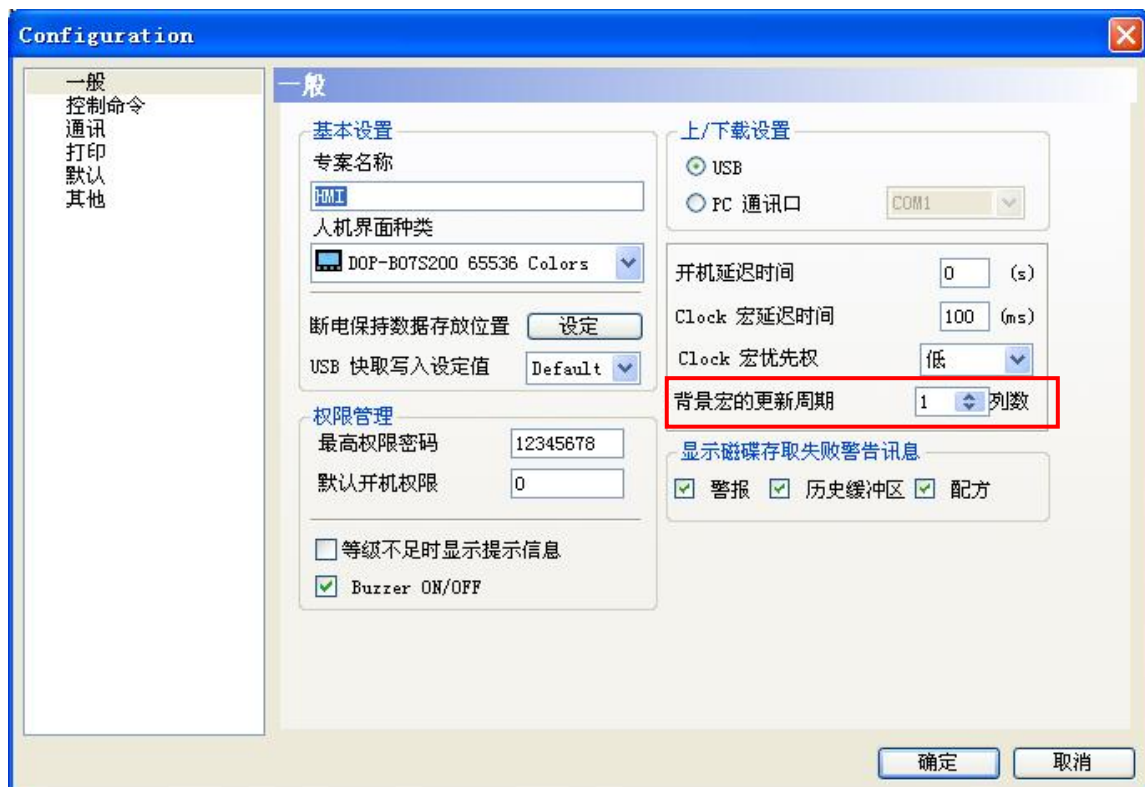


图 3-14-3 Background 宏更新周期设定

若画面上有 25 个元件地址，Background 宏内有 5 行程序，且设定一次执行 2 行（背

景宏的更新周期设定为 2)，则 Background 宏执行程序如下图所示：



3.14.1.10 Clock 宏

一个专案（整个人机程序）中有一个 Clock 宏，Clock 宏在人机运作过程中会一直重复执行。与 Background 宏不同的是，Clock 宏会一次执行完所有的程序，而非一次执行一行或数行。而 Clock 宏类似画面 Cycle 宏，依照所设定的 Clock 宏延迟时间来重复执行。用户可以在模块参数中设定 Clock 宏延迟时间（图 3-14-4），也就是说每一次 Clock 宏执行结束后，延迟多久时间再重新开始执行，系统预设时间为 100ms。

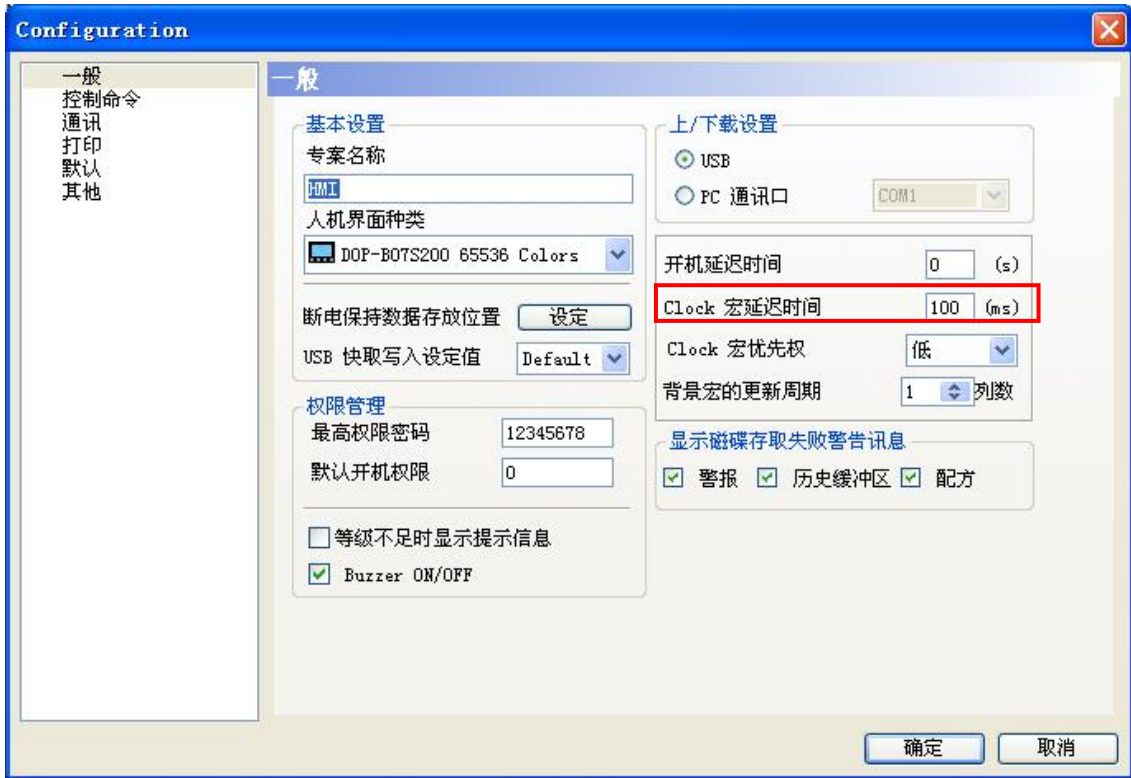
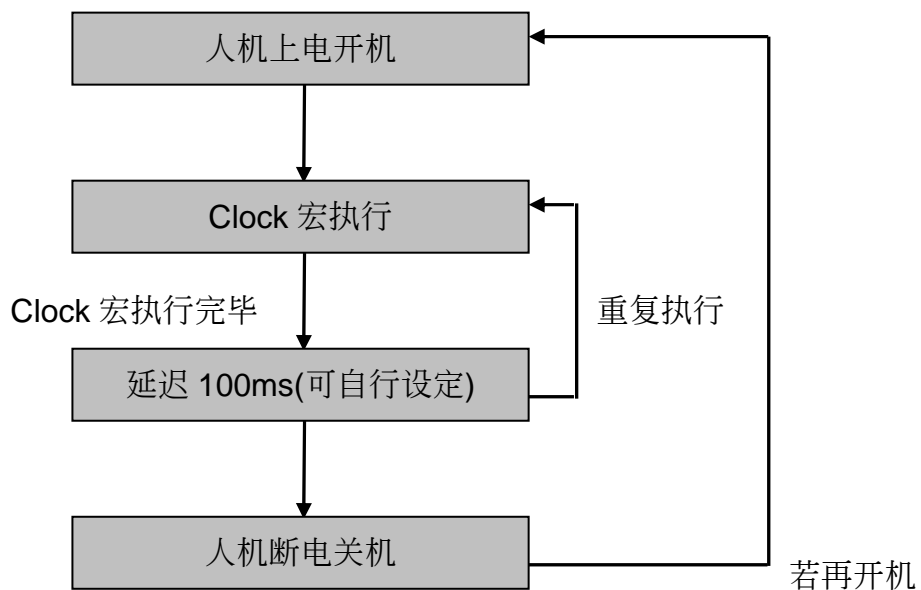


图 3-14-4 Clock 宏延迟时间设定

Clock 宏执行程序如下图所示：



3.14.1.11 子宏

一个专案（整个人机程序）中有 512 个子宏，编号分别为 1 — 512（图 3-14-5、图 3-14-6），子宏类似程序语言中的子程序一样，用户可以把重复性高的动作或是功能写入子宏中，需要用到时再呼叫即可。如此不但可以节省写宏的时间，而且维护容易。而要使用子宏时只需要在程序中呼叫子宏的编号即可，例如要使用第二个子宏，程序中写入” CALL 2” 即完成呼叫第二个子宏的动作。在子宏程序中也可以再 CALL 另一个子宏程序，要特别注意的是子宏呼叫子宏动作请勿超过六层。另外每个子宏都可由用户自由编写名称来表示此子宏功能（图 3-14-5、图 3-14-6），以方便用户管理（子宏预设名称为 Sub-macro (n)，n 为 1 到 512）。

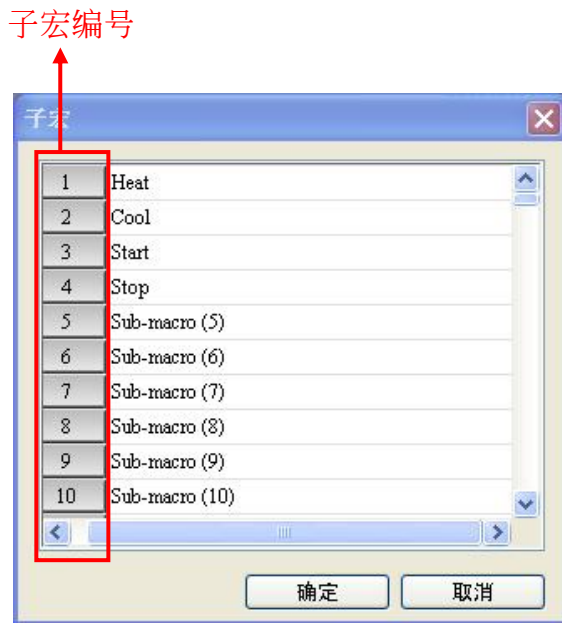


图 3-14-5 子宏画面 I

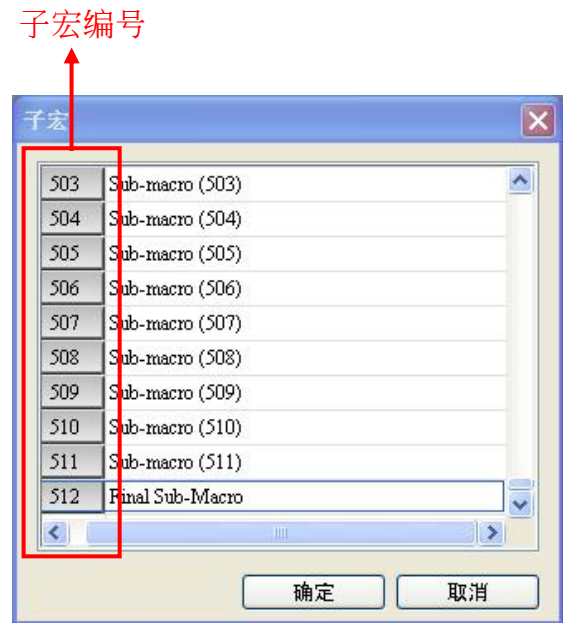


图 3-14-6 子宏画面 II

若在画面开启宏（图 3-14-7）以及子宏 1（图 3-14-8）编辑程序，则画面开启宏执行程序如下图 3-14-9 所示：

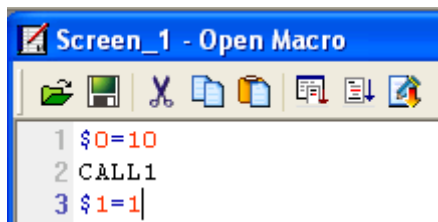


图 3-14-7 画面开启宏画面

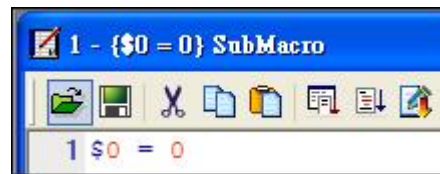


图 3-14-8 子宏 1 画面

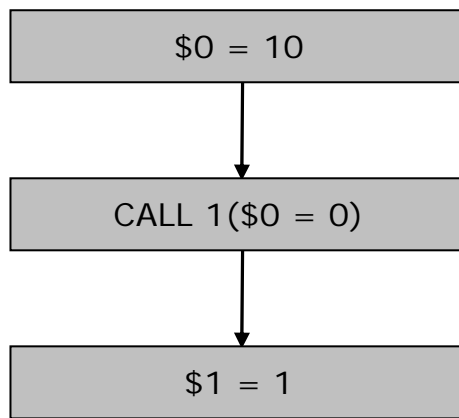


图 3-14-9 画面开启宏执行程序

画面开启宏执行至第二行 CALL 1，亦即呼叫子宏 1，此时会执行子宏内的程序，执行完毕后会继续执行画面开启宏，直到结束。此时 $\$0 = 0$ 且 $\$1 = 1$ 。

NOTE

任何宏的编辑都需要特别注意到宏语法的逻辑性，尤其是在循环及判断式的使用，若在宏中编写了一个无穷循环或需要等到某个条件满足才会继续执行的判断式，都有可能使得人机动作无法正常的执行，甚至造成人机触碰无反应的情况，因此用户编辑宏时必须格外谨慎。

3.14.2 宏的编辑方式

3.14.2.1 宏编辑画面与工具栏

进入要编辑的宏画面后(图 3-14-10)即可开始编写宏。特别要注意的是,每一种(类)宏能编写的行数最多皆是 512 行,每行可写的字数最多皆是 128 个字。

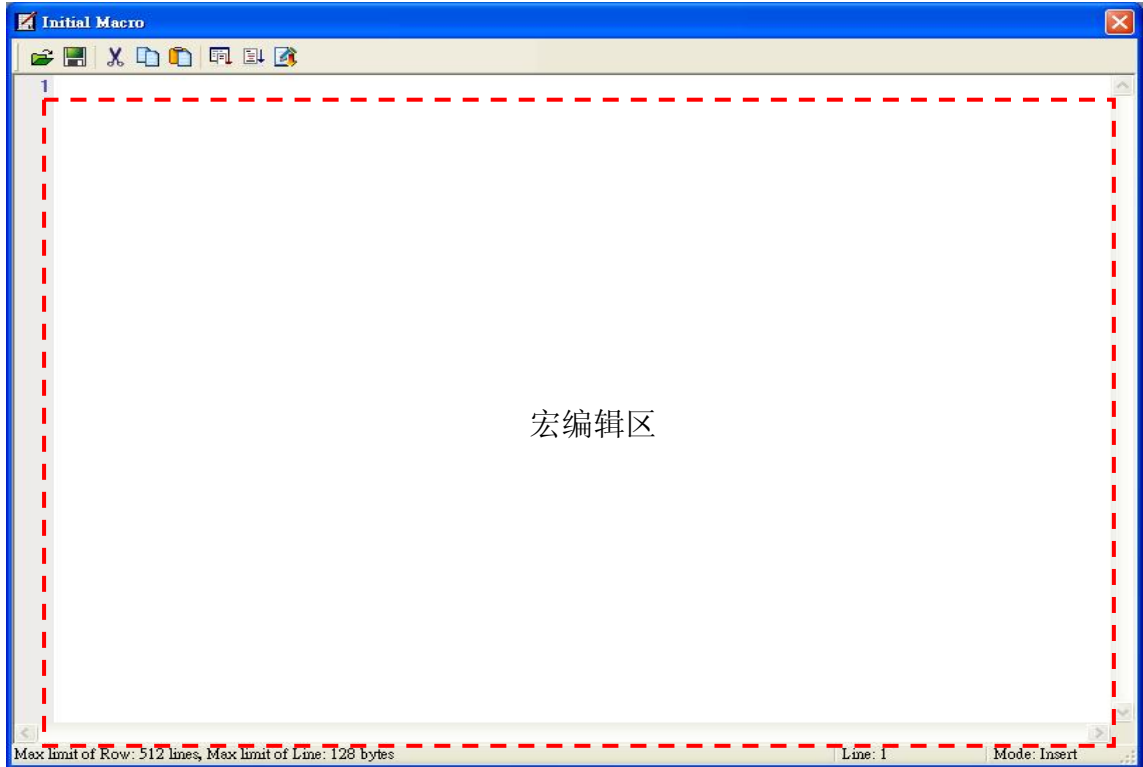


图 3-14-10 宏编辑窗口

宏编辑工具栏(图 3-14-11)详细介绍如下:



图 3-14-11 宏编辑工具栏

汇入

为了方便用户编辑宏，我们提供了汇入的功能，用户可使用汇入功能将先前所储存（汇出）的宏开启，如此一来重复性高的宏文件便不需要重新编辑，大大的缩减编辑宏的时间，开启的窗口请参考图 3-14-12。

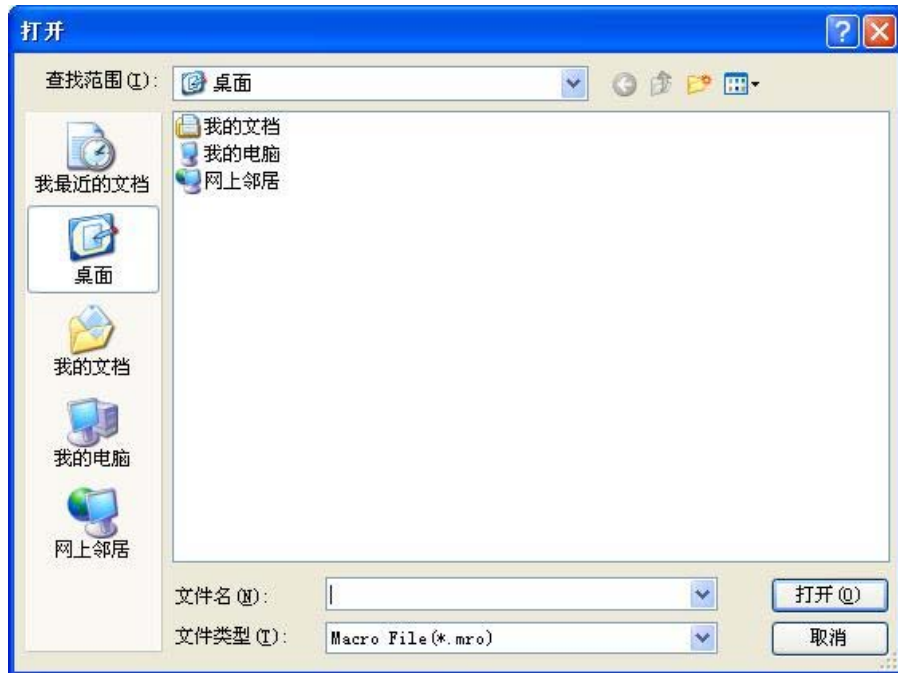


图 3-14-12 汇入画面

汇出

为了方便用户编辑宏，我们提供了汇出的功能，用户可使用汇出功能将宏储存起来，作为备份或提供其它画面程序使用，汇出的窗口请参考图 3-14-13。

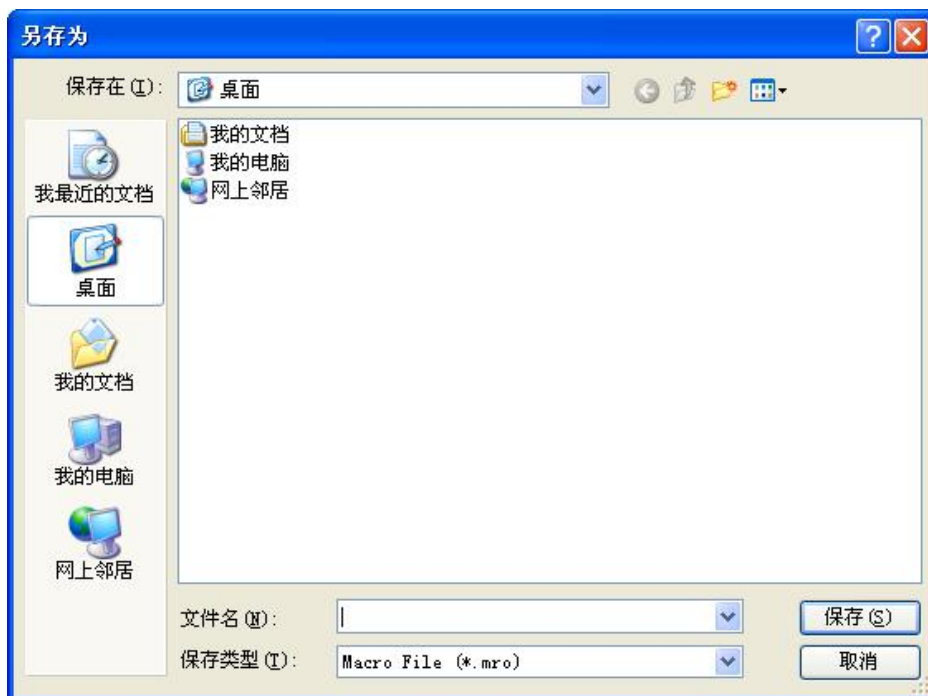
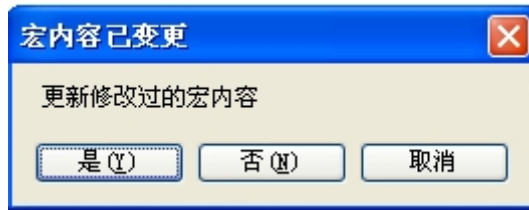


图 3-14-13 汇出画面

更新

宏编辑完成后按下此更新钮，程序会立刻进行语法检查的动作，语法检查完成无误后此宏立即储存于人机程序中。若宏编辑完成后未按下此更新钮就作关闭宏编辑画面的动作，则会跳出以下确认窗口：



按下”是”后，程序会立刻进行语法检查的动作，语法检查完成无误后此宏立即储存于人机程序中。

语法检查时若发现错误，则会跳出以下信息窗口，此时请检查宏语法。

**剪下、复制、粘贴**

此类功能的操作方式与 Microsoft Office 软件操作方式一样，可剪切、复制与粘贴宏指令。

热键设定：Ctrl+X 为剪切功能；Ctrl+C 为复制功能；Ctrl+V 为粘贴功能。

语法检查

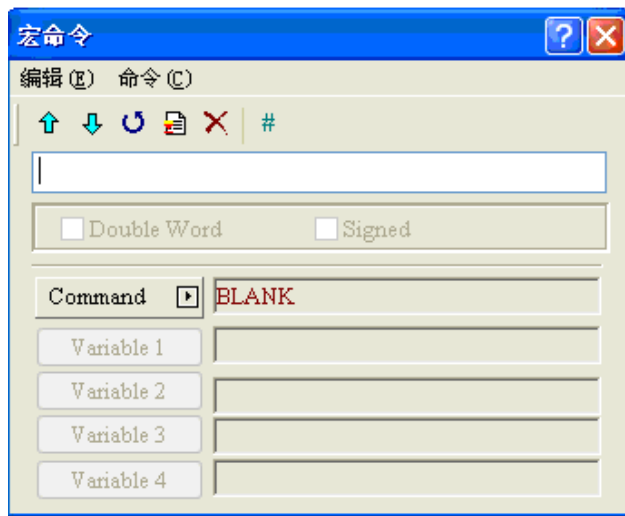
宏编辑完成后按下此语法检查按钮，程序会立即执行语法检查，语法检查时若发现错误，则会跳出以下信息窗口，此时请检查宏语法。



要特别注意的是，执行语法检查的动作并非等同于宏编译的动作，宏的编译会在人机画面程序编译时同时执行。

使用宏输入对话框

宏输入对话框如下图所示：



宏编辑的方式有两种，第一种是使用宏输入对话框的方式来编辑宏；第二种是使用手动输入的方式来编辑宏。再此我们建议用户以第一种方式来输入宏，较不易发生语法错误。而详细的宏编辑方式分别于下面二节详述。

3.14.2.2 宏输入对话框编辑方式


在宏编辑工具栏上点选 ，宏输入对话框便会自动弹出，用户可借此对话框选择并编辑宏指令，如图 3-14-14 所示。



图 3-14-14 宏输入对话框

编辑：

点选[编辑]选单，选单内包含了宏编辑工具，可帮助用户编辑宏（图 3-14-15）。而宏编辑工具也在宏输入对话框窗口中同步以工具栏的方式呈现，如图 3-14-16。

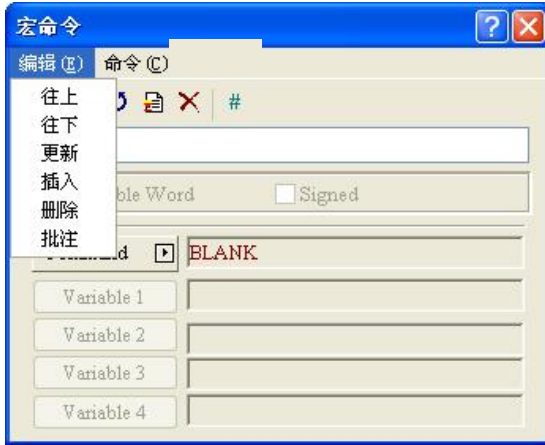


图 3-14-15 编辑选单

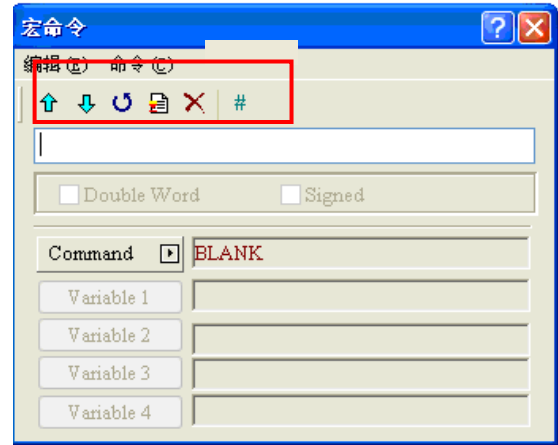


图 3-14-16 宏命令编辑工具栏

往上

往上 (宏编辑画面) 光标往上移动。



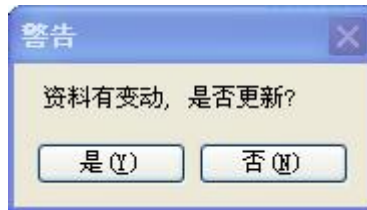
往下

往下 (宏编辑画面) 光标往下移动。



更新

宏编辑完成后按下此更新钮，程序会立刻进行语法检查的动作，语法检查完成无误后，程序会自动将宏输入至宏编辑画面。若宏编辑完成后未按下此更新钮就作关闭宏输入对话框的动作，则会跳出以下确认窗口：



更新



按下”是”后，程序会立刻进行语法检查的动作，语法检查完成无误后此宏立即输入至宏编辑画面。

语法检查时若发现错误，则会跳出以下信息窗口，此时请检查宏语法。



插入

插入 将目前所编辑的宏，插入两行宏之中，其详细动作为取代原本的那一行，将原本的宏往下移动一行。



删除

删除 将目前这所点选的这一行宏是指直接删除，下面如果还有宏，则往上递补。



注解

用户可利用此批注功能增加宏的可读性及维护性。而使用方法是直接在宏编辑画面中键入符号#后再编写注解或宏程序，或者是使用宏输入对话框上的选项或按钮直接在宏程序前加上符号#。另外要注意的是，注解在宏中是不会被执行的。

批注
#

命令：

点选[命令]选单，选单内包含了数种类型的宏指令，提供用户编辑宏（图 3-14-17）。而宏指令也可由宏输入对话框窗口中的 Command 按钮选取，如图 3-14-18。

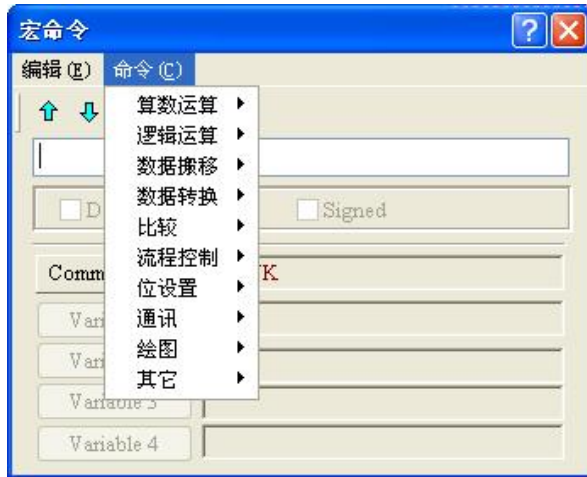


图 3-14-17 命令菜单



图 3-14-18 Command 菜单

图 3-14-20 ~ 图 3-14-29 为所有的宏指令，其详细的语法将于 3.14.3 节说明。

- 算数运算 ▶
- 逻辑运算 ▶
- 数据擦移 ▶
- 数据转换 ▶
- 比较 ▶
- 流程控制 ▶
- 位设置 ▶
- 通讯 ▶
- 绘图 ▶
- 其它 ▶

图 3-14-19 宏指令类型

- +
-
- *
- /
- %
- MUL64
- ADDSUMW

- FADD
- FSUB
- FMUL
- FDIV
- FMOD

- SIN
- COS
- TAN
- COT
- SEC
- CSC

图 3-14-20 算术运算

```

|
&&
^
NOT
<<
>>
    
```

图 3-14-21 逻辑运算

```

MOV
EMOV
FILL
FILLASC
-----
FMOV
    
```

图 3-14-22 数据搬移

| | |
|---------|-------|
| BCD | XCHG |
| BIN | MAX |
| TODWORD | MIN |
| TOWORD | TOHEX |
| TOBYTE | TOASC |
| SWAP | |
| | ----- |
| | FCNV |
| | ICNV |

图 3-14-23 数据转换

```

IF ... THEN GOTO ▶
IF ... ▶
ELSEIF ... ▶
ELSE
ENDIF
-----
FCMP
    
```

图 3-14-24 比较

```

GOTO
LABEL
CALL
RET
FOR
NEXT
END
    
```

图 3-14-25 流程控制

```

BITON
BITOFF
BITNOT
GETB
    
```

图 3-14-26 位设置

```

INITCOM
ADDSUM
XORSUM
PUTCHARS
GETCHARS
SELECTCOM
CLEARCOMBUFFER
CHRCHKSUM
LOCKCOM
UNLOCKCOM
STATIONON
STATIONOFF
    
```

图 3-14-27 通讯

```

RECTANGLE
LINE
POINT
CIRCLE
    
```

图 3-14-28 绘图

```

Time Tick
GETLASTERROR
Comment
Delay
GETSYSTEMTIME
SETSYSTEMTIME
GETHISTORY
EXPORT
    
```

图 3-14-29 其他

选定宏指令后（图 3-14-30、图 3-14-31），指令内所需设定的变量（Var1 - Var4）可经由变量设定按钮来作设定：

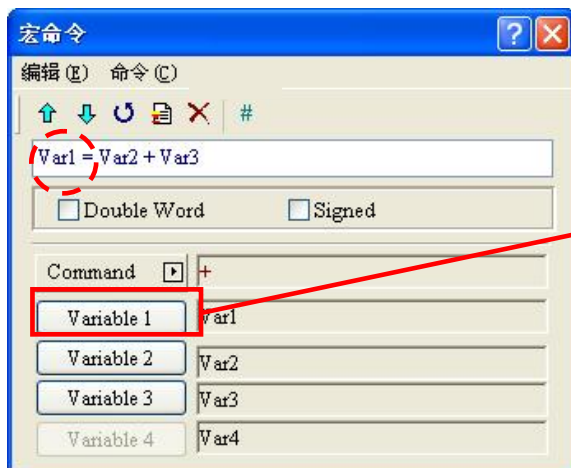


图 3-14-30 指令选取画面



图 3-14-31 变量设定画面

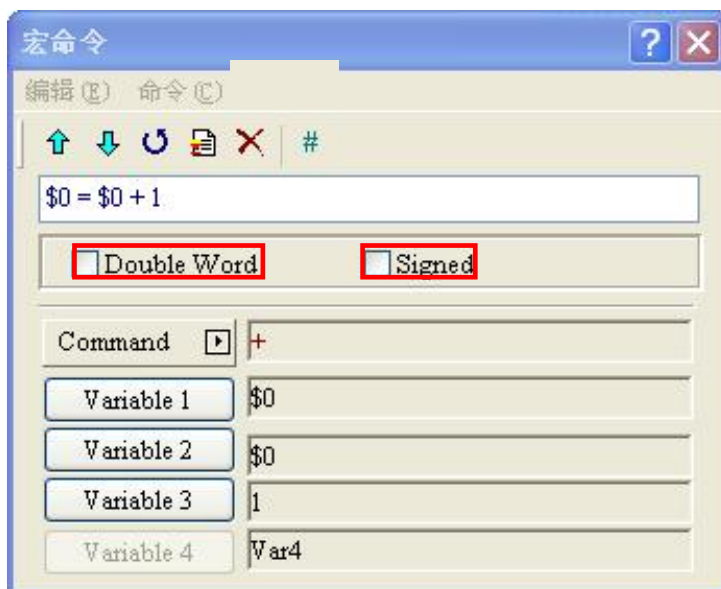


图 3-14-32 Double Word 与 Signed 选项

某些宏指令提供 32 位及有符号数运算，如图 3-14-32 所示，若用户勾选 Signed，表示该行指令是以有符号数做运算，若无勾选 Signed，则是以无符号数作运算。若用户勾选 Double Word，表示该行指令是以 32 位做运算，若无勾选 Double Word，则是以 16 位做运算。

| | |
|------------------------|--------|
| Unsigned | 无符号数 |
| Signed | 有符号数 |
| WORD | 16 位资料 |
| DW (DOUBLE WORD、DWORD) | 32 位资料 |

表 3-14-2 宏名词定义

在此要特别注意的是，当用户设定宏指令为 Double Word 格式时，指令内每个存储区位置皆占两个寄存器。举一数据搬移例子来作说明， $S0 = S2 (DW)$ ，代表把人机内部存储区 $S2$ 及 $S3$ 内的数据搬移至人机内部存储区 $S0$ 及 $S1$ 中。

设定完成后，按下更新钮，宏指令立即输入至宏编辑画面，如图 3-14-33 所示：

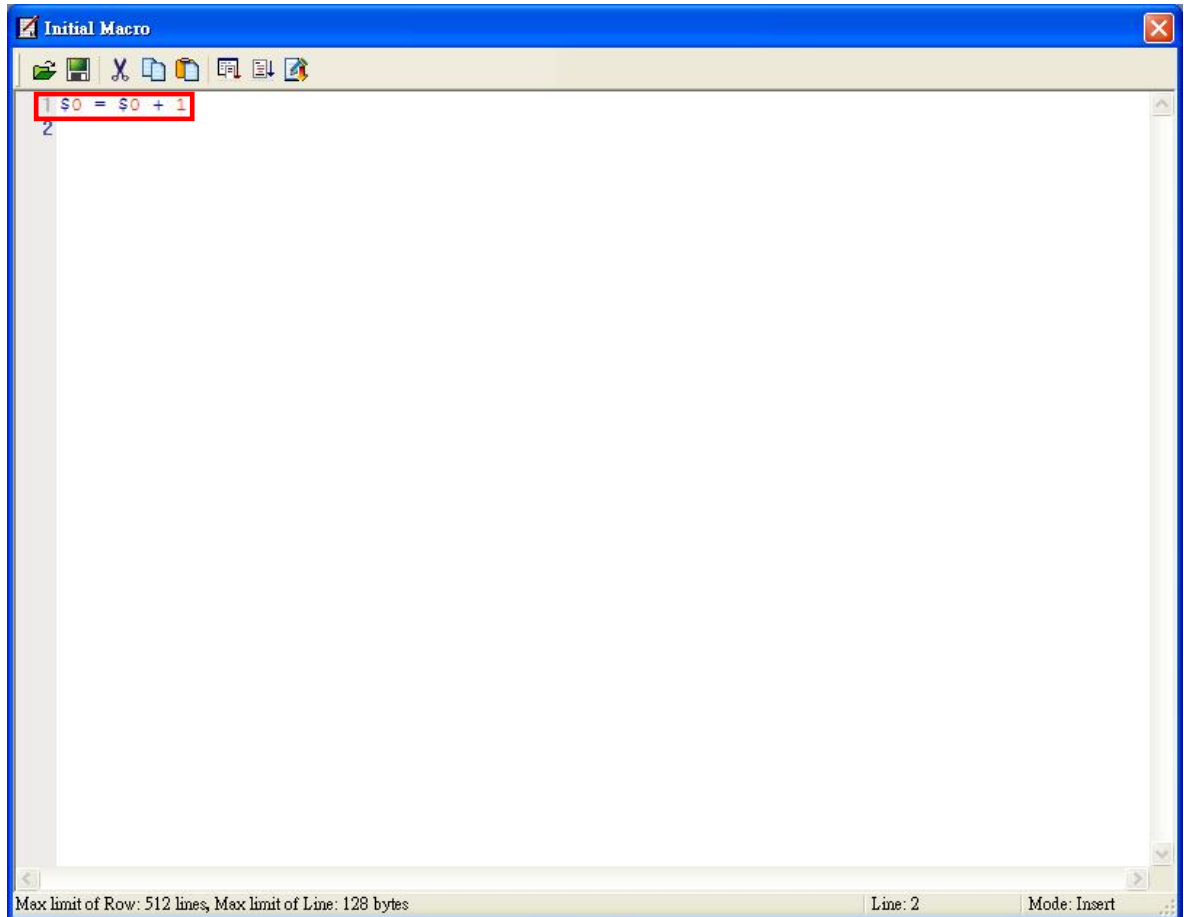


图 3-14-33 宏指令输入完成画面

宏编辑完成后，需进行语法检查与更新动作后方能离开宏编辑画面，而语法检查或更新钮说明请参考 3.14.2.1 节。

3.14.2.3 手动输入编辑方式

手动输入编辑方式是指用户可在宏编辑画面中自行输入宏指令来编辑宏，编辑过程中若输入关键字，宏指令选取方块便会自动弹出（弹出方块不影响原先编写流程，用户可选择使用方块或继续自行输入），提供用户选取宏指令，如图 3-14-34 所示。

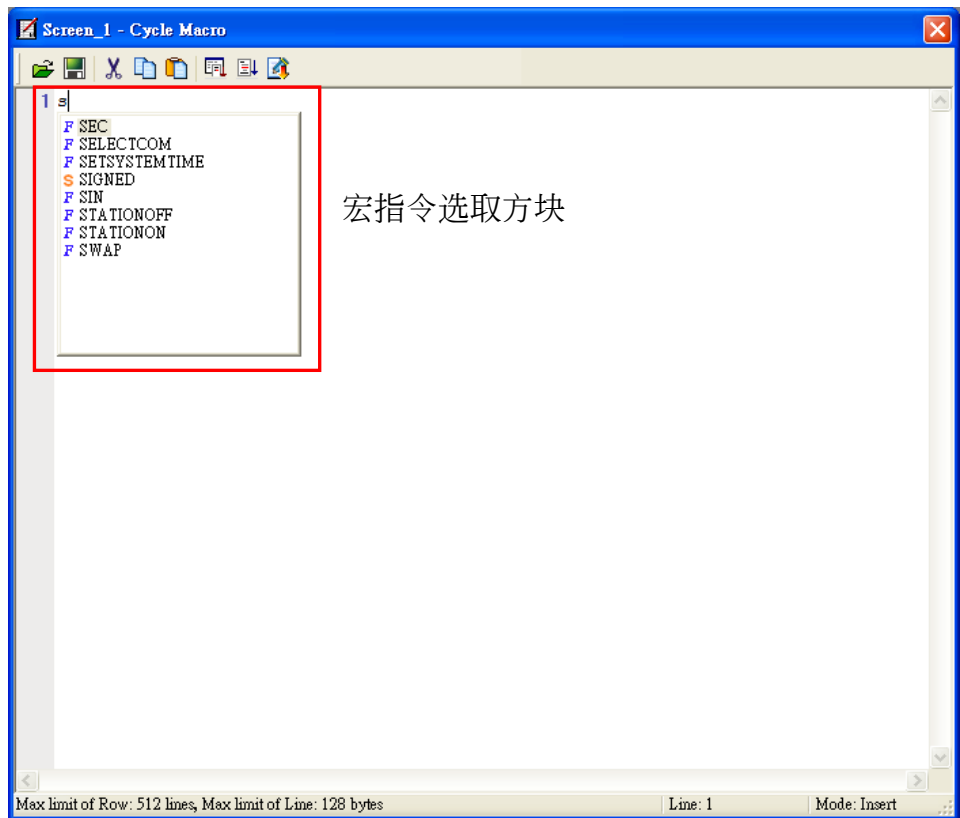


图 3-14-33 自行输入宏指令画面

输入完成，如图 3-14-34 所示。

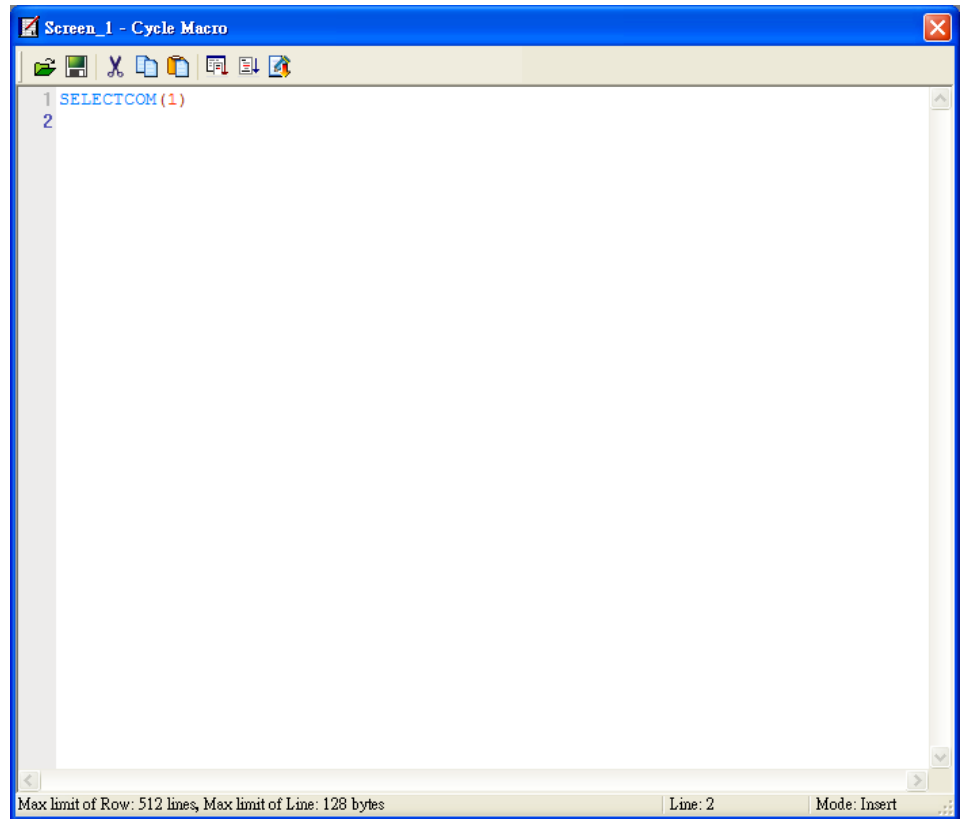


图 3-14-34 宏指令输入完成画面

宏编辑完成后，需进行语法检查与更新动作后方能离开宏编辑画面，而语法检查或更新说明请参考 3.14.2.1 节。

3.14.3 宏的编辑语法

本章节中宏的编辑语法将以宏输入对话框来作说明，有关宏输入对话框编辑方式请参考 3.14.2.2 节。

3.14.3.1 算术运算

| | |
|---------|--|
| + | |
| - | |
| * | |
| / | |
| % | |
| MUL64 | |
| ADDSUMW | |
| <hr/> | |
| FADD | |
| FSUB | |
| FMUL | |
| FDIV | |
| FMOD | |
| <hr/> | |
| SIN | |
| COS | |
| TAN | |
| COT | |
| SEC | |
| CSC | |

算术运算包含了整数运算与浮点数运算，其详细说明如下：

■ + (加法运算)

| 句型 | 代表意义 | 备注 |
|---|---------------------------|---|
| $\text{Var1} = \text{Var2} + \text{Var3}$ | Var2 和 Var3 相加运算结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|---|--|
| <p>将内部存储区\$0 加上 1 (无符号数 16 位运算)</p> |  |
| <p>将内部存储区\$1 与\$2 的 值相加后存放到\$3 (有符 号数 16 位运算)</p> |  |
| <p>将内部存储区\$4 与\$6 的 值相加后存放到\$8 (有符 号数 32 位运算)</p> |  |

■ - (减法运算)

| 句型 | 代表意义 | 备注 |
|--------------------|---------------------------|---|
| Var1 = Var2 - Var3 | Var2 和 Var3 相减运算结果存在 Var1 | <ul style="list-style-type: none"> ▶ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD，溢位的部分将会被忽略。 ▶ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|---|--|
| <p>将内部存储区\$0 减掉 1 (无符号数 16 位运算)</p> |  |
| <p>将内部存储区\$1 与\$2 的值相减后存放到\$3 (有符号数 16 位运算)</p> |  |
| <p>将内部存储区\$4 与\$6 的值相减后存放到\$8 (有符号数 32 位运算)</p> |  |

■ * (乘法运算)

| 句型 | 代表意义 | 备注 |
|--------------------|---------------------------|---|
| Var1 = Var2 * Var3 | Var2 和 Var3 相乘运算结果存在 Var1 | <ul style="list-style-type: none"> ▶ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD，溢位的部分将会被忽略。 ▶ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

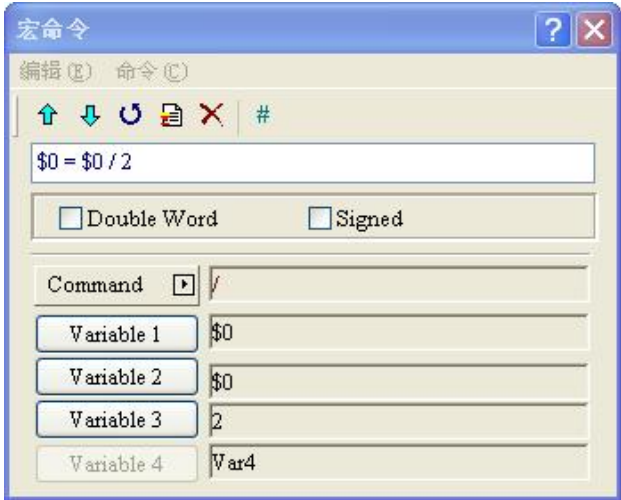

范例:

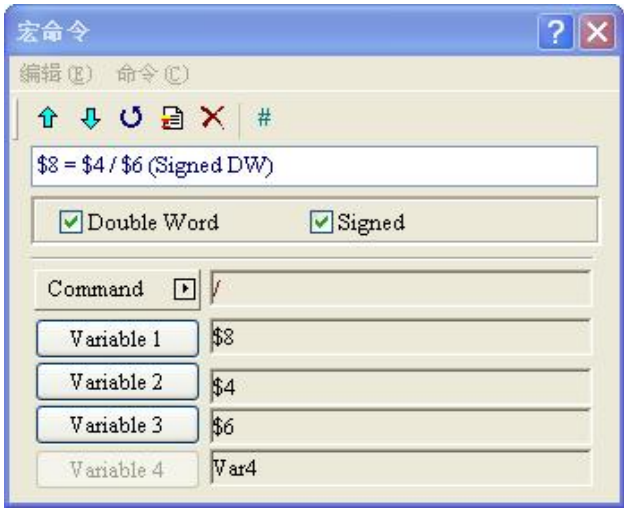
| | |
|---|--|
| <p>将内部存储区\$0 乘上 2 (无符号数 16 位运算)</p> |  |
| <p>将内部存储区\$1 与\$2 的值相乘后存放到\$3 (有符号数 16 位运算)</p> |  |
| <p>将内部存储区\$4 与\$6 的值相乘后存放到\$8 (有符号数 32 位运算)</p> |  |

■ / (除法运算)

| 句型 | 代表意义 | 备注 |
|--------------------|---------------------------|--|
| Var1 = Var2 / Var3 | Var2 和 Var3 相除运算结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 ➤ Var3 不可为 0。 |

范例：

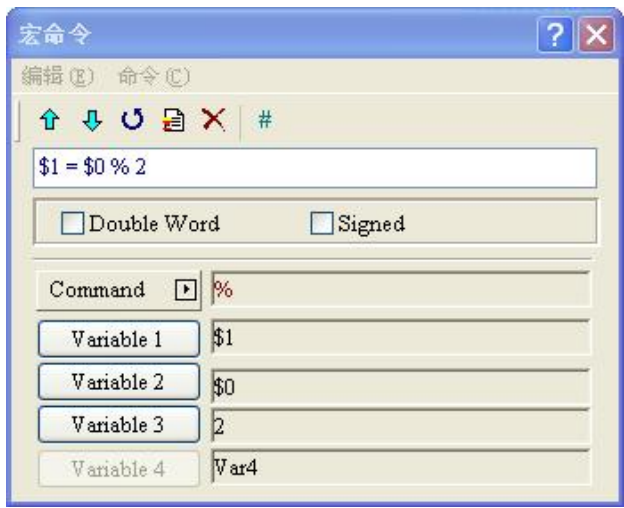
| | |
|---|--|
| <p>将内部存储区\$0 除上 2 (无符号数 16 位运算)</p> |  |
| <p>将内部存储区\$1 与\$2 的值相除后存放到\$3 (有符号数 16 位运算)</p> |  |

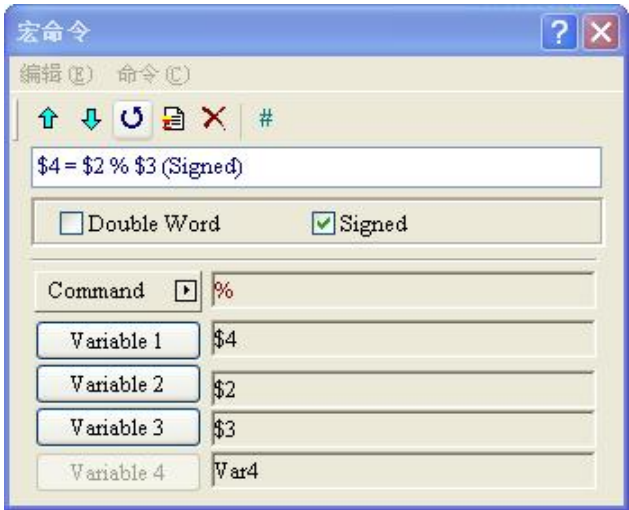
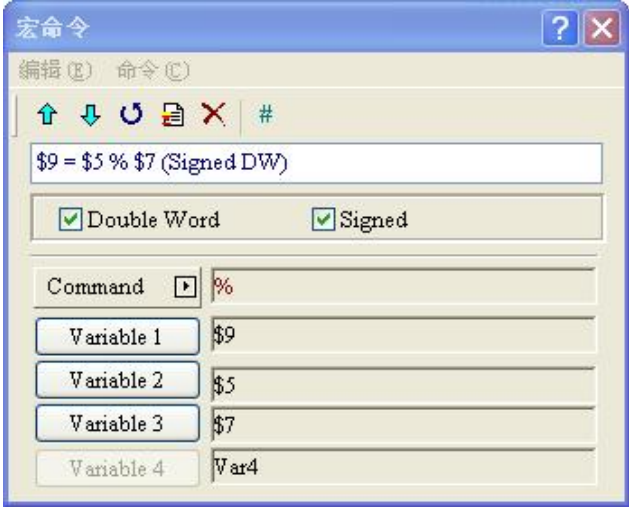
| | |
|---|--|
| <p>将内部存储区\$4 与\$6 的值相除后存放到\$8 (有符号数 32 位运算)</p> |  |
|---|--|

■ % (余数运算)

| 句型 | 代表意义 | 备注 |
|-----------------------|--------------------------|--|
| $Var1 = Var2 \% Var3$ | Var2 和 Var3 相除的余数存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 ➤ Var3 不可为 0。 |

范例:

| | |
|---|--|
| <p>将内部存储区\$0 内的值除以 2, 所得的余数存放在\$1 (无符号数 16 位运算)</p> |  |
|---|--|

| | |
|--|---|
| <p>将内部存储区\$2 与\$3 内的值相除，所得的余数存放在\$4（有符号数 16 位运算）</p> |  |
| <p>将内部存储区\$5 与\$7 内的值相除，所得的余数存放在\$9（有符号数 32 位运算）</p> |  |

■ MUL64（64 位乘法运算）

| 句型 | 代表意义 | 备注 |
|--------------------------------------|----------------------------------|--|
| <p>Var1 = MUL64 (Var2, Var3)</p> | <p>Var2 和 Var3 相乘运算结果存在 Var1</p> | <ul style="list-style-type: none"> ➢ 此运算式固定以 32 位来作运算。 ➢ 计算结果可以存放成有号或无符号数的 DOUBLE WORD。 ➢ Var1 固定占四个 Word，Var2 与 Var3 皆固定占两个 Word。 ➢ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|---|---|
| <p>将内部存储区\$0 与\$2 的值相乘后存放到\$4 (无符号数 32 位运算)</p> |  |
| <p>将内部存储区\$8 与\$10 的值相乘后存放到\$12 (有符号数 32 位运算)</p> |  |

■ ADDSUMW (累加)

| 句型 | 代表意义 | 备注 |
|--|--|--|
| <p>Var1 = ADDSUMW (Var2, Var3)</p> | <p>从 Var2 开始连续累加 Var3 长度的地址, 并将结果存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以无符号数来作运算。 ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1、Var2 只能是内部存储区, Var3 可以是内部存储区或是常数。 |

范例:

| | |
|--|---|
| <p>将内部存储区\$0、\$1、\$2、\$3 与\$4 的值相加后存放到\$5（无符号数 16 位运算）</p> |  |
| <p>将内部存储区\$6、\$8 与\$10 的值相加后存放到\$12（无符号数 32 位运算）</p> |  |

■ FADD（浮点加法运算）

| 句型 | 代表意义 | 备注 |
|-------------------------------------|----------------------------------|--|
| <p>Var1 = FADD (Var2, Var3)</p> | <p>Var2 和 Var3 相加运算结果存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|---|---|
| <p>将内部存储区\$0 加上 1.0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区\$2 与\$4 的值相加后存放到\$6 (有符号数 32 位运算)</p> |  |

■ FSUB (浮点减法运算)

| 句型 | 代表意义 | 备注 |
|-------------------------------------|----------------------------------|--|
| <p>Var1 = FSUB (Var2, Var3)</p> | <p>Var2 和 Var3 相减运算结果存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|--|---|
| <p>将内部存储区\$0 减掉 1.0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区\$2 与\$4 的 值相减后存放到\$6 (有符号数 32 位运算)</p> |  |

■ FMUL (浮点乘法运算)

| 句型 | 代表意义 | 备注 |
|-------------------------------------|---------------------------------------|--|
| <p>Var1 = FMUL (Var2, Var3)</p> | <p>Var2 和 Var3 相乘运算 结果存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|---|---|
| <p>将内部存储区\$0 乘上 2.0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区\$2 与\$4 的值相乘后存放到\$6 (有符号数 32 位运算)</p> |  |

■ FDIV (浮点除法运算)

| 句型 | 代表意义 | 备注 |
|-------------------------------------|---------------------------------------|---|
| <p>Var1 = FDIV (Var2, Var3)</p> | <p>Var2 和 Var3 相除运算 结果存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 ➤ Var3 不可为 0。 |

范例:

| | |
|---|---|
| <p>将内部存储区\$0 除上 2.0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区\$2 与\$4 的值相除后存放到\$6 (有符号数 32 位运算)</p> |  |

■ FMOD (浮点余数运算)

| 句型 | 代表意义 | 备注 |
|-------------------------------------|---------------------------------|---|
| <p>Var1 = FMOD (Var2, Var3)</p> | <p>Var2 和 Var3 相除的余数存在 Var1</p> | <ul style="list-style-type: none"> ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 ➤ Var3 不可为 0。 |

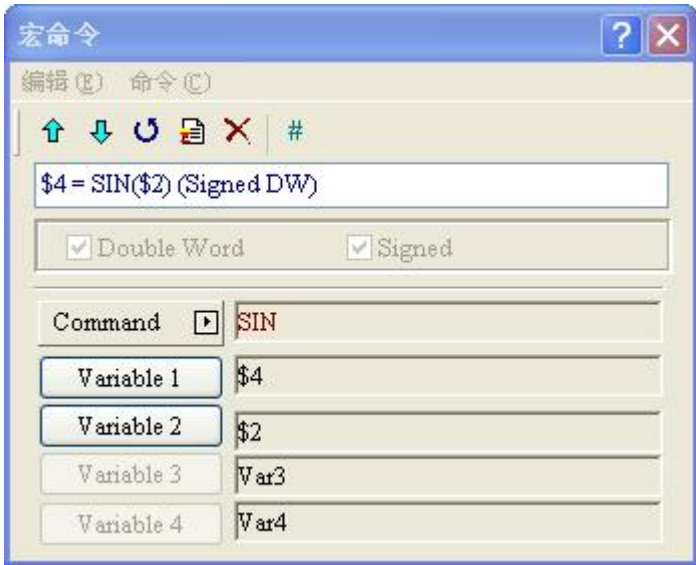
范例:

| | |
|--|---|
| <p>将内部存储区\$0 内的值除以 2.0，所得之余数存放在\$2（有符号数 32 位运算）</p> |  |
| <p>将内部存储区\$4 与\$6 内的值相除，所得之余数存放在\$8（有符号数 32 位运算）</p> |  |

■ SIN（正弦函数运算）

| 句型 | 代表意义 | 备注 |
|--------------------------|----------------------------------|---|
| <p>Var1 = SIN (Var2)</p> | <p>将 Var2 内的值作正弦运算后，存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |

范例：

| | |
|--|---|
| <p>将 SIN30° 的值存放在内部存储区\$0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区\$2 内的值作正弦运算后, 存放在\$4 (有符号数 32 位运算)</p> |  |

■ COS (余弦函数运算)

| 句型 | 代表意义 | 备注 |
|--------------------------|-----------------------------------|---|
| <p>Var1 = COS (Var2)</p> | <p>将 Var2 内的值作余弦运算后, 存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |



范例:

| | |
|---|---|
| <p>将 $\text{COS}30^\circ$ 的值存放在内部存储区 \$0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区 \$2 内的值作余弦运算后, 存放在 \$4 (有符号数 32 位运算)</p> |  |

■ TAN (正切函数运算)

| 句型 | 代表意义 | 备注 |
|---|-----------------------------------|---|
| <p>$\text{Var1} = \text{TAN}(\text{Var2})$</p> | <p>将 Var2 内的值作正切运算后, 存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |

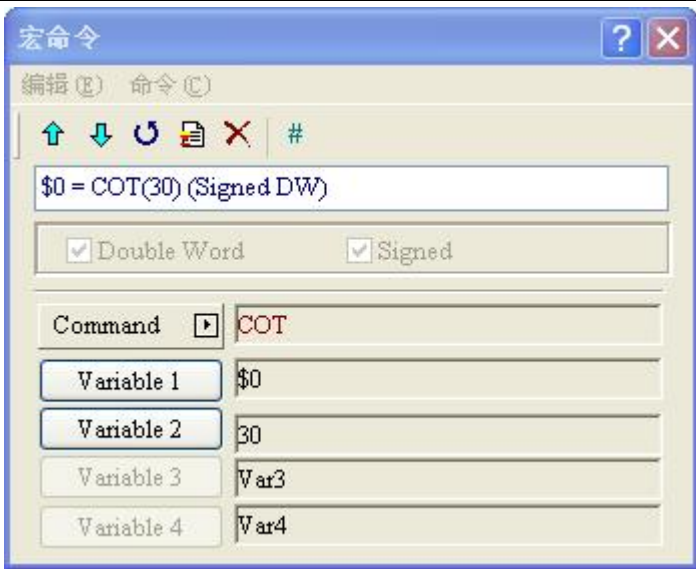
范例:

| | |
|--|---|
| <p>将 TAN30° 的值存放在内部存储区\$0（有符号数 32 位运算）</p> |  |
| <p>将内部存储区\$2 内的值作正切运算后，存放在\$4（有符号数 32 位运算）</p> |  |

■ COT（余切函数运算）

| 句型 | 代表意义 | 备注 |
|--------------------------|----------------------------------|---|
| <p>Var1 = COT (Var2)</p> | <p>将 Var2 内的值作余切运算后，存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |



范例：

| | |
|--|---|
| <p>将 $COT30^\circ$ 的值存放在内部存储区 \$0 (有符号数 32 位运算)</p> |  |
| <p>将内部存储区 \$2 内的值作余切运算后, 存放在 \$4 (有符号数 32 位运算)</p> |  |

■ SEC (正割函数运算)

| 句型 | 代表意义 | 备注 |
|--------------------------|-----------------------------------|---|
| <p>Var1 = SEC (Var2)</p> | <p>将 Var2 内的值作正割运算后, 存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD, 溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区, Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |

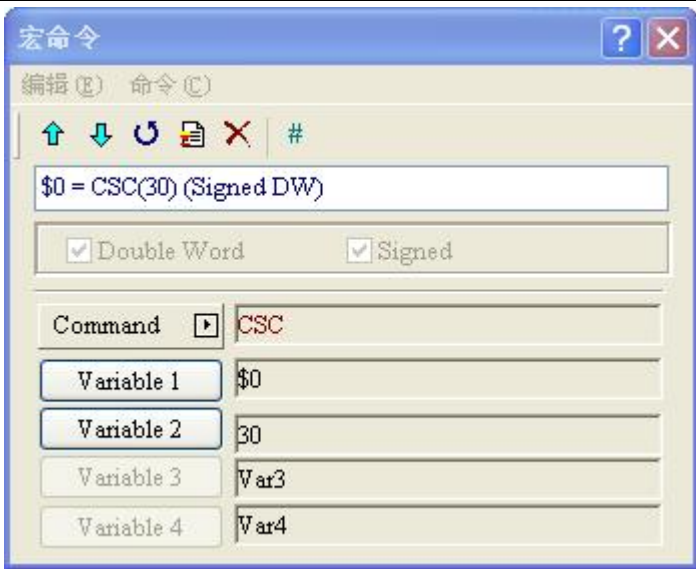
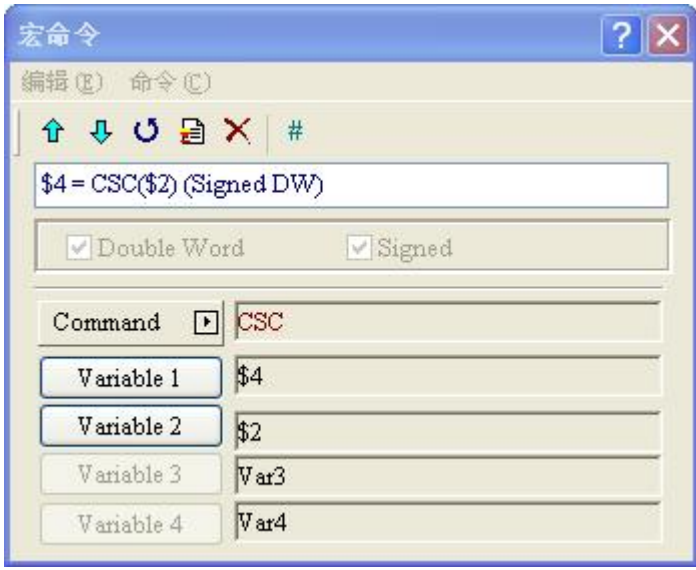
范例:

| | |
|--|---|
| <p>将 SEC30° 的值存放在内部存储区\$0（有符号数 32 位运算）</p> |  |
| <p>将内部存储区\$2 内的值作正割运算后，存放在\$4（有符号数 32 位运算）</p> |  |

■ CSC（余割函数运算）

| 句型 | 代表意义 | 备注 |
|--------------------------|----------------------------------|---|
| <p>Var1 = CSC (Var2)</p> | <p>将 Var2 内的值作余割运算后，存放在 Var1</p> | <ul style="list-style-type: none"> ➤ Var2 物理意义为角度而非弧度。 ➤ 此运算式固定以有符号数 32 位来作运算。 ➤ 计算结果可以存放成有符号数的 DOUBLE WORD，溢位的部分将会被忽略。 ➤ Var1 只能是内部存储区，Var2 可以是内部存储区或是常数。 ➤ 显示元件变数格式必为 Floating。 ➤ 输入元件变数必为 Signed Decimal 且不可以有小数点。 |

范例：

| | |
|--|---|
| <p>将 CSC30° 的值存放在内部存储区\$0（有符号数 32 位运算）</p> |  |
| <p>将内部存储区\$2 内的值作余割运算后，存放在\$4（有符号数 32 元运算）</p> |  |

3.14.3.2 逻辑运算

| |
|-----|
| |
| && |
| ^ |
| NOT |
| << |
| >> |

程序中提供六种逻辑运算子，其详细说明如下：

■ | (OR 逻辑运算)

| 句型 | 代表意义 | 备注 |
|--------------------|----------------------------------|---|
| Var1 = Var2 Var3 | 将 Var2 与 Var3 作 OR 运算后，结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|--|---|
| <p>将\$0 与\$1 作 OR 运算后, 结果存在\$2 (无符号数 16 位运算)</p> <p>若设定 \$0 = F000H, \$1 = 0F00H, 则\$2 = FF00H</p> |  |
| <p>将\$3 与\$5 作 OR 运算后, 结果存在\$7 (无符号数 32 位运算)</p> <p>若设定 \$3 = F000F000H, \$1 = 0F000F00H, 则\$2 = FF00FF00H</p> |  |

■ && (AND 逻辑运算)

| 句型 | 代表意义 | 备注 |
|---------------------|------------------------------------|--|
| Var1 = Var2 && Var3 | 将 Var2 与 Var3 作 AND 运算后, 结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

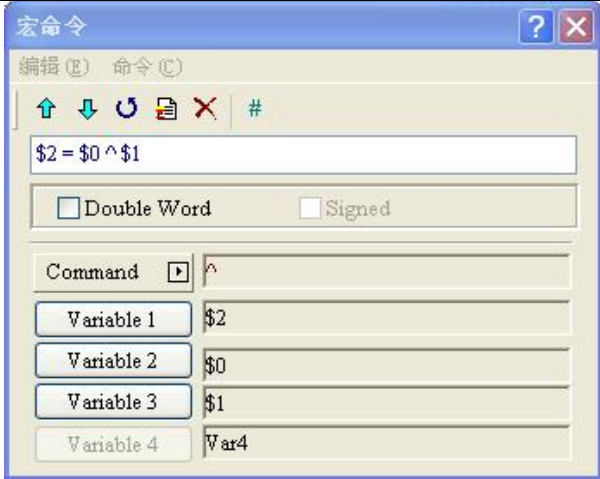
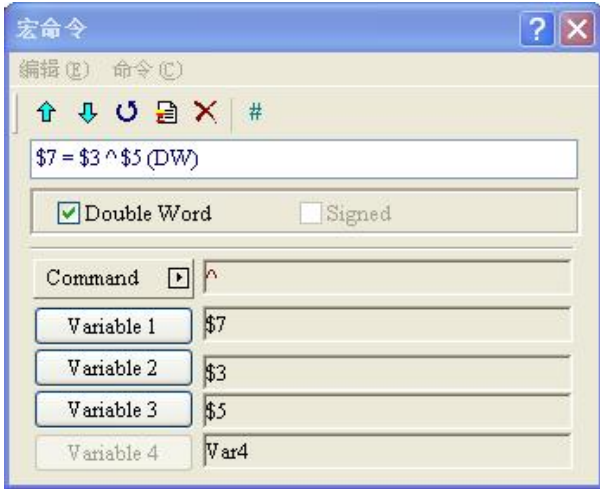
范例:

| | |
|---|---|
| <p>将\$0 与\$1 作 AND 运算后, 结果存在\$2 (无符号数 16 位运算)</p> <p>若设定 \$0 = F000H, \$1 = 0F00H, 则\$2 = 0000H</p> |  |
| <p>将\$3 与\$5 作 AND 运算后, 结果存在\$7 (无符号数 32 位运算)</p> <p>若设定 \$3 = F000F000H, \$1 = 0F000F00H, 则\$2 = 00000000H</p> |  |

■ ^ (XOR 逻辑运算)

| 句型 | 代表意义 | 备注 |
|--------------------|------------------------------------|--|
| Var1 = Var2 ^ Var3 | 将 Var2 与 Var3 作 XOR 运算后, 结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|---|---|
| <p>将\$0与\$1作XOR运算后,结果存在\$2(无符号数16位运算)</p> <p>若设定 \$0 = F100H, \$1 = 0F00H, 则\$2 = FE00H</p> |  |
| <p>将\$3与\$5作XOR运算后,结果存在\$7(无符号数32位运算)</p> <p>若设定 \$3 = F100F100H, \$1 = 0F000F00H, 则\$2 = FE00FE00H</p> |  |

■ NOT (NOT 逻辑运算)

| 句型 | 代表意义 | 备注 |
|-----------------|----------------------------|---|
| Var1 = NOT Var2 | 将 Var2 作 NOT 运算后,结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有号或无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区,Var2 可以是内部存储区或是常数。 |

范例:

| | |
|---|--|
| <p>将\$0作NOT运算后,结果存在\$1(无符号数16位运算)</p> <p>若设定\$0 = F100H, 则\$1 = 0EFFH</p> |  |
|---|--|

| | |
|---|--|
| <p>将\$2 作 NOT 运算后，结果存在\$4（有符号数 32 位运算）</p> <p>若设定 \$2 = F100 F100H, 则\$4 = 0EFF 0EFFH</p> |  |
|---|--|

■ << (SHL 逻辑运算)

| 句型 | 代表意义 | 备注 |
|---------------------|-----------------------------|---|
| Var1 = Var2 << Var3 | 左移 Var2 的位 Var3 个，并存放于 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|--|--|
| <p>将\$0 左移 4 个位后，结果存在\$1（无符号数 16 位运算）</p> <p>若设定\$0 = F100H，则\$1 = 1000H</p> |  |
|--|--|

| | |
|---|--|
| <p>将\$2 左移 4 个位后，结果存在\$4（无符号数 32 位运算）</p> <p>若设定 \$2 = F1000000H，则\$4 = 10000000H</p> |  |
|---|--|

■ >> (SHR 逻辑运算)

| 句型 | 代表意义 | 备注 |
|---------------------|-----------------------------|---|
| Var1 = Var2 >> Var3 | 右移 Var2 的位 Var3 个，并存放于 Var1 | <ul style="list-style-type: none"> ▶ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ▶ 如果位移位数超过暂存器大小，超出暂存器的部分就会被忽略。 ▶ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|---|--|
| <p>将\$0 右移 4 个位后，结果存在\$1（无符号数 16 位运算）</p> <p>若设定\$0 = F100H，则\$1 = 0F10H</p> |  |
| <p>将\$2 右移 4 个位后，结果存在\$4（无符号数 32 位运算）</p> <p>若设定 \$2 = F1000000H，则\$4 = 0F100000H</p> |  |

3.14.3.3 数据搬移

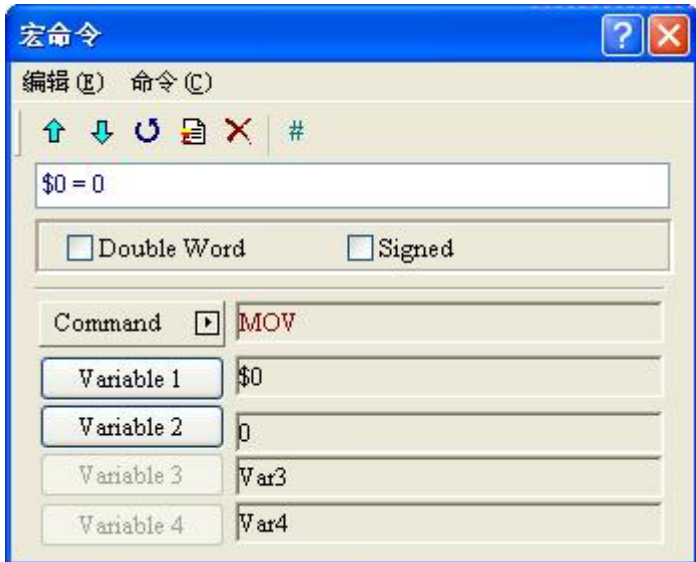
| |
|---------|
| MOV |
| BMOV |
| FILL |
| FILLASC |
| FMOV |

程序中提供五种数据搬移指令，其详细说明如下：

■ MOV (资料指定运算元)

| 句型 | 代表意义 | 备注 |
|-------------|---|---|
| Var1 = Var2 | 将 Var2 的数据复制到 Var1 中，而 Var2 的数据不会因指令而改变 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号或无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 可以是控制器暂存器或内部存储区，Var2 可以是内部存储区或是常数。 |

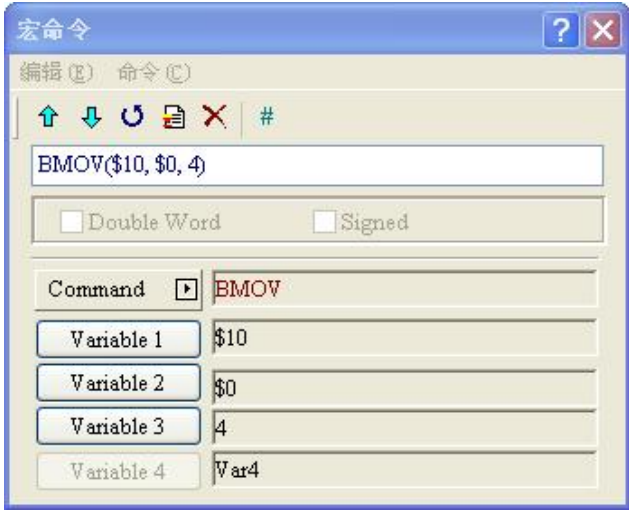
范例：

| | |
|--------------------------------------|--|
| <p>设定\$0 的值为 0 (无符号数 16 位运算)</p> |  |
| <p>设定\$1 的值为 99999 (有符号数 32 位运算)</p> |  |

■ BMOV (复制区块)

| 句型 | 代表意义 | 备注 |
|-------------------------|--|--|
| BMOV (Var1, Var2, Var3) | 从 Var2 的地址上复制 Var3 个 Word 资料到 Var1 地址上 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ 区块的长度如果超过内部存储区或是控制器地址的最大值，编译将不会通过。 ➤ Var1、Var2 可以是控制器寄存器或内部存储区，Var3 可以是内部存储区或是常数。 |

范例：

| | |
|--|---|
| <p>将\$0, \$1, \$2, \$3 四个寄存器内的数据依序储存在\$10, \$11, \$12, \$13 四个寄存器内。(无符号数 16 位运算)</p> |  |
|--|---|

■ FILL (填充区块)

| 句型 | 代表意义 | 备注 |
|-------------------------|-----------------------------------|---|
| FILL (Var1, Var2, Var3) | 从 Var1 的地址开始依序存放 Var2 的数值共 Var3 个 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号或无符号数的 WORD。 ➤ 区块的长度如果超过内部存储区或是控制器地址的最大值，编译将不会通过。 ➤ Var1 只能是内部存储区，Var2、Var3 可以是内部存储区或是常数。 |

范例：

| | |
|---|--|
| <p>将\$0, \$1, \$2, \$3, \$4 五个寄存器内的数据填入 0。（无符号数 16 位运算）</p> |  |
|---|--|

■ FILLASC (将文字转为 ASCII 数值)

| 句型 | 代表意义 | 备注 |
|-----------------------------------|--|---|
| <p>FILLASC (Var1, "Var2")</p> | <p>将 Var2 字符串每个字元依序转成 ASCII 数值放在 Var1 的地址上</p> | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ 字符串的长度最大为 128 个字元。 ➤ 一个寄存器最多只可存放两个字元，多出的字元依序往下一个寄存器存放，转换后的 ASCII 数值高低位会对调。 ➤ Var1 只能是内部存储区，Var2 只能是输入字符串。 |

范例：

| | |
|--|---|
| <p>将 AB12 字符串每个字元依序转成 ASCII 数值放在 \$0 的地址上（无符号数 16 位运算）</p> <p>运算结果 \$0 = 4241H \$1 = 3130H</p> |  |
|--|---|

■ FMOV (浮点数值资料指定)

| 句型 | 代表意义 | 备注 |
|--------------------|--|--|
| Var1 = FMOV (Var2) | 将 Var2 的浮点数值资料复制到 Var1 中, 而 Var2 的资料不会因指令而改变 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号数的 DOUBLE WORD。 ➤ Var1 可以是控制器寄存器或内部存储区, Var2 可以是控制器寄存器、内部存储区或是常数。 |

范例:

| | |
|--|--|
| <p>指定内部存储区\$0 的地址放常数 44.3 的资料 (有符号数 32 位运算)</p> |  |
| <p>指定内部存储区\$2 储存一份与 PLC C200 相同的资料 (有符号数 32 位运算)</p> |  |

3.14.3.4 数据转换

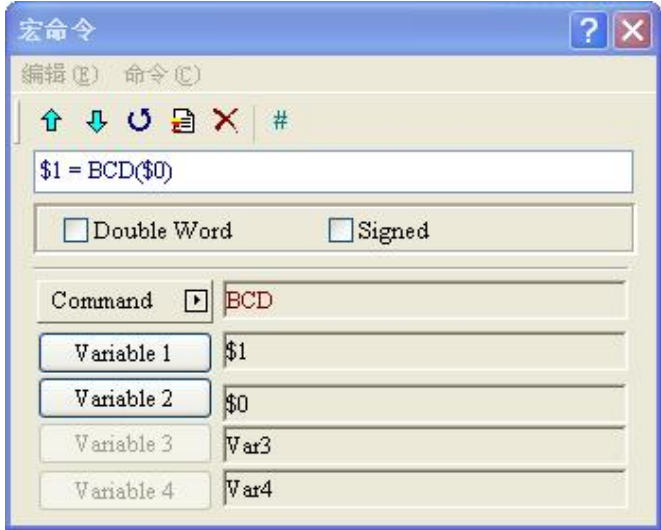
| | |
|---------|-------|
| BCD | XCHG |
| BIN | MAX |
| TODWORD | MIN |
| TOWORD | TOHEX |
| TOBYTE | TOASC |
| SWAP | FCNV |
| | ICNV |

程序中提供数种数据转换宏指令，其详细说明如下：

■ BCD（十进制数值转换为 BCD 格式的数值）

| 句型 | 代表意义 | 备注 |
|-------------------|------------------------------------|---|
| Var1 = BCD (Var2) | 将 Var2 内的十进制数值转换为 BCD 格式数值存放于 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1、Var2 只能是内部存储区。 |

范例：

| | |
|---|---|
| <p>将\$0 内的十进制数值转换为 BCD 格式数值并存放至\$1（无符号数 16 位运算）</p> |  |
|---|---|

■ BIN（BCD 格式数值转换为十进制数值）

| 句型 | 代表意义 | 备注 |
|-------------------|---------------------------------|---|
| Var1 = BIN (Var2) | 将 Var2 BCD 格式数值转换为十进制数值存放于 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1、Var2 只能是内部存储区。 |

范例：

将\$0 内的 BCD 数值转换为十进位格式数值并存放至\$1（无符号数 16 位运算）

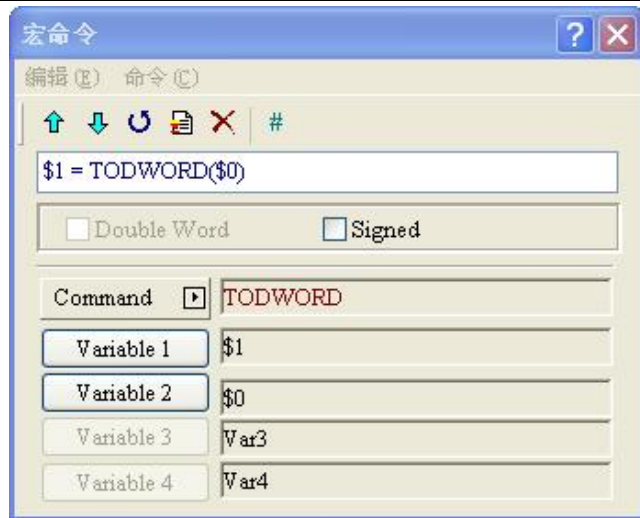


■ TODWORD（将数值从 WORD 转为 DOUBLE WORD）

| 句型 | 代表意义 | 备注 |
|-----------------------|--|--|
| Var1 = TODWORD (Var2) | 将 WORD 整数值 Var2 转换为一个 DOUBLE WORD 整数值存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号或无符号数的 DOUBLE WORD。 ➤ Var1、Var2 只能是内部存储区。 |

范例：

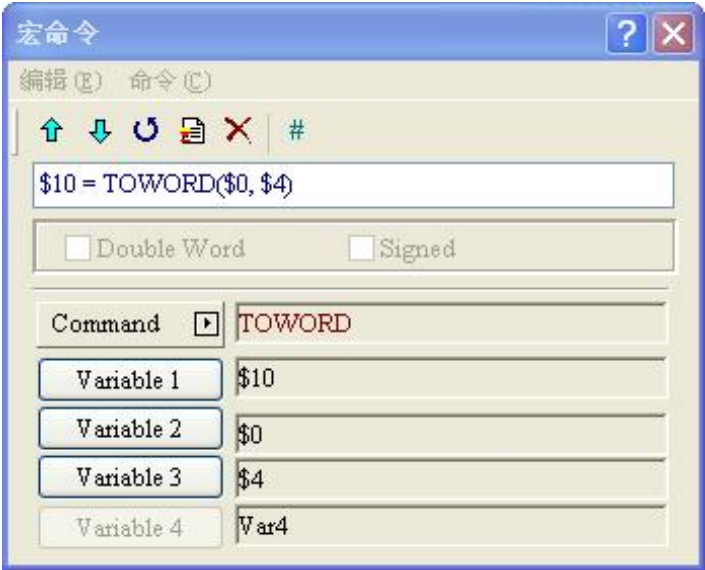
将\$0 内的 Word 整数值转换成 Double Word 整数值并存放至\$1，此时因为已经转成 Double Word 格式，所以实际占用了\$1、\$2 两个寄存器。（无符号数 16 位运算）



■ TOWORD (将数值从 BYTE 转为 WORD)

| 句型 | 代表意义 | 备注 |
|-------------------------------|--|--|
| Var1 = TOWORD (Var2, Var3) | 从 Var2 开始连续 Var3 个 BYTE 转换为 WORD 数值, 转换结果存在 Var1, 且每个 Var1 的高字节补 0 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区, Var3 可以是内部存储区或是常数。 ➤ 由于 Var2 的单位为 WORD, 因此对于 Var2 每一个 WORD 可以转成 2 个 WORD。 ➤ 数据转换后, 转换前后的 WORD 高低字节会对调。 |

范例:

| | |
|---|---|
| <p>从\$0 开始连续 4 个 BYTE 转换为 WORD 数值, 转换结果存在\$10。(无符号数 16 位运算)</p> <p>\$0 = AB12H, \$1 = 34CDH, \$10 = B2W (\$0, 4), 从\$0 读取 2 个 WORD 也就是 4 个 BYTE 的资料, 转换成 4 个 WORD 存放在 \$10~\$13, 所以得到的资料为</p> <p>\$10 = 12H, \$11 = ABH, \$12 = CDH, \$13 = 34H</p> |  |
|---|---|

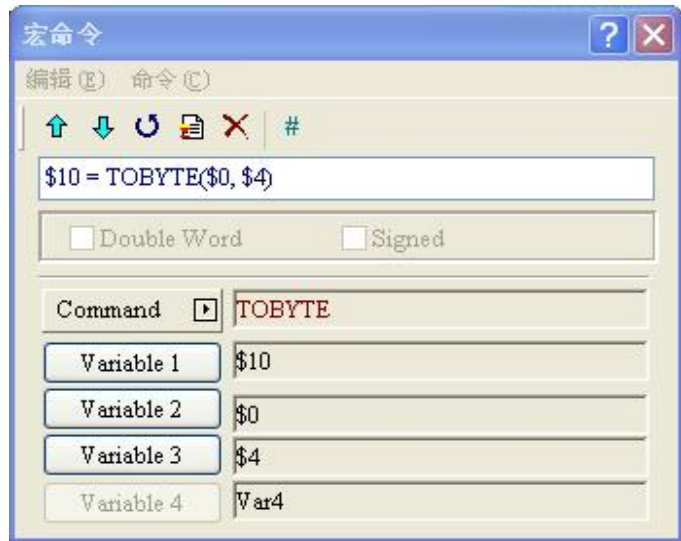
■ TOBYTE (WORD 转换为 BYTE 格式的数值)

| 句型 | 代表意义 | 备注 |
|-------------------------------|--|--|
| Var1 = TOBYTE (Var2, Var3) | 从 Var2 的低字节开始连续 Var3 个 WORD 转换为 BYTE 格式的数值, 且忽略 Var2 的高字节, 转换结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区, Var3 可以是内部存储区或是常数。 ➤ 数据转换后, 转换前后的 WORD 高低字节会对调。 |

范例:

从\$0 开始连续 4 个 BYTE 转换为 WORD 数值，转换结果存在\$10。（无符号数 16 位运算）

```
$0 = 1234H,
$1 = 5678H,
$2 = ABH,
$3 = CDH,
$10 = W2B ($10, 4),
先从$0 读取 4 个 WORD 转换为 4 个 BYTE 格式的数值，转换成 2 个 WORD 存放在$10~$11，所以得到的资料为
$10 = 7834H,
$11 = CDABH
```



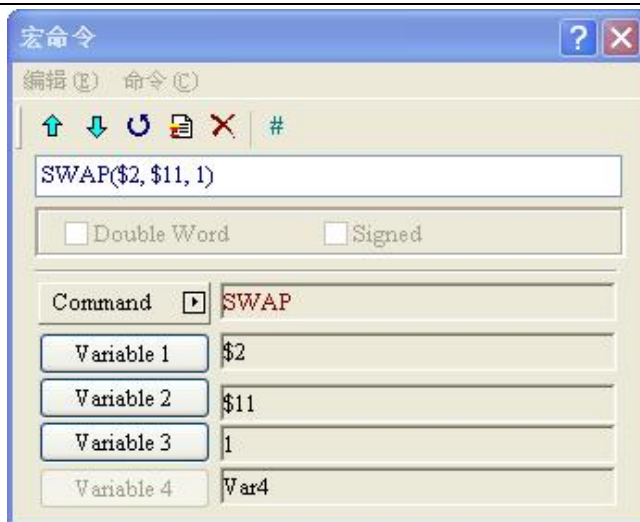
■ SWAP (对调 WORD 高低字节)

| 句型 | 代表意义 | 备注 |
|-------------------------|---|---|
| SWAP (Var1, Var2, Var3) | 开始依序 Var2, Var2+1, Var2+2... Var2+Var3 每个 WORD 高字节和低字节的资料内容对调并存储于 Var1, Var1+1, Var1+2...Var1+Var3 开始的位置 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区，Var3 可以是内部存储区或是常数。 |

范例：

对调\$11 内的高低字节并存放至\$2。（无符号数 16 位运算）

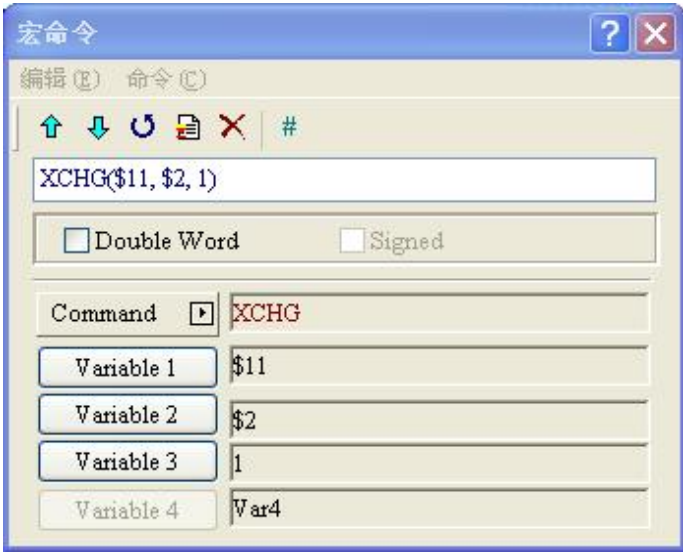
```
SWAP ($2, $11, 1)
假如$11 = 1234H,
经由运算会得到
$2 = 3412H
```



■ XCHG (数值资料对调)

| 句型 | 代表意义 | 备注 |
|----------------------------|---|--|
| XCHG (Var1, Var2, Var3) | 将 Var2, Var2+1, Var2+2..., Var2+Var3 的资料跟 Var1, Var1+1, Var1+2..., Var1+Var3 资料内容对调 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区, Var3 可以是内部存储区或是常数。 ➤ 本指令执行完成后 Var1, Var2 的资料内容将被改变。 |

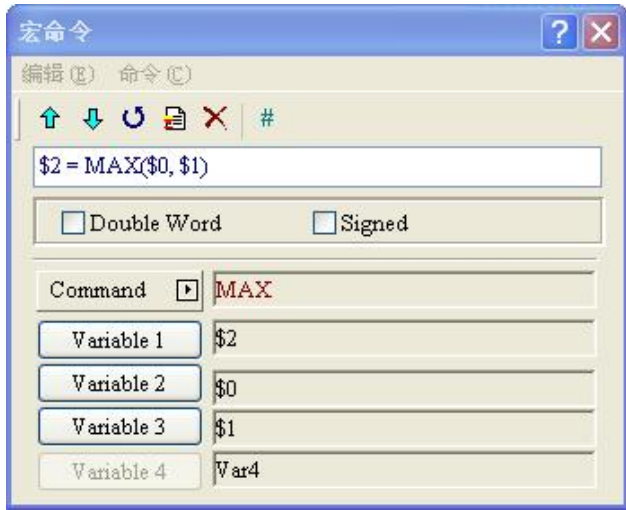
范例:

| | |
|--|---|
| <p>对调\$11 与\$2 的资料内容。(无符号数 16 位运算)</p> <p>XCHG (\$2, \$11, 1)</p> <p>假如</p> <p>\$11 = 1234H,</p> <p>\$2 = 5678H</p> <p>经由运算会得到</p> <p>\$11 = 5678H,</p> <p>\$2 = 1234H</p> |  |
|--|---|

■ MAX (求最大值)

| 句型 | 代表意义 | 备注 |
|----------------------------|-----------------------------|--|
| Var1 = MAX (Var2, Var3) | 从 Var2、Var3 求最大值, 结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号或无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

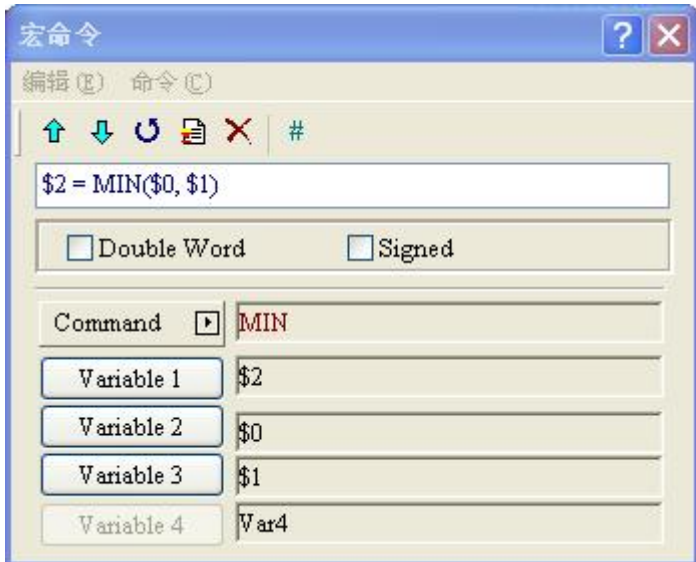
范例:

| | |
|--|---|
| <p>从\$0、\$1 求最大值, 并将结果存在\$2。(无符号数 16 位运算)</p> <p>若\$0 = 2, \$1 =10, 则运算结果为\$2 = 10</p> |  |
|--|---|

■ MIN (求最小值)

| 句型 | 代表意义 | 备注 |
|----------------------------|-----------------------------|--|
| Var1 = MIN (Var2, Var3) | 从 Var2、Var3 求最小值, 结果存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号或无符号数的 WORD 或 DOUBLE WORD。 ➤ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或是常数。 |

范例:

| | |
|---|--|
| <p>从\$0、\$1 求最小值, 并将结果存在\$2。(无符号数 16 位运算)</p> <p>若\$0 = 2, \$1 =10, 则运算结果为\$2 = 2</p> |  |
|---|--|

■ TOHEX (4 个 ASCII 字元转为 WORD 的 16 进制的整数)

| 句型 | 代表意义 | 备注 |
|---------------------|---|---|
| Var1 = TOHEX (Var2) | 从 Var2 开始的 4 个 WORD 的 ASCII 字元转换为整数后存在 Var1 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区。 |

范例：

| | |
|---|---|
| <p>从\$0 开始的 4 个 WORD 的 ASCII 字元转换为整数后存在\$10。(无符号数 16 位运算)</p> <p>\$0 = 0034H, \$1 = 0033H, \$2 = 0036H, \$3 = 0038H, \$10 = A2H (\$0) , 经由以上运算\$10 得数值会变成 4368H</p> |  |
|---|---|

■ TOASC (将 1 WORD 的 16 进制的整数转换为 4 WORD 的 ASCII 字元)

| 句型 | 代表意义 | 备注 |
|---------------------|---|---|
| Var1 = TOASC (Var2) | 从 Var2 的 1 个 WORD 的 16 进位整数转换后存入 Var1 开始的 4 个 WORD 的 ASCII 字元 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区。 |

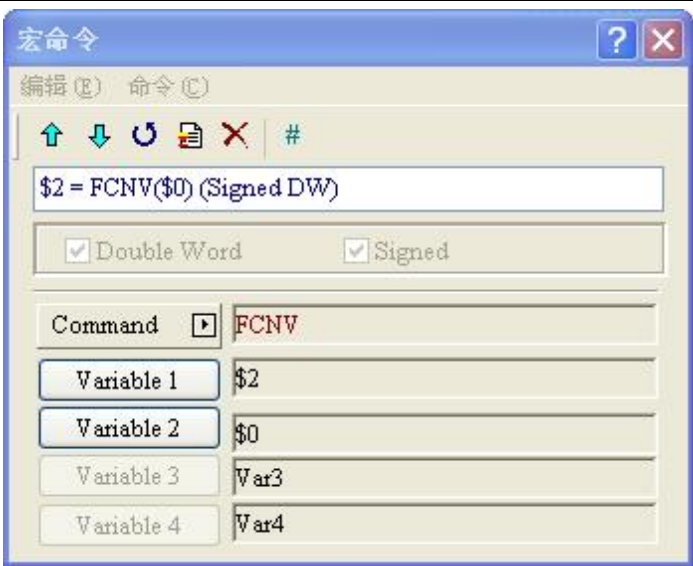
范例:

| | |
|--|---|
| <p>从\$0 的 1 个 WORD 的 16 进位整数转换后存入\$10 开始的 4 个 WORD 的 ASCII 字元。(无符号数 16 位运算)</p> <p>\$0 = 1234H, \$10 = H2A (\$0) , 经由以上运算 \$10 = 0031H, \$11 = 0032H, \$12 = 0033H, \$13 = 0034H</p> |  |
|--|---|

■ FCNV (整数转换浮点数)

| 句型 | 代表意义 | 备注 |
|--------------------|--------------------------|---|
| Var1 = FCNV (Var2) | 将 Var2 整数或浮点数转为 Var1 浮点数 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号数的 DOUBLE WORD。 ➤ Var1 只能是内部存储区, Var2 可以是内部存储区或常数。 |

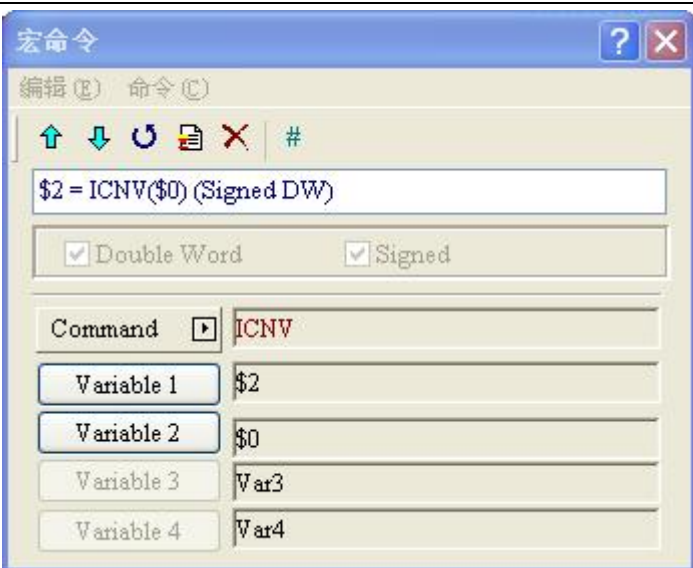
范例:

| | |
|---|--|
| <p>将\$0 整数或浮点数转为浮点数，并存在\$2。（有符号数 32 位运算）</p> <p>若\$0 = 100，则\$2 = 100.0</p> |  |
|---|--|

■ ICNV（浮点数转换整数）

| 句型 | 代表意义 | 备注 |
|--------------------|-------------------------|--|
| Var1 = ICNV (Var2) | 将 Var2 整数或浮点数转为 Var1 整数 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成有符号数的 DOUBLE WORD。 ➤ Var1 只能是内部存储区，Var2 可以是内部存储区或常数。 |

范例：

| | |
|--|--|
| <p>将\$0 整数或浮点数转为整数，并存在\$2。（有符号数 32 位运算）</p> <p>若\$0 = 100.0，则\$2 = 100</p> |  |
|--|--|

3.14.3.5 比较

| | |
|------------------|---|
| IF ... THEN GOTO | ▶ |
| IF ... | ▶ |
| ELSEIF ... | ▶ |
| ELSE | |
| ENDIF | |
| FCMP | |

程序中提供数种比较宏指令，其详细说明如下：

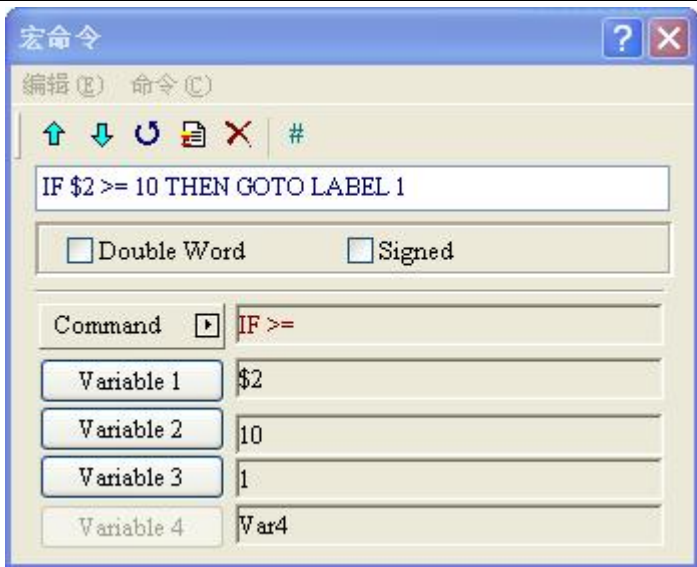
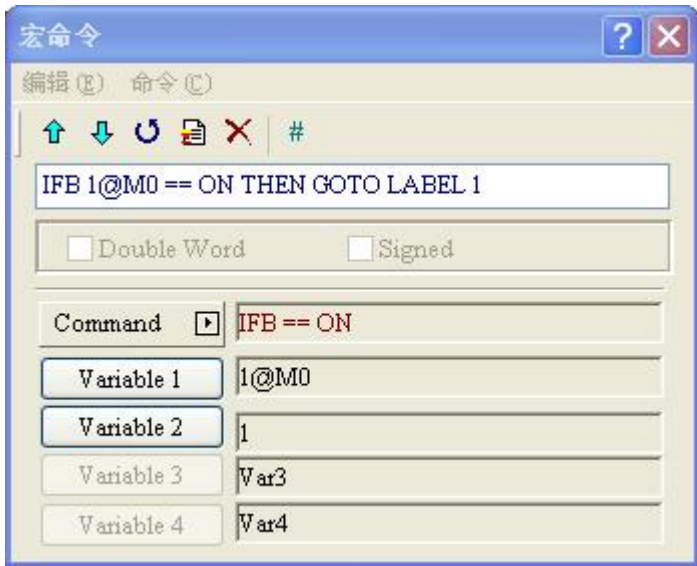
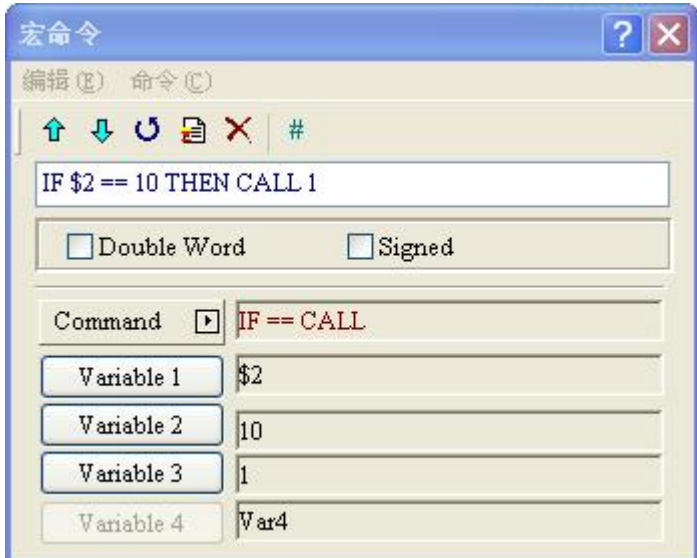
■ IF...THEN GOTO ...

| 句型 | 代表意义 | 备注 |
|---|---|------------------------|
| IF <i>expression</i> THEN GOTO LABEL <i>identifier</i> | 如果 <i>expression</i> 为真就跳到 <i>identifier</i> 的程序段执行 | ➤ Expression 表示法请参阅下表。 |
| IFB <i>expression</i> THEN GOTO LABEL <i>identifier</i> | 如果 <i>expression</i> 为真就跳到 <i>identifier</i> 的程序段执行 (PLC 寄存器使用) | |
| IF <i>expression</i> THEN CALL <i>sub-macro</i> | 如果 <i>expression</i> 为真就呼叫某子宏执行 | ➤ Expression 表示法请参阅下表。 |

Expression 表示法：

| 句型 | 代表意义 | 备注 |
|-------------------|---------------------------|-------------------------------|
| Var1 == Var2 | Var1 等于 Var2 | Var1, Var2 可以为内部存储区或常数。 |
| Var1 != Var2 | Var1 不等于 Var2 | |
| Var1 > Var2 | Var1 大于 Var2 | |
| Var1 >= Var2 | Var1 大于等于 Var2 | |
| Var1 < Var2 | Var1 小于 Var2 | |
| Var1 <= Var2 | Var1 小于等于 Var2 | |
| Var1 && Var2 == 0 | Var1 与 Var2 做 AND 运算等于 0 | |
| Var1 && Var2 != 0 | Var1 与 Var2 做 AND 运算不等于 0 | Var1 可能是 PLC 寄存器或内部存储区 (BIT)。 |
| Var1== ON | Var1 为 ON | |
| Var1== OFF | Var1 为 OFF | |

范例：

| | |
|--|---|
| <p>如果\$2 大于等于 10 就跳到 LABEL 1 的地方继续执行程序</p> |  <p>The screenshot shows the 'Macro Command' dialog box. The command text is 'IF \$2 >= 10 THEN GOTO LABEL 1'. The 'Command' dropdown is set to 'IF >='. The variables are: Variable 1: \$2, Variable 2: 10, Variable 3: 1, Variable 4: Var4. There are checkboxes for 'Double Word' and 'Signed', both of which are unchecked.</p> |
| <p>如果 1@M0 为 ON 就跳到 LABEL 1 的地方继续执行程序</p> |  <p>The screenshot shows the 'Macro Command' dialog box. The command text is 'IFB 1@M0 == ON THEN GOTO LABEL 1'. The 'Command' dropdown is set to 'IFB == ON'. The variables are: Variable 1: 1@M0, Variable 2: 1, Variable 3: Var3, Variable 4: Var4. There are checkboxes for 'Double Word' and 'Signed', both of which are unchecked.</p> |
| <p>如果\$2 等于 10 就呼叫子宏 1</p> |  <p>The screenshot shows the 'Macro Command' dialog box. The command text is 'IF \$2 == 10 THEN CALL 1'. The 'Command' dropdown is set to 'IF == CALL'. The variables are: Variable 1: \$2, Variable 2: 10, Variable 3: 1, Variable 4: Var4. There are checkboxes for 'Double Word' and 'Signed', both of which are unchecked.</p> |

■ IF...ELSEIF...ELSE...ENDIF

| 句型 | 代表意义 | 备注 |
|--|---|----------------------|
| IF <i>expression1</i> <i>Statement1</i> ELSEIF <i>expression2</i> <i>Statement2</i> ELSE <i>Statement3</i> ENDIF | 多重情况的逻辑判断，如果 <i>expression1</i> 成立就执行 <i>Statement1</i> ，反之则再判断 <i>expression2</i> ，若成立就执行 <i>Statement2</i> ，若都不成立则执行 <i>Statement3</i> ，而 IF 循环结尾都需要加上 ENDIF 作为结束 | Expression 表示法请参阅下表。 |

Expression 表示法：

| 句型 | 代表意义 | 备注 |
|-------------------|---------------------------|------------------------|
| Var1 == Var2 | Var1 等于 Var2 | Var1、Var2 可以为内部存储区或常数。 |
| Var1 != Var2 | Var1 不等于 Var2 | |
| Var1 > Var2 | Var1 大于 Var2 | |
| Var1 >= Var2 | Var1 大于等于 Var2 | |
| Var1 < Var2 | Var1 小于 Var2 | |
| Var1 <= Var2 | Var1 小于等于 Var2 | |
| Var1 && Var2 == 0 | Var1 与 Var2 做 AND 运算等于 0 | |
| Var1 && Var2 != 0 | Var1 与 Var2 做 AND 运算不等于 0 | Var1 只能是内部存储区 (BIT)。 |
| Var1== ON | Var1 为 ON | |
| Var1== OFF | Var1 为 OFF | |

范例：

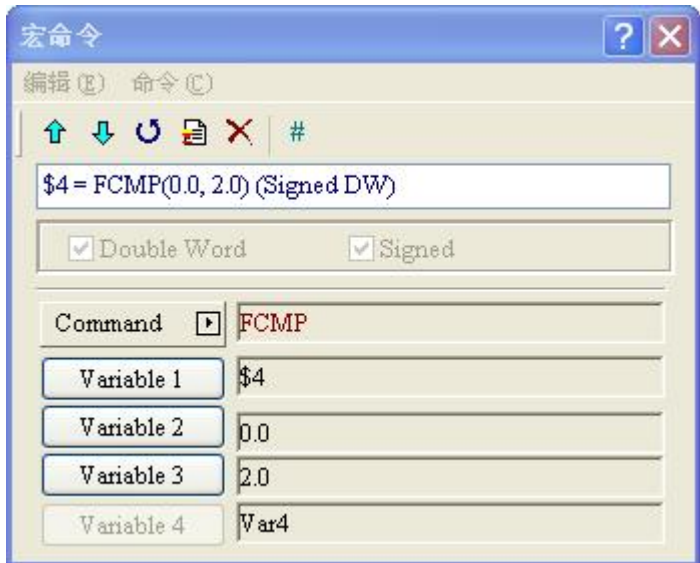
| | | |
|--|----------------|--------------|
| <pre> IF \$0 == 0 \$10 = 0 ELSEIF \$0 == 1 \$10 = 1 ELSE \$10 = 2 ENDIF </pre> | IF \$0 == 0 | 若\$0为0， |
| | \$10 = 0 | 则\$10为0； |
| | ELSEIF \$0 ==1 | 若\$0不为0而为1， |
| | \$10 = 1 | 则\$10为1； |
| | ELSE | 若\$0不为0也不为1， |
| | \$10 = 2 | 则\$10为2 |
| | ENDIF | |

■ FCMP (浮点数数值比较)

| 句型 | 代表意义 | 备注 |
|----|------|----|
|----|------|----|

| | | | |
|-----------------------------|--|-------------------|--|
| Var1 = FCMP (Var2, Var3) | 比较 Var2 与 Var3 的大小 并将 结果 存在 Var1, 而比较方式如下所示: | | ▶ 计算结果可以存放成有符号数的 DOUBLE WORD。 ▶ Var1 只能是内部存储区, Var2、Var3 可以是内部存储区或常数。 |
| | Var1 | FCMP (Var2, Var3) | |
| | 0 | Var2 = Var3 | |
| | 1 | Var2 > Var3 | |
| | 2 | Var2 < Var3 | |

范例:

| | |
|---------------------------------------|---|
| 比较\$0 与\$2 的大小并将结果存在\$4。(有符号数 32 位运算) |  |
|---------------------------------------|---|

3.14.3.6 流程控制


```
GOTO
LABEL
CALL
RET
FOR
NEXT
END
```

程序中提供数种流程控制宏指令，其详细说明如下：

■ GOTO (无条件跳到某个标签 (Label))

| 句型 | 代表意义 | 备注 |
|-----------------|---------------------------|-------------|
| GOTO LABEL Var1 | 无条件跳到该程序内部所指定的 LABEL Var1 | Var1 只能是常数。 |

范例：

| | |
|---------------------------------|--|
| <p>无条件跳跃到 LABEL 1 的位置继续执行程序</p> |  |
|---------------------------------|--|

■ LABEL (标签)

| 句型 | 代表意义 | 备注 |
|------------|------|---|
| LABEL Var1 | 标签代码 | <ul style="list-style-type: none"> ➤ Var1 只能是常数。 ➤ 同一宏程序内部不能够有相同的标签代码 Var1。 |

范例:

| | |
|----------------|--|
| <p>LABEL 1</p> | |
|----------------|--|

■ CALL..RET (呼叫子宏)

| 句型 | 代表意义 | 备注 |
|-----------|-----------|--|
| CALL Var1 | 呼叫子宏 Var1 | ➤ Var1 表示子宏程序的编号从 1 到 512, 且可以是内部存储区或常数。 |
| RET | 离开子宏 | ➤ 须加在子宏末。 |

范例:

| | |
|--|--|
| <p>呼叫子宏会将宏执行控制权跳到子宏 Var1。子宏执行结束后, 须由 RET 指令返回, RET 指令将宏执行控制权跳回到 CALL 指令的下一行指令继续执行。</p> | |
|--|--|

■ FOR...NEXT (程序循环)

| 句型 | 代表意义 | 备注 |
|--------------------------------------|---------------------------|--|
| FOR Var1 <i>Statement</i> NEXT | 将连续执行 Var1 次 Statement | ➤ 可以使用多层循环, 最多可以 3 层。 ➤ Var1 可为常数或内部存储区。 ➤ Statement 为一段宏指令集合。 |

范例:

| | |
|--------------------------------------|-----------------------|
| <pre>FOR 10 \$0 = \$0 + 1 NEXT</pre> | <p>将\$0 累加 10 次 1</p> |
|--------------------------------------|-----------------------|

■ END (结束宏程序)

| 句型 | 代表意义 | 备注 |
|---------------------------------------|-------|---|
| $Statements1$ END $Statements2$ | 结束宏程序 | <ul style="list-style-type: none"> ➤ $Statements1$ 执行完之后不会继续执行 $Statements2$，下次宏执行时会重新从 $Statements1$ 第一行指令开始。 ➤ Statement 为一段宏指令集合。 ➤ END 指令代表宏程序结束，如果在子宏中使用 END 指令，宏指令执行到 END 时便全部结束，不会在跳回原先的宏，若要跳回原先的宏，请使用 RET 指令。 |

范例:

| | |
|---|--|
| <pre> \$1 = 10 \$1 = \$1 + 1 END \$1 = \$1 + 1 </pre> | 经由运算 \$1 = 11，不是 \$1 = 12，因为 END 指令将宏结束。 |
|---|--|

3.14.3.7 位设置

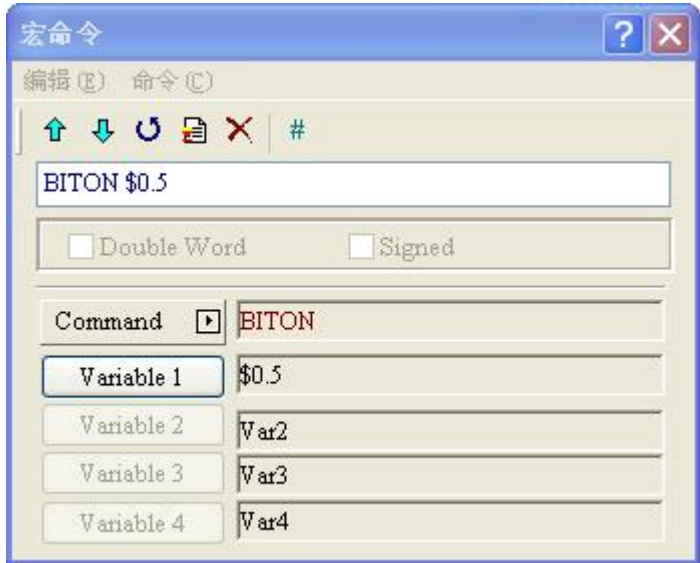
| |
|-----------------------------------|
| BITON BITOFF BITNOT GETB |
|-----------------------------------|

程序中提供数种位设置宏指令，其详细说明如下：

■ BITON (设定为 ON)

| 句型 | 代表意义 | 备注 |
|------------|---------------|--------------------------------|
| BITON Var1 | 设定 Var1 位为 ON | ➤ Var1 可为 PLC 寄存器或内部存储区 (BIT)。 |

范例:

| | |
|--|--|
| <p>设定内部存储区 \$0 的第 5 位为 ON</p> <p>若 \$0 = 0000000000000000, 经运算后结果为 0000000000010000</p> |  |
|--|--|

■ BITOFF (设定位为 OFF)

| 句型 | 代表意义 | 备注 |
|-------------|----------------|--------------------------------|
| BITOFF Var1 | 设定 Var1 位为 OFF | ➤ Var1 可为 PLC 寄存器或内部存储区 (BIT)。 |


范例:

| | |
|--|--|
| <p>设定内部存储区\$0 的第 5 位为 OFF</p> <p>若\$0 = 1111111111111111, 经运算后结果为 1111111111101111</p> |  <p>The screenshot shows a dialog box titled '宏命令' (Macro Command) with a menu bar '编辑(E) 命令(C)'. It contains a command input field with 'BITOFF \$0.5', two checkboxes for 'Double Word' and 'Signed', a 'Command' dropdown set to 'BITOFF', and four variable input fields: 'Variable 1' (\$0.5), 'Variable 2' (Var2), 'Variable 3' (Var3), and 'Variable 4' (Var4).</p> |
|--|--|

■ BITNOT (反相位, ON 变 OFF, OFF 变 ON)

| 句型 | 代表意义 | 备注 |
|-------------|---------------------------------|--------------------------------|
| BITNOT Var1 | 反相 Var1 位, ON 变为 OFF, OFF 变为 ON | ➤ Var1 可为 PLC 寄存器或内部存储区 (BIT)。 |

范例:

| | |
|---|--|
| <p>将内部存储区\$0 的第 5 位作 BITNOT 反向操作</p> <p>若\$0 = 1111111111111111, 经运算后结果为 1111111111101111</p> |  <p>The screenshot shows a dialog box titled '宏命令' (Macro Command) with a menu bar '编辑(E) 命令(C)'. It contains a command input field with 'BITNOT \$0.5', two checkboxes for 'Double Word' and 'Signed', a 'Command' dropdown set to 'BITNOT', and four variable input fields: 'Variable 1' (\$0.5), 'Variable 2' (Var2), 'Variable 3' (Var3), and 'Variable 4' (Var4).</p> |
|---|--|

■ GETB (取得位值)

| 句型 | 代表意义 | 备注 |
|------------------|--------------------|-----------------------------------|
| Var1 = GETB Var2 | 取得 Var2 的位值放到 Var1 | Var1、Var2 可为 PLC 暂存器或内部存储区 (BIT)。 |

范例:

| | |
|---|--|
| <p>将内部存储区\$0 的第 5 位取出并存放至\$1 第 5 位</p> <p>若 \$0 = 111111111111111111, \$1 = 0000000000000000, 经运算后结果为 \$1 = 000000000010000</p> | |
|---|--|

3.14.3.8 通讯

- INITCOM
- ADDSUM
- XORSUM
- PUTCHARS
- GETCHARS
- SELECTCOM
- CLEARCOMBUFFER
- CHRCHKSUM
- LOCKCOM
- UNLOCKCOM
- STATIONON
- STATIONOFF

程序中提供数种通讯宏指令，其详细说明如下：

■ INITCOM (COM PORT 的初始化)

| 句型 | 代表意义 | 备注 |
|-----------------------|-----------------------------------|---|
| Var1 = INITCOM (Var2) | COM PORT 的初始化, 用来开启通讯端口, 并且设定通讯协议 | <ul style="list-style-type: none"> ➤ Var1 为回传值, 成功为 1, 失败为 0。 ➤ Var2 设定方式如下表所示。 |

Var2 设定方式:

| 选项 | 种类 | 数字代号 | 备注 |
|--------|-----------------------|------|---------------|
| 通讯端口 | COM1 | 0 | |
| | COM2 | 1 | |
| | COM3 | 2 | |
| 通讯界面 | RS232 | 0 | |
| | RS422 | 1 | |
| | RS485 | 2 | |
| 数据位 | 7 Bits | 0 | |
| | 8 Bits | 1 | |
| 校验位 | None | 0 | |
| | Odd | 1 | |
| | Even | 2 | |
| 停止位 | 1 Bits | 0 | |
| | 2 Bits | 1 | |
| 波特率 | 300 | 0 | |
| | 600 | 1 | |
| | 900 | 2 | |
| | 1200 | 3 | |
| | 2400 | 4 | |
| | 4800 | 5 | |
| | 9600 | 6 | |
| | 14400 | 7 | |
| | 19200 | 8 | |
| | 28800 | 9 | |
| | 38400 | 10 | |
| | 57600 | 11 | |
| 115200 | 12 | | |
| 流控制 | No Flow Control | 0 | 流控制注意事项如下表所示。 |
| | CTS RTS Flow Control | 1 | |
| | DTR DSR Flow Control | 2 | |
| | Xon Xoff Flow Control | 3 | |

流控制注意事项

No Flow Control

不加以设定。

Flow Control

流控制，当使用串行端口传输数据时，由于即时压缩、纠错等新的传输处理技术，使通讯的速度和正确性大幅提高，但也使电脑和人机间资料传输的速度往往会远大于之间真正的数据传送速度，为确保数据安全及完整传送于电脑与人机之间，因此要有传送流量的控制。

CTS / RTS

为硬件流量控制，由硬件产生的电气脉波经汇流排至内接式数据机或是由连接线至外接式数据机来达成流量控制。

DSR / DTR

也是硬件流量控制，用于电脑跟人机以电缆直接连线。

XON / XOFF

为软件流量控制，通常只用于 2400bps 之 Modem 中，控制方式是由软件产生控制码，并将其加在传送的资料之中。

范例：

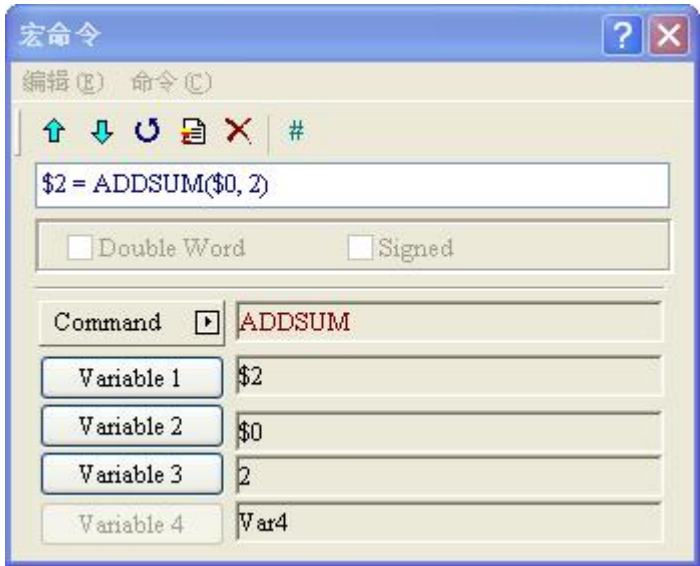
若设定通讯端口、通讯界面、数据位、校验位、停止位、波特率及流量控制为 COM1、RS232、7、Even、1、9600 及 No Flow Control，且将回传值存于 \$0，则设定值如右图所示。



■ ADDSUM (利用加法算出 CHECKSUM)

| 句型 | 代表意义 | 备注 |
|-------------------------------|--|---|
| Var1 = ADDSUM (Var2, Var3) | 利用加法，算出 CHECKSUM。Var1 为计算后的值，Var2 为计算数据的起始地址，Var3 为数据的长度 | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区，Var3 可以是内部存储区或是常数。 |

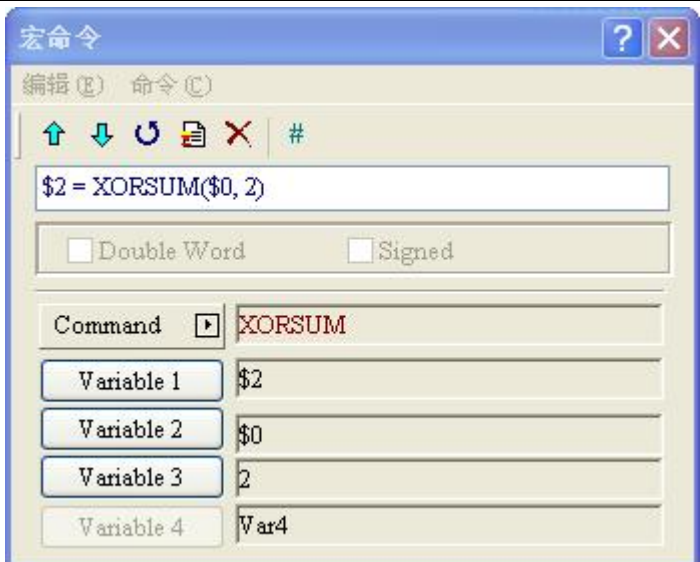
范例:

| | |
|--|--|
| <p>从\$0 开始，长度为 2，利用加法计算 CHECKSUM 并存在\$2（无符号数 16 位运算）</p> |  |
|--|--|

■ XORSUM（利用 XOR 算出 CHECKSUM）

| 句型 | 代表意义 | 备注 |
|-----------------------------------|--|---|
| <p>Var1 = XORSUM (Var2, Var3)</p> | <p>利用 XOR，算出 CHECKSUM, Var1 为计算后的值, Var2 为计算数据的起始地址, Var3 为数据的长度</p> | <ul style="list-style-type: none"> ➤ 计算结果可以存放成无符号数的 WORD。 ➤ Var1、Var2 只能是内部存储区，Var3 可以是内部存储区或是常数。 |

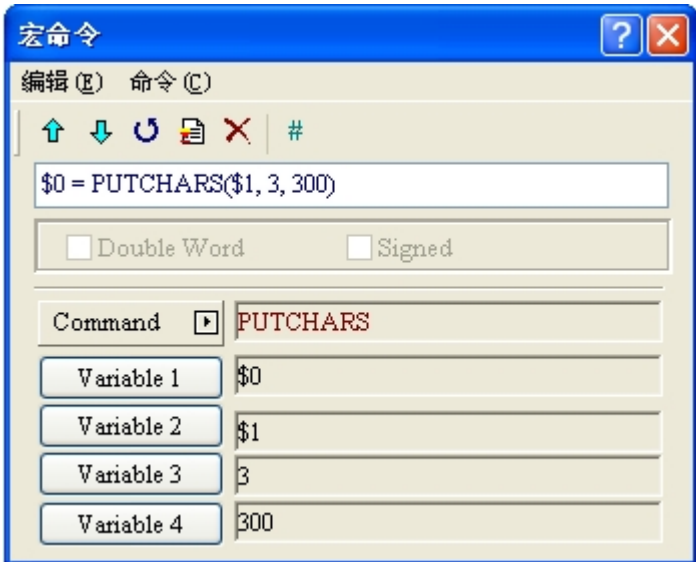
范例:

| | |
|---|--|
| <p>从\$0 开始，长度为 2，利用 XOR 计算 CHECKSUM 并存在\$2（无符号数 16 位运算）</p> |  |
|---|--|

■ PUTCHARS (经由通讯端口输出字)

| 句型 | 代表意义 | 备注 |
|---------------------------------------|---|--|
| Var1 = PUTCHARS (Var2, Var3, Var4) | 经由通讯端口输出字， Var1 为通讯后回传的值， 通讯成功回传 1，通讯失败回传 0， Var2 为传输数据的起始地址， Var3 为资料的长度， Var4 为所允许的最大通讯时间， 其单位为千分之一秒 (ms) | ▶ Var1、Var2 只能是内部存储区， Var3 Var4 可以是内部存储区或是常数。 |

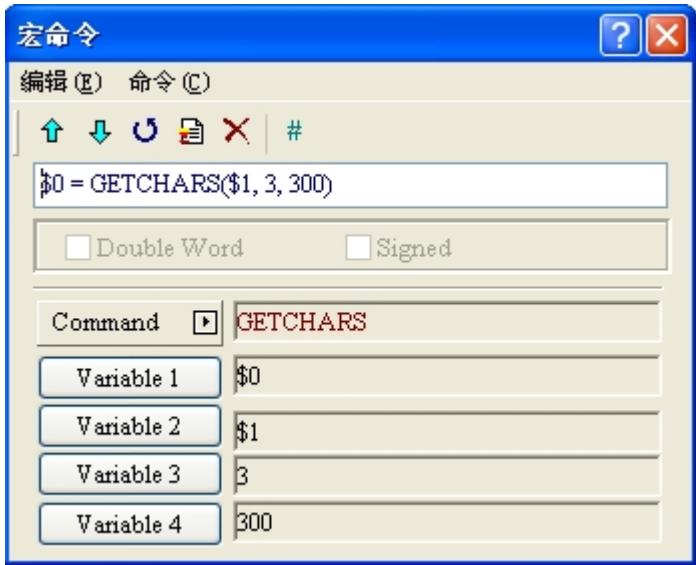
范例：

| | |
|--|---|
| <p>输出从\$1 开始的连续三个 WORD 数据，并将通讯结果存在\$0。</p> |  |
|--|---|

■ GETCHARS (经由通讯端口得到字)

| 句型 | 代表意义 | 备注 |
|---------------------------------------|---|--|
| Var1 = GETCHARS (Var2, Var3, Var4) | 经由通讯端口得到字， Var1 为通讯后回传的值， 通讯成功回传 1，通讯失败回传 0， Var2 为传输数据的起始地址， Var3 为数据的长度， Var4 为所允许的最大通讯时间， 其单位为千分之一秒 (ms) | ▶ Var1、Var2 只能是内部存储区， Var3 Var4 可以是内部存储区或是常数。 |

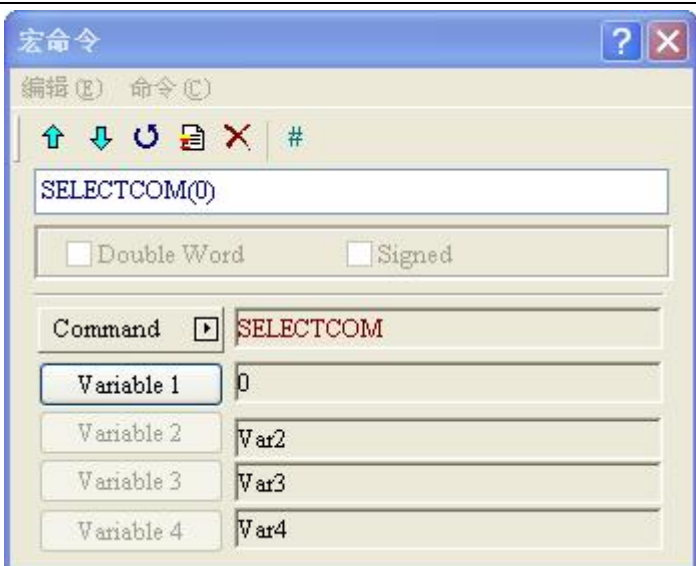
范例:

| | |
|--|--|
| <p>将从通讯端口所得到的资料，存放在从\$1开始的连续三个 WORD，并将通讯结果存在\$0。</p> |  |
|--|--|

■ SELECTCOM (选择通讯端口)

| 句型 | 代表意义 | 备注 |
|------------------|--|---|
| SELECTCOM (Var1) | 经由此项指令，来选定要使用哪一个通讯端口，0 代表 COM1，1 代表 COM2，2 代表 COM3【部分机种支持】 | <ul style="list-style-type: none"> ➢ Var1 只能是常数。 ➢ 使用宏通讯时，指定的通讯端口不能与系统本身所使用的通讯端口冲突所占用。所有的通讯指令将针对所切换的通讯端口作处理，所以不同的宏之间的通讯端口切换并不会互相支持或是干扰。 |

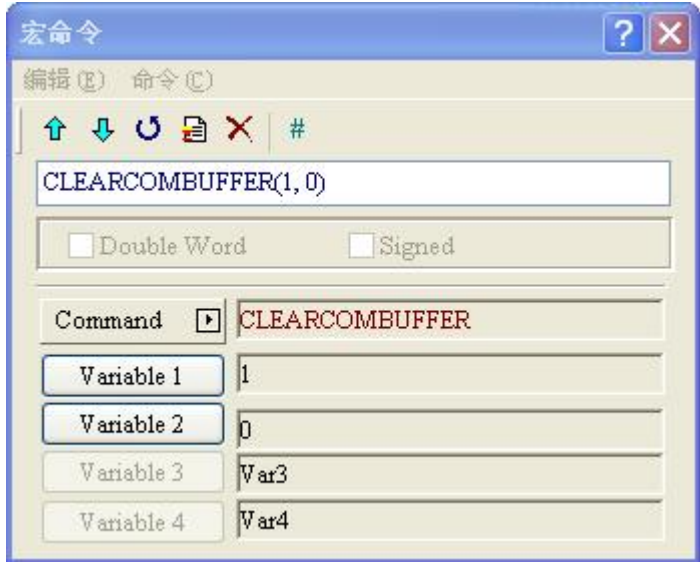
范例:

| | |
|-----------------|--|
| <p>选择 COM1。</p> |  |
|-----------------|--|

■ CLEARCOMBUFFER (清除哪一个通讯端口的缓冲区)

| 句型 | 代表意义 | 备注 |
|--------------------------------|--|------------------|
| CLEARCOMBUFFER (Var1, Var2) | 清除哪一个通讯端口的缓冲区。Var1 为通讯端口代号, 以常数 0 (COM1) 或 1 (COM2) 或 2 (COM3) 【部分机种支持】表示; Var2 为缓冲区类别, 以常数 0 (接收缓冲区) 或 1 (传送缓冲区) 表示 | Var1、Var2 只能是常数。 |

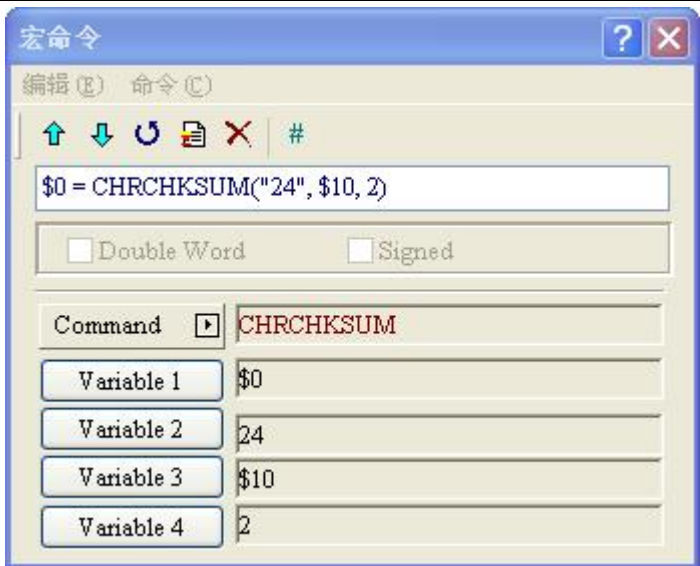
范例:

| | |
|----------------|---|
| 清除 COM2 接收缓冲区。 |  |
|----------------|---|

■ CHRCHKSUM (计算字串的长度和 checksum)

| 句型 | 代表意义 | 备注 |
|---|---|---|
| Var1 = CHRCHKSUM (“Var2”, Var3, Var4) | 计算字串的长度和 checksum。Var2 表示字串, Var3 表示存放 Var2 字串的 checksum 的内部存储区地址, Var4 表示 Var3 存放 Checksum 数据长度 1 代表 Byte, 2 代表 WORD | Var1、Var3 只能是内部存储区, Var2 只能是字串, Var4 只能是常数。 |

范例:

| | |
|---|--|
| <p>计算"24"的字符串长度与 checksum。</p> <p>经由运算\$0 存放 4 表示字符串长度 2 Word, \$10 存放字符串"24"的 ASCII 码 3432, \$11 存放 checksum 值为 66H。</p> |  |
|---|--|

■ LOCKCOM / UNLOCKCOM (锁定 COM Port / 解除 COM Port)

| 句型 | 代表意义 | 备注 |
|------------------------------------|--|--|
| <p>Var1 = LOCKCOM (Var2, Var3)</p> | <p>锁定 COM Port。Var1 为锁定 COM Port 后回传的值，锁定成功回传 1，锁定失败回传 0，Var2 为 COM PORT, 0 为 COM1, 1 为 COM2, 2 为 COM3, Var3 为 time out 值 (单位为 ms)，当到达指定的 time out 值仍无法锁住 COM 时就会回传错误，若设成 0 表示无穷等待</p> | <p>▶ Var1 只能是内部存储区，Var2、Var3 只能是常数。</p> <p>▶ 当通讯指令分布在画面 Cycle 宏、Clock 宏、Background 宏、元件本身宏、画面 On/Off 宏时，便会造成数据互抢的问题，避免的方法就是在通讯指令前后加上 LOCKCOM 与 UNLOCKCOM 宏，其目的就是维持同一时间下，不可以让一段通讯被中断跑去做其他通讯，而保持通讯内容的前后完整性。LOCKCOM 使用范例示意图如下表所示。</p> <p>▶ 若 LOCKCOM 设为无穷等待时 (Var3 = 0)，则在同一宏内连续执行两次 LOCKCOM 将使人机无法回应，此点务必注意。</p> |
| <p>UNLOCKCOM (Var2)</p> | <p>解除 COM Port。Var2 为 COM PORT, 0 为 COM1, 1 为 COM2, 2 为 COM3</p> | |

LOCKCOM 使用范例示意图

| | | |
|---|--|---|
| <p>Background 宏</p> <pre>\$0 = LOCKCOM(0,0) \$1 = PUTCHARS(\$1, 3, 300) UNLOCKCOM(0)</pre> | <p>元件 On 宏</p> <pre>\$0 = LOCKCOM(0,0) \$1 = GETCHARS(\$1, 3, 300) UNLOCKCOM(0)</pre> | <p>画面 Cycle 宏</p> <pre>\$0 = LOCKCOM(0,0) \$1 = PUTCHARS(\$1, 3, 300) UNLOCKCOM(0)</pre> |
|---|--|---|

如上列图示所示，三个宏内皆有通讯指令进行，若此时 Background 宏先行执行，则此时 COM 1 已被锁定，元件 On 宏与画面 Cycle 宏皆是执行到 LUCKCOM (0,0) 这一行时就停住，直到 Background 宏 UNLOCKCOM 为止，此时再执行元件 On 宏或画面 Cycle 宏。如此即可避免数据互抢或误收问题产生。

UNLOCKCOM 使用范例示意图

| | |
|--|---|
| <p>Background 宏</p> <pre>\$0 = LOCKCOM(0,0) \$1 = PUTCHARS(\$1, 3, 300)</pre> | <p>元件 On 宏</p> <pre>UNLOCKCOM(0)</pre> |
|--|---|

如上列图示所示，用户可在 Background 宏内锁定 COM Port 与传输数据，在元件 On 宏内解锁 COM Port，意即锁定 COM Port 与解锁 COM Port 是可以分开写的。


范例：

| | |
|---------------------------|---|
| LOCKCOM 与 UNLOCKCOM 使用范例。 | <pre>\$0 = LOCKCOM(0, 200) \$1 = PUTCHARS(\$1, 3, 300) \$2 = GETCHARS(\$1, 3, 300) UNLOCKCOM(0)</pre> |
|---------------------------|---|

■ STATIONON (站号启动)

| 句型 | 代表意义 | 备注 |
|------------------------|--|--|
| STATIONON (Var1, Var2) | 开启 COM Var1 的第 Var2 站，此时人机便可与该站控制器通讯。其中 Var1 以常数 0 (COM1) 或 1 (COM2) 或 2(COM3)【部分机种支持】表示 | <ul style="list-style-type: none"> ➤ Var1, Var2 可以是内部存储区或是常数。 ➤ 此宏无法与 [设置模块参数] - [通讯] 中的” 通讯中断后取消连线” 功能同时使用。 |

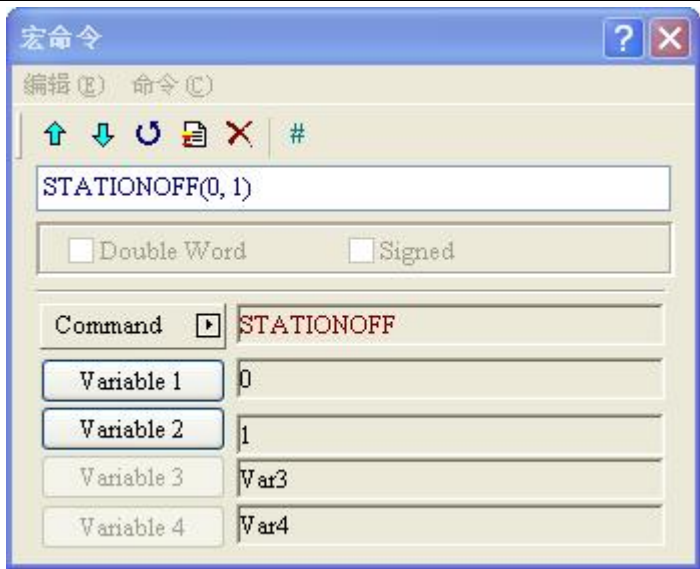
范例:

| | |
|---------------|--|
| 开启 COM1 的第一站。 |  |
|---------------|--|

■ STATIONOFF (站号关闭)

| 句型 | 代表意义 | 备注 |
|-------------------------|---|--|
| STATIONOFF (Var1, Var2) | 关闭 COM Var1 的第 Var2 站，此时人机便无法与该站控制器通讯。其中 Var1 以常数 0 (COM1) 或 1 (COM2) 或 2 (COM3) 【部分机种支持】表示 | Var1, Var2 可以是内部存储区或是常数。 此宏无法与 [设置模块参数] - [通讯] 中的” 通讯中断后取消连线” 功能同时使用。 |

范例:

| | |
|---------------|--|
| 关闭 COM1 的第一站。 |  |
|---------------|--|

3.14.3.9 绘图

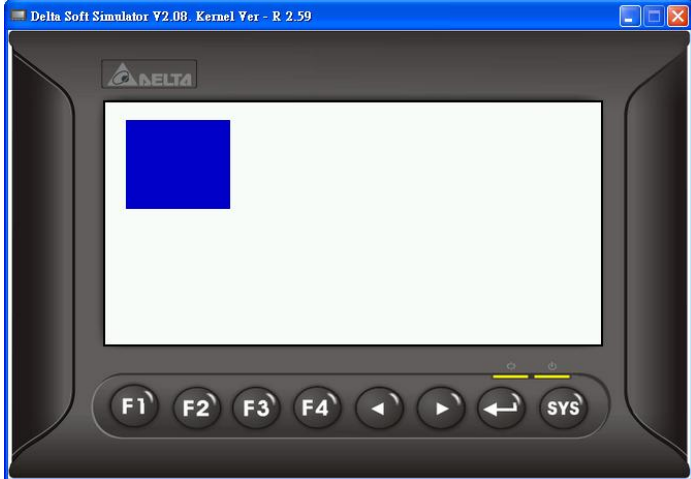
| |
|-----------|
| RECTANGLE |
| LINE |
| POINT |
| CIRCLE |

程序中提供数种绘图宏指令，其详细说明如下：

■ RECTANGLE (矩形)

| 句型 | 代表意义 | 备注 |
|------------------|--|----------------|
| RECTANGLE (Var1) | 绘出矩形， Var1 代表左上 X 座标，Var1+1 代表左上 Y 座标，Var1+2 代表矩形宽度，Var1+3 代表矩形高度，Var1+4 代表矩形颜色 | Var1 只能为内部存储区。 |

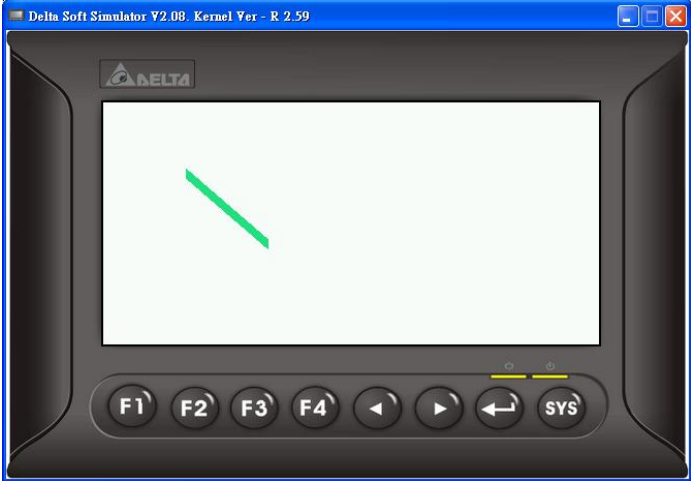
范例：

| | |
|--|---|
| <p>画面开启宏编辑：</p> <pre>\$0 = 20 \$1 = 20 \$2 = 100 \$3 = 100 \$4 = 25</pre> <p>画面 Cycle 宏编辑：</p> <pre>RECTANGLE (\$0)</pre> <p>则结果如右图所示。</p> |  |
|--|---|

■ LINE (线)

| 句型 | 代表意义 | 备注 |
|-------------|---|----------------|
| LINE (Var1) | 绘出线， Var1 代表起点 X 座标，Var1+1 代表起点 Y 座标，Var1+2 代表终点 X 座标，Var1+3 代表终点 Y 座标，Var1+4 代表笔宽， Var1+5 代表颜色 | Var1 只能为内部存储区。 |

范例:

| | |
|---|--|
| <p>画面开启宏编辑:</p> <pre>\$0 = 80 \$1 = 80 \$2 = 160 \$3 = 160 \$4 = 10 \$5 = 10000</pre> <p>画面 Cycle 宏编辑:</p> <pre>LINE (\$0)</pre> <p>则结果如右图所示。</p> |  |
|---|--|

■ POINT (点)

| 句型 | 代表意义 | 备注 |
|--------------|--|------------------|
| POINT (Var1) | 绘出点, Var1 代表点 X 座标, Var1+1 代表点 Y 座标, Var1+2 代表点颜色 | ➤ Var1 只能为内部存储区。 |

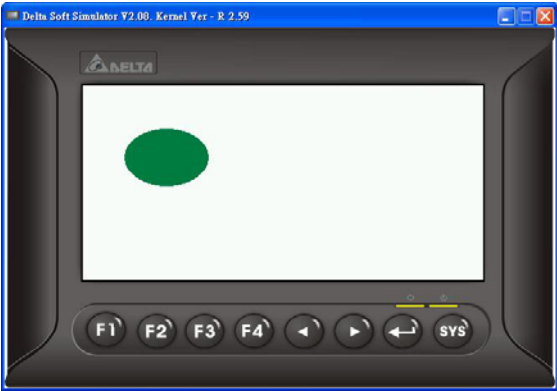
范例:

| | |
|--|--|
| <p>画面开启宏编辑:</p> <pre>\$0 = 80 \$1 = 80 \$2 = 1000</pre> <p>画面 Cycle 宏编辑:</p> <pre>POINT (\$0)</pre> <p>则结果如右图所示。</p> |  |
|--|--|

■ CIRCLE (椭圆)

| 句型 | 代表意义 | 备注 |
|---------------|--|------------------|
| CIRCLE (Var1) | 绘出椭圆, Var1 代表椭圆圆心 X 座标, Var1+1 代表椭圆圆心 Y 座标, Var1+2 代表椭圆长度, Var1+3 代表椭圆宽度, Var1+4 代表椭圆颜色 | ➤ Var1 只能为内部存储区。 |

范例:

| | |
|---|--|
| <p>画面开启宏编辑:</p> <pre>\$0 = 100 \$1 = 100 \$2 = 50 \$3 = 40 \$4 = 1000</pre> <p>画面 Cycle 宏编辑:</p> <pre>CIRCLE (\$0)</pre> <p>则结果如右图所示。</p> |  |
|---|--|

3.14.3.10 其他


- Time Tick
- GETLASTERROR
- Comment
- Delay
- GETSYSTEMTIME
- SETSYSTEMTIME
- GETHISTORY
- EXPORT

其他宏指令详细说明如下:

■ Time Tick (取得系统启动到现在的时间)

| 句型 | 代表意义 | 备注 |
|-----------------|-----------------------|------------------|
| Var1 = TIMETICK | 取得系统启动到现在的时间, 单位为 ms。 | ▶ Var1 只能为内部存储区。 |

范例:

| | |
|------------------------------|--|
| <p>取得系统启动到现在的时间, 并存在\$0。</p> |  |
|------------------------------|--|

■ GETLASTERROR (得到上一道指令的错误值)

| 句型 | 代表意义 | 备注 |
|---------------------|--|------------------|
| Var1 = GETLASTERROR | 得到上一道指令的错误值，如果正确，值为1，如果错误回传负值，每个宏虽然是同时执行，但是其错误信息并不会互相干扰，负值所代表的函义请参考第四节的错误信息。 | ▶ Var1 只能为内部存储区。 |

范例：

| | |
|---------------------|--|
| 得到上一道指令的错误值，并存在\$0。 | |
|---------------------|--|

■ Comment (注解)

| 句型 | 代表意义 | 备注 |
|--------------------|-------------|--------------------------------------|
| # <i>Statement</i> | 注解，可增加宏的可读性 | ▶ <i>Statement</i> 可为一段文字描述或一段宏指令集合。 |

范例：

| | |
|-----------|----------------------------------|
| 将加法宏改成注解。 | <code>#\$0 = \$ 0 + 1</code> |
| 文字注解 | <code>#This is a Comment.</code> |

■ Delay (延迟)

| 句型 | 代表意义 | 备注 |
|--------------|--|--|
| Delay (Var1) | 系统将会在这边延迟用户所设定的时间，但是由于系统是采多工的方式，因此会有所谓的系统延迟，因此所设定的时间可能会因为系统的忙碌而增加，但是绝不会有提早的事情发生。过多的 Delay 设定，会造成人机反应变慢。这边的时间单位为千分之一秒 (ms)。 | <ul style="list-style-type: none"> ➤ Var1 可以为内部存储区或是常数。 ➤ 当执行 Delay 命令时，HMI 将会暂停所有动作，待 Delay 时间结束后，才会继续执行动作。 |

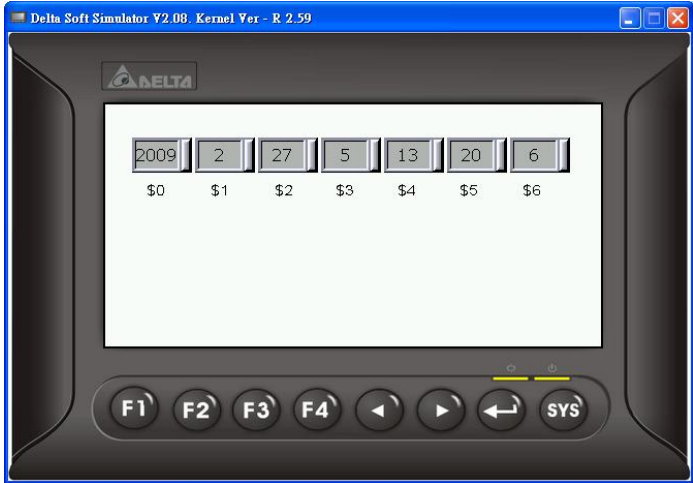
范例：

| | |
|------|-------------|
| 延迟两秒 | Delay(2000) |
|------|-------------|

■ GETSYSTEMTIME (取得系统时间)

| 句型 | 代表意义 | 备注 |
|-------------------------|---|--|
| Var1 = GETSYSTEMTIME | 取得系统时间，Var1 表示内部存储区连续 7 Words 的开始地址。 Var1 代表年， Var1 + 1 代表月， Var1 + 2 代表日， Var1 + 3 代表星期， Var1 + 4 代表时， Var1 + 5 代表分， Var1 + 6 代表秒 | <ul style="list-style-type: none"> ➤ Var1 只能为内部存储区。 |

范例：

| | |
|--|--|
| <p>现在系统时间为 2009/02/27 FRI 13:20:06</p> <p>\$0 = GETSYSTEMTIME</p> <p>则结果如右图所示。</p> |  |
|--|--|

■ SETSYSTEMTIME (设定系统时间)

| 句型 | 代表意义 | 备注 |
|-------------------------|--|---|
| SETSYSTEMTIME (VAR1) | 设定系统时间, Var1 表示内部存储区连续 7 Words 的开始地址。 Var1 代表年, Var1 + 1 代表月, Var1 + 2 代表日, Var1 + 3 代表星期, Var1 + 4 代表时, Var1 + 5 代表分, Var1 + 6 代表秒 | <ul style="list-style-type: none"> ➤ Var1 只能为内部存储区。 ➤ 所输入的数据必须要为合法的数字, 系统才会允许用户变更系统时间。例如: Var1 + 3 为代表存放星期的存储区, 而星期只有一到七的数字输入, 当您所输入的数字大于七之后, 系统判定为不合法。 |

范例:

| | |
|---------------------------------------|--|
| 设定系统时间为 2009/02/27 FRI 13:25:34 | <pre> \$0 = 2009 \$1 = 02 \$2 = 27 \$3 = 5 \$4 = 13 \$5 = 25 \$6 = 34 SETSYSTEMTIME (\$0) </pre> |
|---------------------------------------|--|

■ GETHISTORY (取得历史数据)

| 句型 | 代表意义 | 备注 |
|---|---|---|
| Var1 = GETHISTORY (Var2, Var3, Var4, Var5, Var6) | 取得历史数据, Var1 表示存入数据长度内部存储区, Var2 为历史缓冲区编号, Var3 为读取起始取点位置, Var4 为读取点数, Var5 为数据存入位置, Var6 为读取数据类型 | <ul style="list-style-type: none"> ➤ Var1 只能为内部存储区。 ➤ Var2、Var3 与 Var4 可以为内部存储区或常数。 ➤ Var5 可为控制器暂存器或内部存储区。 ➤ Var6 可为内部存储区或是常数。 ➤ Var6: 0 为数据; 1 为时间; 2 为时间跟数据。 |

范例:

取得历史数据，\$0 = GETHISTORY (\$1, \$2, \$3, \$4, \$5)

■ EXPORT (输出报表)

| 句型 | 代表意义 | 备注 |
|---------------|---|----------------------|
| EXPORT (Var1) | 输出报表， 0 代表 SD 卡， 1 代表 U 盘 2 代表打印机。 | ➤ Var1 可以为内部存储区或是常数。 |

范例：

输出至 SD 卡

3.14.4 宏编译时的错误信息

在程序编译时，为了方便用户能够迅速的找出错误，会将错误列出在输出栏，而有些错误可能是因为用户疏忽少打了一些指令所造成的，为了帮助用户除错，软件再编译程序时提供了错误信息，来提示用户在编辑宏时所犯的语法错误。

编辑时的错误信息

- **代码 —100 无此 LABEL**
此信息表示找不到 GOTO 所要的 LABEL。
- **代码 —101 递归产生**
此信息表示递归产生，此错误信息大部分发生于子宏中，其原因为子宏呼叫子宏本身，不管是直接呼叫，或是辗转几次呼叫后，再呼叫自己，都一样算是递归的发生。原则上子宏不能采用递归的编辑方法，但是如果非用不可的话，请改用 goto 或是 for（无穷次数）的用法。
- **代码 —102 FOR 使用超过 10 个**
此信息表示指令 FOR 的使用超过 10 个。其实这是为了避免用户过度使用 FOR 所限制的，同时也是为了避免存储区的不足所设定的，如果有需要可以使用 GOTO 或是 IF。
- **代码 —103 子宏不存在**
此信息表示所呼叫的子宏不存在。例如编写一行 CALL 5，表示要呼叫子宏 5，但是用户并未编辑子宏 5，为了避免用户的疏忽（可能是输入错误，或是忘记编辑相对应的子宏），而造成不可预知的错误，一样会发生编辑错误的信息，来提醒用户。
- **代码 —104 NEXT 数目少于 FOR**
此信息表示，运算元 NEXT 的次数有少，跟运算元 for 的次数不符合，由于 FOR 跟 NEXT 是搭配使用的，因此一定要成对，如果少了一个 NEXT，程序将不知道要从哪回到 FOR 去执行。
- **代码 —105 FOR 数目少于 NEXT**
此信息表示，运算元 NEXT 的次数有多，跟运算元 FOR 的次数不符合，由于 FOR 跟 NEXT 是搭配使用的，因此一定要成对，如果多了一个 NEXT，程序将不知道要往哪继续执行去执行。
- **代码—106 LABEL 重复**
此信息表示同一个宏里面 LABEL 重复设定，这表示 GOTO 时将产生两个不同的结果，为了避免造成不可预知的错误，会发生编辑错误的信息来提醒用户。
- **代码—107 宏使用 RET**
此信息表示在一个宏里面使用 RET，由于 RET 是为了给子宏返回所呼叫的下一道指

令所设计的，表示程序尚未结束。若需要在宏中标示结束，请使用 END 来表示程序的结束。

人机端的宏错误信息

可以使用宏指令来读取错误信息，但是如果在还没有读取前就又重新执行一道正确的指令，错误信息将会被改变。各个宏执行时，并不会改变其他宏的错误信息。

■ 代码—10 GOTO 错误

此错误信息表示宏的 GOTO 发生错误。

■ 代码—11 堆叠溢位

此错误信息表示宏的堆叠满了，如果发生此错误信息，表示用户呼叫太多层的子宏，使得堆叠不够使用，由于同时间可能有很多不同的宏在执行，这是为了避免存储区不足所采取的保护措施。

■ 代码—12 呼叫空的子宏

此错误信息表示子宏呼叫错误，由于 CALL 的指令可以采取 CALL 内部存储区里面的值所代表的子宏 ID，因此如果用户填入此寄存器地址的值，并没有相对应的子宏可供呼叫。

■ 代码—13 资料读取错误

此错误信息表示资料读取错误，虽然有可能是内部存储区资料发生错误，但大部分应该是控制器资料读取错误。

■ 代码—14 写入错误

此错误信息表示资料写入错误，虽然有可能是内部存储区资料发生错误，但大部分应该是控制器资料写入错误。

■ 代码—15 除数为 0

此错误信息表示除法或是取余数时，除数为 0。

(此页有意留为空白)

3.3 人机内部寄存器说明

人机界面内部设有六种不同功能的寄存器，分别为内部寄存器\$、断电保持内部寄存器\$M、间接寻址寄存器*\$、配方组别寄存器 RCPNO、配方群组别寄存器 RCPG 及配方寄存器 RCP(图 3-3-1)，各个详细说明如下：



图 3-3-1 人机内部寄存器分类

3.3.1 内部寄存器(Read / Write)：\$

Word Access : \$n (n : 0-65535)

Bit Access : \$n.b(n : 0-65535, b : 0-15)

人机提供 65536 个 16 位元内部寄存器(\$0.0 - \$65535.15)。

注意：这段寄存器无断电保持功能，人机断电后寄存器内资料随即消失。

3.3.2 断电保持内部寄存器(Read / Write)：\$M

Word Access : \$Mn (n : 0-1023)

Bit Access : \$Mn.b(n : 0-1023, b : 0-15)

人机提供 1024 个 16 位元断电保持内部寄存器(\$M0.0 - \$M1023.15)，用户可将重要资料或数据记录在此段寄存器。

3.3.3 间接寻址寄存器(Read / Write)：*\$

Word Access: *\$n(n : 0-65535)

间接寻址寄存器*\$n 是先从\$ n 取出值后，把此值当作新地址，再存取此新地址内的值。例如：\$7 = 20, \$20 = 39, 则*\$7 = 39。所以间接寻址的公式表示为：\$n = m, \$m = x, 则

*Sn = x (m 的范围为 0-65535)。

注意：此段寄存器无断电保持功能，人机断电后寄存器内资料随即消失。

3.3.4 配方组别寄存器 (Read / Write) : RCPNO

配方组别寄存器是用来指定配方组别，如下图 3-3-2 所示。配方读取/配方写入即是根据配方编号寄存器记载的配方组别来读/写一组配方。选择第一组配方时，RCPNO = 1，选择第四组配方时，RCPNO = 4。配方的长度与组数是由用户编辑配方表时所设定，其大小限制在下面叙述配方寄存器(RCP)时有详细说明。

注意：此寄存器无断电保持功能，人机断电后寄存器内资料随即消失。



图 3-3-2 位元配方编辑画面

3.3.5 配方群组别寄存器 (Read / Write) : RCPG

配方群组别寄存器是用来指定 32 位元配方群组别，如下图 3-3-3 所示。与 16 位元配方不同的是，32 位元配方多了配方群组别的选择，用户进行配方读取/配方写入时，除了要指定配方组别外，还需指定配方群组别方能读/写一组配方。选择第一群组的第一组配方时，RCPG = 1 且 RCPNO = 1，选择第三群组的第四组配方时，RCPG = 3 且 RCPNO = 4。配方群组数是由用户编辑配方表时所设定，其大小限制在下叙述配方寄存器(RCP)时有详细说明。

注意：此寄存器无断电保持功能，人机断电后寄存器内资料随即消失。

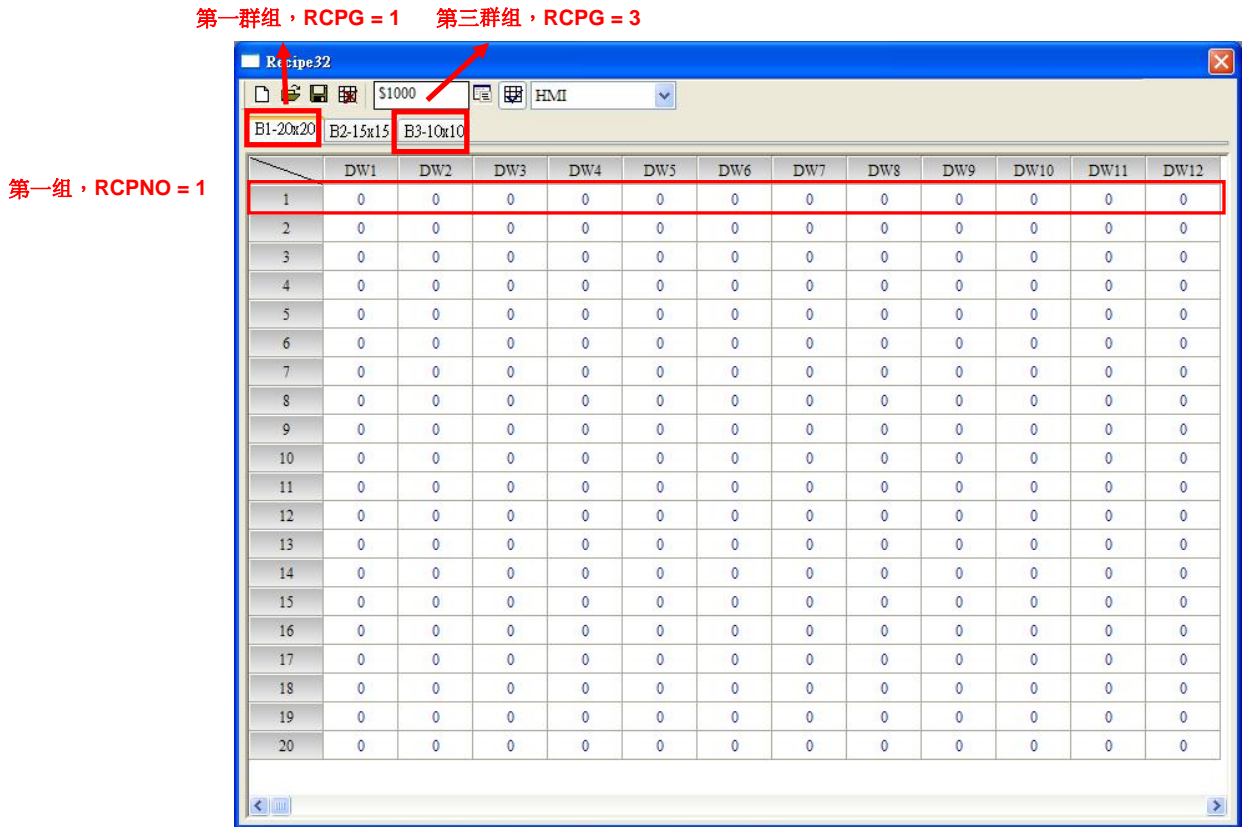


图 3-3-3 位元配方编辑画面

3.3.6 配方寄存器 (Read / Write) : RCP

当用户在画面编辑软件编辑完配方且将之下载至人机后，此配方表立即储存至配方寄存器 (RCP)。而配方寄存器可分成 16 位元及 32 位元：

16 位元配方寄存器

16 位元配方寄存器即代表每个寄存器大小为 16bit(1 Word)。若断电保持区设在外部，例如 U 盘或 SD 卡，则可编辑的 16 位元配方大小为 4MB；若断电保持区设在内部，则可编辑的 16 位元配方大小为 64K。

设配方长度为 L，配方组数为 N，则实际配方大小为 L*N Words(L 乘上 N)，亦即占用了 L*N 个配方寄存器。而人机内部设有配方缓冲区，配置在配方寄存器最前面(表 6-2)，此区提供用户放置所选取的某组配方，且配方缓冲区的长度与所选取的某组配方长度相同，亦即代表配方缓冲区也占用了 L 个配方寄存器，因此一个配方表所占用的配方寄存器数目为 L*(N+1)个，其中 N+1 代表多了一组缓冲区的寄存器数目。由此我们可以得知配方寄存器的范围表示法：

Word Access: RCPn (n : 0-L*(N+1)-1)，其中-1 代表 RCP 表示是从 0 开始。

| | W1 | W2 |
|---|----|----|
| 1 | 36 | 42 |
| 2 | 12 | 33 |
| 3 | 27 | 51 |

表 3-3-1 配方表

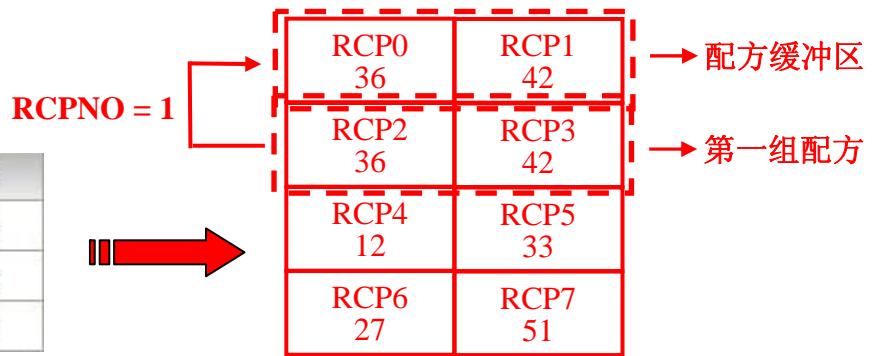


表 3-3-2 配方表寄存器配置

以上表 3-3-1 为例，表中共有三组长度为二的配方，则配方寄存器配置的方式如表 3-3-2 所示(RCP0-RCP7)。因此时所选取的配方组别(RCPNO)为 1，所以配方缓冲区会显示第一组配方值。若所选取的配方组别(RCPNO)改为 3，则配方缓冲区即显示第三组配方值，如表 3-3-4 所示：

| | W1 | W2 |
|---|----|----|
| 1 | 36 | 42 |
| 2 | 12 | 33 |
| 3 | 27 | 51 |

表 3-3-3 配方表

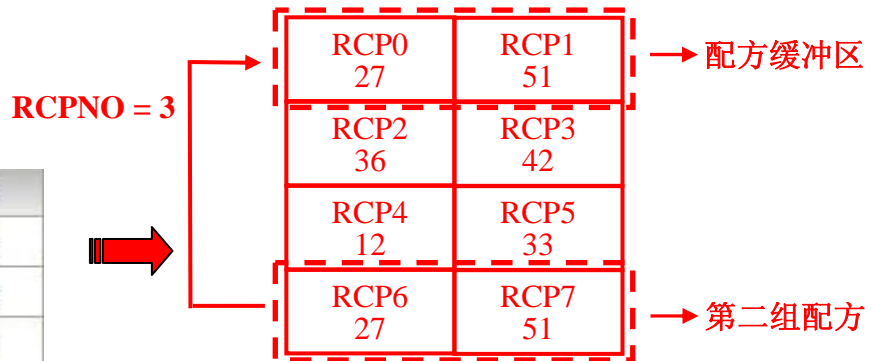


表 3-3-4 配方表寄存器配置

再以下表 3-3-5 为例，表中共有二组长度为三的配方，则配方寄存器配置的方式如表 3-3-6 所示(RCP0-RCP8)。此时若所选取的配方组别(RCPNO)为 2，则配方缓冲区将显示第二组配方值。

| | W1 | W2 | W3 |
|---|----|----|----|
| 1 | 36 | 42 | 1 |
| 2 | 12 | 33 | 76 |

表 3-3-5 配方表

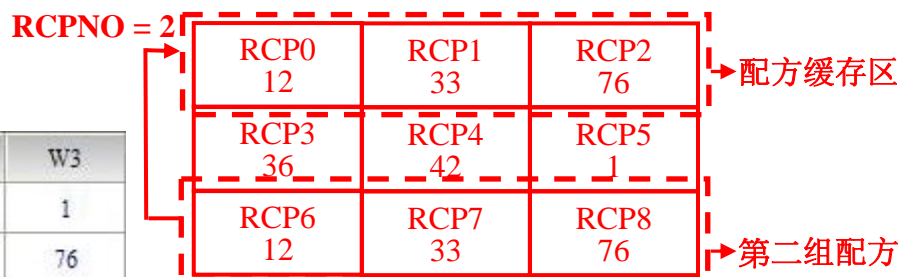


表 3-3-6 配方表寄存器配置

以上为 16 位元配方寄存器(RCP)配置配方缓冲区与配方表的方式。

32 位元配方寄存器

32 位元配方寄存器即代表每个寄存器大小为 32bit (2 Words 或是 Double Word, DW)。若断电保持区设在外部, 例如 U 盘或 SD 卡, 则可编辑的 32 位元配方大小为 50MB; 若断电保持区设在内部, 则可编辑的 32 位元配方大小依不同机型的人机寄存器 (Flash Memory) 规格而定, 请参考型录说明。

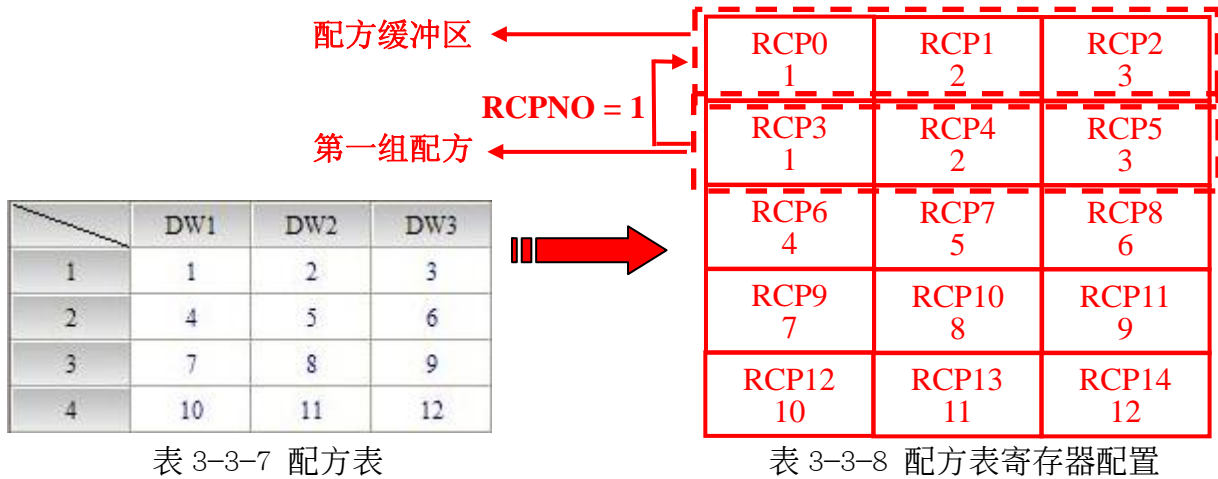
在此必须注意, Flash Memory 里面除存放配方资料外, 还存放了画面资料等其他信息, 因此可编辑的配方大小 (Flash Memory 剩余的空间) 可参考工具栏中的查看 / 寄存器使用清单, 如下图 3-3-4 所示:

| Item | Cost-Bytes |
|------------------|-----------------|
| ROM | 3.13 % Used |
| Total Used | 65536 (64K) |
| Available | 2097152 (2048K) |
| Free | 2031616 (1984K) |
| Detail | |
| Controller | 65536 (64K) |
| Printer | 0 (0K) |
| Screen Data | 0 (0K) |
| Recipe 32 | 0 (0K) |
| SRAM | 0.00 % Used |
| SDRAM | N/A |
| External Storage | |

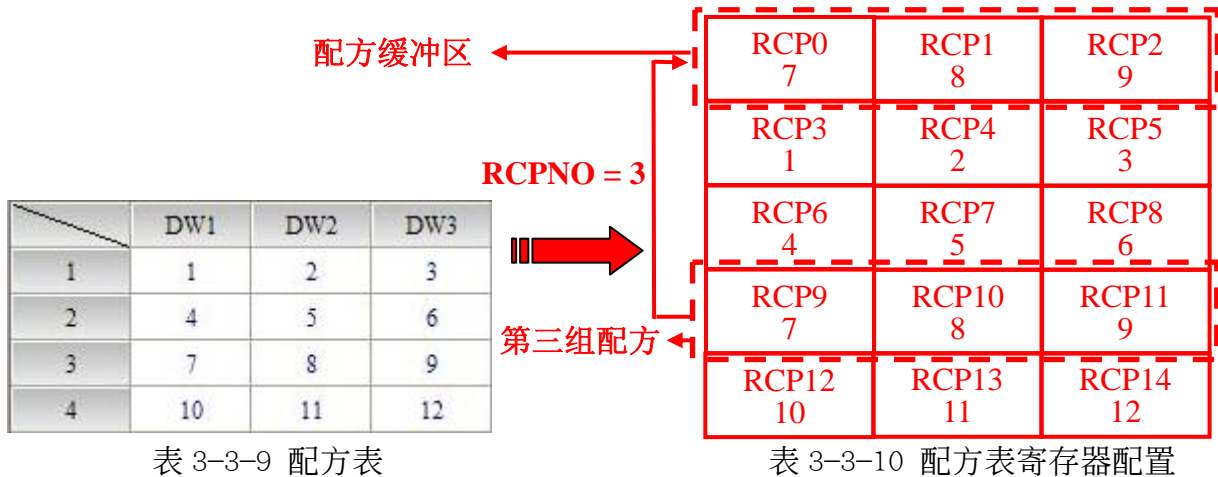
图 3-3-4 寄存器使用清单画面

设配方长度为 L , 配方组数为 N , 则实际配方大小为 $L*N$ DW (L 乘上 N), 亦即占用了 $L*N$ 个配方寄存器。而人机内部设有配方缓冲区, 配置在配方寄存器最前面 (表 3-3-6), 此区提供用户放置所选取的某组配方, 且配方缓冲区的长度与所选取的某组配方长度相同, 亦即代表配方缓冲区也占用了 L 个配方寄存器, 因此一个配方表所占用的配方寄存器数目为 $L*(N+1)$ 个, 其中 $N+1$ 代表多了一组缓冲区的寄存器数目。由此我们可以得知配方寄存器的范围表示法:

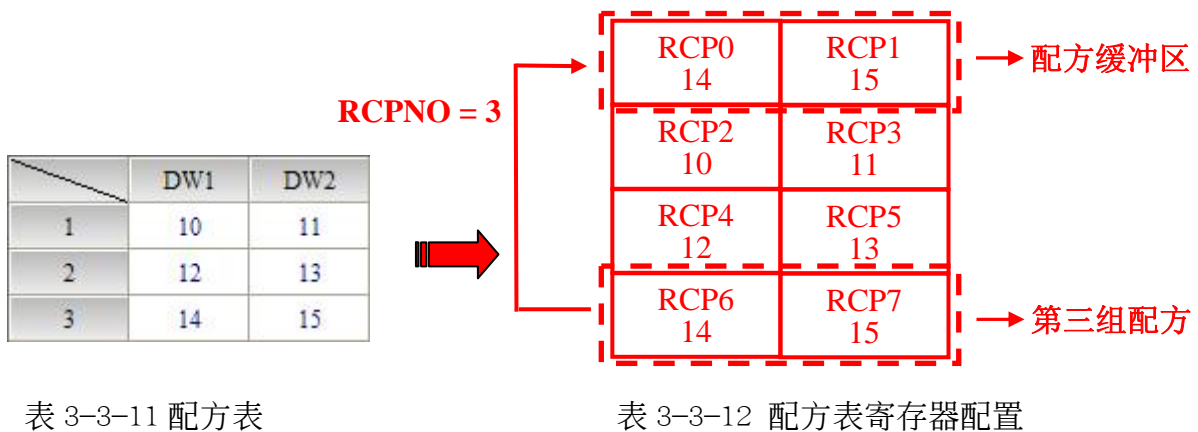
Double Word Access: RCPn ($n : 0-L*(N+1)-1$), 其中 -1 代表 RCP 表示是从 0 开始。



以上表 3-3-7 为例，假设此表建立在第一群组(RCPG=1)，表中共有四组长度为三的配方，则配方寄存器配置的方式如表 3-3-8 所示(RCP0-RCP14)。因此时所选取的配方组别(RCPNO)为 1，所以配方缓冲区会显示第一组配方值。若所选取的配方组别(RCPNO)改为 3，则配方缓冲区即显示第三组配方值，如表 3-3-10 所示：



再以下表 3-3-11 为例，假设此表建立在第一群组(RCPG=1)，表中共有三组长度为二的配方，则配方寄存器配置的方式如表 3-3-12 所示(RCP0-RCP7)。此时若所选取的配方组别(RCPNO)为 3，则配方缓冲区将显示第三组配方值。



以上为 32 位元配方寄存器(RCP)配置配方缓冲区与配方的方式。

3.4 控制区及状态区说明

人机设有系统控制区及状态区两区域，其设定是在 Screen Editor 【选项】 > 【设置模块参数】里的【一般】页面里，如图 3-4-1 及图 3-4-2 所示，详细说明如下列章节所述。

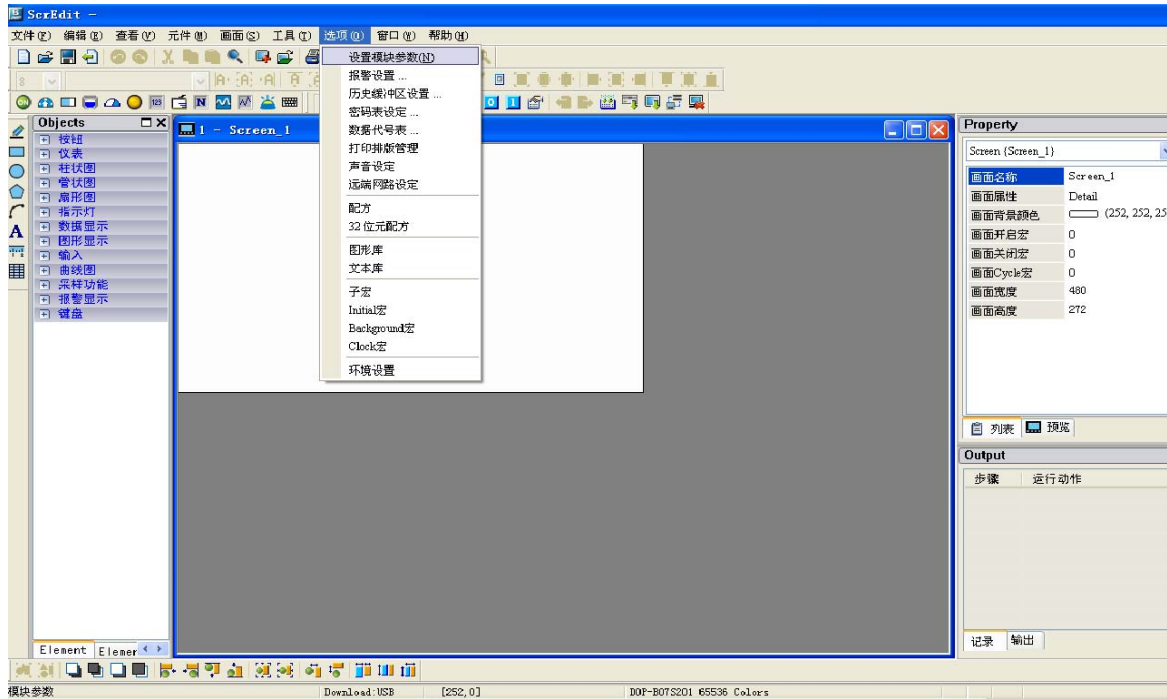


图 3-4-1 设置模块参数

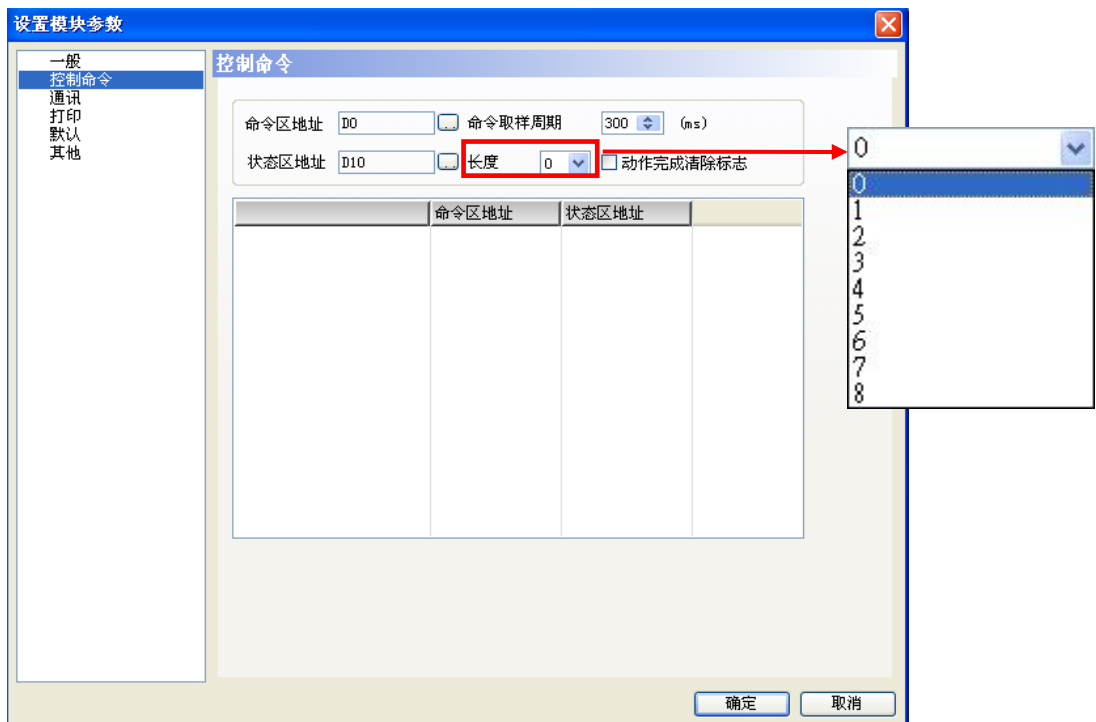


图 3-4-2 设置模块参数

3.4.1 系统控制区

人机设有一可读可写的系统控制区，此区可由用户自行定义在控制器或是人机的某段寄存器地址。用户可借助设定系统控制区来进行人机内部动作，诸如换画面、背灯关闭、目前权限、曲线取样或清除等。而系统控制区是一个以 Word 为单位的连续资料区域，目前可定义的最大长度是 8 Words (图 3-4-2)，但实际长度则是随著用户所使用到的功能多少来决定，例如使用换页功能时(画面编号指定寄存器)，控制区长度至少须开为 1 Word，此时只能使用画面编号指定寄存器；又例如使用历史缓冲区取样功能时(历史缓冲区取样寄存器)，长度至少须开为 4 Words，此时只能使用画面编号指定寄存器、控制标志寄存器、曲线控制寄存器及历史缓冲区取样寄存器；又例如使用多国语言时(系统控制标志寄存器)，长度须开为 8 Words，此时所有控制区寄存器皆可使用。长度设为 0 即代表关闭系统控制区功能。

系统控制区每个寄存器位置的功能及意义说明如下表 3-4-1 所示：

(下表二例，例一是以控制器寄存器 D 为例，起始位置是 D0；例二是以人机内部寄存器 \$ 为例，起始位置是 \$15，用户可自行决定控制区寄存器地址是要存放在控制器内或人机内。)

| 控制区 长度 | 控制区寄存器类别 | 例一(控制器寄存器) | | 例二(人机内部寄存器) | |
|-----------|-------------------|------------|----|-------------|------|
| | | 寄存器 D | 地址 | 寄存器 \$ | 地址 |
| 1 | 画面编号指定寄存器 (SNIR) | Dn | D0 | \$n | \$15 |
| 2 | 控制标志寄存器 (CFR) | Dn+1 | D1 | \$n+1 | \$16 |
| 3 | 曲线控制寄存器 (CUCR) | Dn+2 | D2 | \$n+2 | \$17 |
| 4 | 历史缓冲区取样寄存器 (HBSR) | Dn+3 | D3 | \$n+3 | \$18 |
| 5 | 历史缓冲区清除寄存器 (HBCR) | Dn+4 | D4 | \$n+4 | \$19 |
| 6 | 配方控制寄存器 (RECR) | Dn+5 | D5 | \$n+5 | \$20 |
| 7 | 配方组别指定寄存器 (RBIR) | Dn+6 | D6 | \$n+6 | \$21 |
| 8 | 系统控制标志寄存器 (SCFR) | Dn+7 | D7 | \$n+7 | \$22 |

表 3-4-1 控制区说明

下面将陆续详细介绍各个控制区寄存器的使用方式。

画面编号指定寄存器 (SNIR)

将指定的画面编号写入此寄存器，人机将跳至此指定的画面。

以表 3-4-1 为例，若设定 D0 或 \$15 = 1，则人机将换画面至第一页；若设定 D0 或 \$15 = 7，则人机将换画面至第七页。

控制标志寄存器 (CFR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|-----|---------------------|----------|
| 0 | 0000 0000 0000 000x | 重新通讯开关标志 |

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|------------------|
| 1 | 0000 0000 0000 00x0 | 背光灯开关标志 |
| 2 | 0000 0000 0000 0x00 | 蜂鸣器开关标志 |
| 3 | 0000 0000 0000 x000 | 警报缓冲区（历史警报表）清除标志 |
| 4 | 0000 0000 000x 0000 | 警报计数器（警报频次表）清除标志 |
| 5 | 0000 0000 00x0 0000 | USB 资料即时更新标志 |
| 6-7 | 0000 0000 xx00 0000 | 保留 |
| 8 | 0000 000x 0000 0000 | 设定用户权限标志(权限 1) |
| 9 | 0000 00x0 0000 0000 | 设定用户权限标志(权限 2) |
| 10 | 0000 0x00 0000 0000 | 设定用户权限标志(权限 4) |
| 11-15 | xxxx x000 0000 0000 | 保留 |

■ 重新通讯开关标志

控制人机重新通讯开关。若用户在模块参数中有勾选设定通讯中断重试次数（图 3-4-3），当人机与控制器通讯时通讯中断次数到达设定值后，人机会自动关闭与该控制器的通讯并将此标志设为 ON，此时人机与该控制器通讯中断且不会出现通讯错误的警告视窗(此时并不影响人机与其他控制器的通讯)，用户可将此标志设定为 OFF 来重新开启人机与该控制器的通讯，但此标志只能用于通讯自动中断时恢复通讯用，并无法通过设定此标志为 ON 直接关闭人机与任何控制器之间的通讯(若需要手动关闭或开启与控制器的通讯，可使用 OPENCOM/CLOSECOM 宏命令，请参阅 3.14 节)。

以表 3-4-1 及图 3-4-4 为例，假设人机与 COM2 第二站 PLC 发生通讯异常，且重试次数已满三次，则此时人机自动与该 PLC 通讯中断(但与其他七台 PLC 仍持续通讯)，且不会发生通讯错误之警告视窗。此时若设定 D1 = 0 或 \$16.0 = 0，即通讯开关标志设为 OFF 后，通讯再度被开启，人机会与 COM2 第二站 PLC 重新尝试通讯，重试三次失败后此标志又会被设为 ON。其中 D1 = 0 代表 D1 的第零个 Bit 为 0(0000 0000 0000 0000)。

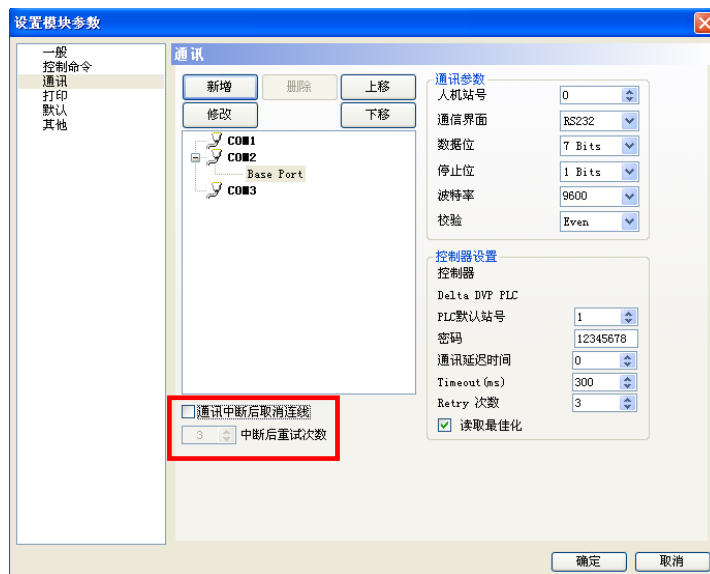


图 3-4-3 通讯参数设定

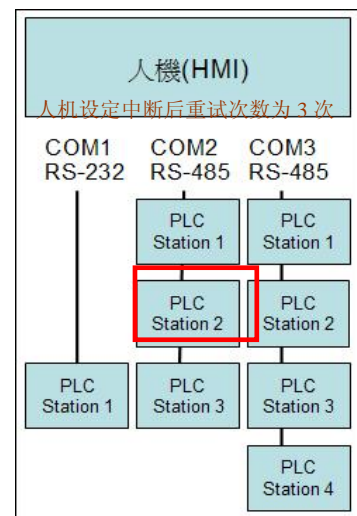


图 3-4-4 接线示意图

■ 背光灯开关标志

控制人机背光灯开关。若此标志被设为 ON，则关闭人机界面背光灯；若此标志被设为 OFF，则开启人机界面背光灯。

以表 3-4-1 为例，若设定 $D1 = 2$ 或 $\$16.1 = 1$ ，则背光灯开关标志即设为 ON，此时人机界面背光灯关闭。其中 $D1 = 2$ 代表 $D1$ 的第一个 Bit 为 1(0000 0000 0000 0010)。

■ 蜂鸣器开关标志

控制人机的蜂鸣器开关。若此标志被设为 ON，则开启人机蜂鸣器；若此标志被设为 OFF，则关闭人机蜂鸣器。

以表 3-4-1 为例，若设定 $D1 = 4$ 或 $\$16.2 = 1$ ，则蜂鸣器开关标志即设为 ON，此时人机蜂鸣器开启。其中 $D1 = 4$ 代表 $D1$ 的第二个 Bit 为 1(0000 0000 0000 0100)。

■ 警报缓冲区（历史警报表）清除控制标志

清除人机的警报缓冲区资料。若此标志被设为 ON，则可将警报记录缓冲区的资料清除。再次触发此标志前必须先将此标志设为 OFF。

以表 3-4-1 为例，若设定 $D1 = 8$ 或 $\$16.3 = 1$ ，则警报缓冲区(历史警报表)清除控制标志即设为 ON，此时会清除警报记录缓冲区的资料。其中 $D1 = 8$ 代表 $D1$ 的第三个 Bit 为 1(0000 0000 0000 1000)。

■ 警报计数器（警报频次表）清除控制标志

清除人机的警报频次表资料。若此标志被设为 ON，则可将警报发生频次表的资料清除。再次触发此标志前必须先将此标志设为 OFF。

以表 3-4-1 为例，若设定 $D1 = 16$ 或 $\$16.4 = 1$ ，则警报计数器（警报频次表）清除控制标志即设为 ON，此时会清除警报计数器的资料。其中 $D1 = 16$ 代表 $D1$ 的第四个 Bit 为 1(0000 0000 0001 0000)。

■ USB 资料即时更新标志

将人机缓冲区内的资料即时更新至 U 盘。若警报、历史缓冲区或配方功能被启用，且断电保持区设在 U 盘，则此标志被设为 ON 后，人机会将暂时存放在缓冲区内的资料即时更新至 U 盘。再次触发此标志前必须先将此标志设为 OFF。在此提醒，人机欲写入 U 盘的资料会先暂时放置在缓冲区，缓冲区的资料量未放满时(缓冲区预设值为 64KB，也可透过”设置模块参数”自行定义)，资料将不会真正写入 U 盘，此方式可避免持续不断的抹写导致 U 盘快速损坏。因此 (1)用户要存取的资料量若小于缓冲区的容量或 (2)为避免断电或其他不当操作导致资料遗失，可借助触发此标志将资料强制写入 U 盘。

以表 3-4-1 为例，若设定 $D1 = 32$ 或 $\$16.5 = 1$ ，则 USB 资料即时更新标志即设为 ON，此时会将存放于缓冲区的资料即时写入 U 盘。其中 $D1 = 32$ 代表 $D1$ 的第五个 Bit 为 1(0000 0000 0010 0000)。

■ 设定用户权限标志

借助 Bit 8、Bit 9、Bit 10 三个标志的触发来改变目前人机用户权限。人机内部权限分成两种，分别是 (1) 权限 0 (最低权限) - 权限 7，以及 (2) 最高权限，用户可借助这三个标志来设定权限 0 (最低权限) - 权限 7，但无法设定最高权限。Bit 8 ON 时用户权限为 1；Bit 9 ON 时用户权限为 2；Bit 10 ON 时用户权限为 4；而详细设定方式如下表所示：

| 权限等级 | 标志控制 | | 二进制表示法 |
|------|--------------------|--------------------|---------------------|
| | ON | OFF | |
| 权限 0 | | Bit 8、Bit 9、Bit 10 | 0000 0000 0000 0000 |
| 权限 1 | Bit 8 | Bit 9、Bit 10 | 0000 0001 0000 0000 |
| 权限 2 | Bit 9 | Bit 8、Bit 10 | 0000 0010 0000 0000 |
| 权限 3 | Bit 8、Bit 9 | Bit 10 | 0000 0011 0000 0000 |
| 权限 4 | Bit 10 | Bit 8、Bit 9 | 0000 0100 0000 0000 |
| 权限 5 | Bit 8、Bit 10 | Bit 9 | 0000 0101 0000 0000 |
| 权限 6 | Bit 9、Bit 10 | Bit 8 | 0000 0110 0000 0000 |
| 权限 7 | Bit 8、Bit 9、Bit 10 | | 0000 0111 0000 0000 |

以表 3-4-1 及上表为例，若设定 $D1 = 1280$ ，或 $\$16.8 = 1$ 及 $\$16.10 = 1$ ，则用户权限即为权限 5。其中 $D1 = 1280$ 代表 $D1$ 的第八及第十个 Bit 为 1 (0000 0101 0000 0000)。

曲线控制寄存器 (CUCR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|----------|
| 0 | 0000 0000 0000 000x | 曲线取样标志 1 |
| 1 | 0000 0000 0000 00x0 | 曲线取样标志 2 |
| 2 | 0000 0000 0000 0x00 | 曲线取样标志 3 |
| 3 | 0000 0000 0000 x000 | 曲线取样标志 4 |
| 4-7 | 0000 0000 xxxx 0000 | 保留 |
| 8 | 0000 000x 0000 0000 | 曲线清除标志 1 |
| 9 | 0000 00x0 0000 0000 | 曲线清除标志 2 |
| 10 | 0000 0x00 0000 0000 | 曲线清除标志 3 |
| 11 | 0000 x000 0000 0000 | 曲线清除标志 4 |
| 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 曲线取样控制标志

人机提供四种曲线取样标志 (曲线取样标志 1-4)，曲线图 (一般曲线图及 X-Y 曲线图) 的图形显示须由曲线取样标志来作控制。而标志的选定，只是作为触发曲线图图形显示的依据，因此若画面中数个曲线图同时使用同一个曲线取样标志，则该标志被触发时，画面中有运用到该标志的曲线图会将所需的绘图资料取样一次，再将资料转换成曲线图显示 (详细曲线图元件设定方式及说明请参阅 3.8.10 节)。再次触发此标志前必须先清除此标志为 OFF。

以表 3-4-1 及图 3-4-5 为例，若设定 $D2 = 1$ 或 $S17.0 = 1$ ，则图 1 与图 2 的曲线将会被绘出，图 3 与图 4 则保持原状态。其中 $D2 = 1$ 代表 $D2$ 的第零个 Bit 为 1 (0000 0000 0000 0001)。

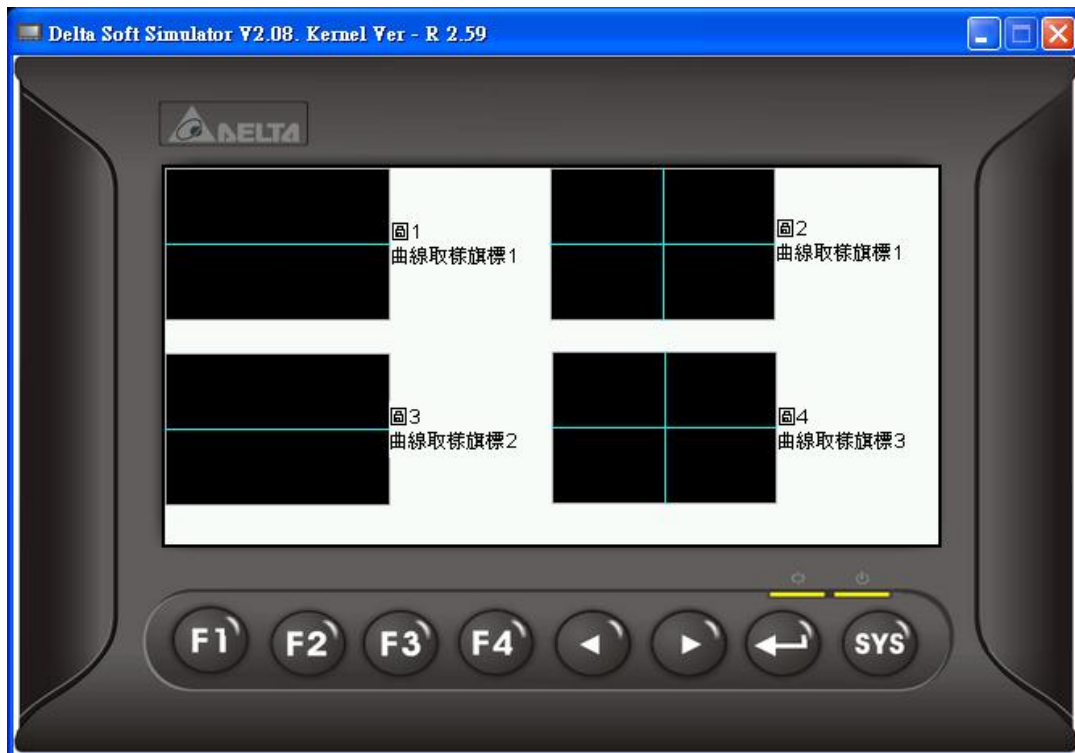


图 3-4-5 曲线元件示意图

■ 曲线清除控制标志

人机提供四种曲线清除标志（曲线清除标志 1-4），曲线图（一般曲线图及 X-Y 曲线图）的图形清除须由曲线清除标志来作控制。而标志的选定，只是作为触发曲线图图形清除的依据，因此若画面中数个曲线图同时使用同一个曲线清除标志，则该标志被触发时，画面中有运用到该标志的曲线图会将图形清除（详细曲线图元件设定方式及说明请参阅 3.8.10 节）。再次触发此标志前必须先清除此标志为 OFF。

以表 3-4-1 及图 3-4-5 为例，若设定 $D2 = 512$ 或 $S17.9 = 1$ ，则图 3 的曲线将会被清除，图 1、图 2 与图 4 则保持原状态。其中 $D2 = 512$ 代表 $D2$ 的第九个 Bit 为 1 (0000 0010 0000 0000)。

历史缓冲区取样寄存器 (HBSR)

人机最多可记录十二组历史缓冲区资料（图 3-4-6），每组缓冲区所对应的取样标志如下表。缓冲区记录触发源除了可设为人机 Timer 触发取样外，也可通过触发此寄存器内的历史缓冲区取样标志执行取样动作。有关历史缓冲区的详细设定请参阅 3.11.3 「历史缓冲区设定」一节。

| 对应缓冲区 | 对应标志 | | |
|-------|------|---------------|----|
| | Bit | 二进制表示法相对位置(x) | 功能 |

| | | | |
|--------|-------|---------------------|--------------|
| 缓冲区 1 | 0 | 0000 0000 0000 000x | 历史缓冲区取样标志 1 |
| 缓冲区 2 | 1 | 0000 0000 0000 00x0 | 历史缓冲区取样标志 2 |
| 缓冲区 3 | 2 | 0000 0000 0000 0x00 | 历史缓冲区取样标志 3 |
| 缓冲区 4 | 3 | 0000 0000 0000 x000 | 历史缓冲区取样标志 4 |
| 缓冲区 5 | 4 | 0000 0000 000x 0000 | 历史缓冲区取样标志 5 |
| 缓冲区 6 | 5 | 0000 0000 00x0 0000 | 历史缓冲区取样标志 6 |
| 缓冲区 7 | 6 | 0000 0000 0x00 0000 | 历史缓冲区取样标志 7 |
| 缓冲区 8 | 7 | 0000 0000 x000 0000 | 历史缓冲区取样标志 8 |
| 缓冲区 9 | 8 | 0000 000x 0000 0000 | 历史缓冲区取样标志 9 |
| 缓冲区 10 | 9 | 0000 00x0 0000 0000 | 历史缓冲区取样标志 10 |
| 缓冲区 11 | 10 | 0000 0x00 0000 0000 | 历史缓冲区取样标志 11 |
| 缓冲区 12 | 11 | 0000 x000 0000 0000 | 历史缓冲区取样标志 12 |
| | 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 历史缓冲区取样标志

用户可借控制器触发历史缓冲区取样标志来决定缓冲区取样时机，每触发一次历史缓冲区取样标志即执行缓冲区取样一次。再次触发此标志前必须先清除此标志为 OFF。

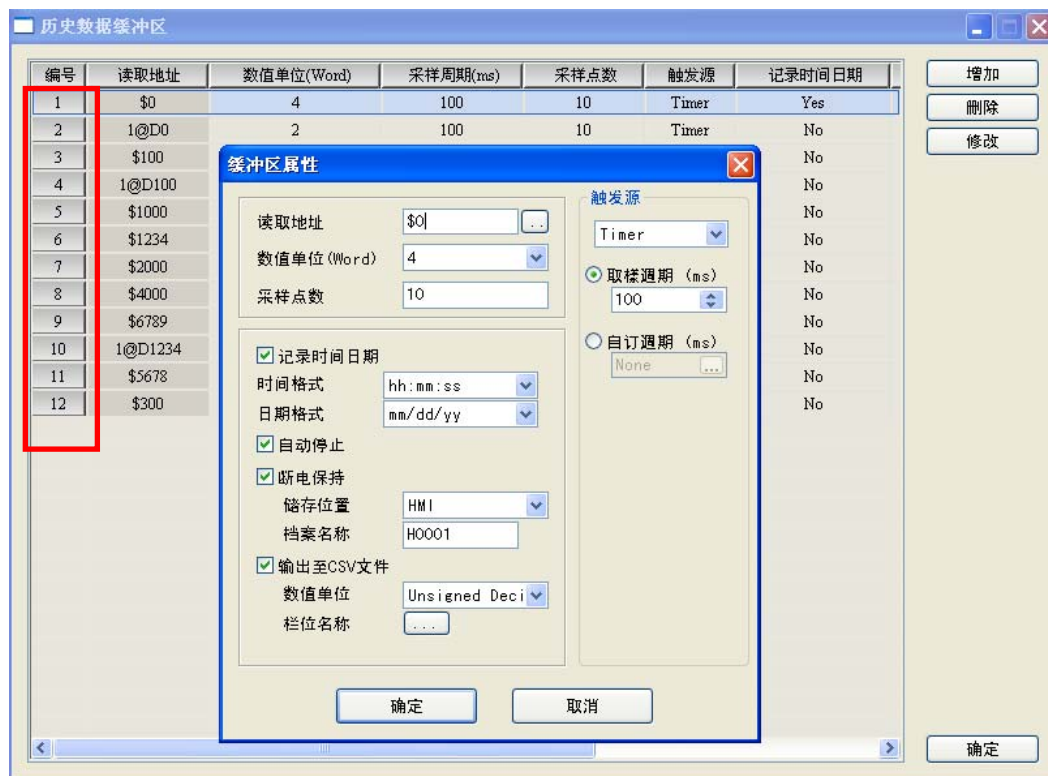


图 3-4-6 历史缓冲区设定

历史缓冲区清除寄存器 (HBCR)

每组缓冲区所对应的清除标志如下表。用户可通过触发此寄存器内的历史缓冲区清除标

志执行缓冲区清除动作

| 对应缓冲区 | 对应标志 | | |
|--------|-------|---------------------|--------------|
| | Bit | 二进制表示法相对位置(x) | 功能 |
| 缓冲区 1 | 0 | 0000 0000 0000 000x | 历史缓冲区清除标志 1 |
| 缓冲区 2 | 1 | 0000 0000 0000 00x0 | 历史缓冲区清除标志 2 |
| 缓冲区 3 | 2 | 0000 0000 0000 0x00 | 历史缓冲区清除标志 3 |
| 缓冲区 4 | 3 | 0000 0000 0000 x000 | 历史缓冲区清除标志 4 |
| 缓冲区 5 | 4 | 0000 0000 000x 0000 | 历史缓冲区清除标志 5 |
| 缓冲区 6 | 5 | 0000 0000 00x0 0000 | 历史缓冲区清除标志 6 |
| 缓冲区 7 | 6 | 0000 0000 0x00 0000 | 历史缓冲区清除标志 7 |
| 缓冲区 8 | 7 | 0000 0000 x000 0000 | 历史缓冲区清除标志 8 |
| 缓冲区 9 | 8 | 0000 000x 0000 0000 | 历史缓冲区清除标志 9 |
| 缓冲区 10 | 9 | 0000 00x0 0000 0000 | 历史缓冲区清除标志 10 |
| 缓冲区 11 | 10 | 0000 0x00 0000 0000 | 历史缓冲区清除标志 11 |
| 缓冲区 12 | 11 | 0000 x000 0000 0000 | 历史缓冲区清除标志 12 |
| | 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 历史缓冲区清除控制标志

用户可借控制器触发历史缓冲区清除标志来决定缓冲区清除时机，每触发一次历史缓冲区清除标志即执行缓冲区清除一次。再次触发此标志前必须先清除此标志为 OFF。

配方控制寄存器 (RECR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|------|---------------------|--------------------------------------|
| 0 | 0000 0000 0000 000x | 配方组别变更标志 |
| 1 | 0000 0000 0000 00x0 | 配方读取标志： 控制器内的配方资料写入人机 (PLC → HMI) |
| 2 | 0000 0000 0000 0x00 | 配方写入标志： 人机内的配方资料写入控制器 (PLC ← HMI) |
| 3 | 0000 0000 0000 x000 | 配方群组别变更标志 |
| 4-7 | 0000 0000 xxxx 0000 | 保留 |
| 8-15 | xxxx xxxx 0000 0000 | 配方群组别指定寄存器 |

■ 配方组别变更标志

用户呼叫或变更配方组别有两种方式，第一种是直接经由人机内部寄存器 RCPNO 变更；第二种是可通过此控制区—配方控制寄存器来变更。用户要通过此控制区下达变更人机配方组别的命令，首先要将要变更的的配方组别写入配方组别指定寄存器内（请见下列配方组别指定寄存器 RBIR 说明），然后再触发配方组别变更标志。当人机侦测到配方组别变更标志被触发时，即会自动更改内部寄存器 RCPNO 的值并变更配方组别。再次触发前必须先清除此标志为 OFF。

以表 3-4-1 为例，若设定 $D6 = 3$ ，且 $D5 = 1$ 或 $S20.0 = 1$ ，则配方组别将变成第三组 (RCPNO=3)。其中 $D6 = 3$ 代表指定的配方组别为第三组； $D5 = 1$ 代表 $D5$ 的第零个 Bit 为 1(0000 0000 0000 0001)。

■ 配方读取标志

用户可借助触发配方读取标志将控制器内配方资料写入至人机内所指定的某组配方资料寄存器。再次触发此标志前必须先清除此标志为 OFF。

以表 3-4-1 为例，若人机内选定的配方组别为第四组 (RCPNO=4)，且设定 $D5 = 2$ 或 $S20.1 = 1$ ，则控制器内的配方资料将立即写入人机内部第四组配方寄存器，原本第四组寄存器内的配方资料立即被更新。其中 $D5 = 2$ 代表 $D5$ 的第一个 Bit 为 1(0000 0000 0000 0010)。

■ 配方写入标志

用户可借助触发配方写入标志将人机内所指定的某组配方资料写入至控制器寄存器。再次触发此标志前必须先清除此标志为 OFF。

以表 3-4-1 为例，若人机内选定的配方组别为第二组 (RCPNO=2)，且设定 $D5 = 4$ 或 $S20.2 = 1$ ，则人机内部第二组配方资料将立即写入控制器寄存器，原本控制器寄存器内的配方资料立即被更新。其中 $D5 = 4$ 代表 $D5$ 的第二个 Bit 为 1(0000 0000 0000 0100)。

■ 配方群组别变更标志

用户呼叫与变更配方群组别有两种方式，第一种直接经由人机内部寄存器 RCPG 变更；第二种是可通过此控制区—配方控制寄存器来变更。用户要透过此控制区下达变更人机配方群组别的命令，首先要将要变更的的配方群组别写入配方群组别指定寄存器内(配方控制寄存器高位元组：Bit 8 –Bit 15)，然后再触发配方群组别变更标志。当人机侦测到配方群组别变更标志被触发时，即会自动更改内部寄存器 RCPG 的值并变更配方群组别。再次触发此标志前必须先清除此标志为 OFF。

以表 3-4-1 为例，若设定 $D5 = 520$ ，或 $S20.3 = 1$ 及 $S20.9 = 1$ ，则配方群组别将变成第二组(RCPG=2)。其中 $D5 = 520$ 代表 $D5$ 的第三个及第九个 Bit 为 1(0000 0010 0000 1000)。

■ 配方群组别指定寄存器

将指定的配方群组别写入配方群组别指定寄存器，也就是写入配方控制寄存器高位元组 (Bit 8 –Bit 15)，并配合触发配方群组别变更标志，则人机将会自动更改内部寄存器 RCPG 的值并变更配方群组别。

以表 3-4-1 为例，若设定 $D5 = 520$ ，或 $S20.3 = 1$ 及 $S20.9 = 1$ ，则配方群组别将变成第二组(RCPG=2)。其中 $D5 = 520$ 代表 $D5$ 的第三个及第九个 Bit 为 1(0000 0010 0000 1000)。详细配方群组别指定方法如下所述。

配方群组别指定方法：

| | |
|------------------------------------|-----------------------------------|
| D5 = 520(0000 0010 0000 1000) | |
| High Byte(配方群组别指定寄存器) 0000 0010 | Low Byte(其他配方相关控制标志) 0000 1000 |

将 D5 寄存器分成高低位元组来看，决定配方群组别的是高位元组的部分(与低位元组无关)，当高位元组设定为 0000 0010 时(如上表)，代表选定的配方群组别为第二组；若将高位元组改为 0000 0011 时，代表选定的配方群组别为第三组，依此类推。

配方组别指定寄存器 (RBIR)

将指定的配方组别写入配方组别指定寄存器，并配合触发配方组别变更标志，则人机将会自动更改内部寄存器 RCPNO 的值并变更配方组别。

以表 3-4-1 为例，若设定 D6 = 3，且 D5 = 1 或 \$20.0 = 1，则配方组别将变成第三组 (RCPNO=3)。其中 D6 = 3 代表指定的配方组别为第三组；D5 = 1 代表 D5 的第零个 Bit 为 1(0000 0000 0000 0001)。

系统控制标志寄存器 (SCFR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|------------|
| 0-7 | 0000 0000 xxxx xxxx | 多国语言设定值寄存器 |
| 8 | 0000 000x 0000 0000 | 打印标志 |
| 9 | 0000 00x0 0000 0000 | 打印换页标志 |
| 10-15 | xxxx xx00 0000 0000 | 保留 |

■ 多国语言设定值寄存器

人机最多提供十六国语言的编辑(详细设定方式及说明请参阅 3.10 节)，而用户可透过切换多国语言设定值寄存器的设定值(图 3-4-7)，达到切换语言的目的。

以表 3-4-1 及图 3-4-7 为例，设定中文语言之设定值为 1，且设定 D7 = 1 或 \$22.0 = 1，则人机将显示中文操作界面。其中 D7 = 1 代表 D7 的第零个 Bit 为 1(0000 0000 0000 0001)。详细多国语言设定值指定方法如下所述。

多国语言设定值指定方法：

| D7 = 1(0000 0000 0000 0001) | |
|------------------------------------|-----------------------------------|
| High Byte(其他系统控制相关标志) 0000 0000 | Low Byte(多国语言设定值寄存器) 0000 0001 |

将 D7 寄存器分成高低位元组来看，决定多国语言设定值的是低位元组的部分(与高位元组无关)，当低位元组设定为 0000 0001 时(如上表)，代表选定的多国语言设定值为 1，对应到图 3-4-7 的语言便是中文 chinese；若将低位元组改为 0000 0011 时，代表选定的多国语言设定值为 3，对应到图 3-4-7 的语言便是日文 japan，依此类推。设定值的范围为 0-255。

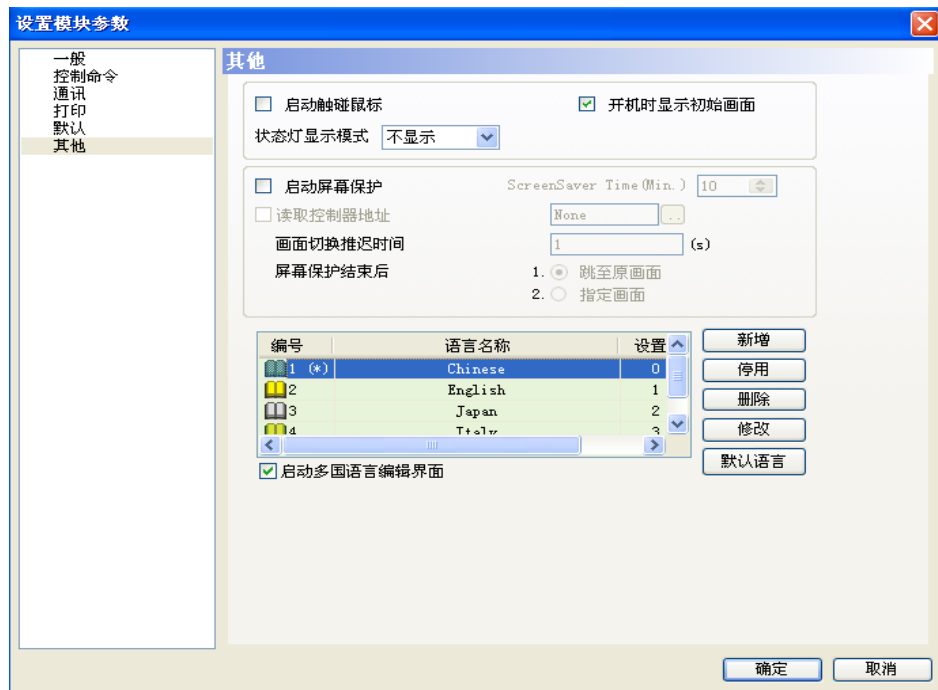


图 3-4-7 多国语言设定

■ 打印标志

打印功能选项可分为一般画面打印(Hard Copy)与画面排版打印。此两种打印方式同时只能有一种成立，且会以排版打印画面为优先。当此打印标志设为 ON 时，会执行当时人机所设定的一般画面打印(Hard Copy)或画面排版打印。打印标志设为 OFF 时则无动作。

以表 3-4-1 为例，若设定 $D7 = 256$ 或 $\$22.8 = 1$ ，则人机将执行打印动作。其中 $D7 = 256$ 代表 $D7$ 的第八个 Bit 为 1(0000 0001 0000 0000)。

■ 打印换页标志

打印换页标志设为 ON 时，打印机纸张自动退纸换页。打印换页标志设为 OFF 时则无动作。

以表 3-4-1 为例，若设定 $D7 = 512$ 或 $\$22.9 = 1$ ，则人机将执行退纸换页动作。其中 $D7 = 512$ 代表 $D7$ 的第九个 Bit 为 1(0000 0010 0000 0000)。

3.4.2 系统状态区

人机设有一只读的系统状态区，此区可由用户自行定义在控制器或是人机的某段寄存器地址。用户可借助设定系统状态区来得知人机内部动作是否已完成，诸如换画面、背灯关闭、目前权限、曲线取样或清除等。而系统状态区是一个以 Word 为单位的连续资料区域，且此区域一旦打开，长度即固定是 8 Words，软件并未开放选择长度功能（图 3-4-2）。特别要注意的是，系统控制区若关闭，系统状态区功能也随之关闭，且系统控制区与系统状态区的地址不可设定为相同地址。

系统状态区每个寄存器位置的功能及意义说明如下表 3-4-2 所示：

（下表二例，例一是以控制器寄存器 D 为例，起始位置是 D10；例二是以人机内部寄存器 \$ 为例，起始位置是 \$25，用户可自行决定控制区寄存器地址是要存放在控制器内或人机内。）

| Word | 控制区寄存器类别 | 例一(控制器寄存器) | | 例二(人机内部寄存器) | |
|------|---------------------|------------|-----|-------------|------|
| | | 寄存器 D | 地址 | 寄存器 \$ | 地址 |
| 0 | 一般控制状态寄存器 (GCSR) | Dm | D10 | \$m | \$25 |
| 1 | 画面编号状态寄存器 (SNSR) | Dm+1 | D11 | \$m+1 | \$26 |
| 2 | 曲线控制状态寄存器 (CCSR) | Dm+2 | D12 | \$m+2 | \$27 |
| 3 | 历史取样状态寄存器 (HSSR) | Dm+3 | D13 | \$m+3 | \$28 |
| 4 | 历史清除状态寄存器 (HCSR) | Dm+4 | D14 | \$m+4 | \$29 |
| 5 | 配方状态寄存器 (RESR) | Dm+5 | D15 | \$m+5 | \$30 |
| 6 | 配方组别状态寄存器 (RBSR) | Dm+6 | D16 | \$m+6 | \$31 |
| 7 | 一般控制状态寄存器 2 (GCSR2) | Dm+7 | D17 | \$m+7 | \$32 |

表 3-4-2 状态区说明

一般控制状态寄存器 (GCSR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|--------------------|
| 0 | 0000 0000 0000 000x | 画面切换状态标志 |
| 1-2 | 0000 0000 0000 0xx0 | 保留 |
| 3 | 0000 0000 0000 x000 | 警报缓冲区(历史警报表)清除状态标志 |
| 4 | 0000 0000 000x 0000 | 警报计数器(警报频次表)清除状态标志 |
| 5-7 | 0000 0000 xxx0 0000 | 保留 |
| 8 | 0000 000x 0000 0000 | 用户权限标志(权限 1) |
| 9 | 0000 00x0 0000 0000 | 用户权限标志(权限 2) |
| 10 | 0000 0x00 0000 0000 | 用户权限标志(权限 4) |
| 11-15 | xxxx x000 0000 0000 | 保留 |

■ 画面切换状态标志

画面切换时，此标志会设 ON，画面切换完毕后，此标志即设为 OFF。

■ 警报缓冲区（历史警报表）清除状态标志

人机清除警报缓冲区时，此标志会设 ON，当警报缓冲区清除完毕后，此标志即设为 OFF。

■ 警报计数器（警报频次表）清除状态标志

人机清除警报计数器时，此标志会设 ON，当警报计数器清除完毕后，此标志即设为 OFF。

■ 用户权限标志

透过 BIT 8 至 BIT 10 可得知目前操作人机的用户权限等级。

| 权限等级 | 标志控制 | | 二进制表示法 |
|------|--------------------|--------------------|---------------------|
| | ON | OFF | |
| 权限 0 | | Bit 8、Bit 9、Bit 10 | 0000 0000 0000 0000 |
| 权限 1 | Bit 8 | Bit 9、Bit 10 | 0000 0001 0000 0000 |
| 权限 2 | Bit 9 | Bit 8、Bit 10 | 0000 0010 0000 0000 |
| 权限 3 | Bit 8、Bit 9 | Bit 10 | 0000 0011 0000 0000 |
| 权限 4 | Bit 10 | Bit 8、Bit 9 | 0000 0100 0000 0000 |
| 权限 5 | Bit 8、Bit 10 | Bit 9 | 0000 0101 0000 0000 |
| 权限 6 | Bit 9、Bit 10 | Bit 8 | 0000 0110 0000 0000 |
| 权限 7 | Bit 8、Bit 9、Bit 10 | | 0000 0111 0000 0000 |

画面编号状态寄存器（SNSR）

人机界面在每次更换画面后，会将最后开启的画面编号（包含子画面）写到画面编号状态寄存器（表 3-4-2 中的 D11 或 S26）存放。

曲线控制状态寄存器（CCSR）

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|------------|
| 0 | 0000 0000 0000 000x | 曲线取样状态标志 1 |
| 1 | 0000 0000 0000 00x0 | 曲线取样状态标志 2 |
| 2 | 0000 0000 0000 0x00 | 曲线取样状态标志 3 |
| 3 | 0000 0000 0000 x000 | 曲线取样状态标志 4 |
| 4-7 | 0000 0000 xxxx 0000 | 保留 |
| 8 | 0000 000x 0000 0000 | 曲线清除状态标志 1 |
| 9 | 0000 00x0 0000 0000 | 曲线清除状态标志 2 |
| 10 | 0000 0x00 0000 0000 | 曲线清除状态标志 3 |
| 11 | 0000 x000 0000 0000 | 曲线清除状态标志 4 |
| 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 曲线取样状态标志

人机在一般曲线图或 X-Y 曲线图元件执行资料取样动作时，会将曲线图相对应的曲线取样状态标志设为 ON，取样完成后曲线取样状态标志随即设为 OFF。其中控制区曲线取样标志 1 对应至状态区曲线取样状态标志 1；控制区曲线取样标志 2 对应至状态区曲线取样状态标志 2，依此类推。

■ 曲线清除状态标志

人机在一般曲线图或 X-Y 曲线图元件执行资料清除动作时，会将曲线图相对应的曲线清除状态标志设为 ON，取样完成后曲线清除状态标志随即设为 OFF。其中控制区曲线清除标志 1 对应至状态区曲线清除状态标志 1；控制区曲线清除标志 2 对应至状态区曲线清除状态标志 2，依此类推。

历史取样状态寄存器 (HSSR)

| 对应缓冲区 | 对应标志 | | |
|--------|-------|---------------------|----------------|
| | Bit | 二进制表示法相对位置(x) | 功能 |
| 缓冲区 1 | 0 | 0000 0000 0000 000x | 历史缓冲区取样状态标志 1 |
| 缓冲区 2 | 1 | 0000 0000 0000 00x0 | 历史缓冲区取样状态标志 2 |
| 缓冲区 3 | 2 | 0000 0000 0000 0x00 | 历史缓冲区取样状态标志 3 |
| 缓冲区 4 | 3 | 0000 0000 0000 x000 | 历史缓冲区取样状态标志 4 |
| 缓冲区 5 | 4 | 0000 0000 000x 0000 | 历史缓冲区取样状态标志 5 |
| 缓冲区 6 | 5 | 0000 0000 00x0 0000 | 历史缓冲区取样状态标志 6 |
| 缓冲区 7 | 6 | 0000 0000 0x00 0000 | 历史缓冲区取样状态标志 7 |
| 缓冲区 8 | 7 | 0000 0000 x000 0000 | 历史缓冲区取样状态标志 8 |
| 缓冲区 9 | 8 | 0000 000x 0000 0000 | 历史缓冲区取样状态标志 9 |
| 缓冲区 10 | 9 | 0000 00x0 0000 0000 | 历史缓冲区取样状态标志 10 |
| 缓冲区 11 | 10 | 0000 0x00 0000 0000 | 历史缓冲区取样状态标志 11 |
| 缓冲区 12 | 11 | 0000 x000 0000 0000 | 历史缓冲区取样状态标志 12 |
| | 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 历史缓冲区取样状态标志

人机在执行历史缓冲区取样动作时，会将缓冲区相对应的历史缓冲区取样状态标志设为 ON，取样完成后历史缓冲区取样状态标志随即设为 OFF。

历史清除状态寄存器 (HCSR)

| 对应缓冲区 | 对应标志 | | |
|-------|------|---------------------|---------------|
| | Bit | 二进制表示法相对位置(x) | 功能 |
| 缓冲区 1 | 0 | 0000 0000 0000 000x | 历史缓冲区清除状态标志 1 |
| 缓冲区 2 | 1 | 0000 0000 0000 00x0 | 历史缓冲区清除状态标志 2 |
| 缓冲区 3 | 2 | 0000 0000 0000 0x00 | 历史缓冲区清除状态标志 3 |

| 对应缓冲区 | 对应标志 | | |
|--------|-------|---------------------|----------------|
| | Bit | 二进制表示法相对位置(x) | 功能 |
| 缓冲区 4 | 3 | 0000 0000 0000 x000 | 历史缓冲区清除状态标志 4 |
| 缓冲区 5 | 4 | 0000 0000 000x 0000 | 历史缓冲区清除状态标志 5 |
| 缓冲区 6 | 5 | 0000 0000 00x0 0000 | 历史缓冲区清除状态标志 6 |
| 缓冲区 7 | 6 | 0000 0000 0x00 0000 | 历史缓冲区清除状态标志 7 |
| 缓冲区 8 | 7 | 0000 0000 x000 0000 | 历史缓冲区清除状态标志 8 |
| 缓冲区 9 | 8 | 0000 000x 0000 0000 | 历史缓冲区清除状态标志 9 |
| 缓冲区 10 | 9 | 0000 00x0 0000 0000 | 历史缓冲区清除状态标志 10 |
| 缓冲区 11 | 10 | 0000 0x00 0000 0000 | 历史缓冲区清除状态标志 11 |
| 缓冲区 12 | 11 | 0000 x000 0000 0000 | 历史缓冲区清除状态标志 12 |
| | 12-15 | xxxx 0000 0000 0000 | 保留 |

■ 历史缓冲区清除状态标志

人机在执行历史缓冲区清除动作时，会将缓冲区相对应的历史缓冲区清除状态标志设为 ON，清除完成后历史缓冲区清除状态标志随即设为 OFF。

配方状态寄存器 (RESR)

| Bit | 二进制表示法相对位置(x) | 功能 |
|------|---------------------|--|
| 0 | 0000 0000 0000 000x | 配方组别变更状态标志 |
| 1 | 0000 0000 0000 00x0 | 配方读取状态标志： 控制器内的配方资料写入人机 (PLC → HMI) |
| 2 | 0000 0000 0000 0x00 | 配方写入状态标志： 人机内的配方资料写入控制器 (PLC ← HMI) |
| 3 | 0000 0000 0000 x000 | 配方群组别变更状态标志 |
| 4~7 | 0000 0000 xxxx 0000 | 保留 |
| 8~15 | xxxx xxxx 0000 0000 | 配方群组别状态寄存器 |

■ 配方组别变更状态标志

控制器控制配方组别变更时，配方组别变更状态标志会被设为 ON，当人机变更组别完成且更新 RCPNO 的内容值后，配方组别变更状态标志随即设为 OFF。

■ 配方读取状态标志

当人机由控制器读回 1 组配方资料时，配方读取状态标志会被设为 ON，当配方由控制器读回且储存完成，配方读取状态标志随即设为 OFF。

■ 配方写入状态标志

当人机传送 1 组指定配方资料至控制器时，配方写入状态标志会被设为 ON，当配方传送至写至控制器完成，配方写入状态标志随即设为 OFF。

■ 配方群组别变更状态标志

控制器控制配方组别变更时，配方组别变更状态标志会被设为 ON。当人机变更群组别完成且更新 RCPG 的内容值后，配方群组别变更状态标志随即设为 OFF。

■ 配方群组别状态寄存器

无论是由控制器指定变更或由人机界面变更配方群组别指定寄存器 (RCPG) 的值，只要有所变更，配方群组别状态寄存器就会更改为新值，以方便控制器通过此寄存器得知目前的配方群组别编号。

配方组别状态寄存器 (RBSR)

无论是由控制器指定变更或由人机界面变更配方组别指定寄存器 (RCPNO) 的值，只要有所变更，配方组别状态寄存器就会更改为新值，以方便控制器通过此寄存器得知目前的配方组别编号。

一般控制状态寄存器 2 (GCSR2)

| Bit | 二进制表示法相对位置(x) | 功能 |
|-------|---------------------|----------|
| 0-7 | 0000 0000 xxxx xxxx | 多国语言状态值 |
| 8 | 0000 000x 0000 0000 | 打印状态标志 |
| 9 | 0000 00x0 0000 0000 | 打印换页状态标志 |
| 10-15 | xxxx xx00 0000 0000 | 保留 |

■ 多国语言状态值

表示人机正在显示此指定的语言文字对应的设定值。

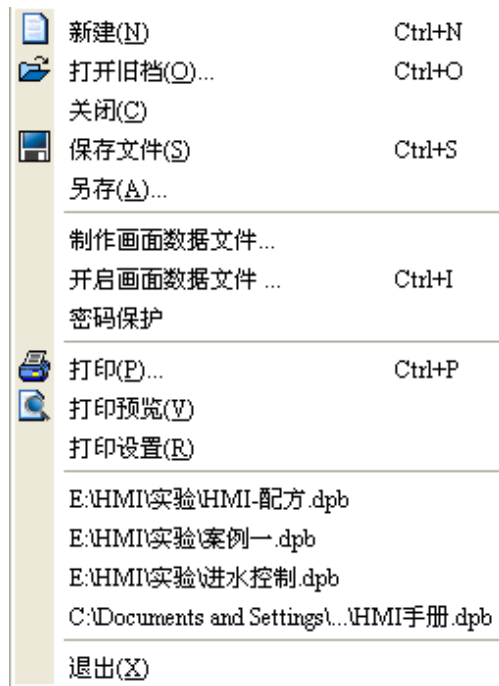
■ 打印状态标志

打印状态标志设为 ON 时，打印机正在打印当时人机所显示的画面或正在打印所编辑排版的画面；打印状态标志设为 OFF 时，打印机无动作。


■ 打印换页状态标志

打印换页状态标志设为 ON 时，打印机正在执行纸张自动退纸换页；打印换页状态标志设为 OFF 时，打印机无动作。

3.5 功能菜单—文件



3.5.1 新建

开启一个新项目时，可直接点击文件里面的新建选项，或是点击工具栏中的  图标，或是使用系统所设定的热键 Ctrl+N。

如果是新开启并且没有其他旧项目的情况下，选择新建后，将会直接出现新项目对话框（图 3-5-1）。



图 3-5-1 新项目对话框

如果视窗中有其他的项目时，那么再开新项目中，程序会先询问是否要储存文件，这时将

出现储存对话框（图 3-5-2），用户可以选择储存或是不储存之后，才会显示新项目对话框，或者按下取消结束开新项目的动作。

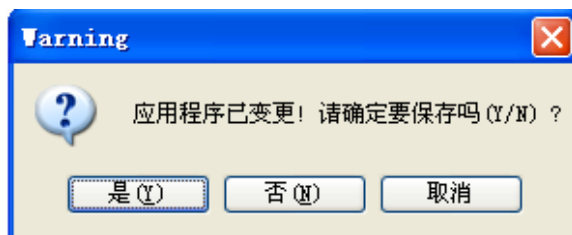


图 3-5-2 储存对话框

输入项目名称及画面名称，并选择人机界面种类与 Base Port 控制器型号（图 3-5-3）后按下确定的按钮。

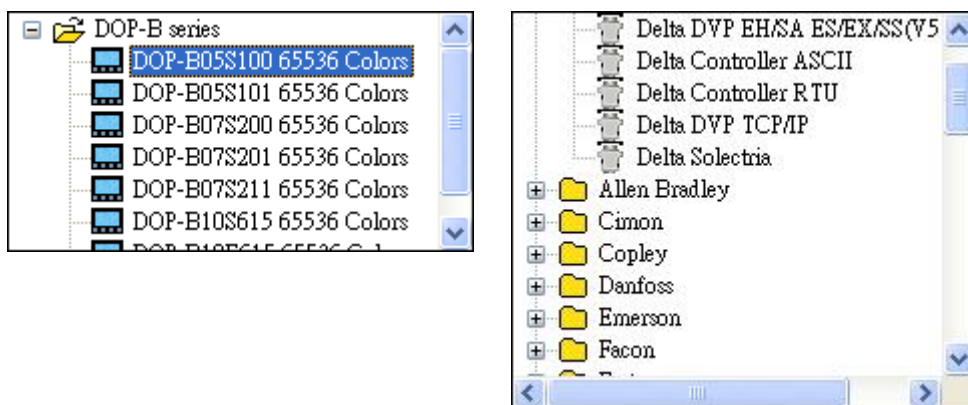



图 3-5-3 选择人机界面种类与 Base Port 控制器型号

3.5.2 打开旧档

打开已储存的项目文件。可在文件里面的”打开旧档”选项中选择或是按下工具栏  中的图标，或是使用系统所设定的热键 Ctrl+O。

如果视窗中已有编辑修改过的项目时，那么在打开已有文件前，程序会先询问是否要储存文件（图 3-5-2），这时将出现储存对话框，用户可以选择储存或是不储存之后，才会显示打开已有文件对话框（图 3-5-4）。

如果储存完毕或是无其他旧项目，则会直接出现打开已有文件对话框（图 3-5-4）。

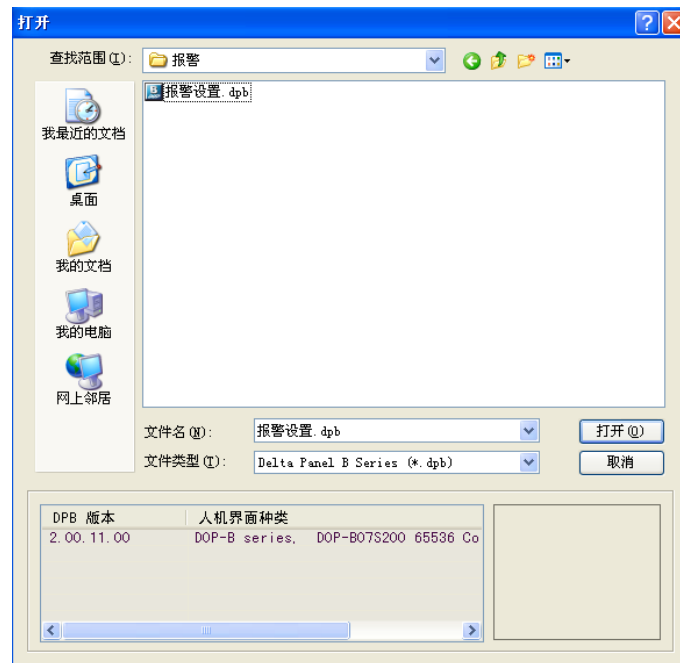


图 3-5-4 打开已有文件对话框

3.5.3 关闭

结束目前编辑的项目文件。只能直接选取文件中的关闭选项。

如果视窗中已有编辑修改过的项目时，将会出现储存对话框（图 3-5-5），并询问是否要储存此项目之后，才会关闭此项目，您也可以按下取消终止关闭文件的动作。

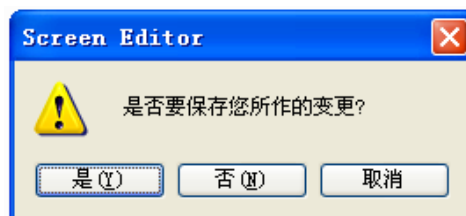



图 3-5-5 储存对话框

3.5.4 保存文件

将目前编辑的项目，直接储存到磁盘中已指定的文件名称，其后缀名为 .dpb。可点击文件中的储存文件选项或是直接点击工具栏中的  图标，或是使用系统所设定的热键 Ctrl + S。如果此项目为新开的项目，将会出现另存对话框（图 3-5-6）。如果为旧文件则立刻储存，并不会出现图 3-5-6 的对话框。

3.5.5 另存

将目前编辑的画面资料储存到磁盘中另外指定的文件名称。此项只能选择文件中的另存选项，无其他方法可以选择。不管是新文件或是旧文件，都会出现另存对话框（图 3-5-6），储存完毕后项目的路径会被指定到新储存的路径。除非有另选后缀名的名称，否则后缀名一律为 dpb。

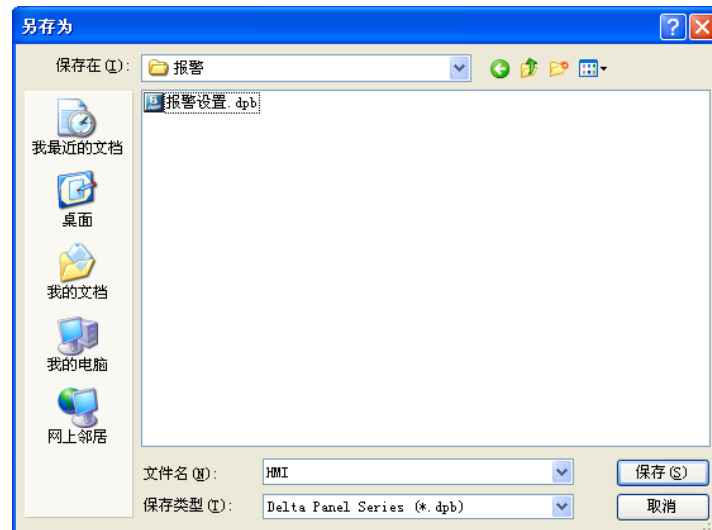


图 3-5-6 另存对话框

3.5.6 制作画面数据文件

在选择此功能菜单之前,要先将目前编辑的画面先作编译。若没有编译, Screen Editor 无法制作画面数据文件,这时会显示错误的信息对话框(图 3-5-7)。编译后按下此功能菜单,将会出现目录选择对话框,决定目录名称后, Screen Editor 会将编译后的画面数据文件复制至指定的目录下,通常这个目录为读卡机上一个支援 SD 卡的磁盘代号或是一个 U 盘(图 3-5-8)。制作好的 SD 卡或是 U 盘可直接插入人机界面,人机会直接读取其中的资料开机。

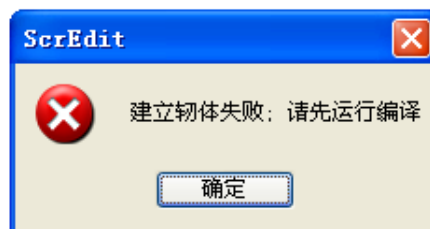


图 3-5-7 制作画面数据文件发生错误的信息对话框

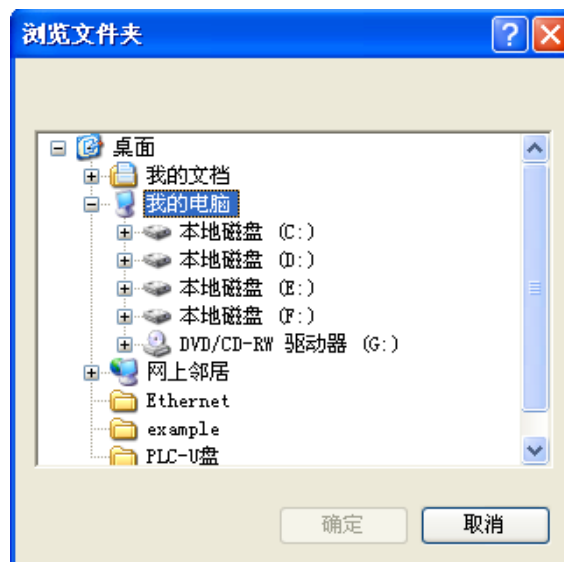


图 3-5-8 选择制作画面数据的目录对话框

3.5.7 开启画面数据文件

通过此功能可以开启制作好的画面数据文件。

3.5.8 密码保护

由【文件】功能中的选项去选取，启动或是关闭密码保护时，系统都会通知用户（图 3-5-9、图 3-5-10）。如果此项有出现 的符号，表示所储存的文件（所编辑的元件文件 .dpb 档）会有密码保护。当开启此文件时必须输入密码，密码错误的话将无法开启文件，而该密码为【选项】功能中【设置模块参数】里的最高权限密码（图 3-5-11）。

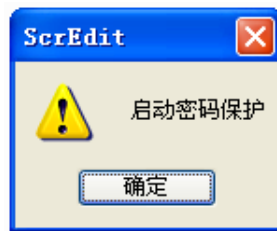


图 3-5-9 密码保护启动



图 3-5-10 密码保护关闭

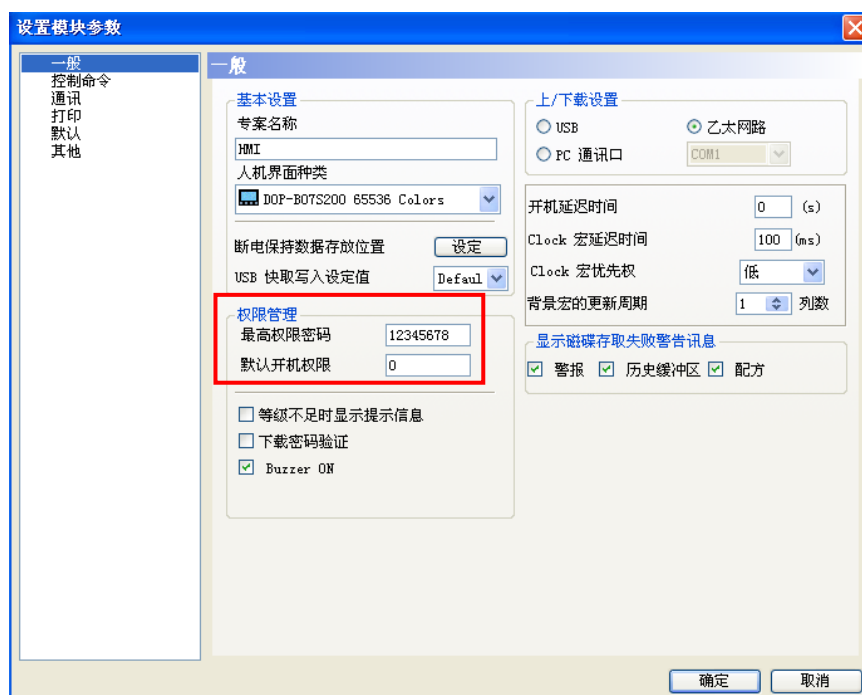



图 3-5-11 设定密码的地方 — 权限管理

3.5.9 打印

将目前编辑的画面输出至打印机。可点击文件里面的打印选项或是点击工具栏中的  图标，或是使用系统所设定的热键 Ctrl + P。

3.5.10 打印预览

只能由文件的项目里面选取，此功能可以预览打印时整页的图形。

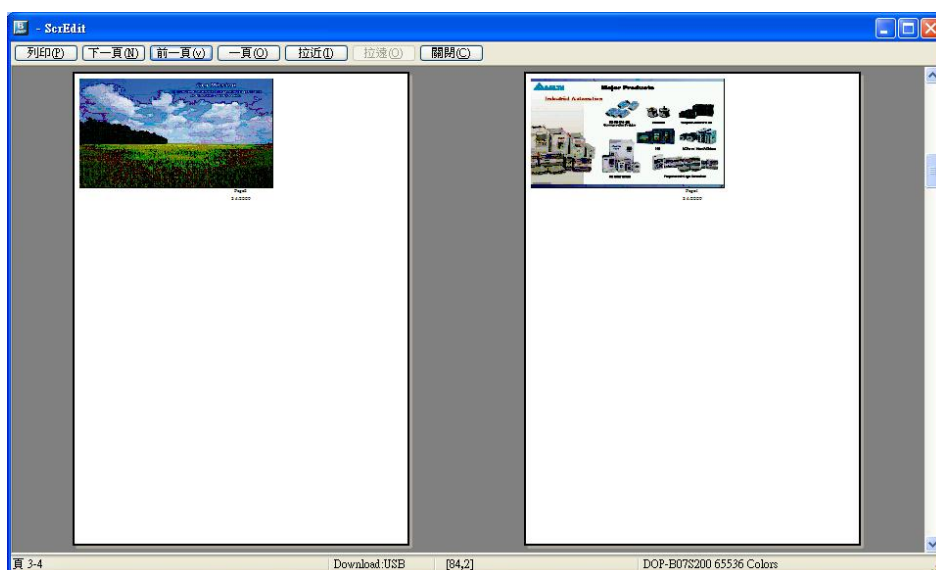


图 3-5-12 打印预览

3.5.11 打印设置

只能由文件的项目里面选取。此功能可以设定打印机的属性,包含纸张种类,打印方向等等 (图 3-5-13)。



图 3-5-13 打印机属性设置对话框

3.5.12 最近使用文件清单

此处存放着最近使用过的四个项目路径，点击其中一个后，其功能就有如打开已有文件一般，用户可参考之前的打开旧档说明（图 3-5-14）。若路径过长，以 … 显示其路径名，并仍保留 dpb 文件名全名称

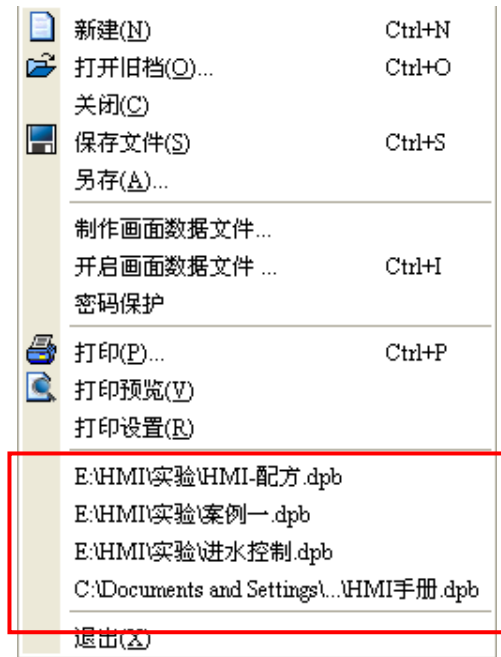


图 3-5-14 最近使用文件清单


3.5.13 退出

结束应用程序并且保存文件。同样也只能通过点击文件之后进入选项选取。如果文件有变更或是尚未存档，则会出现对话框（图 3-5-2），询问是否储存变更。此时如果选择取消，程序便不会结束；如果是选择是（Y）或是否（N）之后，都会关闭程序；如果是新编辑的项目选择：是（Y），便会出现另存对话框（图 3-5-4）之后，选保存或是取消都会关闭程序。


3.6 功能菜单—编辑




3.6.1 撤消

复原上一次编辑的动作。除了进入编辑选择复原的选项外，也可以点击工具栏  的图标或是使用系统预定的热键 Ctrl + Z，在视窗右下角输出栏中记录了所做的动作。


3.6.2 重复

重复前一个复原的动作。此动作可以选取编辑里面的选项，也可以点击工具栏里面  的图标或是使用系统设定的热键，热键为 Ctrl + Y。


3.6.3 剪切

将所选的元件剪下来，以便贴上新的位置。除了可以选择编辑里面的剪切选项外，还可以选择点击工具栏  的图标，或是使用系统设定的热键 Ctrl + X。

3.6.4 复制

可以将所选定的元件复制起来，那么便不需要再重新制作一个新的元件，只要贴上去就可以得到一个一模一样的元件了，除了可以选择编辑里面的复制选项外，还可以选择点击工具栏  的图标或是使用系统设定的热键 Ctrl + C。

3.6.5 粘贴

贴上已复制的元件，除点击编辑选项外，可点击工具栏的  图标，或是使用系统设定的热键 Ctrl + V。

3.6.6 删除元件

删除选取的元件。点击元件后，在编辑中选取删除元件，或是使用系统设定的热键 Del。

3.6.7 全选

将所有画面上的元件作圈选的动作，可选取编辑里面的全选。全选时，最后一个被建立的元件(亦即被建立显示在顶端的)是呈现白框蓝底的方块，其余的则会出现黑框白底；这是为了提供”对齐”跟”使同大小”功能的基准。也就是说，同时选择 2 个以上的元件时，会有一个**基准元件**，所有的”对齐”跟”使同大小”功能会以它为参考基准。全选热键为 Ctrl + A。

3.6.8 查找

搜寻指定的项目，可选取编辑里面的查找。先输入搜寻的内容，然后在查找选项中选择目前画面以及全部画面。查找类型可以选择寻找元件的文字、读取地址、写入地址或是所有存储器地址(图 3-6-1)。确定查找类型后，按下查找，系统便依照设定开始寻找符合的内容。找到后，会将所找到的元件输出到输出栏的选项，点击输出栏的选项，光标会自动选定到此元件(图 3-6-2)。热键为 CTRL + F。

查找功能说明

查找内容：输入您的元件所要搜寻的资料内容。

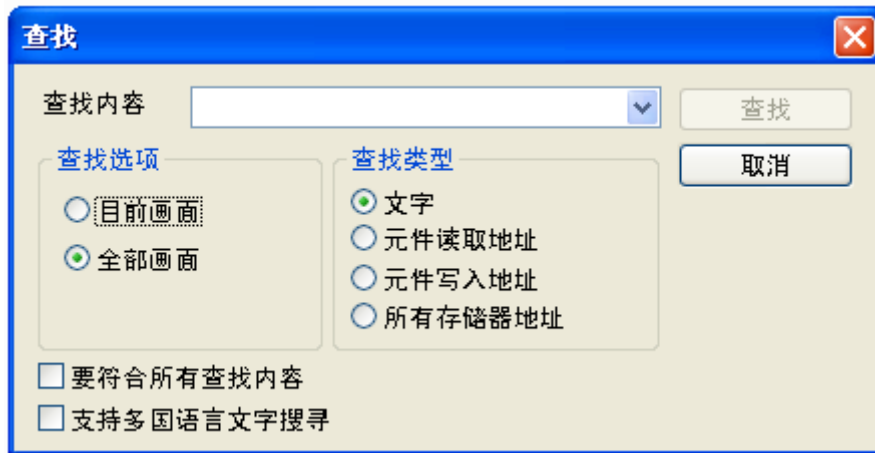


图 3-6-1

查找选项：目前画面

查找过程中仅查找目前编辑的画面，并比对此画面的所有元件，符合的会被输出到输出栏的视窗中。您可以在输出的视窗里以双击的方式来自动圈选被查找到的元件。如下图所示：

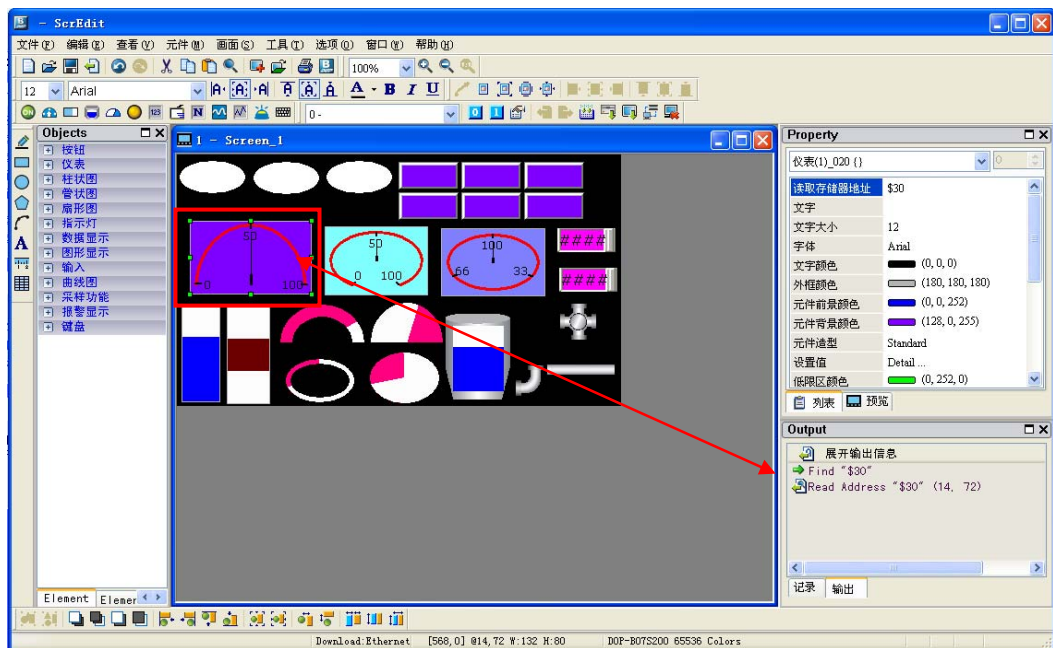


图 3-6-2

全部画面

查找过程中会查找所有的画面，并比对所有元件，符合的会被输出至输出栏的视窗中。同样地您也可以输出的视窗里以双击的方式来自动圈选被查找到的元件上。

查找类型：文字

比对元件输入文字。

元件读取地址

比对元件输入的读取地址。

元件写入地址

比对元件输入的写入地址。

所有存储器地址

比对元件的读取与写入地址。

选项：其中若勾选**要符合所有查找内容**，寻找时会比对所有输入的内容，若没勾选只要所输入的内容有部份吻合就算找到。而**支持多国语言文字搜寻**选项只有查找类型为文字时才会被开启，若被勾选，比对方式将不被限定只找目前编辑语言的文字，而是所有语言的文字将会被比对。

3.6.9 替换

替换指定的项目，可选取编辑里面的替换。分别输入查找的内容以及替换内容，然后在查找选项中选择目前画面或是全部画面。替换类型可以选择替换文字、读取地址或是写入地址。数值单位及过滤条件是当替换类型为读取地址或是写入地址才会开启，数值单位可以选择 BIT、WORD 或是 DWORD，过滤条件可选择要替换地址的是在元件、宏、控制区/状态区、历史缓冲区、警报、还是配方。热键为 CTRL + R。

替换功能说明

查找内容：输入您的元件所要查找的资料内容。

替换内容：输入您的元件所要替换的资料内容。



图 3-6-3

查找选项：目前画面

查找过程中仅查找目前编辑的画面，并比对此画面的所有元件，符合的会依序被替换。

全部画面

查找过程中会查找所有的画面，并比对所有元件，符合的会依序被替换。

替换类型：文字

替换寻找到符合的元件文字。

读取地址

替换寻找到符合的元件读取地址。

写入地址

替换寻找到符合的元件写入地址。

数值单位：Bit、WORD、DWORD

数值单位是当替换类型为读取地址或是写入地址才会开启，可以选择 BIT、WORD 或是 DWORD。

过滤条件：元件、宏、控制区/状态区、历史缓冲区、警报、配方

过滤条件是当替换类型为读取地址或是写入地址才会开启，可以选择元件、宏、命令区、历史缓冲区、警报、配方。

替换/全部替换：以替换 WORD 写入地址为例，并查找所有画面。将读取地址\$0 替换为 \$1234，按下**替换**按钮后(图 3-6-4)，系统会自动寻找符合该条件下的元件并显示在图 3-6-5 视窗中，询问您哪些元件要执行替换的动作，用户可依照其需要自行勾选所需要的元件进行替换的动作。或在图 3-6-4 视窗中直接选择全部替换，直接替换所有符合条件的元件。



图 3-6-4 替换视窗

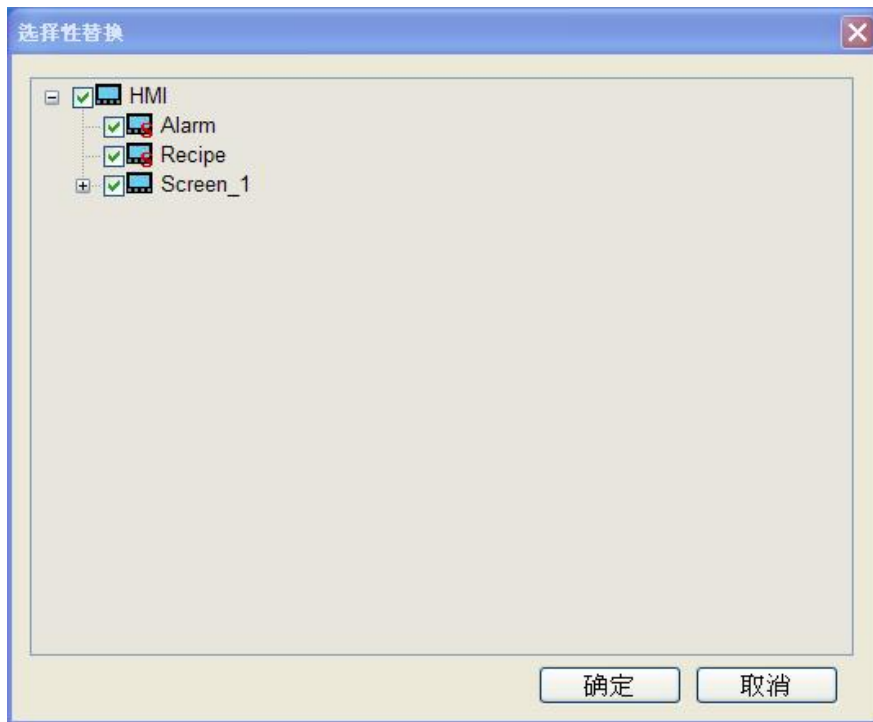


图 3-6-5


3.6.10 站号替换

可替换指定的站号。




图 3-6-6

3.6.11 组合

将所选择的元件设成组合。先选取要设定组合的所有元件之后，选取编辑里面组合的选项或是点击工具栏的  图示。设定组合后，不管有多少的元件一律当成是同一元件处理并且移动时一起移动，并且可以进行组合元件大小的缩放。

3.6.12 解除组合


将原本有设组合的组合解除。可以选取编辑选项里面的解除组合或是直接点取工具栏上的  图标。


3.6.13 层次


设定元件里面的层次，而元件的层次会随着建立元件的先后次序而不同。而层次的不同将会影响被遮盖的顺序，用户可选项里面的层次或是点击工具栏里面的图标




其中


 上推至最顶层




 送至最下层



 上推一层

 后推一层

3.6.14 对齐

将元件坐标对齐。用户可选项里面的对齐或是点击工具栏里面的 (：靠左对齐)

(：靠右对齐) (：靠上对齐) (：靠下对齐) (：垂直置中) (：水平置中)

(：横向等间距) (：纵向等间距)；靠左对齐、靠右对齐、靠上对齐以及靠下对齐，

都是需要圈选两个以上的元件去执行才会有动作。因为其对齐方式为所圈选基准元件的最左、最右、最上、最下的坐标来作为所有元件的新坐标；而垂直置中跟水平置中，就可以单一元件使用。使用后会根据元件的长宽将坐标改为新的位置，在垂直置中使得元件上下的距离一样，水平置中则使得左右的距离一样。横向等间距与纵向等间距则必须圈选三个元件以上，程序会计算出横向元件彼此间的等宽距离，并重新排列。若是执行纵向等间距，程序会计算出纵向元件彼此间的等高距离，并重新排列。

3.6.15 使同大小

将元件大小对齐。用户可选项里面的使同大小或是点击工具栏里面的图标。这里必须选择两个以上的元件，以所选到的基准元件当标准，执行造成相同宽度、造成相同高度或是造成相同尺寸。

3.6.16 文字处理

除了可以在此选择元件文字的状态外，也可以利用文字工具栏设定。另外在此可以选取从文本库汇入文字的方式将文字内容输入到元件上。在文本库的对话框里您可以决定是否要使用文本库里所设定的字体，若选择使用文本库里所设定的字体则在汇入文字后，相对的该元件的文字字体也同时被设定为文本库上的字体。文本库的建立请参照【选项】>>【文本库】。

3.6.17 位图处理

选取编辑里面的位图处理选项。而前面的 标志表示此元件图形的状态。除了可以在此选择元件图形的状态外，也可以利用图形工具栏设定。

3.6.18 多重复制

选取一个特定元件来做多重复制。点击后会出现视窗（图 3-6-7），在复制数量上面选取水平个数 X 以及垂直个数 Y 来得到总数为 $X * Y$ 的元件。如果只想单一选择水平或是垂直方向的复制，可以利用前面的打勾选项来启动或是关闭。而间隔则是输入每个元件间的间隔，输入后会在新元件复制完成后自动依设定隔开。偏移地址此一选项，则是依照所设定的地址，来依序依照水平或是垂直方向来递增。如果元件以 word 为单位，就以 word 单位来递增；如果是 bit 单位，则依照 bit 单位来递增，反之若指定递减选项，则是依照所设定的地址，来依序依照水平或是垂直方向来递减。请参考范例（图 3-6-8、图 3-6-9）。



图 3-6-7 多重复制对话框

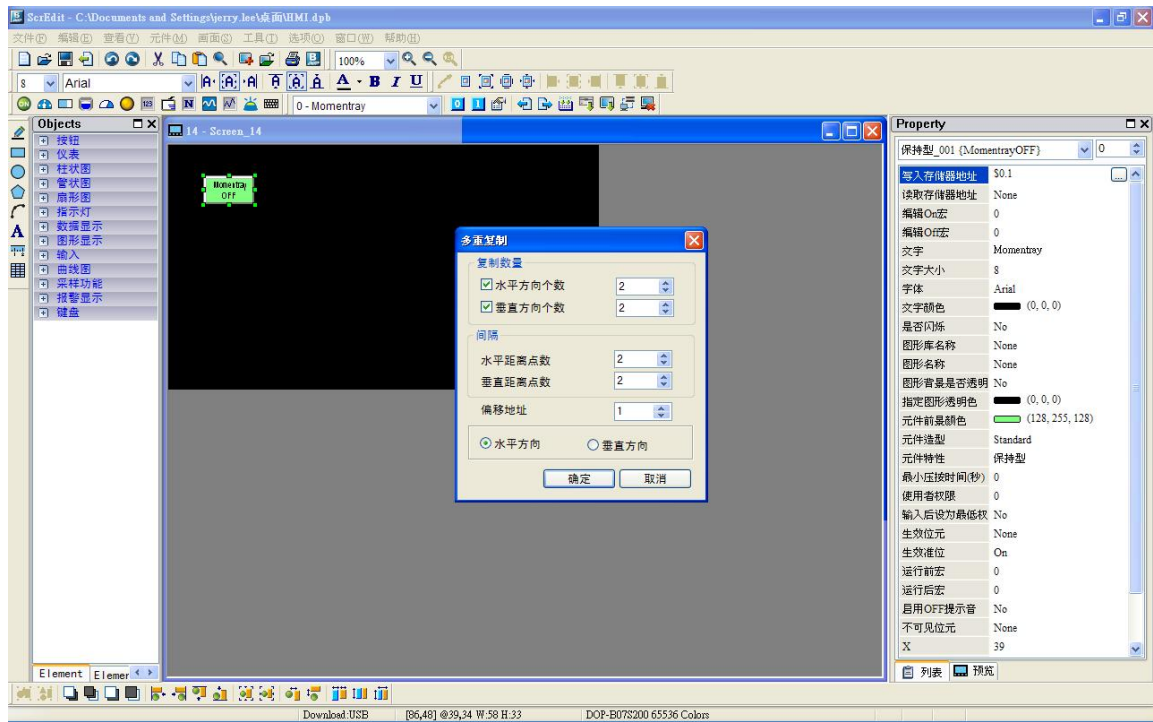


图 3-6-8 多重复制范例

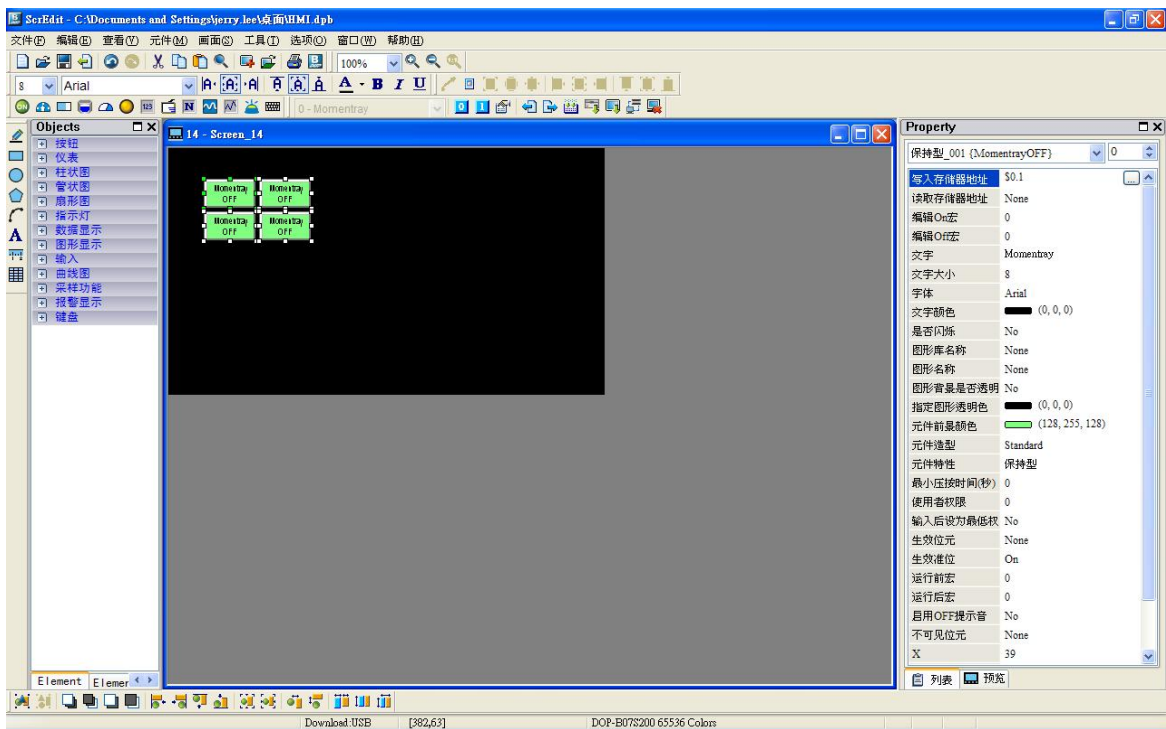


图 3-6-9 多重复制范例

3.7 功能菜单—查看



3.7.1 工具栏

标准工具栏



| 图形 | 名称 | 功能 |
|----|-------|----------------------------------|
| | 新建 | 开启新的编辑文件。 |
| | 打开旧档 | 开启已有的编辑文件。 |
| | 保存文件 | 保存正在编辑的文件。 |
| | 汇出 | 将目前编辑区的画面数据以 BMP 图档的格式保存。 |
| | 撤消 | 撤消上一个动作（有些动作无法撤消）。 |
| | 重复 | 重复上一个动作。 |
| | 剪切 | 将元件剪下。 |
| | 复制 | 复制所选取的元件。 |
| | 粘贴 | 粘贴所复制或剪切的元件。 |
| | 查找 | 查找特定的文字、写入地址以及读取地址。 |
| | 建立新画面 | 开启一个新画面。 |
| | 打开旧画面 | 打开旧画面。 |
| | 打印 | 打印编辑内容。 |
| | 关于 | 版本说明。 |

表格 3-7-1 标准工具栏

状态栏

此栏会显示目前编辑的状态（图 3-7-1）。

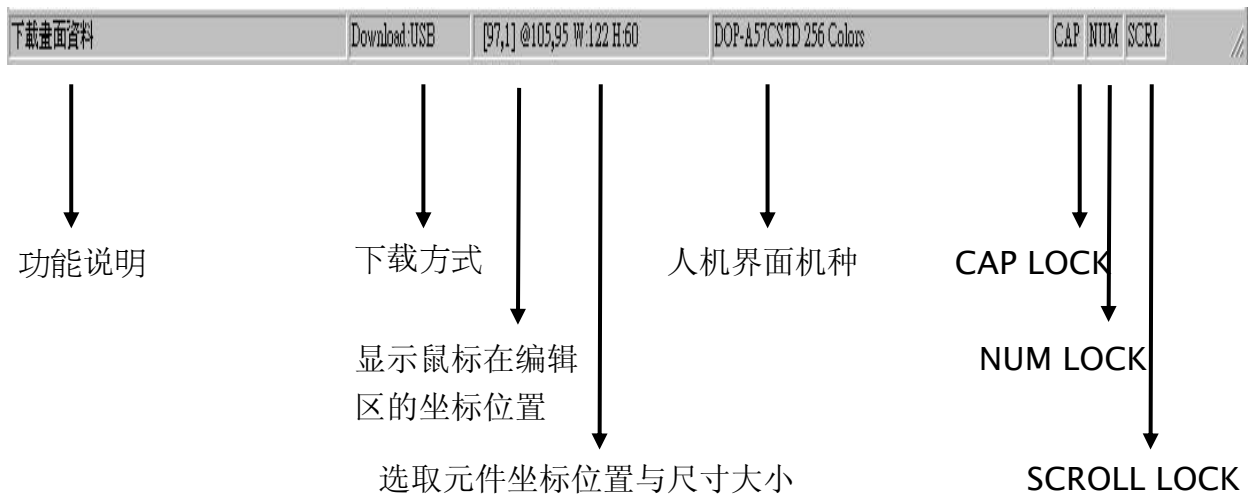
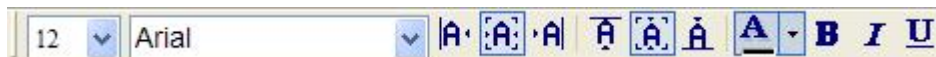


图 3-7-1 状态栏

文字工具栏



| 图形 | 名称 | 功能 |
|----|--------|-----------|
| | 字体大小 | 显示并改变文字大小 |
| | 字体 | 所选择的字体 |
| | 文字靠左对齐 | 文字靠左 |
| | 文字水平置中 | 文字置中 |
| | 文字靠右对齐 | 文字靠右 |
| | 文字靠上对齐 | 文字靠上 |
| | 文字垂直置中 | 文字垂直置中 |
| | 文字靠下对齐 | 文字靠下 |
| | 字体颜色 | 改变字体颜色 |
| | 文字粗体 | 文字粗体 |
| | 文字斜体 | 文字斜体 |
| | 文字加底线 | 文字加底线 |

表格 3-7-2 文字工具栏

图形工具栏



| 图形 | 名称 | 功能 |
|----|------------------|--|
| | 选取图形透明色 | 以吸管的方式选择图形上的哪个颜色要变成透明 |
| | 处理所有状态 图形模式切换 | 若此功能被启动，执行元件上图形的缩放功能并不仅仅只有目前的状态，元件的所有状态的汇入图形者皆会完成指 |

| 图形 | 名称 | 功能 |
|---|----------|------------------|
| | | 定的缩放功能 |
|  | 延展图形 | 将所选图形延展至该元件范围 |
|  | 保持比例延展图形 | 按照原图形的长宽比作等比例的缩放 |
|  | 实际图形大小 | 将所选图形恢复原实际尺寸大小 |
|  | 图形靠左对齐 | 将所选图形靠左 |
|  | 水平置中 | 将所选图形置中 |
|  | 图形靠右对齐 | 将所选图形靠右 |
|  | 图形靠上对齐 | 将所选图形靠上 |
|  | 图形垂直置中 | 将所选图形垂直置中 |
|  | 图形靠下对齐 | 将所选图形靠下对齐 |

表格 3-7-3 图形工具栏

元件工具栏



| 图形 | 名称 | 展开项目 |
|---|----|--|
|  | 按钮 | <ul style="list-style-type: none">  设On  设Off  保持型  交替型  复状态  设值  设常数值  加值  减值  换画面  回前页  系统时间日期  设置密码表  密码输入  调整对比亮度  设为最低权限  系统目录  输出报表  撷取画面  移除USB  汇入/汇出配方  触碰校正 |
|  | 仪表 | <ul style="list-style-type: none">  仪表(1)  仪表(2)  仪表(3) |

| 图形 | 名称 | 展开项目 |
|---|------|--|
|  | 柱状图 | <ul style="list-style-type: none">  一般型  差量型 |
|  | 管状图 | <ul style="list-style-type: none">  管状图(1)  管状图(2)  管状图(3)  管状图(4)  管状图(5)  管状图(6)  管状图(7) |
|  | 扇形图 | <ul style="list-style-type: none">  扇形图(1)  扇形图(2)  扇形图(3)  扇形图(4) |
|  | 指示灯 | <ul style="list-style-type: none">  状态指示灯  数值范围指示灯  简易指示灯 |
|  | 数据显示 | <ul style="list-style-type: none">  数值显示  文数值显示  日期显示  时间显示  星期显示  一般型信息显示  走马灯信息显示 |
|  | 图形显示 | <ul style="list-style-type: none">  状态图显示  动画  动态线条  动态矩形  动态椭圆形  即时图显示 |
|  | 输入 | <ul style="list-style-type: none">  数值输入  文数字输入  Barcode输入 |
|  | 曲线图 | <ul style="list-style-type: none">  一般曲线图  X-Y曲线图  X-Y即时分布图 |
|  | 取样功能 | <ul style="list-style-type: none">  历史趋势图  历史数值数据表  历史信息表 |

| 图形 | 名称 | 展开项目 |
|---|------|--|
|  | 警报显示 | <ul style="list-style-type: none">  历史报警表  当前报警表  报警频次表  报警信息走马灯 |
|  | 键盘 | <ul style="list-style-type: none">  键盘 (1)  键盘 (2)  键盘 (3) |

表格 3-7-4 元件工具栏所有元件

绘图工具栏



| 图形 | 名称 | 功能 |
|---|-----|-------|
|  | 线 | 线绘图 |
|  | 矩形 | 矩形绘图 |
|  | 圆 | 圆绘图 |
|  | 多边形 | 多边形绘图 |
|  | 弧 | 弧绘图 |
|  | 文字 | 文字方块 |
|  | 刻度 | 绘制刻度 |
|  | 表格 | 绘制表格 |

表格 3-7-5 缩放工具栏

元件规划工具栏



| 图形 | 名称 | 功能 |
|---|------------|---------------|
|  | 目前状态的文字 | 显示目前编辑元件状态的文字 |
|  | 查看状态 0/OFF | 切换并查看状态 0/OFF |
|  | 查看状态 1/ON | 切换并查看状态 1/ON |
|  | 查看所有元件读写地址 | 查看所有元件读写地址 |

| 图形 | 名称 | 功能 |
|---|-----------|--------------------------|
|  | 上一个视窗 | 选择上一个视窗 |
|  | 下一个视窗 | 选择下一个视窗 |
|  | 编译 | 编译所编辑的元件与画面 |
|  | 下载画面数据与配方 | 下载画面数据与配方 |
|  | 下载画面数据 | 下载画面数据 |
|  | 线上模拟 | 在 PC 端测试元件编辑后的档案，必须连接控制器 |
|  | 离线模拟 | 在 PC 端测试元件编辑后的档案，不必连接控制器 |

表格 3-7-6 上层规划工具栏



| 图形 | 名称 | 功能 |
|---|--------|--------------------|
|  | 组合 | 将所圈选的元件设成组合 |
|  | 取消组合 | 将组合取消 |
|  | 上推至最顶层 | 将所选元件上推至最顶层 |
|  | 下推至最底层 | 将所选元件下推至最底层 |
|  | 上推一层 | 将所选元件上推一层 |
|  | 下推一层 | 将所选元件下推一层 |
|  | 靠左对齐 | 所选元件靠左对齐 |
|  | 靠右对齐 | 所选元件靠右对齐 |
|  | 靠上对齐 | 所选元件靠上对齐 |
|  | 靠下对齐 | 所选元件靠下对齐 |
|  | 垂直置中 | 所选元件垂直置中 |
|  | 水平置中 | 所选元件水平置中 |
|  | 横向等间距 | 所选元件横向间距设为相等 |
|  | 纵向等间距 | 所选元件纵向间距设为相等 |
|  | 造成相同宽度 | 将所选元件的宽度设成和基准元件一样 |
|  | 造成相同高度 | 将所选元件的高度设成和基准元件一样 |
|  | 造成相同尺寸 | 将所选元件的长跟宽设成和基准元件一样 |

表格 3-7-7 下层规划工具栏

缩放工具栏



| 图形 | 名称 | 功能 |
|---|------|--|
|  | 显示比例 | 显示缩小放大比例，有 25%、50%、75%、100%、150%、200%、300% |

| 图形 | 名称 | 功能 |
|----|-----|---------------------------|
| | 放大 | 放大显示的比例, 有 150%、200%、300% |
| | 缩小 | 缩小显示的比例, 有 25%、50%、75% |
| | 1:1 | 恢复原始大小比例 100% |

表格 3-7-8 缩放工具栏

3.7.2 属性表

提供各种元件列表的属性设定与编辑画面预览 (图 3-7-2)。



图 3-7-2 属性表：元件列表的属性设定与编辑画面预览

3.7.3 输出

此项功能会将编译时的过程显示在此栏位, 以让用户了解程序编译的结果, 便于追查编辑时发生错误的问题 (图 3-7-3、图 3-7-4、图 3-7-5、图 3-7-6)。



图 3-7-2 记录栏



图 3-7-4 输出栏



图 3-7-5 编译结果输出栏



图 3-7-6 集中错误输出栏

3.7.4 拉近

将整个编辑画面放大，以方便用户编辑。请参考（图 3-7-7、图 3-7-8）。

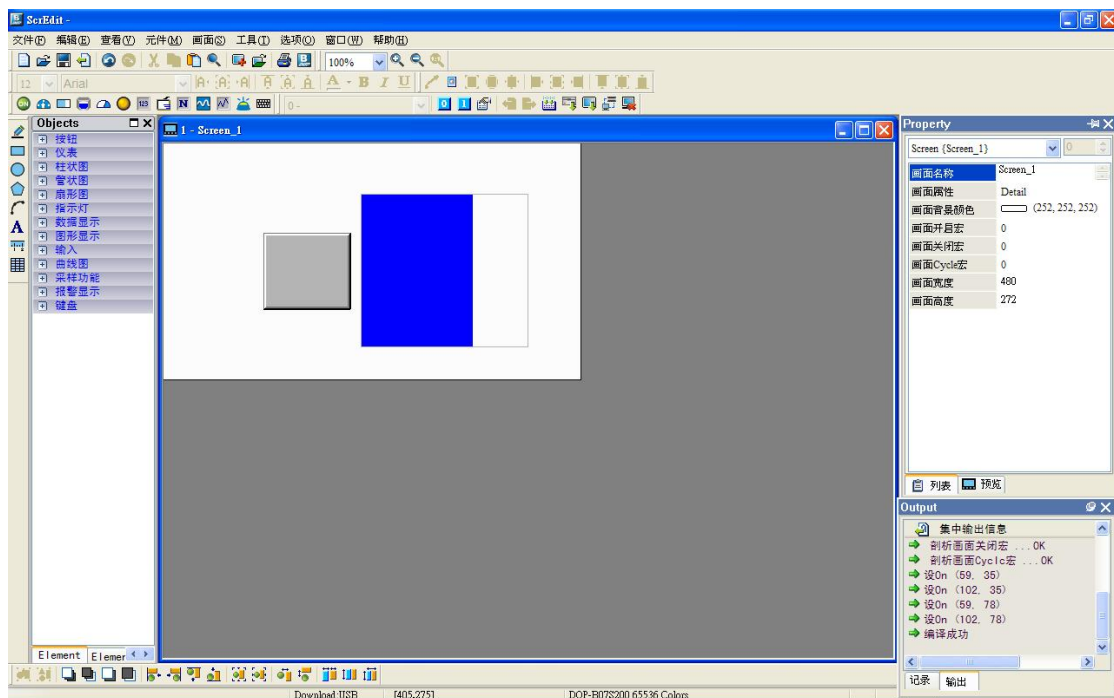


图 3-7-7 未拉近前编辑视窗大小 100%

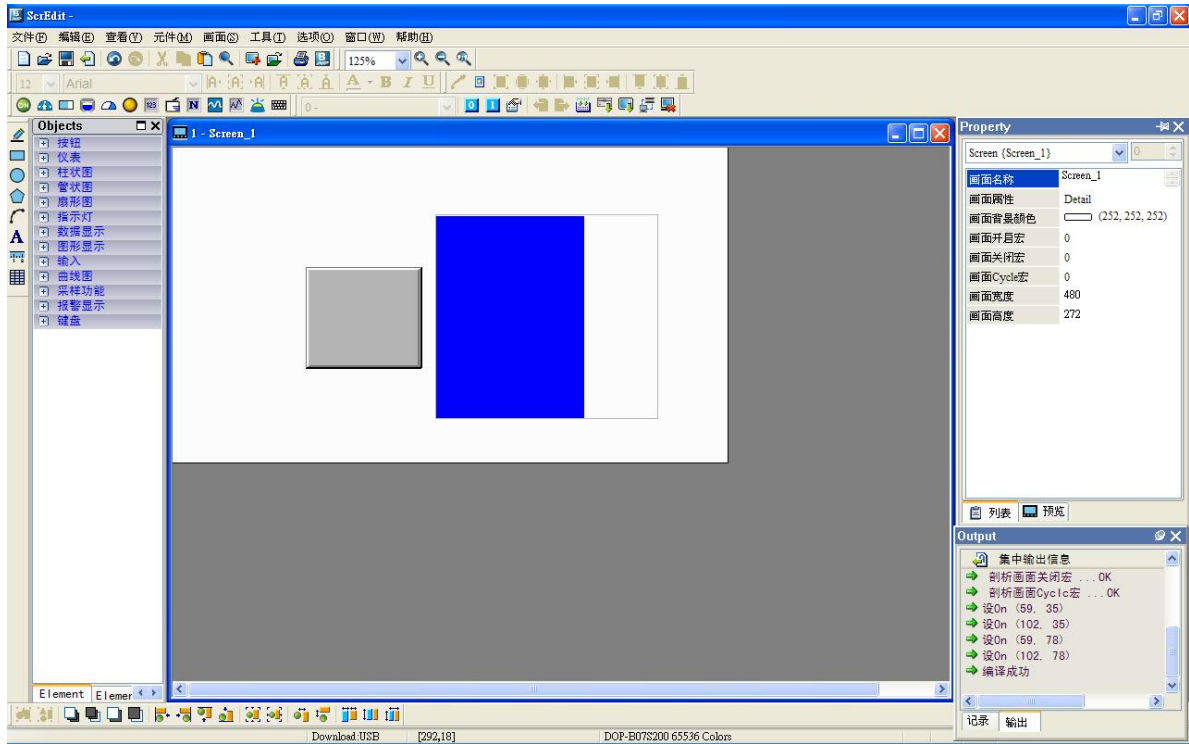


图 3-7-8 拉近后编辑视窗大小 125%

3.7.5 拉远

将整个编辑画面缩小，请参考（图 3-7-9）。

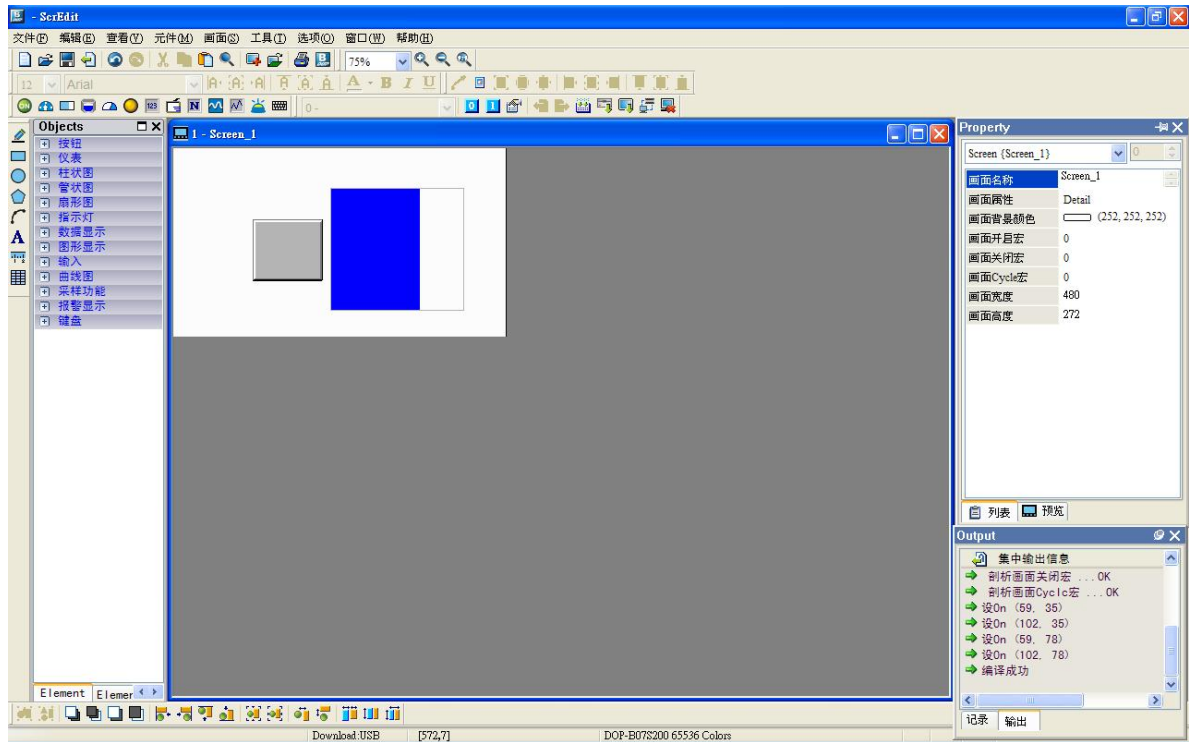




图 3-7-9 拉远后编辑视窗大小 75%

3.7.6 实际大小

立即恢复 100%的比例，此比例是相对于人机屏幕的大小。

不管是拉近或是拉远，其缩放比例从 25%、50%、75%、100%、150%、200%、300%，当然您也可以从查看工具栏上直接按下  拉近或按下  拉远查看或直接选择其缩放比例（图 3-7-10）。

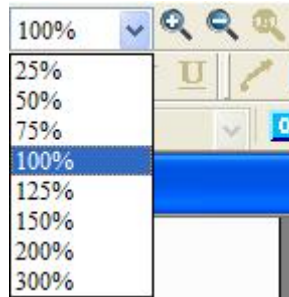


图 3-7-10 选择缩放比例

3.7.7 全屏幕

如图 3-7-11。将编辑的画面以全屏幕显示，并显示此画面有使用的宏

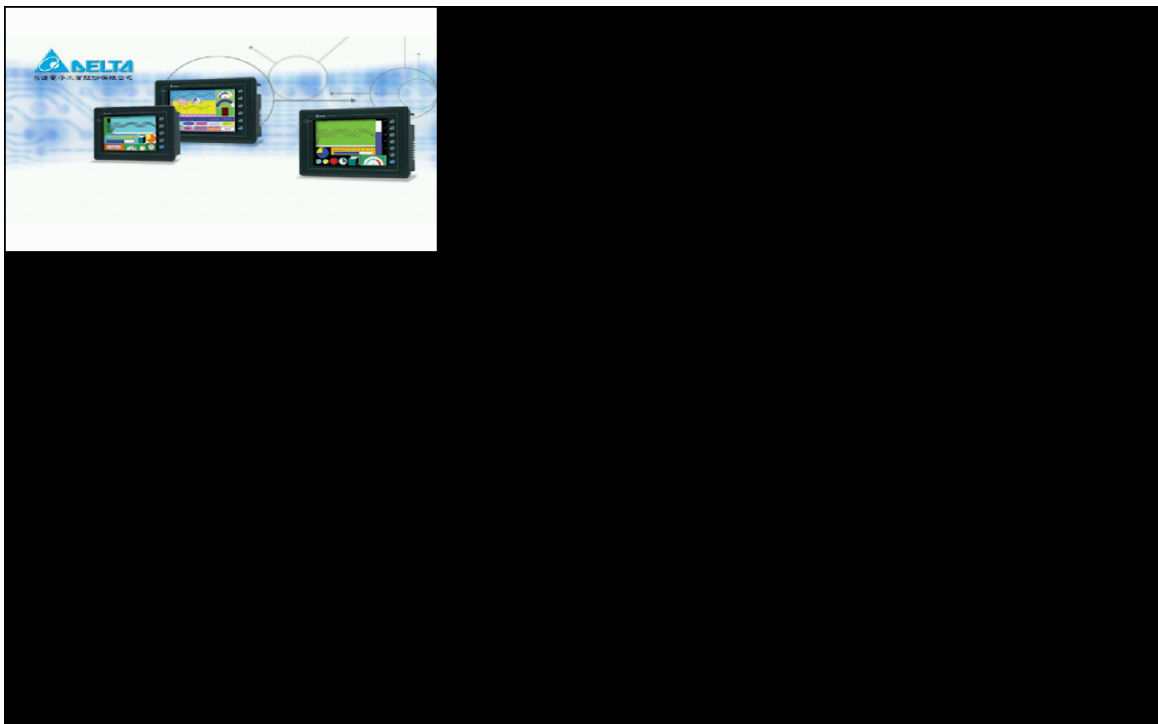


图 3-7-11 选取全屏幕后编辑视窗画面，按 Esc 键或鼠标左键离开

3.7.8 全屏幕及输出点

如图 3-7-12。此与全屏幕功能差别在于此全屏幕及输出点功能会将元件的读写地址一并显示出来，同样地也会将此画面有设定的宏行数显示出来。



图 3-7-12 选取全屏幕及输出点后编辑视窗画面，按 Esc 键或鼠标左键离开

3.7.9 格子设置

其主要用途是帮助用户在编辑时能够很容易的将各元件对齐，而对齐格点的间距可由用户自由设定（图 3-7-13、图 3-7-14）。



图 3-7-13 格子设置对话框

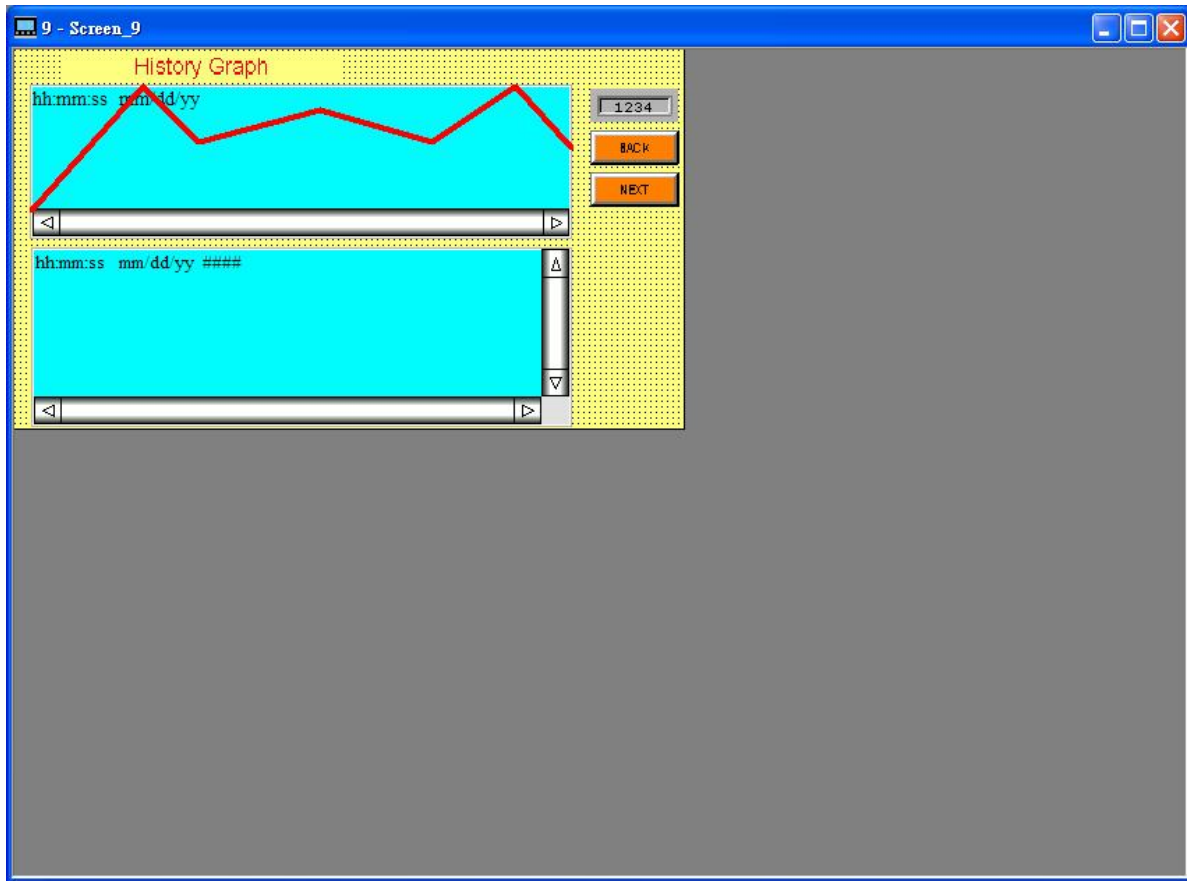


图 3-7-14 编辑元件时启动显示格点的状况

3.7.10 地址映射表

在规划众多读写地址元件时，往往会在规划地址过程中忘记或重覆使用到相同的地址。所以在 Screen Editor 中提供了地址映射表的功能，能查看出目前被选择到的元件其读写地址与哪些元件或者宏或者控制区等等地址有相互关联，如图 3-7-15，程序中警报地址与宏皆使用到内部存储器\$3。



图 3-7-15 地址映射表

3.7.11 元件地址清单

开启元件地址清单，程序会将目前编辑画面上的所有元件作分类，您可以用点击页签的方式切换您想查看的分类。每一分类里会列出所属该分类里元件的相关地址属性（图 3-7-16）（以按钮类别来看分别有写入地址、读取地址、触发地址、生效地址），其中也包括了一些较重要的属性，譬如：触发方式、生效准位等等。这些所列出的栏位您可以直接更改其属性同时您也可以鼠标双击(Double-Click)该列，让程序自动将此元件圈选起来并于属性表编辑该元件详细的属性。

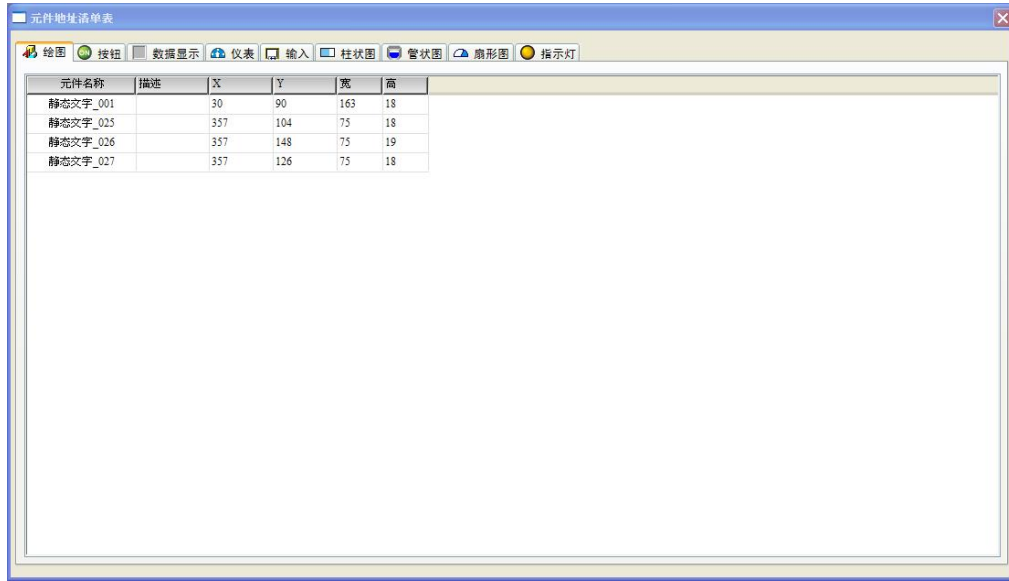


图 3-7-16 元件地址清单对话框

3.7.12 存储器使用清单

存储器使用清单可分为以下四个部分：

1. ROM：实际下载到人机所占的文件空间(包含配方，画面资料等)。
2. SRAM：断电保持资料存放位置预设为存放在 SRAM 静态存储器。当用户如果项目中有规划到历史、警报等资料，则可于此区域查看空间的运用状态。
3. SDARM：人机实际执行项目的地方，用户必须先于项目建立后并进行编译才会得到此信息。
4. External Storage：指外部存储器使用空间。当断电保持资料存放位置设为外部存储设备如 U 盘，此时 U 盘就会将原本存在静态存储器 SRAM 的资料区块搬移到此区域。

| 名称 | 断电保持 | 储存位置 | 档案名称 |
|----|------|----------|------|
| 警报 | No | HMI | |
| 配方 | Yes | HMI | |
| | | HMI | |
| | | USB Disk | |
| | | SD | |

以下举几个例子来说明此一部分。

(1). 画面汇入图形所占用空间

(2). 建立历史资料所占用空间

(1). 画面汇入图形

此部分会使用到 ROM 与 SDRAM 的存储器。

当开启一个新专案时，系统计算出来会使用的存储器空间如上图所示

ROM=3.13%

SDAM=0.71%

当汇入一张图形的时候，存储器的配置就会出现变化，如下图所示。



此时可以注意当汇入图形后存储器空间的变化，

ROM=7.29%

SDAM=3.98%

ROM 的部分多出了 256K (384K-128K)，SDRAM 的部分多出了 600K (600K-0K)。

(2). 建立历史数据

我们使用刚刚的专案继续来规划。

单独看建立历史资料的部分会使用到 SRAM 的存储器。

开启上述专案时，显示的存储器空间为

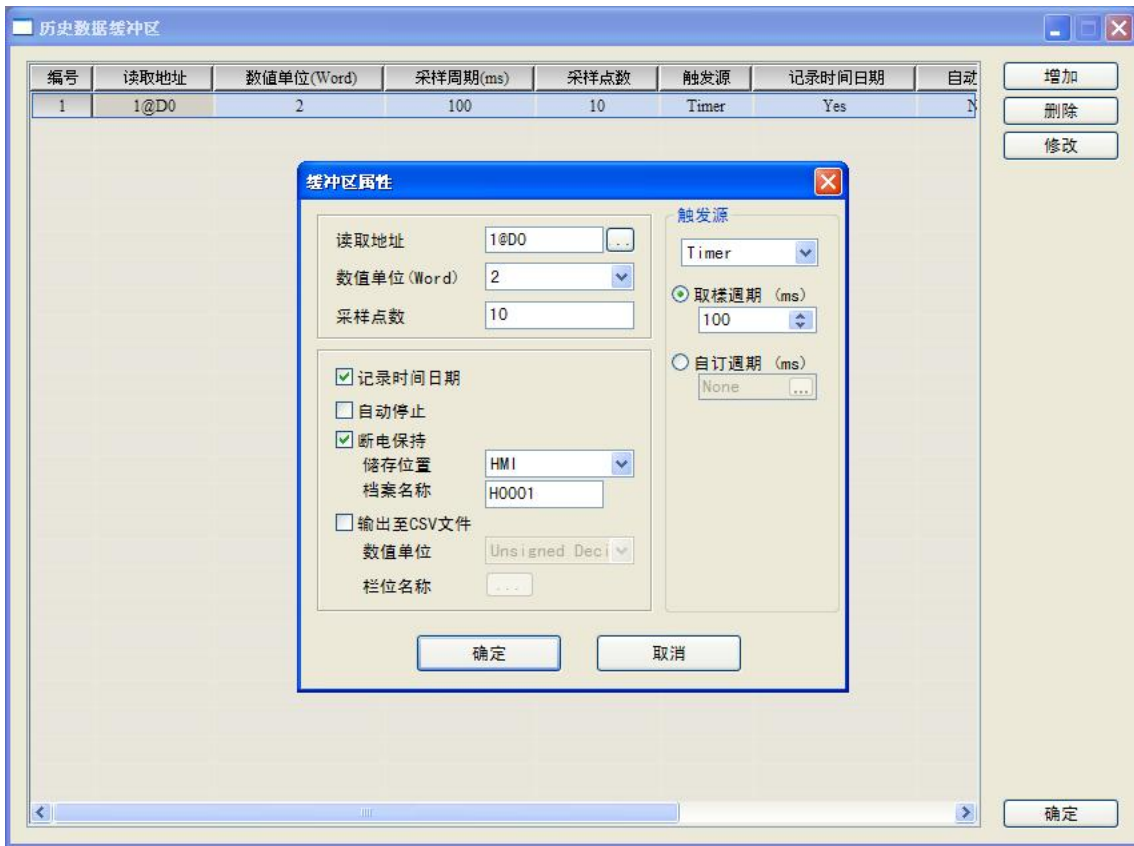
ROM=7.29%

SDAM=3.98%

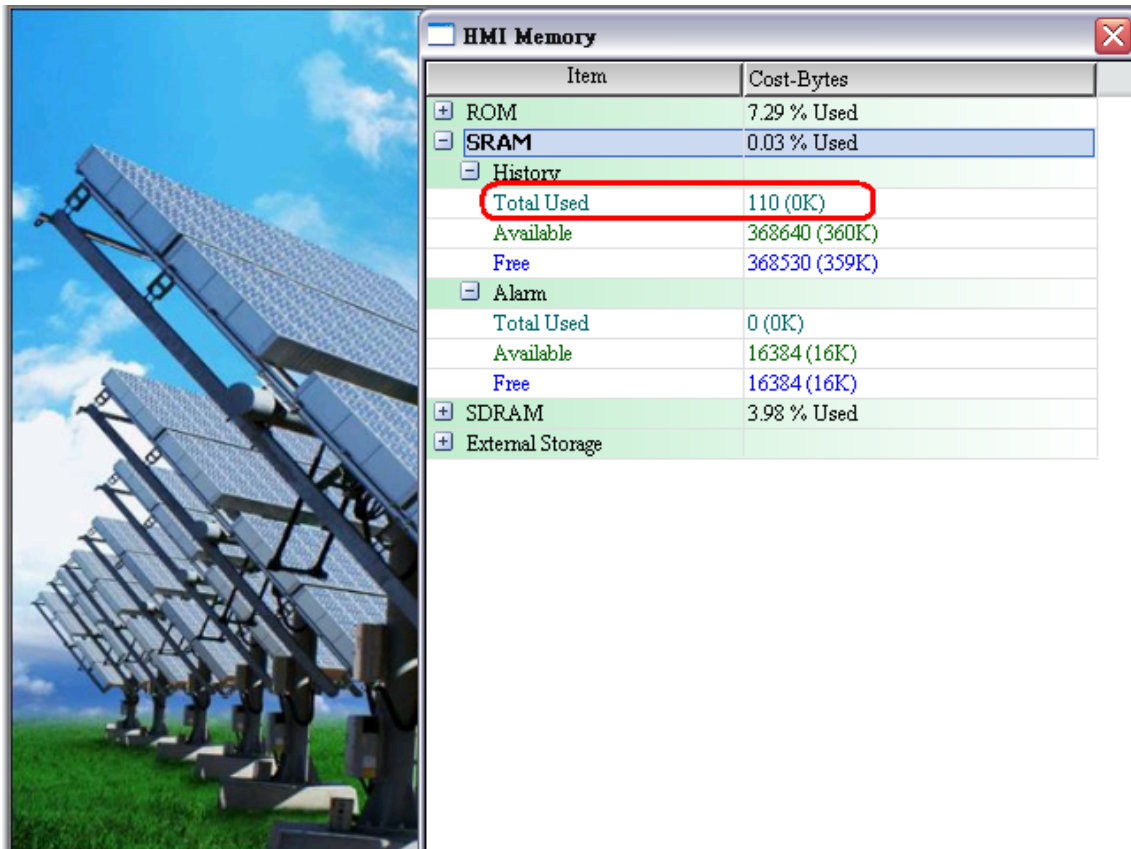
SRAM=0%

External Storage=0%

当建立一个历史资料缓冲区条件如下，



长度为 2word、取样点数 10 笔、记录时间日期并勾选将历史资料存放在断电保持区。断电保持资料存放位置目前设定在 SRAM 静态存储器。设定完成后，内存的配置就会出现以下变化。



每储存一笔皆须记录时间日期需占用 6Byres

每储存一笔会占用 2word (4Byres)

SRAM= (6+4) x 10 笔+10 Byres (系统使用) =110 Byres

此时可以看到存储器空间为

ROM=7.29%

SDAM=3.98%

SRAM=0.03%

External Storage=0%

当将断电保持资料存放位置设为 U 盘时

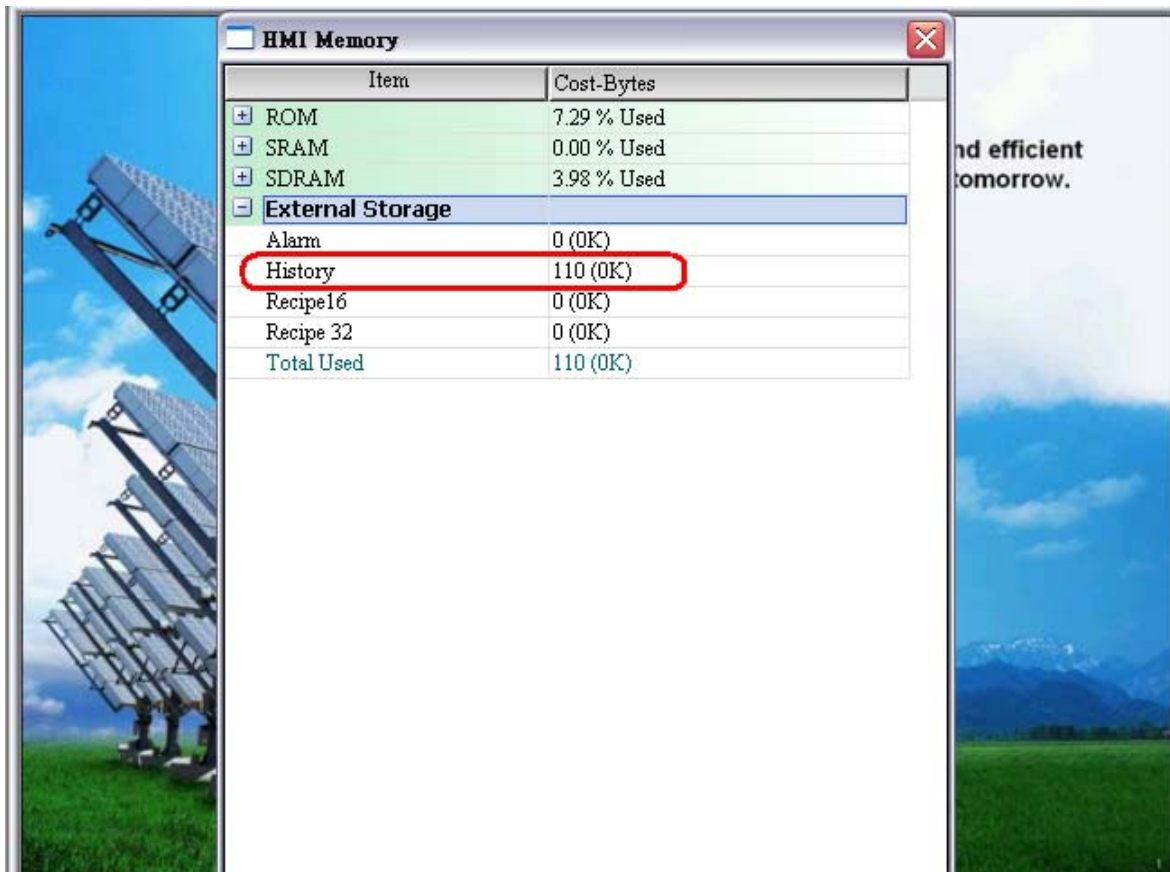
存储器的变化如下

ROM=7.29%

SDAM=3.98%

SRAM=0%

External Storage=110Bytes



详细关于警报与历史区域空间使用的计算，请参阅本手册第 3.11.3 节中有详述。

3.8 功能菜单—元件

3.8.1 元件的建立



在编辑元件时，可使用下列三种方式开启元件功能：

- 在画面编辑区按下鼠标右键，如图 3-8-1。
- 点选元件功能菜单，如图 3-8-2。
- 点选元件工具栏，如图 3-8-3。
- 点选元件库，如图 3-8-4。

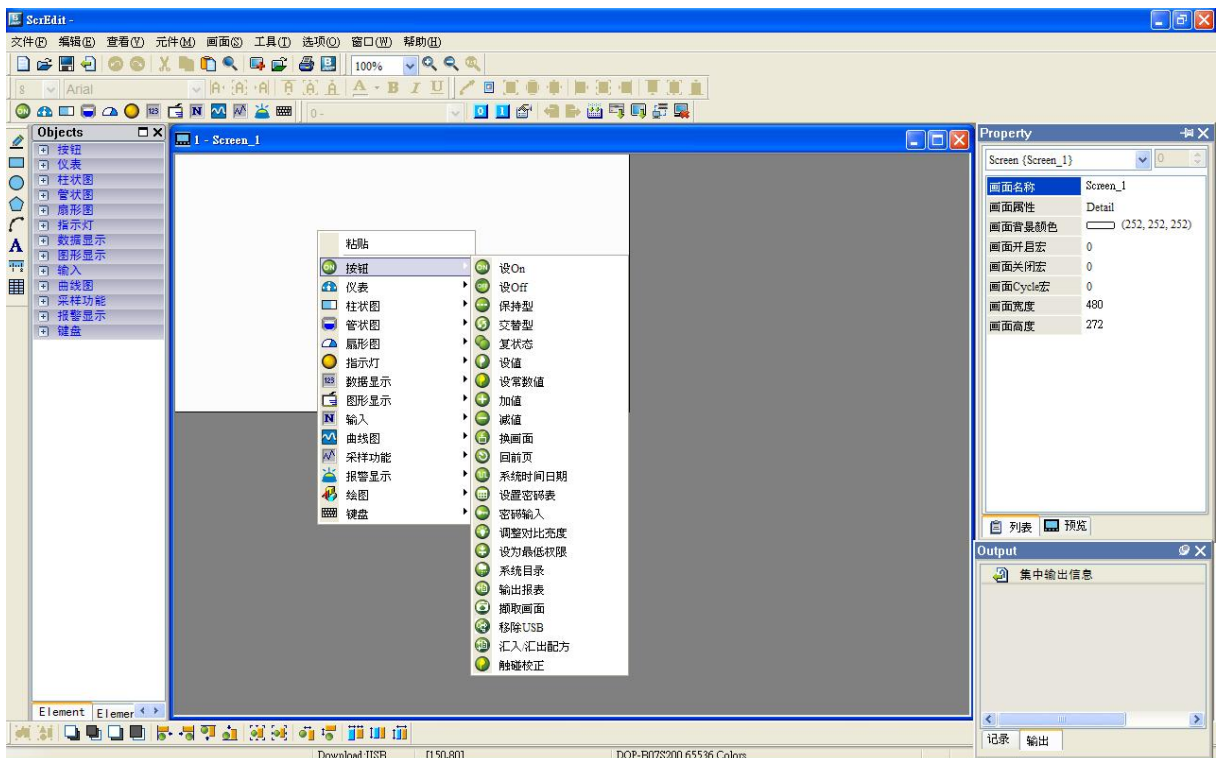


图 3-8-1 编辑视窗按鼠标右键就会出现选项

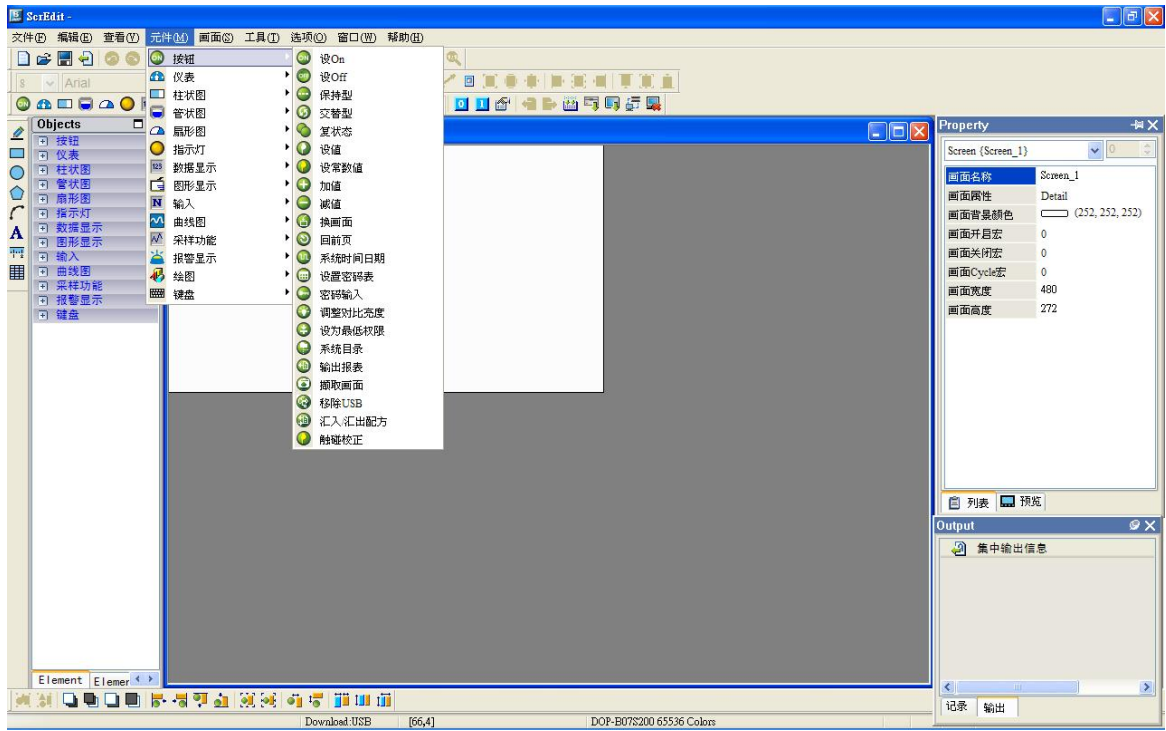


图 3-8-2 直接点元件菜单

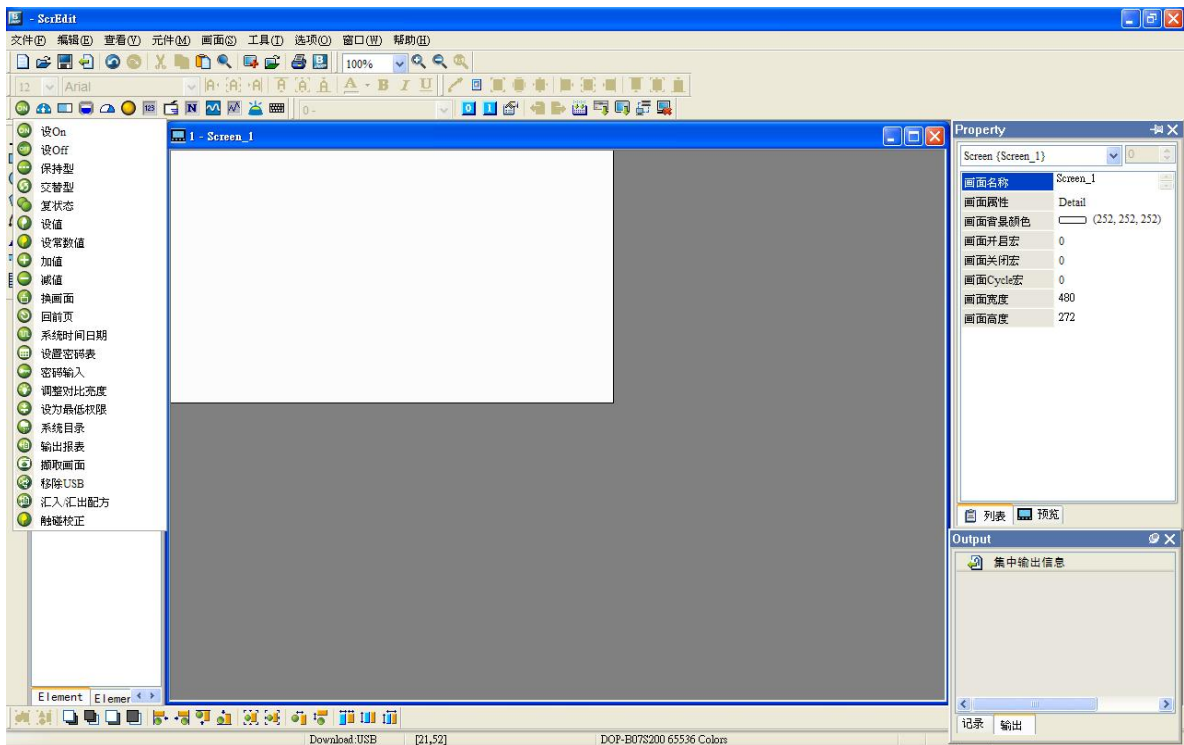


图 3-8-3 直接点选元件工具栏

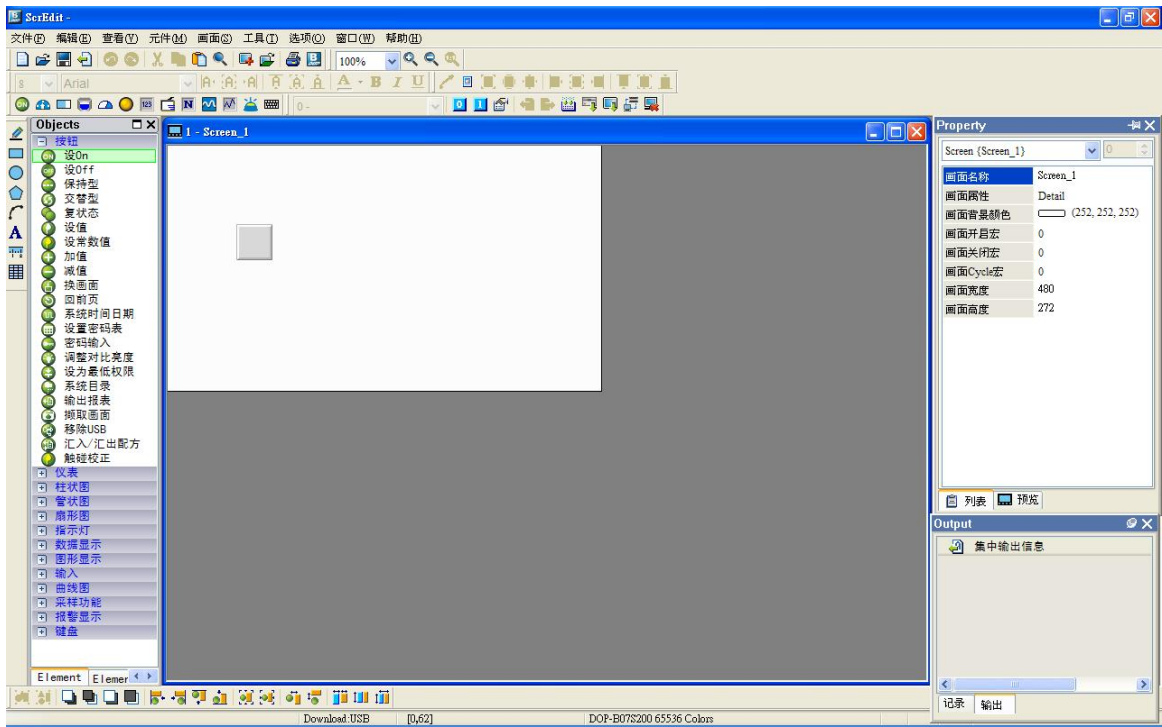


图 3-8-4 直接点选元件库

选择要使用的元件种类后，再使用鼠标按住左键，拖曳出元件范围即能建立一新元件。

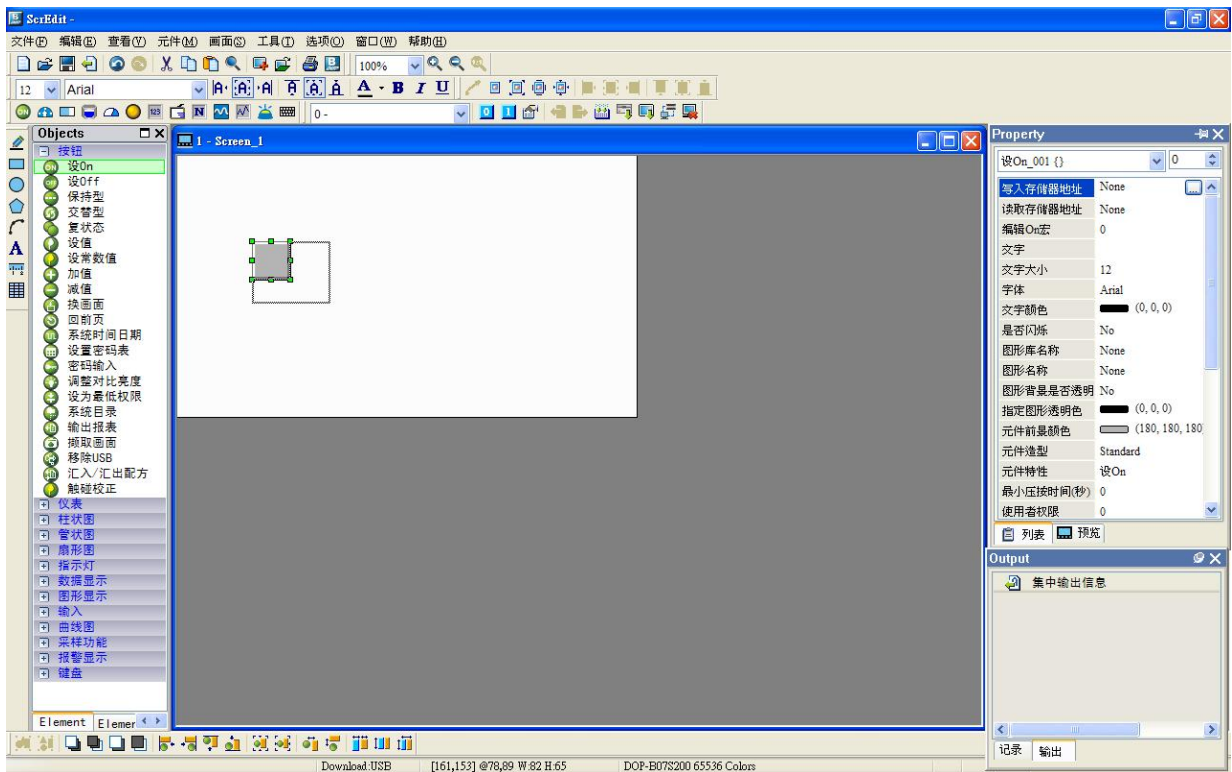


图 3-8-5 以鼠标拖曳的方式决定元件大小

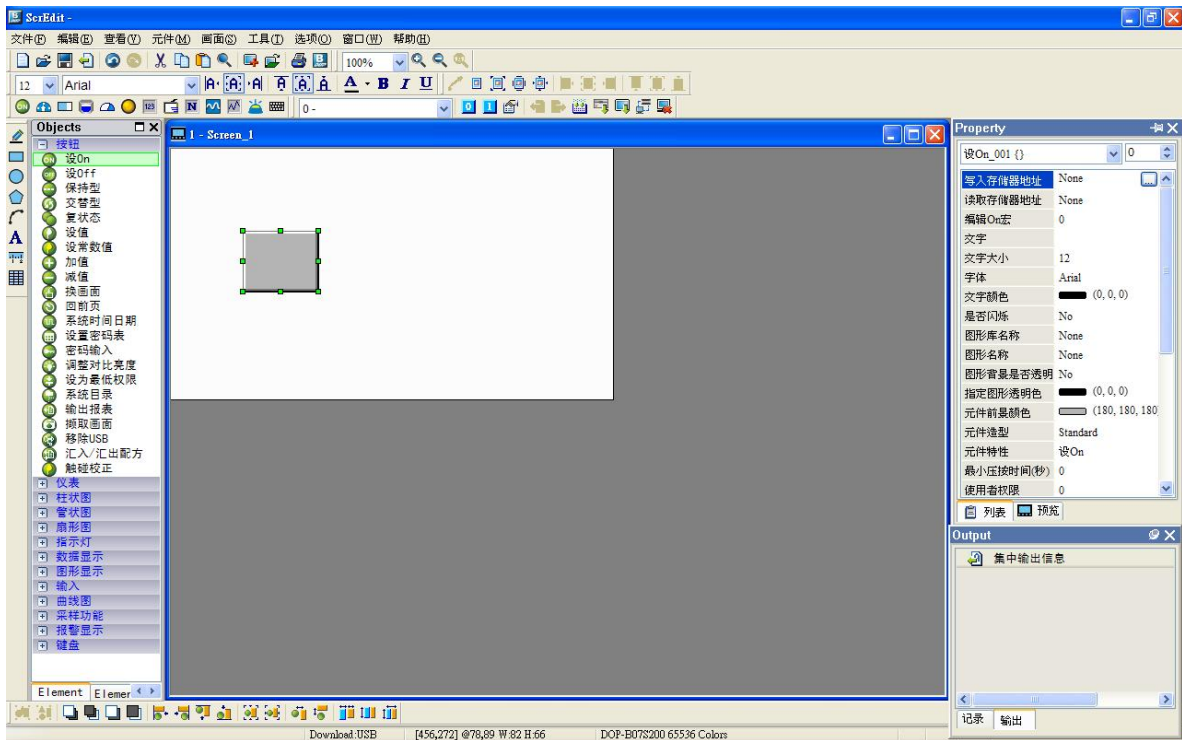


图 3-8-6 绘出元件

元件的建立：

利用鼠标在编辑区的时候，按下鼠标的右键将会出现元件的选项。此时便可以点选自己想要的元件，点选元件选项之后，按住鼠标的左键拖曳出范围，如此便能轻松的建立好自己想要的元件。然后再去元件属性表中设定所要的属性，那么一个实用的元件便如此轻松的建立好了。范例请参考（图 3-8-7、图 3-8-8、图 3-8-9、图 3-8-10）。

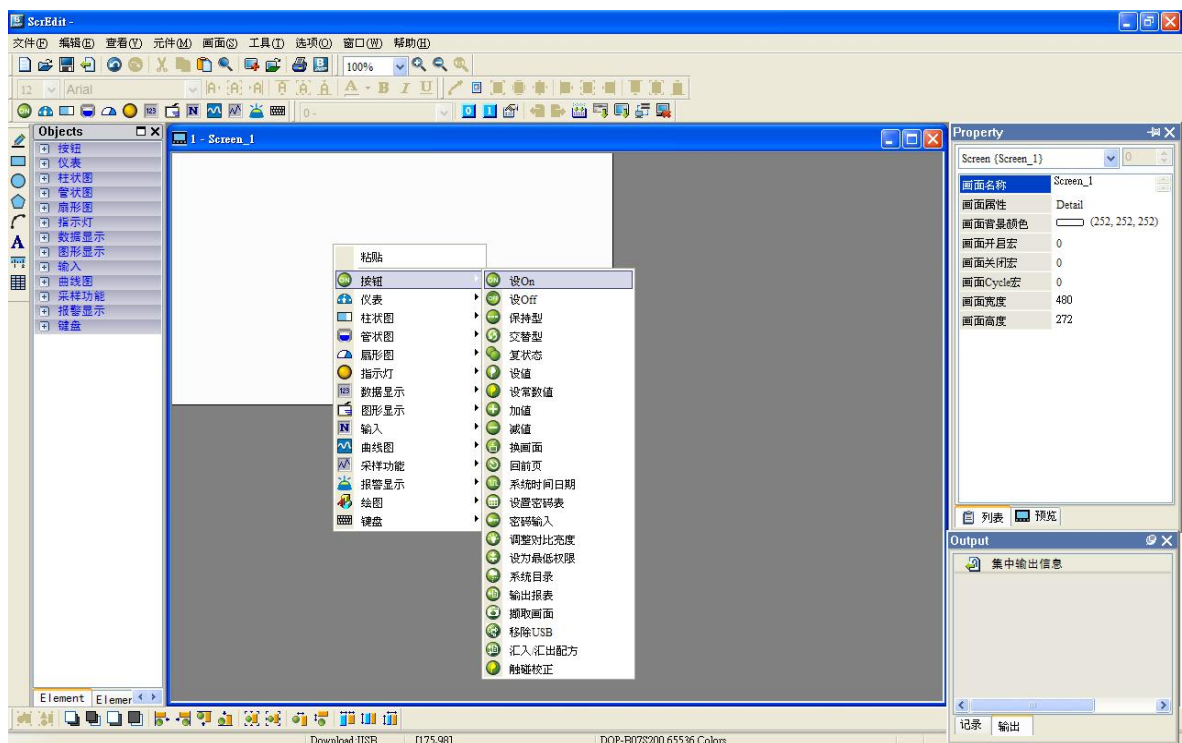


图 3-8-7 按右键后再点选所需的元件

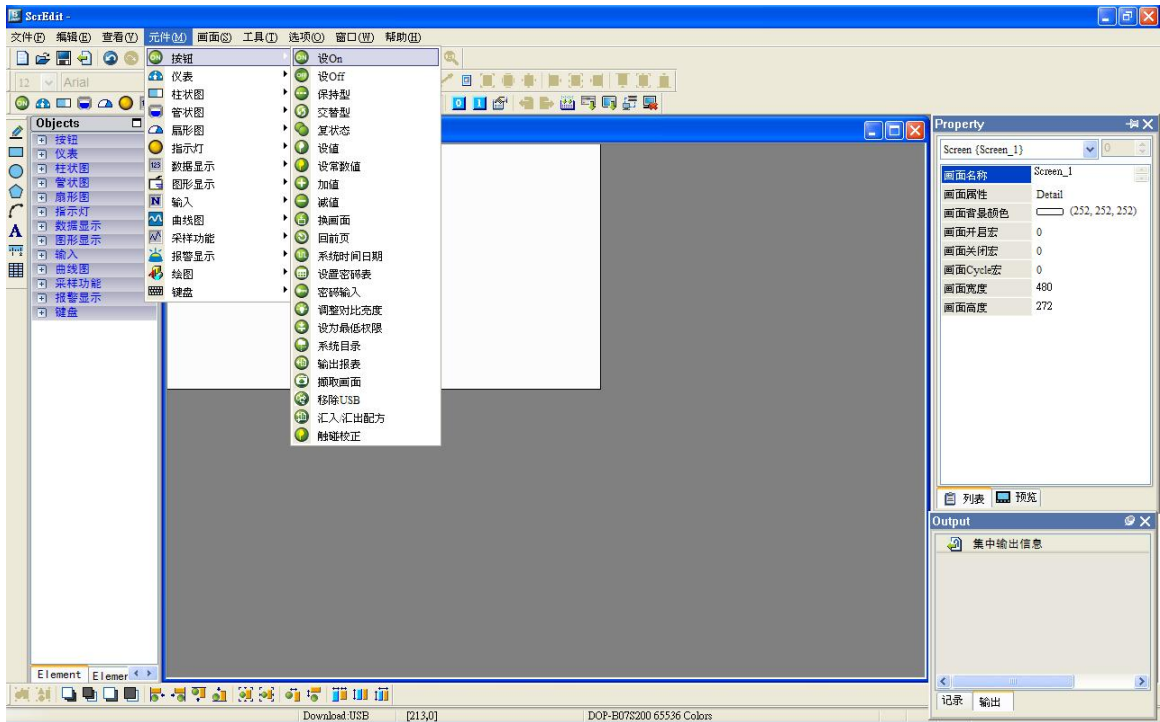


图 3-8-8 选取选项里面的元件选项后再选取所需元件

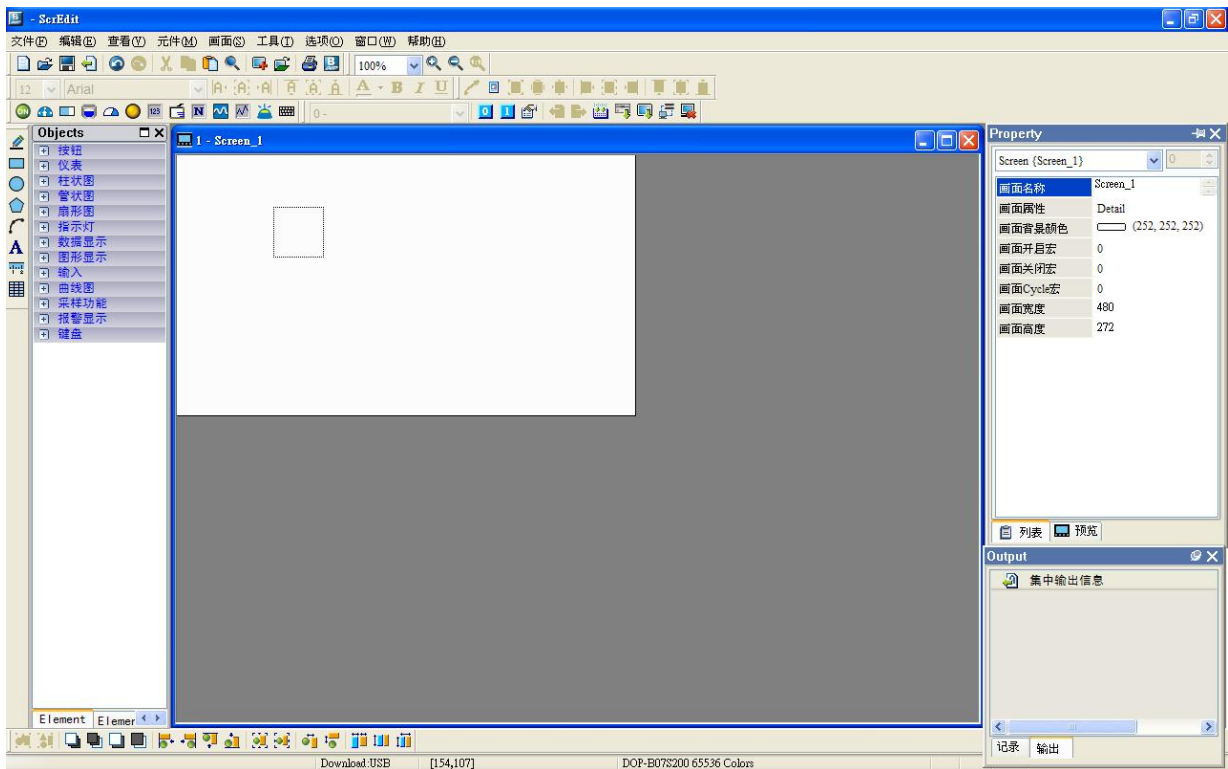


图 3-8-9 按住鼠标左键再拖曳出范围后放开鼠标左键

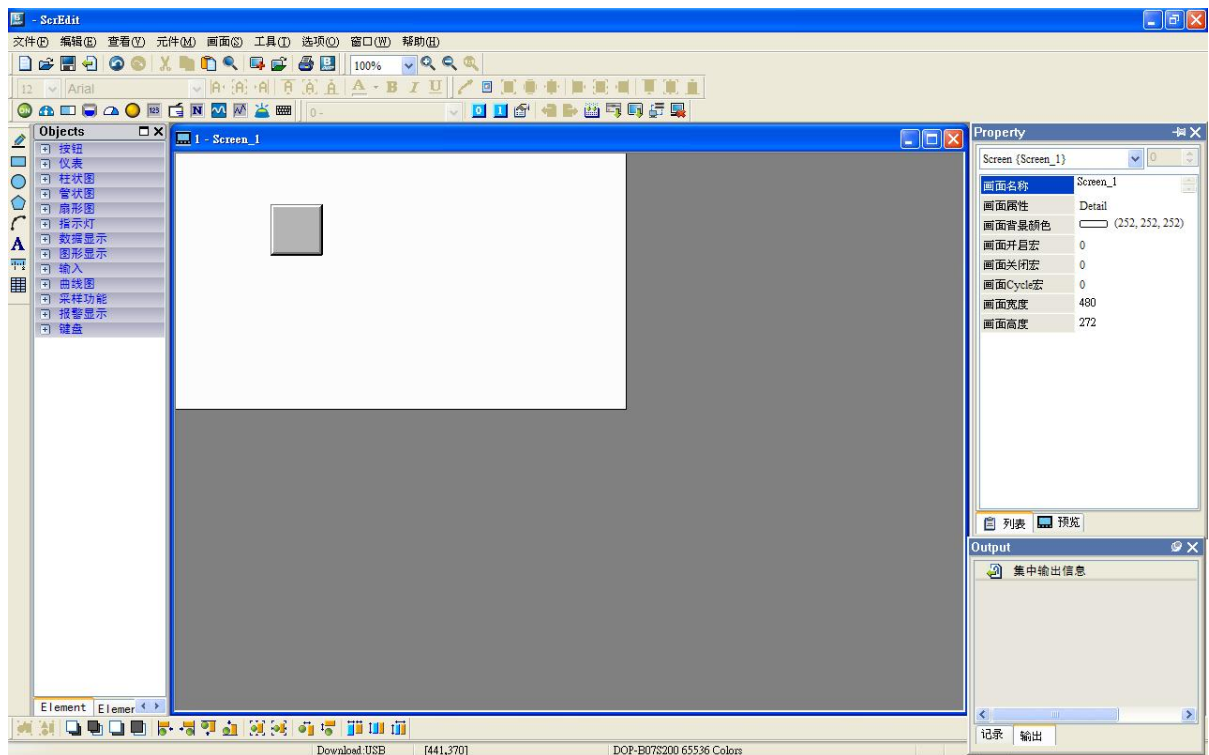



图 3-8-10 元件建立完成

元件的移动:

鼠标的操作，不管是按左键或是按右键，都跟 Windows®使用方法一样。当鼠标在元件上变成  的符号时，按住鼠标左键移动鼠标，就可以移动元件（图 3-8-11）。

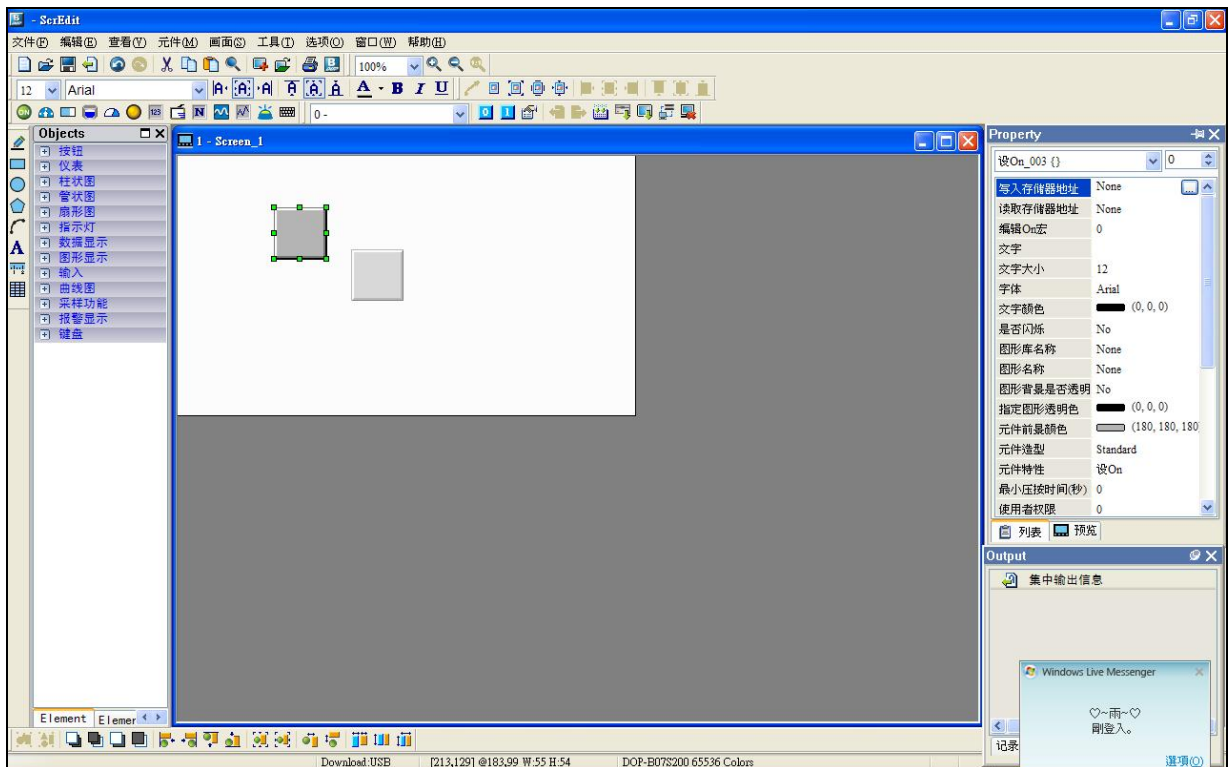


图 3-8-11 移动元件位置

元件的宽度改变：

当鼠标变成 \leftrightarrow 的符号时，按住鼠标左键移动鼠标，就可以改变元件的左右范围(图 3-8-12)。

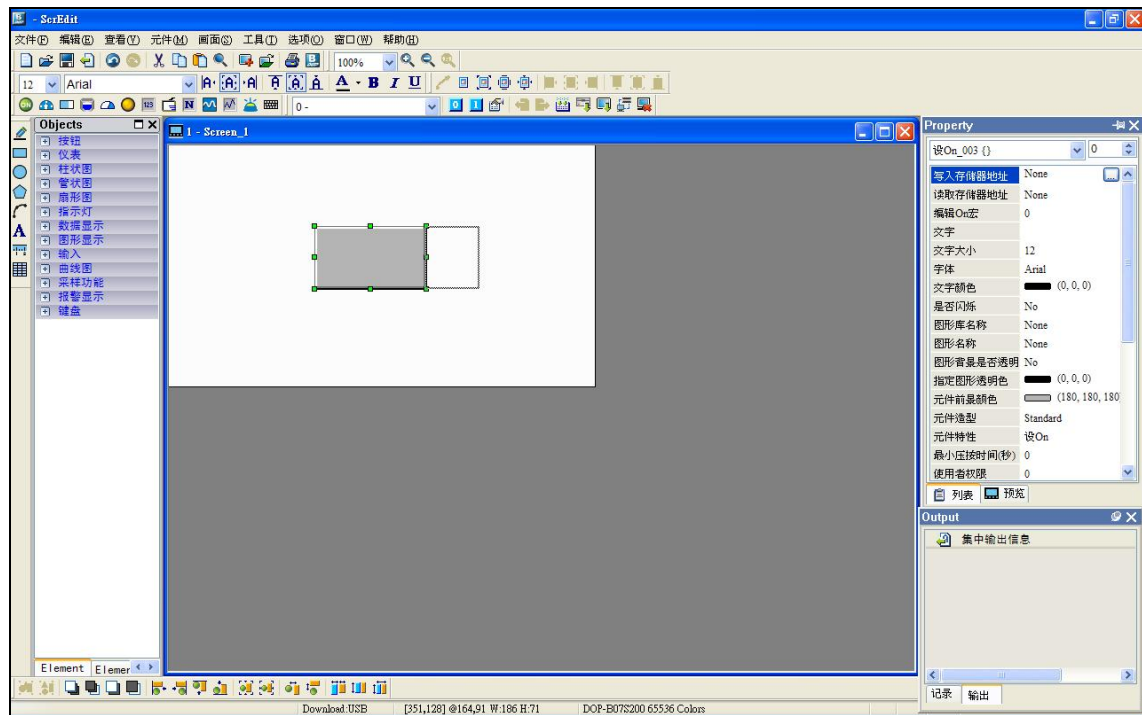


图 3-8-12 改变元件宽度

元件的高度改变：

当鼠标变成 \updownarrow 的符号时，按住鼠标左键移动鼠标，就可以改变元件的上下范围(图 3-8-13)。

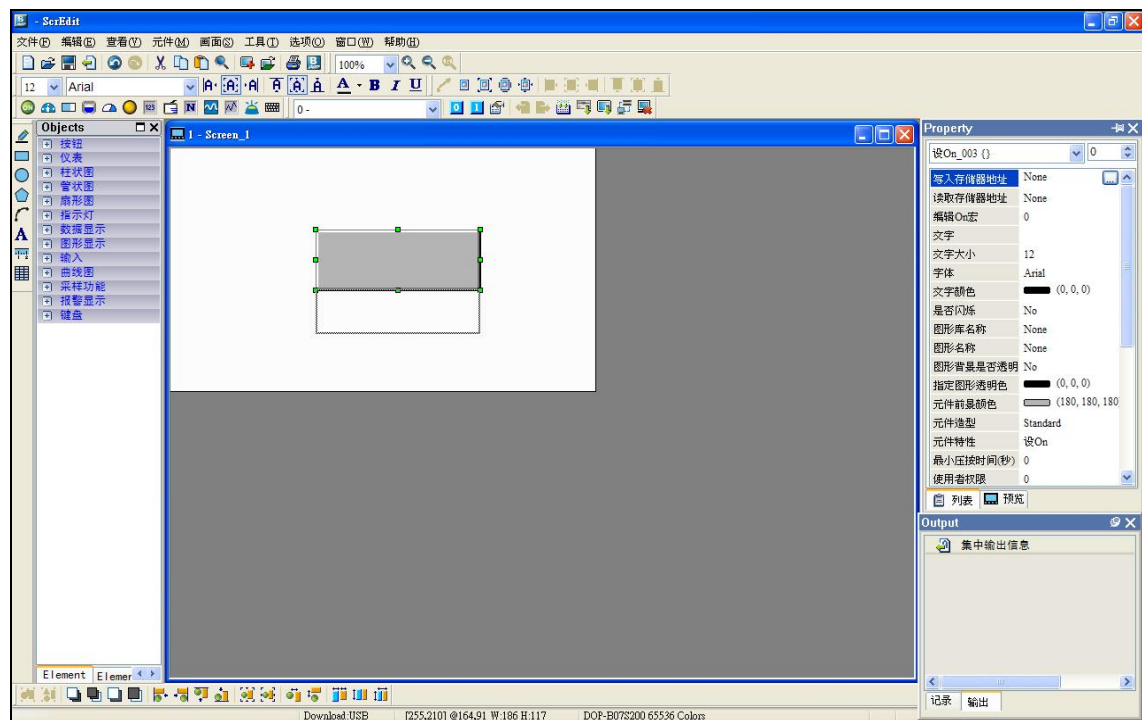


图 3-8-13 改变元件高度

元件的宽高同时改变：

当鼠标变成 \nwarrow 或是 \swarrow 的符号时，按住鼠标左键移动鼠标，就可以改变元件的上下左右范

围 (图 3-8-14)。

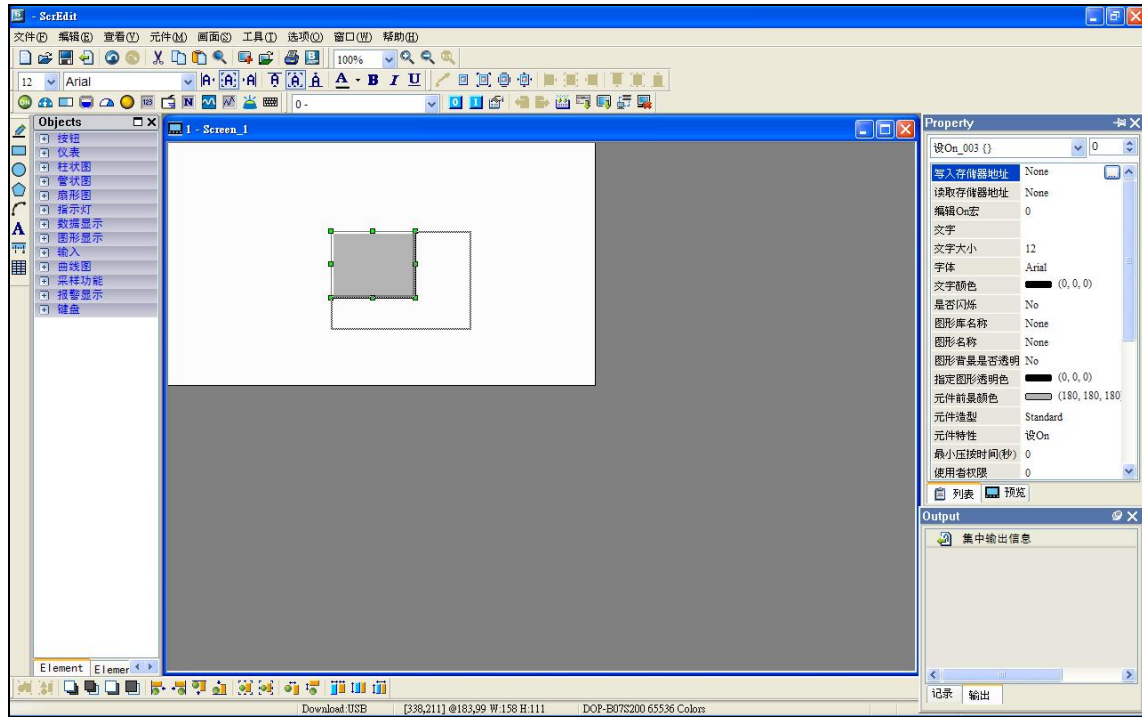




图 3-8-14 同时改变元件宽高

字串的输入：

在该元件属性表中选择【文字】栏后，当鼠标变成  的符号时，就可以在 （光标）闪烁的地方，输入任何 Windows®可以输入的字元（图 3-8-15）。

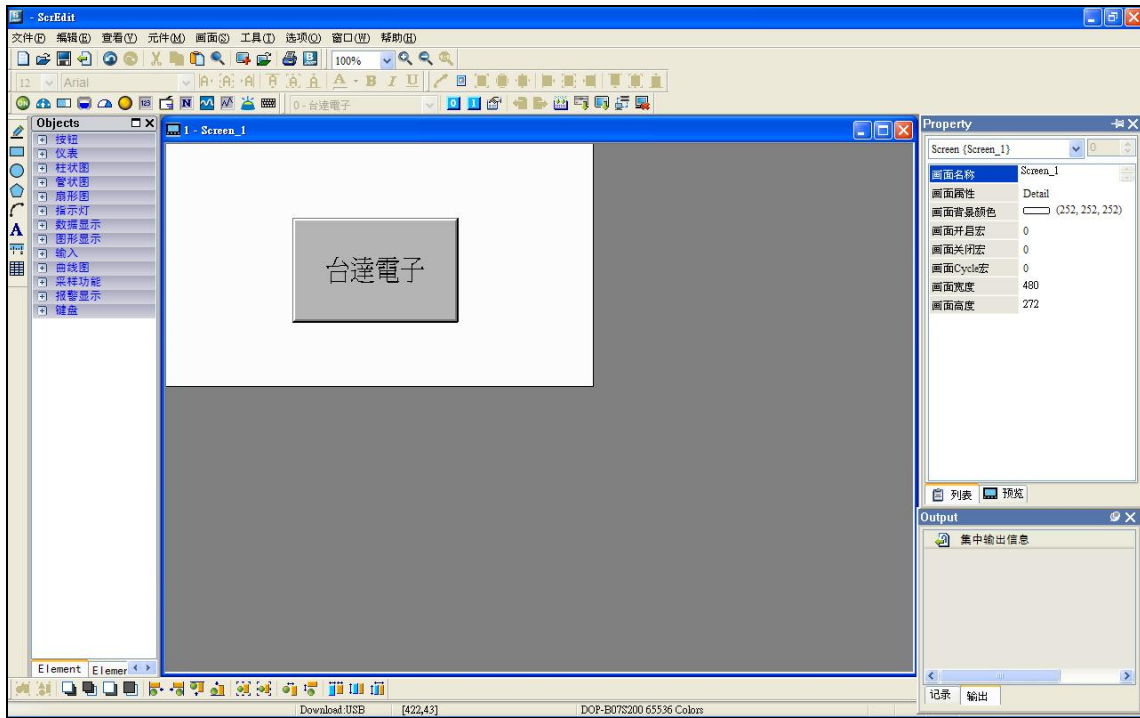


图 3-8-15 输入字符串

按鼠标右键：

如果您是按鼠标右键的话，在不同的地方会有不同的功能（图 3-8-16、图 3-8-17、图 3-8-18）。

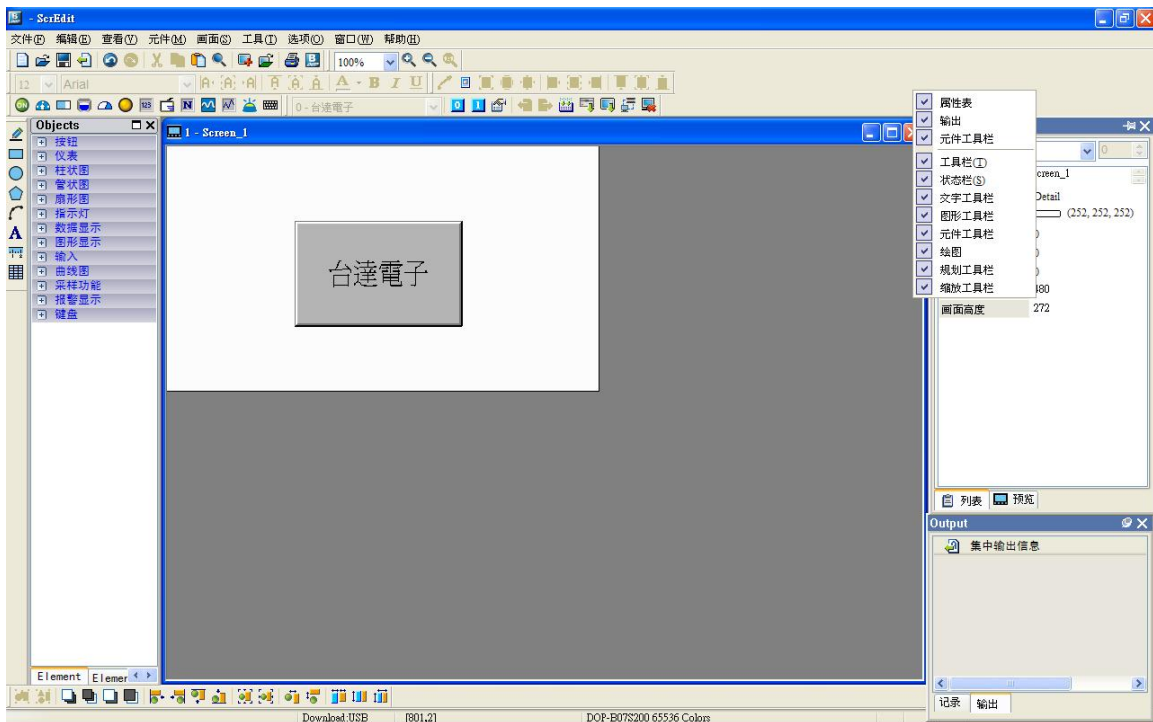


图 3-8-16 在工具栏按鼠标右键的画面－工具栏 Docking 目录

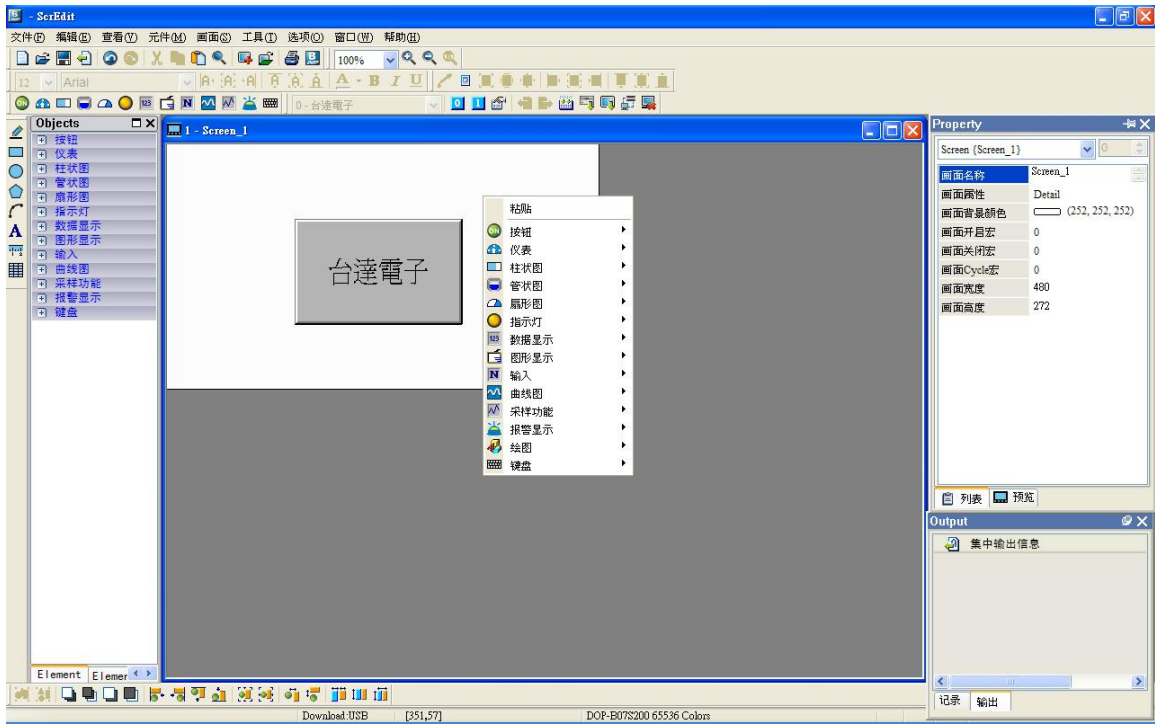


图 3-8-17 在画面编辑区按鼠标右键的画面—元件选择

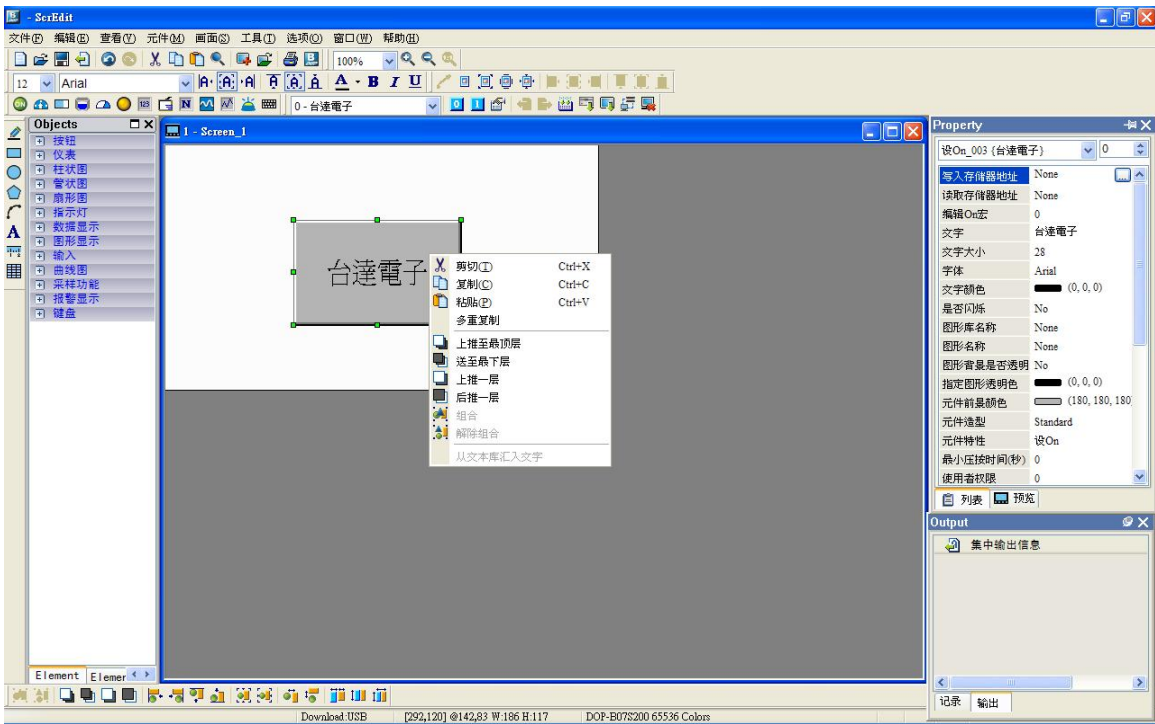


图 3-8-18 在元件上按鼠标右键的画面—规划工具

3.8.2 按钮



图 3-8-19 按钮

| 按钮类别 | 宏 | 读 | 写 | 功能 |
|-------|-----------|---|---|--|
| 设 On | ON | V | V | 按下此按钮后, 会将所设定的 Bit 地址永远被保持在 On 的状态。如果有编写 On 宏, 便会一并执行。 |
| 设 OFF | OFF | V | V | 按下此按钮后, 会将所设定的 Bit 地址永远被保持在 Off 的状态。如果有编写 Off 宏, 便会一并执行。 |
| 保持型 | ON OFF | V | V | 按下此按钮后, 会将所设定的 Bit 地址设为 ON, 放开则变为 OFF。如果有编写 On 或是 Off 宏, 便会一并执行。 |
| 交替型 | ON OFF | V | V | 按一次此按钮会将所设定的 Bit 地址设为 ON, 并执行 On 宏, 再按一次才会被设为 OFF, 同时执行 Off 宏。 |
| 复状态 | X | V | V | 可自行设定 1~256 个状态, 也可以设定其顺序是往前还是往后。 |
| 设值 | X | X | V | 点取该按钮后, 人机会将系统内建的输入键盘显示于屏幕上, 在输入数值而且按下 ENTER 后, 人机就会将数值送到所设定的地址。 |
| 设常数值 | X | X | V | 点此按钮, 人机会将指定的数值, 写入所设定的地址。 |
| 加值 | X | V | V | 点此按钮, 人机会将所设定的地址里的值加上所设定的常数值。 |

| 按钮类别 | 宏 | 读 | 写 | 功能 |
|---------|---|---|---|--|
| 减值 | X | V | V | 点此按钮，人机会将所设定的地址里的值减去所设定的常数值。 |
| 换画面 | X | X | X | 按一次该按钮，切换到所指定的画面。 |
| 回前页 | X | X | X | 回到前一个画面。例如画面有三页，编号分别 1、2、3。当我们依序由第一页换画面到第二页，再换画面到第三页，此时触碰第三页面【回前页】的按钮，人机便回到第二页面；相同的情形，触碰第二页面【回前页】的按钮，人机回到第三页面。 |
| 系统时间日期 | 0 | X | X | 设定人机系统时间日期（年一月一日 时：分：秒） |
| 设定密码表 | 0 | X | X | 设定所有权限以内的密码 |
| 密码输入 | 0 | X | X | 输入密码功能 |
| 调整对比亮度 | 0 | X | X | 调整人机显示的对比亮度 |
| 设为最低权限 | 0 | X | X | 按一次该按钮人机就将用户权限切换为最低等级(LEVEL 0) |
| 系统目录 | 0 | X | X | 按一次该按钮，人机就切换到系统设定画面 |
| 输出报表 | 0 | X | X | 输出数据到特定的装置，此功能按钮通常用于打印功能 |
| 撷取画面 | 0 | X | X | 撷取 HMI 所在页面输出至 U 盘。此按钮功能输出路径设定在 U 盘上，因此没有连接上述装置，则人机无法输出此撷取画面。 |
| 移除 USB | 0 | X | X | 使用此按键可执行安全移除 U 盘的动作。 |
| 汇入/汇出配方 | 0 | X | X | 汇入/汇出 Excel CSV 配方表到人机外部存储器。 |
| 触碰校正 | 0 | X | X | 触碰校正 |

3.8.2.1 设 On / 设 Off / 保持型 / 交替型

在人机上触碰按钮，人机会对按钮元件所设定的 Bit 地址送出信号给控制器相对应接点 ON 或 OFF。按钮共有四类可选择：一、设 ON 按钮；二、设 OFF 按钮；三、交替型按钮；四、保持型按钮。以下将对按钮元件中所使用的属性加以说明：

| 一般按钮属性说明 | |
|--------------------|---|
| 写入存储器地址 读取存储器地址 | 可选择内部存储器或控制器寄存器地址，若只有设定写入存储器地址而没有设定读取存储器地址，人机会自动读取写入存储器地址的数值。 地址设定视窗请参照下图： |

一般按钮属性说明



连线种类预设会有一个 Base Port (控制器) 以及 Internal Memory (内部存储器), 若您有新增多组连线设定, 下拉式菜单就会多加入您新增的连线名称。在选择连线与元件种类并输入正确的地址后, 按下确认按钮, 对应的数值数据会被记录在您选择的元件上。其中元件种类如下表叙述:

| | | | |
|-----|-----------|-------|------------------------|
| \$ | 内部存储器 | RCP | 配方寄存器 |
| \$M | 内部断电保持存储器 | RCPNO | 配方组别寄存器区 |
| *\$ | 间接寻址存储器 | RCPG | 配方组合别寄存器区 |
| | | Other | 其它控制器支持的元件名称。请参考通讯连线手册 |

编辑 On/Off 宏

当用户触碰按钮后更改状态为 ON 人机会执行 On 宏内的指令。当用户触碰按钮后更改状态为 OFF 人机会执行 Off 宏内的指令。但若按钮的状态不是使用触碰方式更改(使用宏更改或是外部信号)时, 并不会执行 On/Off 宏指令。宏编辑方法请参考 3.14 节。

一般按钮属性说明

文字
文字大小
字体
文字颜色

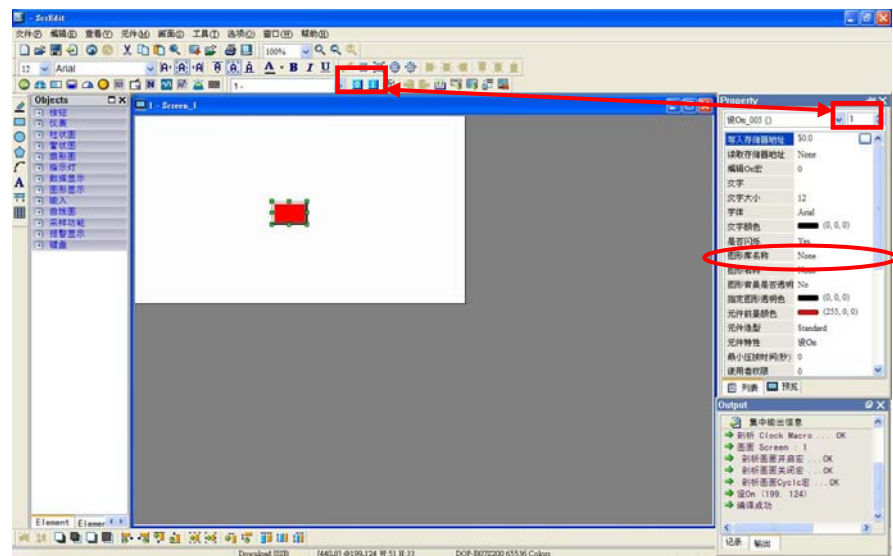
用户可依 Windows®所提供的文字大小、字体与颜色功能，设定该元件文字显示型态。其中点选字体属性设定时会出现如下的对话框



在字体属性设定对话框里您不但可以设定上述的功能，还可以指定字体的缩放比例，经由预览可以看到字体的变化。请注意在上图里由于我们设定了多国语言的编辑，您可以通过点选页签的方式分别作各国语言字体的设定。

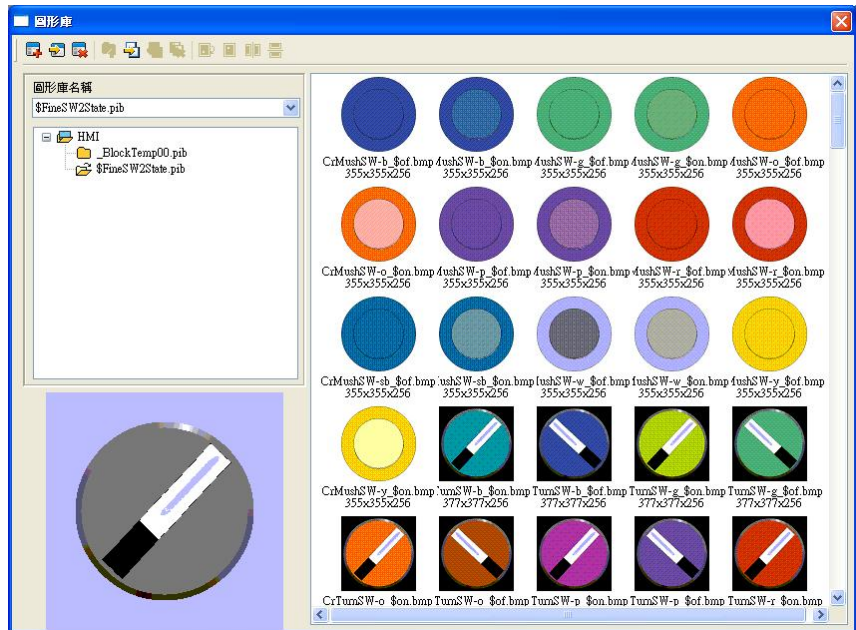
是否闪烁

以闪烁的显示方式提醒用户。
用户可以自行指定当按钮在 ON 或是 OFF 状态时，是否需要闪烁，闪烁的颜色为该状态之相对色。切换 ON 或是 OFF 状态的方式如下图所示。



一般按钮属性说明

点选**图形库名称**栏位(图行库名称若已选定, 可点选**图形名称**栏位), 选择已存在的图形库便会出现如下图的对话框:



图形库名称
图形名称



在这选择图形的对话框里, 您可以以双击的方式点选您的元件所要汇入的一张图形, 您也可以以多选的方式(Ctrl+鼠标左键), 并依照您所圈选的顺序, 圈选多张图形一并汇入到您指定的元件上。所选的多张图形会依序存放至元件的每个状态里。若所选的图形个数大于元件的状态数, 程序仅会以目前元件的状态个数汇入图形, 之后被圈选剩馀的图形会被忽略掉。

从文件汇入
图形


不必再由【菜单】里去启动【图形库】来汇入磁盘里的图形资料, 在此按下此按钮就可直接将磁盘里的图形文件汇入至您选择的图形库里。

图形背景是否透明
指定图形透明色

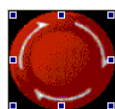
可将图形背景透明, 或是指定图形中某一颜色将其变成透明。

点选工具栏的  决定图形透明色, 也可以直接使用下拉式颜色菜单指定透明色, 但在某些相近色时, 指定透明色会无效果。例如 RGB(0,0,0)代表黑色, RGB(1,1,1)也是黑色, 但是图形内的黑色却是 RGB(0,1,0), 此时虽然您在下拉式颜色菜单选择黑色 RGB(0,0,0), 一样会无法达到透明色的效果。此时建议直接点选工具栏的  去点选图形中要变成透明的部份。

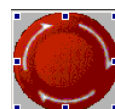
请确定您选的图形是不是对应所使用人机机型可支持的色彩数。比方来说, 当使用 256 色机种时, 如果您汇入的是一张 65536 色的图片, 某些颜色将无法顺利转换成透明色。

点选工具栏的  决定图形透明色, 如下图所示:

图形背景透明前



图形背景透明后



| 一般按钮属性说明 | | | | | | | | | |
|---|---|---|---|-------|-----------|---|---|---|---|
| 元素前景颜色 元素型式 | <p>用户可依照下图改变元件样式并设定元件的显示颜色。</p> <table border="1"> <tr> <td>Standard</td> <td>Raised</td> <td>Round</td> <td>Invisible</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> | Standard | Raised | Round | Invisible |  |  |  |  |
| Standard | Raised | Round | Invisible | | | | | | |
|  |  |  |  | | | | | | |
| 元素特性 | 可直接修改元素特性，而不必重新建立元件。能改变的特性有 <u>设 ON 按钮</u> 、 <u>设 OFF 按钮</u> 、 <u>交替型按钮</u> 与 <u>保持型按钮</u> 。 | | | | | | | | |
| 最小压按时间(秒) | 利用此功能可设定该元件按压动作时间。当按压超过按压动作时间该元件才会动作，如此可避免误动作。范围从 0~10 秒。 | | | | | | | | |
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。 | | | | | | | | |
| 输入后设为最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低，可预防 Operator 的误触。 | | | | | | | | |
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 | | | | | | | | |
| 执行前宏 | 当用户碰触按钮元件后，会先执行此宏内指令，才会执行按钮的动作。但若按钮的状态不是使用触碰方式更改(使用宏更改或是外部信号)时，并不会执行宏指令。 | | | | | | | | |
| 执行后宏 | 当用户碰触按钮元件后，会先执行按钮的动作后，才会执行此宏内指令。但若按钮的状态不是使用触碰方式更改(使用宏更改或是外部信号)时，并不会执行宏指令。 | | | | | | | | |
| 启用确认视窗 | <p>当选定启用确认视窗功能并执行此按钮时出现的提示信息。</p>  | | | | | | | | |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | |
| Y | 元件左上角 Y 座标 | | | | | | | | |
| 宽 | 元件宽度 | | | | | | | | |
| 高 | 元件高度 | | | | | | | | |

3.8.2.2 复状态

| 复状态按钮属性说明 |
|---|
| <p>复状态按钮可以支持三种数值单位，如果数值单位为 Word，可以设定 1-256 个状态；LSB 可以设定 16 个状态；Bit 只能设定 2 个状态。如果是设定 Word 或是 LSB，存储器地址就以 Word 为单位；反之如果数值单位为 Bit，存储器地址就会是以 Bit 为单位。如需新增或删除状态总数，只须增加跟减少属性表中状态总数即可。</p> |

| 复状态按钮属性说明 | |
|-------------------------|---|
| 写入存储器地址 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能, 设定该元件文字显示型态。(参阅一般按钮说明) |
| 是否闪烁 | 以闪烁的显示方式提醒用户。(参阅一般按钮说明) |
| 图形库名称 图形名称 | (参阅一般按钮说明) |
| 图形背景是否 透明 指定图形透明色 | (参阅一般按钮说明) |
| 元件前景颜色 元件型式 | (参阅一般按钮说明) |
| 用户权限 | 利用此功能可设定该元件按压动作的权限, 只有高于或等于设定权限才能使用。可预防元件被任意操作。 |
| 数值单位 | Bit 复状态元件可以有 2 个状态。 |
| | Word 复状态元件可以有 256 个状态。 |
| | LSB 复状态元件可以有 16 个状态。 |
| 数值格式 | 复状态提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式。 |
| 新增删除状态数 | 设定复状态按钮的状态总数。如果数值单位为 Word, 则可以设定 1-256 个状态; LSB 就可以设定 16 个状态; Bit 只能设定 2 个状态。 |
| 切换顺序 | 设定复状态按钮切换状态顺序 (<u>前一状态/次一状态</u>)。 |
| 输入后设为 最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低, 可预防元件被任意操作。 |
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时 (由生效准位决定), 此按键才能操作。 |
| 执行前宏 | 执行按钮动作之前, 会先执行此宏。 |
| 执行后宏 | 执行按钮动作之后, 再执行此宏。 |
| 启用确认视窗 | 当选定启用确认视窗功能并执行此按钮时会出现的提示信息。 |
| 不可见位元 | 当此位元被设为 On 时, 按钮元件将被隐藏。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

复状态实例说明

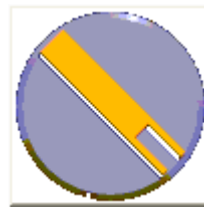
当数值单位选择 LSB 时 (D100.0~D100.3)



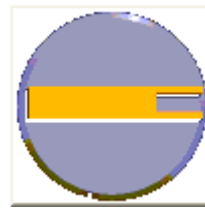
S0=D100.0 ON



S1=D100.1 ON



S2=D100.2 ON




S3=D100.3 ON

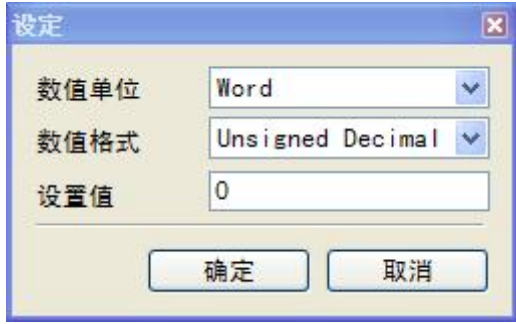
复状态显示实例

3.8.2.3 设值

| 设值按钮属性说明 | |
|---|---|
| 在人机上触碰此按钮，人机会显示系统内建的 TEN-KEY 键盘在屏幕上提供给用户输入数值。当按下 ENTER 时，人机就会送出数值给对应的寄存器，且输入的最大值与最小值可由用户自行决定，同时还可以设定输入前或是输入后触发地址来触发指定的控制器某一 Bit 地址。 | |
| 写入存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)。 |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) |
| 图形库名称 图形名称 | (参阅一般按钮说明) |
| 图形背景是否透明 指定图形透明色 | (参阅一般按钮说明) |
| 元件前景颜色 元件型式 | (参阅一般按钮说明) |
| 触发 触发方式 | 用户可选择数值写入前后触发指定的控制器 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 |
| 设定范围值 | <div data-bbox="550 1556 981 2016" data-label="Image"> </div> <p>数值单位： 提供 16bits Word 与 32bits Double Word。 最小值/最大值： 用户可设定输入值之最小与最大值，限制输入数值之范围。</p> |

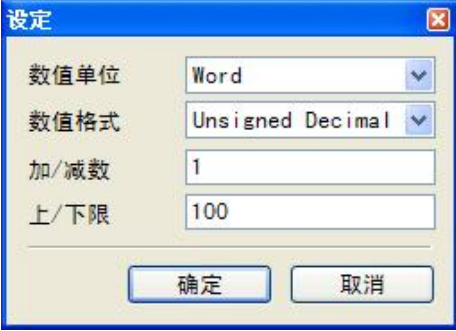
| 设置按钮属性说明 | | | | | | | | | | | | | | | | | |
|----------------------|--|-------------|-------------|--------|--------|---------------|---------------|-------------------|-------------------|---------------------|---------------------|--------|--------|-----------|-----------|--|-------------|
| 设定范围值 | <p>选择不同的数值单位会有不同的数值格式可供选择如下表</p> <table border="1"> <thead> <tr> <th>Word</th> <th>Double Word</th> </tr> </thead> <tbody> <tr> <td>1. BCD</td> <td>1. BCD</td> </tr> <tr> <td>2. Signed BCD</td> <td>2. Signed BCD</td> </tr> <tr> <td>3. Signed Decimal</td> <td>3. Signed Decimal</td> </tr> <tr> <td>4. Unsigned Decimal</td> <td>4. Unsigned Decimal</td> </tr> <tr> <td>5. Hex</td> <td>5. Hex</td> </tr> <tr> <td>6. Binary</td> <td>6. Binary</td> </tr> <tr> <td></td> <td>7. Floating</td> </tr> </tbody> </table> | Word | Double Word | 1. BCD | 1. BCD | 2. Signed BCD | 2. Signed BCD | 3. Signed Decimal | 3. Signed Decimal | 4. Unsigned Decimal | 4. Unsigned Decimal | 5. Hex | 5. Hex | 6. Binary | 6. Binary | | 7. Floating |
| | Word | Double Word | | | | | | | | | | | | | | | |
| | 1. BCD | 1. BCD | | | | | | | | | | | | | | | |
| 2. Signed BCD | 2. Signed BCD | | | | | | | | | | | | | | | | |
| 3. Signed Decimal | 3. Signed Decimal | | | | | | | | | | | | | | | | |
| 4. Unsigned Decimal | 4. Unsigned Decimal | | | | | | | | | | | | | | | | |
| 5. Hex | 5. Hex | | | | | | | | | | | | | | | | |
| 6. Binary | 6. Binary | | | | | | | | | | | | | | | | |
| | 7. Floating | | | | | | | | | | | | | | | | |
| <p>整数位数 小数位数</p> | <p>决定输入的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示格式，只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。</p> | | | | | | | | | | | | | | | | |
| | <p>您输入的最小值与最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式、整数位数与小数位数作数值范围的检查。</p> | | | | | | | | | | | | | | | | |
| 用户权限 | <p>利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。可预防元件被任意操作。</p> | | | | | | | | | | | | | | | | |
| 输入后设为最低权限 | <p>强制在执行按压动作之后将目前的用户权限设为最低，可预防元件被任意操作。</p> | | | | | | | | | | | | | | | | |
| 生效位元生效准位 | <p>当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。</p> | | | | | | | | | | | | | | | | |
| 执行前宏 | <p>执行按钮动作之前，会先执行此宏。</p> | | | | | | | | | | | | | | | | |
| 执行后宏 | <p>执行按钮动作之后，再执行此宏。</p> | | | | | | | | | | | | | | | | |
| 启用确认视窗 | <p>当选定启用确认视窗功能并执行此按钮时会出现的提示信息。</p> | | | | | | | | | | | | | | | | |
| 超过范围信息提示 | <p>启用此功能时，当输入超过所设定的范围值时 HMI 会出现提示信息视窗。</p>  | | | | | | | | | | | | | | | | |
| 不可见位元 | <p>当此位元被设为 On 时，按钮元件将被隐藏。</p> | | | | | | | | | | | | | | | | |
| X | <p>元件左上角 X 座标</p> | | | | | | | | | | | | | | | | |
| Y | <p>元件左上角 Y 座标</p> | | | | | | | | | | | | | | | | |
| 宽 | <p>元件宽度</p> | | | | | | | | | | | | | | | | |
| 高 | <p>元件高度</p> | | | | | | | | | | | | | | | | |

3.8.2.4 设常数值

| 设常数值按钮属性说明 | | |
|--|---|---|
| 在人机上触碰此按钮，人机会立即送出指定的常数值到相对应的寄存器。与设值按钮相同可以在设定常数值后触发一个 Bit 地址。 | | |
| 写入存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） | |
| 图形库名称 图形名称 | （参阅一般按钮说明） | |
| 图形背景是否透明 指定图形透明色 | （参阅一般按钮说明） | |
| 元件前景颜色 元件型式 | （参阅一般按钮说明） | |
| 触发 触发方式 | 用户可选择数值写入前后触发指定的控制器 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 | |
| 设定范围值 |  | |
| | 数值单位 | 提供 16bits Word、与 32bits Double Word。 |
| | 数值格式 | 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex |
| 设定值 | 决定要写入的常数值。您输入的常数值在按下确定按钮后程序会参照您选择的数值单位与数值格式作数值范围的检查。 | |
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。可预防元件被任意操作。 | |
| 输入后设为 最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低，可预防元件被任意操作。 | |
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 | |

| 设常数值按钮属性说明 | |
|------------|-----------------------------|
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 |
| 启用确认视窗 | 当选定启用确认视窗功能并执行此按钮时会出现的提示信息。 |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

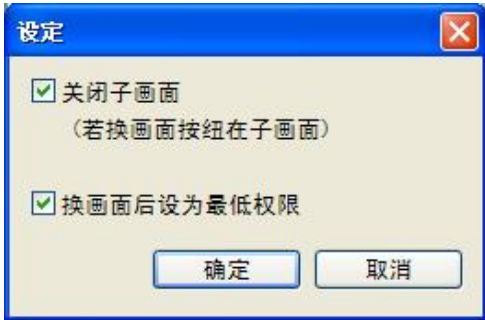

3.8.2.5 加/减值

| 加/减值按钮属性说明 | |
|---|--|
| 在人机上触碰此按钮，人机会先读取寄存器的数值并加上/减去所设定的常数值，尔后再将运算结果写至相对应的寄存器。如果增加/减少后的值超过所设定的上/下限值，加/减值按钮将维持上/下限值于对应的寄存器里。 | |
| 写入存储器地址 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） |
| 图形库名称 图形名称 | （参阅一般按钮说明） |
| 图形背景是否透明 指定图形透明色 | （参阅一般按钮说明） |
| 元件前景颜色 元件型式 | （参阅一般按钮说明） |
| 元件特性 | 直接修改元件特性，而不必重新建立元件。能改变的特性有 加值按钮 与 减值按钮 两种。 |
| 触发 触发方式 | 用户可选择数值写入前后触发指定的控制器 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 |
| 设定范围值 |  |
| 数值单位 | 提供 16bits Word、与 32bits Double Word。 |

| 加/减按钮属性说明 | | |
|-----------|--|---|
| 设定范围值 | 数值格式 | 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex |
| | 加/减数 | 每次压按时所需递增/递减的数值。 |
| | 上/下限 | 设定递增/递减的范围极限值。您输入的加/减数值与上/下限值会在按下确定按钮后程序会参照您选择的数值单位与数值格式作数值范围的检查。 |
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。可预防元件被任意操作。 | |
| 输入后设为最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低，可预防元件被任意操作。 | |
| 生效位元生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 | |
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 | |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 | |
| 启用确认视窗 | 当选定启用确认视窗功能并执行此按钮时会出现的提示信息。 | |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |


3.8.2.6 换画面/回前页

| 换画面/回前页按钮属性说明 | |
|---|--|
| 换画面按钮的类型共有二种可选择： 1. 换画面：在人机上触碰此按钮，人机会切换至所设定的画面。 2. 回前页：在人机上触碰此按钮，人机就会切换回屏幕先前所显示的画面。 | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） |
| 图形库名称 图形名称 | （参阅一般按钮说明） |
| 图形背景是否透明 指定图形透明色 | （参阅一般按钮说明） |

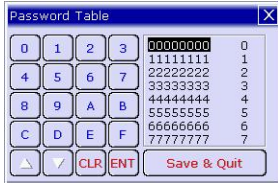



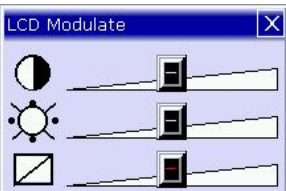
| 换画面/回前页按钮属性说明 | |
|----------------|--|
| 元件前景颜色 元件型式 | (参阅一般按钮说明) |
| 元件特性 | 直接修改元件特性，而不必重新建立元件。能改变的特性有 换画面按钮 、 回前页按钮 与 上一页按钮 三种。 |
| 触发 触发方式 | 用户可选择数值写入前后触发指定的控制器 Bit 地址为 ON 注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 |
| 设定 | 只有在元件特性为 换画面按钮 时此栏位属性才会出现。  |
| | 关闭子画面 当换画面按钮是被建立于子画面里时此选项才能被启动。其动作为当按下换画面时会连带将目前 (Active) 的子画面关闭。 |
| | 换画面后设为最低权限 强制在执行按压动作之后将目前的用户权限设为最低，可预防元件被任意操作。 |
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。可预防元件被任意操作。 |
| 改变画面 | <p>点此属性您会看到如下图所示：</p>  <p>在此画面存取对话框里，您可以在左侧选择清单方块里点选您要切换的画面，右侧会显示其预览图，按下确定按钮后即可。</p> |

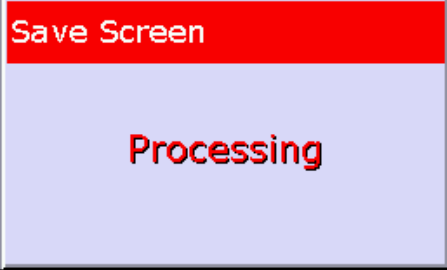
| 换画面/回前页按钮属性说明 | |
|---------------|---|
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 |
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 |
| 启用确认视窗 | 当选定启用确认视窗功能并执行此按钮时会出现的提示信息。 |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.2.7 系统时间日期/设定密码表/密码输入/调整对比亮度/设为最低权限/系统目录/输出报表/截取画面/移除 USB/汇入汇出配方/触碰校正


| 系统功能按钮属性说明 | |
|---------------------|---|
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） |
| 图形库名称 图形名称 | （参阅一般按钮说明） |
| 图形背景是否透明 指定图形透明色 | （参阅一般按钮说明） |
| 元件前景颜色 元件型式 | （参阅一般按钮说明） |
| 元件特性 | 系统时间日期 可以直接在人机上设定系统的日期与时间。与按下 SYS 键进入人机系统设定画面中 Time 的功能一样。  |
| | 设定密码表 【密码表】的功能可以在软件或硬件上去做设定。 1-硬件：此按钮设定后，根据设定属性表里面的用户权限来决定开启的等级。下载到人机之后，如果用户的权限小于所设定的权限，将无法开启此密码表而会显示输入密码的视窗。能否开启则取决于输入的密码等级是否大于等于所设定的权 |

| 系统功能按钮属性说明 | |
|------------|--|
| | 限，就算开启了，也只是表示可以修改自己等级及以下的密码，如果是高于自己等级的密码，不但无法修改也无法观看如下图： |

| 系统功能按钮属性说明 | |
|------------|---|
| 元件特性 | <p>设定密码表</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>最高权限</p> </div> <div style="text-align: center;">  <p>权限 4 时</p> </div> </div> <p>2-软件：在软件下拉式功能菜单选取【选项】\【密码表设定】此功能就会出现下方如图所示各权限密码的设定，当设定完成后，随着下载程序到人机界面上就可以使用。</p> <div style="text-align: center;">  </div> |
| | <p>密码输入</p> <p>此按钮提供人机端输入密码的界面。输入密码的同时也会根据输入的密码等级，而开启相对应的权限。输入密码等级越高，权限便越高，如右图所示：</p> <div style="text-align: right;">  </div> |
| | <p>调整对比亮度</p> <p>此按钮提供人机在对比亮度上的调整。此按钮点选后会出现“LCD Modulate”框，来让用户调整人机的对比以及亮度。其中 Set for default Contrast 会将人机的当时的对比值设为中间值 (TFT LCD 机种无此设定)。</p> <div style="text-align: right;">  </div> |
| | <p>设为最低权限</p> <p>此按钮元件将会将用户权限变为最低 (LEVEL</p> |

| | | |
|------------|--------|---|
| | | 0), 可以让用户将当前的人机用户权限设为最低。因为所有的元件都可以设定用户权限, 如此就能保护控制系统参数不被他人窜改或是随便操纵, 造成系统运转错误 (换画面按钮元件中也提供此项功能)。 |
| 系统功能按钮属性说明 | | |
| 元件特性 | 系统目录 | 按一次该按钮人机就切换到系统目录画面, 只要按下 Run 功能或重新开机即能回到运行画面。 |
| | 输出报表 | 此按钮元件有多方面的功能, 看您的 报表装置 属性如何设定。 |
| | 撷取画面 | <p>此按钮提供人机所在页面的撷取功能, 按下此按钮, 人机会输出 BMP 图形文件到预设路径 U 盘上, 因此若没有连接上述装置, 或格式错误, 则无法顺利输出此撷取画面。</p> <p>当顺利输出此撷取画面会在此装置上建立一个当天的文件夹为 20090101(意指 2009 年 1 月 1 日), 建立的同时人机画面会出现下方视窗</p> <div style="text-align: center;">  </div> <p>表示正在写入资料到 U 盘上, 而此文件夹里就会储存你所输出的 BMP 图形文件, 并按照数字从 00001 开始往后依序排列, 请注意此排列顺序会根据用户建立的专案画面顺序为储存的依据, 并不是依操作者所触发此按钮所在页面的顺序所建立。</p> <p>举例来说, 专案画面有三个页面, 此三个页面个别有撷取画面元件, 当操作者依序触发此元件由第三页开始到第二页到第一页, 此时由 U 盘上看到输出的图形文件会依序由 00001(专案画面一)~00003(专案画面三)。</p> <p>※使用 U 盘格式必为 FAT32, 详情请参考 1.4 注意事项。</p> |
| | 移除 USB | <p>将 HMI 缓冲区的资料顺利取出, 以防止连接在 HMI 上的 U 盘资料丢失。</p> <p>※使用 U 盘格式必为 FAT32, 详情请参考 1.4 注意事项。</p> |

| | | |
|--|---------|---|
| | 汇入/汇出配方 | 汇入/汇出配方包含 16 位元与 32 位元两个资料。使用此功能可将储存在人机内部存储器配方区资料汇出到外部存储器，或由外部存储设备汇入配方表进入人机存储器。 |
|--|---------|---|

| 系统功能按钮属性说明 | | |
|--------------|--|--|
| 元件特性 | 汇入/汇出配方 | 详细配方 CSV 档案格式内容请参阅 P2-86 【配方 CSV 编辑】说明。 ※使用 U 盘格式必为 FAT32，详情请参考 1.4 注意事项。 |
| 触发 触发方式 | | 用户可选择数值写入前后触发指定的控制器 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 |
| 报表装置 | 只有在您系统按键元件特性选择 输出报表 时此栏位属性方能设定。 | |
| |  | |
| | 点选此属性将出现如上图的对话框，您可以指定输出装置到 SD 卡或 U 盘或 Printer。其中 U 盘与 Printer 只有在您选择部份的 AE/AS 系列機種下才可设定。 | |
| | SD 卡 | 将历史数据、警报数据输出至 SD 卡上。 |
| U 盘 | 将历史数据、警报数据输出至 U 盘上。 | |
| | Printer | 会先检查人机是否已有规划排版打印，若有规划人机会将已规划的排版画面输出到打印机上；若无规划人机会将目前画面作 Hard Copy 打印。 |
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。可预防元件被任意操作。 | |
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 | |
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 | |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 | |
| 启用确认视窗 | 当选定启用确认视窗功能并执行此按钮时会出现的提示信息。 | |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 | |
| X | 元件左上角 X 座标 | |

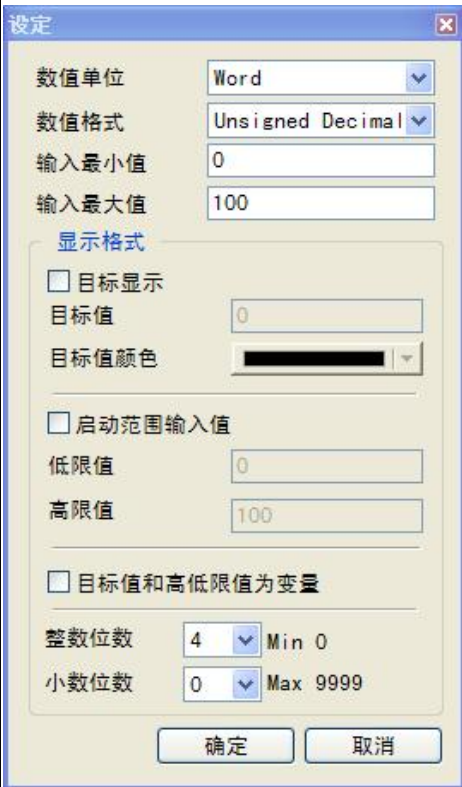
| 系统功能按钮属性说明 | |
|------------|------------|
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

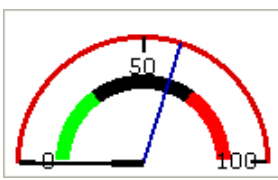
3.8.3 仪表

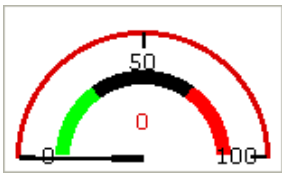


图 3-8-20 仪表

| 仪表属性说明 | | | | | | | |
|---|--|----------|--------|--------|--|--|--|
| 设定最大/最小值、高低限值跟颜色、指针颜色以及刻度颜色跟数目等等。是用来显示设定地址的计量大小是否超出上限还是低于下限，并且以不同颜色来区分以利用户分辨。 | | | | | | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)。 | | | | | | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) | | | | | | |
| 外框颜色 | 设定仪表元件边框颜色。 | | | | | | |
| 元件背景颜色 | 设定仪表背景颜色。 | | | | | | |
| 元件型式 | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Standard</td> <td>Raised</td> <td>Sunken</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> | Standard | Raised | Sunken | | | |
| Standard | Raised | Sunken | | | | | |
| | | | | | | | |
| 设定值 | 目标值、输入最小值与输入最大值在按下确定按钮后，程序会参照您选择的数值单位、数值格式、整数位数与小数位数作数值范围检查。 | | | | | | |

| 仪表属性说明 | |
|---|---|
|  | <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择 Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal <p>输入最小值/输入最大值 显示区间用的最小值与最大值。</p> |

| 仪表属性说明 | | |
|--------|-------------|--|
| 设定值 | 目标值设定 | <p>您可以决定是否要显示目标值，设定目标值及其颜色后仪表会从中心点的位置拉出一条目标线指到您设定的目标值上如下图所示：（这里我们目标值设为 60，目标值颜色为蓝色）</p>  |
| | 启动范围输入值 | 请参考下面低限值颜色与高限值颜色的说明。 |
| | 目标值和高低限值为变量 | 当设定目标值与高低限值为变量时，低限值地址为 <u>读取存储器地址+1</u> ；高限值地址为 <u>读取存储器地址+2</u> ；目标值地址为 <u>读取存储器地址+3</u> 。 |
| | 整数位数 | 决定输入的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示样式。 |
| | 小数位数 | |

| 仪表属性说明 | |
|----------------|--|
| 低限值颜色 高限值颜色 | 在 <u>设定值</u> 属性里有勾选 <u>启动范围输入值</u> 才会显示。下图为将低限值设为 30、低限值颜色设为绿色，高限值设为 70、高限值颜色设为红色执行结果。  |
| 指针颜色 | 设定仪表指针颜色。 |
| 刻度颜色 | 设定仪表刻度颜色。 |
| 刻度区间数目 | 设定刻度区间数目，利用点选上下的按钮来增加或是减少刻度区间数目。范围从 1 个区间~10 个区间。 |
| 尺规颜色 | 尺规颜色 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |











3.8.4 柱状图

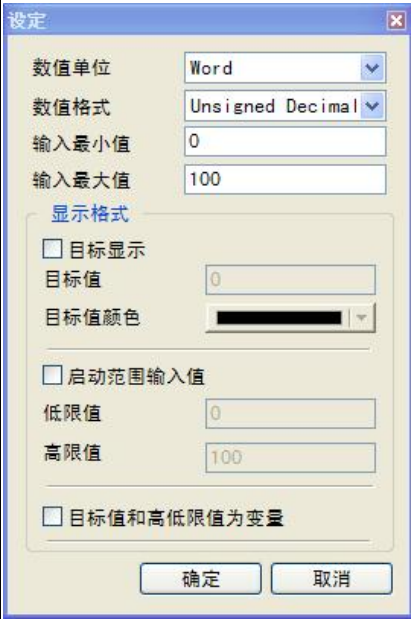






图 3-8-21 柱状图














3.8.4.1 一般型

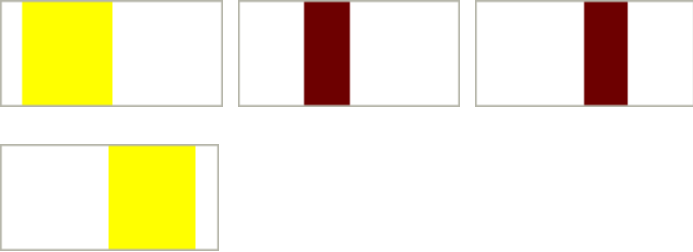
| 一般型柱状图属性说明 | |
|---|--|
| 人机读取控制器对应的寄存器的数值。将数值以图形进度表的方式，显示在人机屏幕上。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） |

| 一般型柱状图属性说明 | | | | | | | |
|---|--|---|--------|--------|---|--|---|
| 外框颜色 | 设定一般型柱状图元件边框颜色。 | | | | | | |
| 元件前景颜色 元件背景颜色 | 设定一般型柱状图前景与背景颜色。下图为前景颜色设为绿色，背景颜色设为黄色的显示结果：  | | | | | | |
| 元件型式 | <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Standard</th> <th>Raised</th> <th>Sunken</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> | Standard | Raised | Sunken |  |  |  |
| Standard | Raised | Sunken | | | | | |
|  |  |  | | | | | |
| 显示格式 | Left | 显示的进度方向是由右至左递增(递减)。 | | | | | |
| | Right | 显示的进度方向是由左至右递增(递减)。 | | | | | |
| | Top | 显示的进度方向是由下至上递增(递减)。 | | | | | |
| | Bottom | 显示的进度方向是由上至下递增(递减)。 | | | | | |
| 设定值 | 目标值、低限值、高限值、输入最小值与输入最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。 | | | | | | |

| 一般型柱状图属性说明 | |
|---|--|
| 设定值 | <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 1; padding-left: 10px;"> <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex <p>输入最小值/输入最大值 显示区间用的最小值与最大值。</p> </div> </div> |
| | <p>目标值设定</p> <p>您可以决定是否要显示目标值，设定目标值及其颜色后一般型柱状图会从整个元件区块里参考输入最大值及输入最小值找出适宜的位置绘出参考线如下图所示：（这里我们目标值设为 50，目标值颜色为红色，输入最大与最小值分别为 100 与 0）。</p> <div style="text-align: center;">  </div> |
| | <p>启动范围输入值</p> <p>请参考下面低限区颜色与高限值颜色的说明。</p> |
| | <p>目标值和高低限值为变量</p> <p>当设定目标值与高低限值为变量时，低限值地址为读取存储器地址+1；高限值地址为读取存储器地址+2；目标值地址为读取存储器地址+3。</p> |
| <p>低限区颜色 高限区颜色</p> <p>在设定值属性里有勾选启动范围输入值才会显示。下图为将低限值设为 30、低限区颜色设为绿色，高限值设为 70、高限区颜色设为红色执行结果。（输入最小值与输入最大值分别为 0、100）</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>值为 20 时</p> </div> <div style="text-align: center;">  <p>值为 50 时</p> </div> <div style="text-align: center;">  <p>值为 80 时</p> </div> </div> | |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.4.2 偏差型

| 偏差型柱状图属性说明 | | | | | | | |
|---|--|---|---|--------|---|--|---|
| 人机读取控制器对应的寄存器的数值与所设定的标准值相减，会得到一个偏差值，并且将数值以图形差量表的方式，显示在人机屏幕上。 | | | | | | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 | | | | | | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） | | | | | | |
| 外框颜色 | 设定偏差型柱状图元件边框颜色。 | | | | | | |
| 元件前景颜色 元件背景颜色 | 设定偏差型柱状图前景与背景颜色。下图为前景颜色设为绿色，背景颜色设为黄色的显示结果：  | | | | | | |
| 元件型式 | <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Standard</td> <td>Raised</td> <td>Sunken</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> | Standard | Raised | Sunken |  |  |  |
| Standard | Raised | Sunken | | | | | |
|  |  |  | | | | | |
| 显示格式 | Horizontal: 以水平方式显示与标准值的偏差量。 Vertical: 以垂直方式显示与标准值的偏差量。 | | | | | | |
| 设定值 | <table border="1" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;">  </td> <td style="width: 50%; vertical-align: top;"> 数值单位 提供 Word、Double Word。 数值格式 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex </td> </tr> </table> |  | 数值单位 提供 Word、Double Word。 数值格式 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex | | | | |
| |  | 数值单位 提供 Word、Double Word。 数值格式 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex | | | | | |
| | 目标值 | 用来计算偏差量的基准值。 | | | | | |
| | 输入最小值 输入最大值 | 偏差柱状图两端的最大最小值。 | | | | | |
| | 显示偏差 | 勾选后，才能设定 偏差上限值 ，并依指定的 颜色 来显示与标准值之间的偏差量，若不勾选则忽略 偏差上限值 及 颜色 ，而以 元件前景颜色 直接显示偏差量。 | | | | | |
| 标准值和偏差值上限为变量 | 当设定标准值与偏差值上限为变量时，标准值地址为 读取存储器地址+1 ；偏差值上限的地址为 读取存储器地址+2 。 | | | | | | |



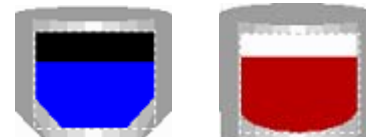
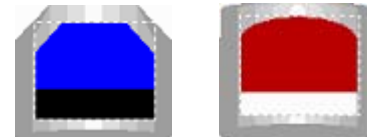
| 偏差型柱状图属性说明 | |
|------------|---|
| 设定值 | 标准值、输入最小值/输入最大值与偏差上限值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。 |
| | 选取 word/unsigned decimal, 标准值 设 50, 输入最小值 设 0, 输入最大值 设 100, 偏差上限 为 20 时, 执行结果如下图所示: |
| |  |
| | <p style="text-align: center;">值为 10 时 值为 30 时 值为 70 时 值为 90 时</p> |
| | |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |


3.8.5 管状图



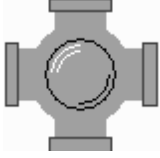
图 3-8-22 管状图

3.8.5.1 管状图(1)/管状图(2)

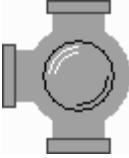
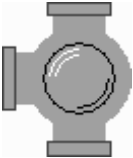
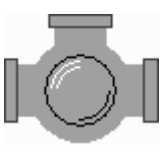
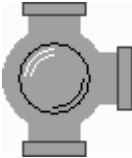
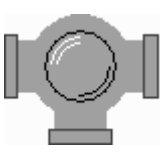
| 管状图(1)/管状图(2)属性说明 | |
|--|--|
| 人机读取对应寄存器的数值，将数值转换为容器的水位容量，显示于人机的管状图(1)/管状图(2)元件上。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) |
| 水位颜色 筒内颜色 | <p>设定管状图(1)/管状图(2)元件水位颜色与容器筒内未填满水时的颜色。</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>管状图(1) 水位为蓝色、筒内为黑色 管状图(2) 水位为红色、筒内为白色</p> |
| 元件型式 | <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Standard</p>  </div> <div style="text-align: center;"> <p>Rotation 180</p>  </div> </div> |

| 管状图(1)/管状图(2)属性说明 | | | | | | | | | |
|-------------------|---|----------------------|---------------------|-------|----------------|---------|----------------------|-----------------|---|
| 设定值 | <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 1; padding-left: 10px;"> <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex </div> </div> | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">输入最小值 输入最大值</td> <td>容器内能存放水位的最小单位与最大单位。</td> </tr> <tr> <td>目标值设定</td> <td>您可以决定是否要显示目标值。</td> </tr> <tr> <td>启动范围输入值</td> <td>请参考下面低限区颜色与高限区颜色的说明。</td> </tr> <tr> <td>目标值和高低限值 为变量</td> <td>当设定目标值与高低限值为变量时，低限值地址为<u>读取存储器地址+1</u>；高限值地址为<u>读取存储器地址+2</u>；目标值地址为<u>读取存储器地址+3</u>。</td> </tr> </table> <p>目标值、低限值、高限值、输入最小值与输入最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。</p> | 输入最小值 输入最大值 | 容器内能存放水位的最小单位与最大单位。 | 目标值设定 | 您可以决定是否要显示目标值。 | 启动范围输入值 | 请参考下面低限区颜色与高限区颜色的说明。 | 目标值和高低限值 为变量 | 当设定目标值与高低限值为变量时，低限值地址为 <u>读取存储器地址+1</u> ；高限值地址为 <u>读取存储器地址+2</u> ；目标值地址为 <u>读取存储器地址+3</u> 。 |
| | 输入最小值 输入最大值 | 容器内能存放水位的最小单位与最大单位。 | | | | | | | |
| | 目标值设定 | 您可以决定是否要显示目标值。 | | | | | | | |
| | 启动范围输入值 | 请参考下面低限区颜色与高限区颜色的说明。 | | | | | | | |
| 目标值和高低限值 为变量 | 当设定目标值与高低限值为变量时，低限值地址为 <u>读取存储器地址+1</u> ；高限值地址为 <u>读取存储器地址+2</u> ；目标值地址为 <u>读取存储器地址+3</u> 。 | | | | | | | | |
| 低限区颜色 高限区颜色 | <p>在设定值属性里有勾选启动范围输入值才会显示。下图为低限值设为 30、低限区颜色设为绿色，高限值设为 70、高限区颜色设为红色执行结果。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>值为 20 时</p> </div> <div style="text-align: center;">  <p>值为 50 时</p> </div> <div style="text-align: center;">  <p>值为 80 时</p> </div> </div> | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | |
| Y | 元件左上角 Y 座标 | | | | | | | | |
| 宽 | 元件宽度 | | | | | | | | |
| 高 | 元件高度 | | | | | | | | |

3.8.5.2 管状图(3)

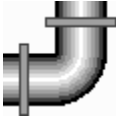
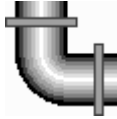
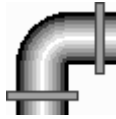
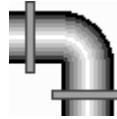
| 管状图(3)属性说明 | |
|---|------------|
| 连接水管元件用。如下图所示 | |
|  | |
| 管口口径：设定口径大小，可选择的口径大小为 1~5。口径 1 代表水管的宽度至少 13 个 pixels，口径 2 代表水管的宽度 26 个 pixels。其它以此类推。 | |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |

3.8.5.3 管状图(4)

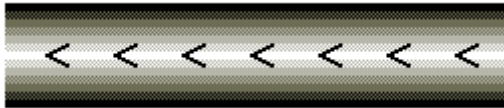
| 管状图(4)属性说明 | | | |
|---|---|--|---|
| 连接水管元件用。如下图所示 | | | |
|  | | | |
| 管口口径：设定口径大小，可选择的口径大小为 1~5。口径 1 代表水管的宽度至少 13 个 pixels，口径 2 代表水管的宽度 26 个 pixels。其它以此类推。 | | | |
| 元件型式： | | | |
| Standard | Rotation 90 | Rotation 180 | Rotation 270 |
|  |  |  |  |
| X | 元件左上角 X 座标 | | |
| Y | 元件左上角 Y 座标 | | |

3.8.5.4 管状图(5)

| 管状图(5)属性说明 | |
|---|--|
| 连接水管元件用。如下图所示 | |
|  | |
| 管口口径：设定口径大小，可选择的口径大小为 1~5。口径 1 代表水管的宽度至少 13 个 pixels，口径 2 代表水管的宽度 26 个 pixels。其它以此类推。 | |

| 管状图(5)属性说明 | | | |
|---|--|--|---|
| 元件型式: | | | |
| Standard  | Rotation 90  | Rotation 180  | Rotation 270  |
| X | 元件左上角 X 座标 | | |
| Y | 元件左上角 Y 座标 | | |

3.8.5.5 管状图(6)/管状图(7)

| 管状图(6)/管状图(7)属性说明 | |
|-------------------|---|
| 水平与垂直水管可显示水流动向。 | |
| 读取存储器地址 | <p>可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)。此元件可选择是否要输入读取存储器位置。如果有设定读取位置,则水管元件就会有水流流动的效果。例如:人机内部存储器\$0。当\$0=1时,配合流动游标颜色设定,此时管状图元件的水流方向是由右至左;当\$0=2时,管状图元件的水流方向是由左至右。当\$0=1或2以外的数字时,则管状图无呈现任何水流状态。同样的若选管状图元件7,例如:人机内部存储器\$1,当\$1=1时,水流方向是由下至上;当\$1=2时,水流方向是由上至下,当\$0=1或2以外的数字时,则管状图无呈现任何水流状态。</p>  |
| 流动游标颜色 | 如果有设定读取位置,则水管元件就会有水流流动的效果。您可以设定此流动游标的颜色。 |
| 管口口径 | 设定口径大小,可选择的口径大小为1~5。口径1代表水管的宽度至少13个pixels,口径2代表水管的宽度26个pixels。其它以此类推。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |

3.8.6 扇形图

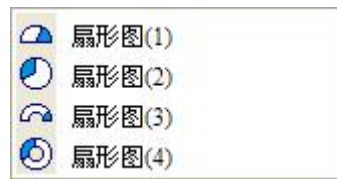










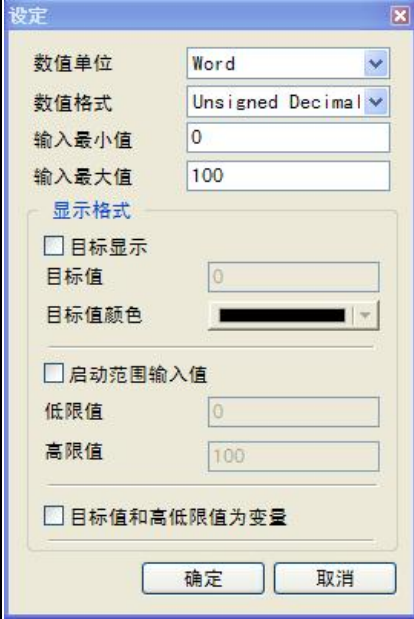
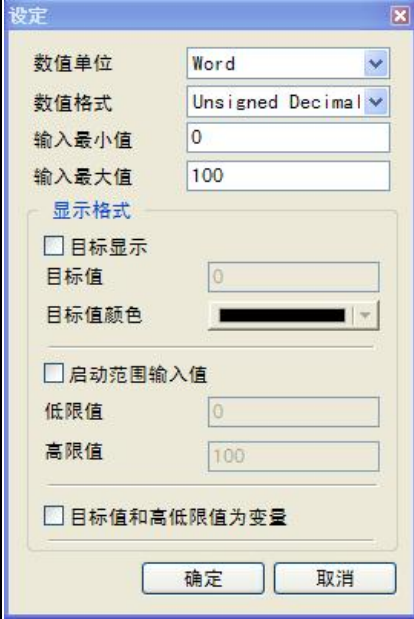
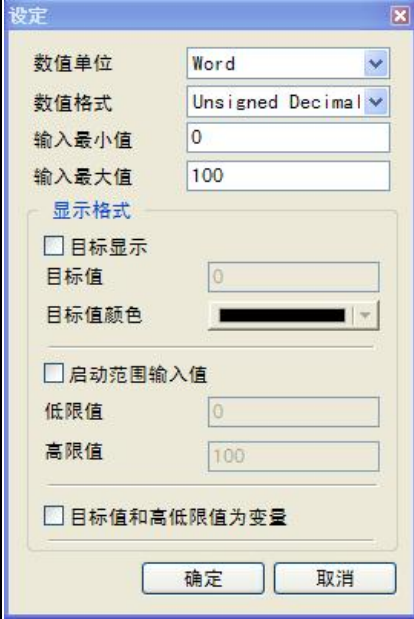
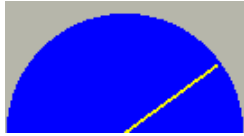



图 3-8-23 扇形图

| 扇形图属性说明 | | | | | | | | | |
|--|--|--|--|--|-------------|---|---|--|---|
| 扇形图共有四种可选择，可设定最大最小值、高低限的值跟颜色等等，用来显示设定地址的计量大小，随着面积的增加减少快速判别数量。如果设定地址的值低于下限或是高于上限，会变色以利用户分辨，并有警示的作用。 | | | | | | | | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 | | | | | | | | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） | | | | | | | | |
| 外框颜色 元件前景颜色 元件背景颜色 | 设定扇形图外框、前景与背景颜色。下图为扇形图外框颜色设为蓝色、前景颜色设为绿色，背景颜色设为黄色的显示结果：  | | | | | | | | |
| 元件型式 | <table border="1"> <tr> <td>Standard</td> <td>Raised</td> <td>Sunken</td> <td>Transparent</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> | Standard | Raised | Sunken | Transparent |  |  |  |  |
| Standard | Raised | Sunken | Transparent | | | | | | |
|  |  |  |  | | | | | | |
| 设定值 | <table border="1"> <tr> <td>  </td> <td> <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex <p>输入最小值/输入最大值 显示区间用的最小值与最大值。</p> </td> </tr> <tr> <td colspan="2"> 目标值、低限值、高限值、输入最小值与输入最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。 </td> </tr> </table> |  | <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex <p>输入最小值/输入最大值 显示区间用的最小值与最大值。</p> | 目标值、低限值、高限值、输入最小值与输入最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。 | | | | | |
|  | <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex <p>输入最小值/输入最大值 显示区间用的最小值与最大值。</p> | | | | | | | | |
| 目标值、低限值、高限值、输入最小值与输入最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式作数值范围的检查。 | | | | | | | | | |

| 扇形图属性说明 | |
|------------------------|--|
| 设定值 | <p>目标值设定</p> <p>您可以决定是否要显示目标值，设定目标值及其颜色后扇形图会从中心点的位置拉出一条目标线指到您设定的目标值上如下图所示：（这里我们目标值设为 80，目标值颜色为黄色）</p>  |
| | <p>启动范围输入值</p> <p>请参考下面低限值颜色与高限值颜色的说明。</p> |
| | <p>目标值和高低限值为变量</p> <p>当设定目标值与高低限值为变量时，低限值地址为<u>读取存储器地址+1</u>；高限值地址为<u>读取存储器地址+2</u>；目标值地址为<u>读取存储器地址+3</u>。</p> |
| <p>低限值颜色 高限值颜色</p> | <p>在<u>设定值</u>属性里有勾选<u>启动范围输入值</u>才会显示。下图为低限值设为 30、低限值颜色设为绿色，高限值设为 70、高限值颜色设为红色执行结果。（输入最小值与输入最大值分别为 0、100）</p>  <p style="text-align: center;"> 值为 20 时 值为 50 时 值为 80 时 </p> |
| X | 元件左上角 X 坐标 |
| Y | 元件左上角 Y 坐标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.7 指示灯

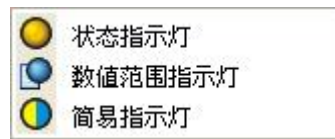



图 3-8-24 指示灯

3.8.7.1 状态指示灯

| 状态指示灯属性说明 | | |
|--|--|---------------------|
| <p>状态指示灯的作用在于指示某一个地址的状态，不管是 Bit、LSB 或是 Word，都会提醒用户状态的改变。如果此地址是一个很重要的指标或是代表很重要的信息或是警示，则利用立即改变显示状态的方式或是借由不同状态文字的设置，来立即告诉用户状态的变化，甚至随著不同状态的改变，让用户知道更多的信息，使得用户能在第一时间完成相对应状态的处理。</p> | | |
| 读取存储器地址 | <p>可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。当您所设定的读取存储器地址为控制器的接点时（ON 或 OFF），状态指示灯会依照您所规划的状态作变化。例如值为 1 时显示“Start”的状态、值为 0 时显示“Stop”的状态，您也可以为状态指示灯的每一个状态加入图形的显示效果。</p> | |
| 文字/文字大小 字体/文字颜色 | <p>用户可依 Windows® 所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明）</p> | |
| 是否闪烁 | <p>以闪烁的显示方式提醒用户。（参阅一般按钮说明）</p> | |
| 图形库名称 图形名称 | <p>（参阅一般按钮说明）</p> | |
| 图形背景是否透明 指定图形透明色 | <p>（参阅一般按钮说明）</p> | |
| 元件前景颜色 元件型式 | <p>（参阅一般按钮说明）</p> | |
| 数值单位 | Bit | 状态指示灯元件可以有 2 个状态。 |
| | Word | 状态指示灯元件可以有 256 个状态。 |
| | LSB | 状态指示灯元件可以有 16 个状态。 |
| 数值格式 | <p>状态指示灯提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来显示读取到的存储器内容。</p> | |
| 新增删除状态数 | <p>设定状态指示灯的状态总数。如果数值单位为 Word，则可以设定 1-256 个状态；LSB 就可以设定 16 个状态；Bit 只能设定 2 个状态。</p> | |
| 不可见位元 | <p>当此位元被设为 On 时，按钮元件将被隐藏。</p> | |
| X | <p>元件左上角 X 座标</p> | |
| Y | <p>元件左上角 Y 座标</p> | |

| 状态指示灯属性说明 | |
|-----------|------|
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.7.2 数值范围指示灯

| 数值范围指示灯属性说明 | | | | | | | | | | | | |
|---|--|------|------|------|------|------|------|--|--|-----|----|----|
| 数值范围指示灯的作用，在于指示某一个地址的状态。人机读取对应的寄存器的数值，以此数值对应此元件与所设定的范围值，最后将对应的状态显示于人机屏幕上。 | | | | | | | | | | | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。您也可以为指示灯的每一个状态加入图形的显示效果。 | | | | | | | | | | | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。（参阅一般按钮说明） | | | | | | | | | | | |
| 是否闪烁 | 以闪烁的显示方式提醒用户。（参阅一般按钮说明） | | | | | | | | | | | |
| 图形库名称 图形名称 | （参阅一般按钮说明） | | | | | | | | | | | |
| 图形背景是否透明 指定图形透明色 | （参阅一般按钮说明） | | | | | | | | | | | |
| 元件前景颜色 元件型式 | （参阅一般按钮说明） | | | | | | | | | | | |
| 新增删除状态数 | 设定数值范围指示灯的状态总数。最多可设定 256 个状态。 | | | | | | | | | | | |
| 设定值 | <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 1; padding-left: 10px;"> <p>数值单位 提供 Word、Double Word。</p> <p>数值格式 提供以下的数值格式可供选择： Word/Double Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal </div> </div> | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">范围</th> <th style="width: 15%;">常量</th> <th style="width: 15%;">范围 0</th> <th style="width: 15%;">范围 1</th> <th style="width: 15%;">范围 2</th> <th style="width: 15%;">范围 3</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>100</td> <td>50</td> <td>33</td> <td>1</td> </tr> </tbody> </table> <p>以建立后的预设的 5 个 State 来设定范围值。n 个 State 会有 n-1 个范围值可以输入。将状态 0、1、2、3、4 的元件前景颜色分别设为红、绿、蓝、黄、紫。</p> | 范围 | 常量 | 范围 0 | 范围 1 | 范围 2 | 范围 3 | | | 100 | 50 | 33 |
| 范围 | 常量 | 范围 0 | 范围 1 | 范围 2 | 范围 3 | | | | | | | |
| | | 100 | 50 | 33 | 1 | | | | | | | |

| 数值范围指示灯属性说明 | | | |
|-------------|------------|----|--|
| 设定值 | 范围 | 常量 | 当读取的存储器地址数值为大于等于 100 时数值范围指示灯会呈现红色, 当读取的存储器地址数值为大于等于 50, 数值范围指示灯会呈现绿色, 其它以此类推。 |
| | | 变量 | 当范围被设定为变量时, 数值范围指示灯元件会以读取存储器的地址后 n-1 个地址当作范围下限值, 其中 n 为数值范围指示灯的状态总数。例如: 若读取存储器地址为\$0, 元件的状态总数为 5, 范围 0 的下限值即为\$1、范围 1 的下限值即为\$2, 其它以此类推。 |
| X | 元件左上角 X 座标 | | |
| Y | 元件左上角 Y 座标 | | |
| 宽 | 元件宽度 | | |
| 高 | 元件高度 | | |

3.8.7.3 简易指示灯








| 简易指示灯属性说明 | |
|---|--|
| 提供基本的两个状态 (on/off), 给用户方便作底图的 XOR 颜色交错变化。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能, 设定该元件文字显示型态。(参阅一般按钮说明) |
| XOR 颜色 | 指定与底图 XOR 的颜色。 |
| 是否重绘 | 开启重绘功能【Yes】可使此元件重叠放置在动态元件上, 也能顺利读取动态元件变化的资料, 倘若关闭重绘功能【No】则无法顺利显示动态元件变化的资料。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8 资料显示



图 3-8-25 资料显示

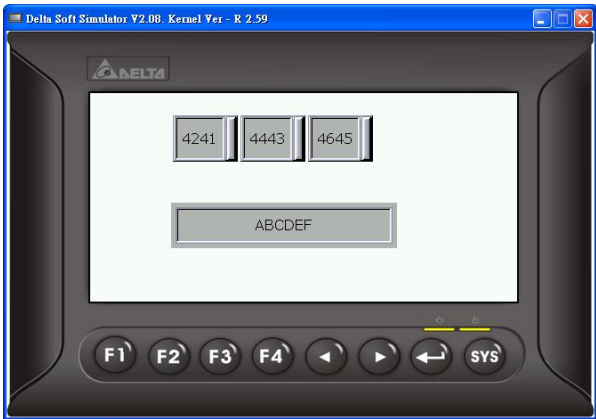
3.8.8.1 数值显示

| 数值显示属性说明 | | | | |
|----------------------------------|---|---|---|--|
| 此元件会读取所设定地址的值，依照用户所设定的格式，即时显示出来。 | | | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） | | | |
| 文字大小 文字颜色 | 设定显示文数字的大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 | | | |
| 外框颜色 元件背景颜色 | 当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为绿色，外框颜色设为红色的 Sunken 数值显示元件。  （当元件型式为 Transparent，外框颜色与元件背景颜色是无作用） | | | |
| 元件型式 | Standard  | Raised  | Sunken  | Transparent  |
| 前位数值补零 | 下两张图为补零与不补零的变化。（请注意整数位数设为 4 位）  | | | |
| 设定值 |  | | | |

| 数值显示属性说明 | | | | | | | | | | | | | | | | | | |
|-----------------------------|---|---|------|-------------|--------|--------|---------------|---------------|-------------------|-------------------|---------------------|---------------------|--------|--------|-----------|-----------|--|-------------|
| 设定值 | 数值单位 | 提供 16bits Word、与 32bits Double Word。 | | | | | | | | | | | | | | | | |
| | 数值格式 | 选择不同的数值单位会有不同的数值格式可供选择如下： <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Word</td> <td style="width: 50%;">Double Word</td> </tr> <tr> <td>1. BCD</td> <td>1. BCD</td> </tr> <tr> <td>2. Signed BCD</td> <td>2. Signed BCD</td> </tr> <tr> <td>3. Signed Decimal</td> <td>3. Signed Decimal</td> </tr> <tr> <td>4. Unsigned Decimal</td> <td>4. Unsigned Decimal</td> </tr> <tr> <td>5. Hex</td> <td>5. Hex</td> </tr> <tr> <td>6. Binary</td> <td>6. Binary</td> </tr> <tr> <td></td> <td>7. Floating</td> </tr> </table> | Word | Double Word | 1. BCD | 1. BCD | 2. Signed BCD | 2. Signed BCD | 3. Signed Decimal | 3. Signed Decimal | 4. Unsigned Decimal | 4. Unsigned Decimal | 5. Hex | 5. Hex | 6. Binary | 6. Binary | | 7. Floating |
| | Word | Double Word | | | | | | | | | | | | | | | | |
| | 1. BCD | 1. BCD | | | | | | | | | | | | | | | | |
| 2. Signed BCD | 2. Signed BCD | | | | | | | | | | | | | | | | | |
| 3. Signed Decimal | 3. Signed Decimal | | | | | | | | | | | | | | | | | |
| 4. Unsigned Decimal | 4. Unsigned Decimal | | | | | | | | | | | | | | | | | |
| 5. Hex | 5. Hex | | | | | | | | | | | | | | | | | |
| 6. Binary | 6. Binary | | | | | | | | | | | | | | | | | |
| | 7. Floating | | | | | | | | | | | | | | | | | |
| 整数位数 小数位数 | 决定输入的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示格式，只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。 | | | | | | | | | | | | | | | | | |
| 增益 (a) 偏移 (b) 运算后四舍五入 | 套用 $y=(a)x+(b)$ 公式决定显示的变化。例如增益 (a) 我们设 2、偏移 (b) 设 3，当读取存储器地址值为 3 时，显示的数值将变为 $9=(2)\times 3+(3)$ 。若设定的增益或偏移数值为小数时，请将数值格式设定为 Floating。 | | | | | | | | | | | | | | | | | |
| 显示快速更新 | 针对元件显示数值作即时的更新。 请注意，此功能是针对设定元件作较频繁的通讯动作，以达到快速更新数值的效果，所以一个画面中最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能，会拖慢人机正常执行速率，所以如非特殊需要，建议不要开启此功能。 您可以从菜单里的【画面】>>【画面属性】设定快速更新频率，分为高、中、低三个等级。 | | | | | | | | | | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | | | | | | | | | | |
| Y | 元件左上角 Y 座标 | | | | | | | | | | | | | | | | | |
| 宽 | 元件宽度 | | | | | | | | | | | | | | | | | |
| 高 | 元件高度 | | | | | | | | | | | | | | | | | |

3.8.8.2 文数值显示

| 文数值显示属性说明 | |
|--|-------------------------------|
| 此元件会读取所设定地址的值，转换成字元的型式显示出来，因此所读取的值必须是 ASCII 的显示格式，否则用户将无法看到所显示的字元。（字串长度最长 28 个字） | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 |
| 文字大小 | 设定显示文数字的大小与颜色，这里所使用的字体是采用人机系统 |

| 文数值显示属性说明 | | | | | | | | | | | | | | | | | |
|----------------------|---|----------------------|--|-------|-----|---|----|---|----|---|----|---|----|---|----|---|----|
| 文字颜色 | 里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 | | | | | | | | | | | | | | | | |
| 元件型式 | (参阅数值显示说明) | | | | | | | | | | | | | | | | |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) | | | | | | | | | | | | | | | | |
| 字符串长度 | <p>可显示的字符串长度范围为 1~28 个字。</p> <p>我们将文数值显示元件的读取地址设为内部存储器\$0，字符串长度设为 6，并建立三个数值输入元件，数值格式设定为 Hex，并输入数值为</p> <p style="text-align: center;">\$0 = 4241 (Hex) \$1 = 4443 (Hex) \$2 = 4645 (Hex)</p>  <p>会得到如上图的执行结果，请注意文数值显示元件是读取位元组资料(Byte)，而一个数值输入元件预设数值格式为 Word，是由两个位元组(Byte)所组成，所以一个 Word 可代表两个文数字。但是文数值显示元件读取到的数值需将高低位元组对换才会是实际显示的文数字，所以\$0 为 4241 (Hex)，经过高低位元组对换后，会显示为 AB，以此类推。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">ASCII Code / Hex 对应表</th> </tr> <tr> <th>ASCII</th> <th>Hex</th> </tr> </thead> <tbody> <tr><td>A</td><td>41</td></tr> <tr><td>B</td><td>42</td></tr> <tr><td>C</td><td>43</td></tr> <tr><td>D</td><td>44</td></tr> <tr><td>E</td><td>45</td></tr> <tr><td>F</td><td>46</td></tr> </tbody> </table> | ASCII Code / Hex 对应表 | | ASCII | Hex | A | 41 | B | 42 | C | 43 | D | 44 | E | 45 | F | 46 |
| ASCII Code / Hex 对应表 | | | | | | | | | | | | | | | | | |
| ASCII | Hex | | | | | | | | | | | | | | | | |
| A | 41 | | | | | | | | | | | | | | | | |
| B | 42 | | | | | | | | | | | | | | | | |
| C | 43 | | | | | | | | | | | | | | | | |
| D | 44 | | | | | | | | | | | | | | | | |
| E | 45 | | | | | | | | | | | | | | | | |
| F | 46 | | | | | | | | | | | | | | | | |
| 显示快速更新 | <p>针对元件显示数值作即时的更新。请注意，此功能是针对设定元件作较频繁的通讯动作，以达到快速更新数值的效果，所以一个画面最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能，会拖慢人机正常执行速率，所以如非特殊需要，建议不要开启此功能。您可以从菜单里的【画面】>>【画面属性】设定快速更新频率，分为高、中、低三个等级。</p> | | | | | | | | | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | | | | | | | | | |

| 文数值显示属性说明 | |
|-----------|------------|
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8.3 日期显示

| 日期显示属性说明 | |
|-----------------------------|--|
| 此元件会显示人机端内部所设定的日期。可选择显示的格式。 | |
| 文字大小 文字颜色 | 设定显示文数字的大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供 8~64 的系统字体大小可供选择。 |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) |
| 元件型式 | (参阅数值显示说明) |
| 日期显示格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY 三种显示格式。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8.4 时间显示

| 时间显示属性说明 | |
|-------------------------------|--|
| 此元件会显示人机端内部所设定的时间。可选择时间显示的格式。 | |
| 文字大小 文字颜色 | 设定显示文数字的大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供 8~64 的系统字体大小可供选择。 |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) |
| 元件型式 | (参阅数值显示说明) |
| 时间显示格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8.5 星期显示

| 星期显示属性说明 | |
|--|--|
| 可显示星期中的每一天，总共有七个状态，用户可依照需要自订每天所显示的文字、字体及颜色，这些属性您可以在星期显示元件的属性栏位中改变。 | |
| 文字大小 文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) |
| 元件型式 | (参阅数值显示说明) |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8.6 一般型信息显示

| 一般行信息显示属性说明 | |
|--|--|
| 设定状态的总数以及各种状态的文字，如此依照所设定的读取地址的值，来改变显示其对应的状态文字。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)。 |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) |
| 元件型式 | (参阅数值显示说明) |
| 数值单位 | Bit 复状态元件可以有 2 个状态。 |
| | Word 复状态元件可以有 256 个状态。 |
| | LSB 复状态元件可以有 16 个状态。 |
| 数值格式 | 一般型信息显示提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 |
| 新增删除状态数 | 设定一般型信息显示之状态总数。如果数值单位为 Word，则可以显示 1-256 个状态文字；LSB 就可以显示 16 个状态文字；Bit 只能显示 2 个状态文字。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.8.7 走马灯信息显示

| 走马灯信息显示属性说明 | | |
|---|--|-------------------|
| 文字显示会有以走马灯的方式循环移动，此外还可以设定每次移动点数、显示方向以及间隔时间，来决定走马灯的显示方式。 | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)。 | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) | |
| 外框颜色 元件背景颜色 | (参阅数值显示说明) | |
| 元件型式 | (参阅数值显示说明) | |
| 数值单位 | Bit | 复状态元件可以有 2 个状态。 |
| | Word | 复状态元件可以有 256 个状态。 |
| | LSB | 复状态元件可以有 16 个状态。 |
| 数值格式 | 走马灯信息显示提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 | |
| 新增删除状态数 | 设定一般型信息显示之状态总数。如果数值单位为 Word，则可以显示 1-256 个状态文字；LSB 就可以显示 16 个状态文字；Bit 只能显示 2 个状态文字。 | |
| 显示方向 | Left | 显示的步进方向是由右至左。 |
| | Right | 显示的步进方向是由左至右。 |
| | Top | 显示的步进方向是由下至上。 |
| | Bottom | 显示的步进方向是由上至下。 |
| 每次移动点数 | 文字步进的量，单位为 Pixel，范围从 1~50 Pixels。 | |
| 间隔时间(毫秒) | 文字步进与下次文字步进时的间隔时间，单位为 ms，范围从 50~3000 ms。 | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |



3.8.9 图形显示



图 3-8-26 图形显示

3.8.9.1 状态图显示

| 状态图显示属性说明 | |
|--|---|
| 在人机使用时，您可以借由状态图显示元件来设定多个状态图形并由所设定的寄存器读取地址所读取到的数值转换为所要切换的状态，将个别的状态图形显示到人机屏幕上。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明）。 |
| 图形库名称 图形名称 | (参阅数值显示说明) |
| 图形背景是否透明 指定图形透明色 | (参阅数值显示说明) |
| 元件前景颜色 | (参阅数值显示说明) |
| 数值单位 | Bit 复状态元件可以有 2 个状态。 |
| | Word 复状态元件可以有 256 个状态。 |
| | LSB 复状态元件可以有 16 个状态。 |
| 数值格式 | 状态图元件提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 |
| 新增删除状态数 | 设定一般型信息显示的状态总数。如果数值单位为 Word，则可以显示 1-256 个状态文字；LSB 就可以显示 16 个状态文字；Bit 只能显示 2 个状态文字。 |
| 自动变换图形 | No 由设定的寄存器地址数值来决定显示切换到哪一个状态图。例如：\$0 为 0 时切换至 State 0，\$0 为 5 时切换至 State 5。 |
| | Yes 当设定的寄存器地址数值为非零的值时状态图显示元件会自动变换图形。例如：\$0 存储器位置的内容值为 1 以上，元件就会根据您所设定的图形变换时间，自动变换图形。此 \$0 存储器位置的内容值为 0，则此状态图复原最初状态并且不动作 |

| 状态图显示属性说明 | | | |
|-----------|---|--|---|
| 自动变换图形 | Variation | 以 <u>读取存储器地址</u> 为作状态图形的状态切换寄存器，而 <u>读取存储器地址+1</u> 用来设定是否要自动变换图形；举例来说，当\$0 为状态图显示元件的寄存器地址，\$1 则为切换自动变换图形寄存器，当\$1=0，需利用\$0 切换状态，此时元件的动作与上图【自动变换图形 (No)】是一样的。当\$1=1 以上，此时元件的动作与上图【自动变换图形 (Yes)】是一样的。 | |
| 透明色 | 选择 Yes 状态图显示元件会以透明状呈现，通常是配合图形透明色来同步使用，请参阅下列图形，元件前景颜色为红色，画面背景颜色为白色： | | |
| |  |  |  |
| | 未决定任何透明色 | 指定元件透明色 | 指定图形透明色 |
| | 请注意此属性在设定为 Yes 后，元件前景颜色的属性是无作用的。 | | |
| X | 元件左上角 X 座标 | | |
| Y | 元件左上角 Y 座标 | | |
| 宽 | 元件宽度 | | |
| 高 | 元件高度 | | |

状态图显示功能实例说明

指定存储器读地址=D100，各状态与存储器内容值



3.8.9.2 动画

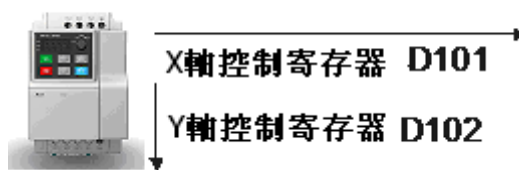
| 动画属性说明 | | |
|---|------------------------------|------------------------------|
| 可以借由动画元件来设定多个状态图形并由设定的寄存器读取地址所读取到的数值转换为元件所要切换的状态，同时可控制元件移动的位置，将个别的状态图形以及移动位置显示在人机屏幕上。 | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） | |
| | 读取存储器地址 | 以 <u>读取存储器地址</u> 的值当作动画的状态图形 |

| 动画属性说明 | | |
|---------------------|---|--|
| | | 的切换。 |
| 读取存储器地址 | 读取存储器地址+1 | 以 <u>读取存储器地址+1</u> 的值作为动画元件水平 (Horizontal) 轴的位置。 |
| | 读取存储器地址+2 | 以 <u>读取存储器地址+2</u> 的值作为动画元件垂直 (vertical) 轴的位置。 |
| 图形库名称 图形名称 | (参阅一般按钮说明) | |
| 图形背景是否透明 指定图形透明色 | (参阅一般按钮说明) | |
| 是否清除图形 | 移动过程中以及切换状态图片时是否要清除上一个状态的图形。 | |
| 数值单位 | Word | 动画元件可以有 256 个状态。 |
| | LSB | 动画元件可以有 16 个状态。 |
| 数值格式 | 动画元件提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 | |
| 新增删除状态数 | 设定动画元件的状态总数。如果数值单位为 Word，则可以显示 1-256 个状态图形；LSB 就可以显示 16 个状态图形；Bit 只能显示 2 个状态图形。 | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

动画元件功能实例说明

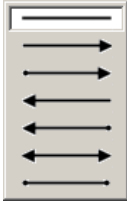
指定存储器读地址=D100，各状态与存储器内容值如下：

状态控制寄存器 D100



3.8.9.3 动态线条

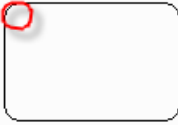
| 动态线条属性说明 | | |
|--------------------------------|------------------------------|---|
| 控制所绘制的线条能依据寄存器值作线条的延展与线条颜色的变化。 | | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) | |
| | 读取存储器地址 | 当 <u>位置可变</u> 属性为 Yes 时此 <u>读取存储器地址</u> 的值方能作用，其意义为元件左上角水平 (Left) 位置也就是屏幕第一点的 X 座标 (X ₁)。 |

| | | |
|----------|---|---|
| | 读取存储器地址+1 | 当 <u>位置可变</u> 属性为 Yes 时此 <u>读取存储器地址+1</u> 的值方能作用，其意义为元件左上角垂直 (Top) 位置也就是屏幕第一点的 Y 座标 (Y_1)。 |
| 动态线条属性说明 | | |
| 读取存储器地址 | 读取存储器地址+2 | 当 <u>位置可变</u> 属性为 Yes 时此 <u>读取存储器地址+2</u> 的值方能作用，其意义为元件右下角水平 (Right) 位置也就是屏幕第二点的 X 座标 (X_2)。 |
| | 读取存储器地址+3 | 当 <u>位置可变</u> 属性为 Yes 时此 <u>读取存储器地址+3</u> 的值方能作用，其意义为元件右下角垂直 (Bottom) 位置也就是屏幕第二点的 Y 座标 (Y_2)。 |
| | 读取存储器地址+4 | 当 <u>颜色可变</u> 属性为 Yes 时此 <u>读取存储器地址+4</u> 的值方能作用，其意义为线条颜色，数值范围依照机型不同而不同，16 色灰阶机种为 0~15，256 色机种为 0~255，65536 色机种为 0~65535。 |
| | 读取存储器地址+5 | 当是否闪烁属性为 Yes 时此 <u>读取存储器地址+5</u> 的值方能作用，其内容值大于 1 时，则动态线条元件呈现闪烁状态，其内容值 0 时，则不闪烁。 |
| 线条颜色 | 可设定动态线条的显示颜色。 | |
| 是否闪烁 | 以闪烁的显示方式提醒用户。 | |
| 线条宽度 | 可设 1~8 Pixels 的线宽。 | |
| 数值格式 | 动态线条元件提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 | |
| 线条型式 | 提供下图的线条型式可供选择  | |
| 位置可变 | (参阅动态线条读取存储器地址栏位说明) | |
| 颜色可变 | (参阅动态线条读取存储器地址栏位说明) | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.9.4 动态矩形

| 动态矩形属性说明 | |
|----------------------------------|------------------------------|
| 控制所绘制的矩形能依据寄存器值作矩形的延展与矩形前景颜色的变化。 | |
| 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) |

| 动态矩形属性说明 | | |
|----------|--|--|
| 读取存储器地址 | 读取存储器地址 | 当 位置可变 属性为 Yes 时此 读取存储器地址 属性的值方能作用, 其意义为元件左上角水平(Left)位置也就是屏幕第一点的 X 座标(X_1)。 |
| | 读取存储器地址+1 | 当 位置可变 属性为 Yes 时此 读取存储器地址+1 属性的值方能作用, 其意义为元件左上角垂直(Top)位置也就是屏幕第一点的 Y 座标(Y_1)。 |
| | 读取存储器地址+2 | 当 大小可变 属性为 Yes 时此 读取存储器地址+2 属性的值方能作用, 其意义为元件右下角水平(Right)位置以第一点为基准向水平位置延伸, 也就是宽。 |
| | 读取存储器地址+3 | 当 大小可变 属性为 Yes 时此 读取存储器地址+3 属性的值方能作用, 其意义为元件右下角垂直(Bottom)位置以第一点为基准向垂直位置延伸, 也就是高。 |
| | 读取存储器地址+4 | 当 颜色可变 属性为 Yes 时此 读取存储器地址+4 属性的值方能作用, 其意义为矩形前景颜色值。数值范围依照机型不同而不同, 16 色灰阶机种为 0~15, 256 色机种为 0~255, 65536 色机种为 0~65535。 |
| | 读取存储器地址+5 | 当 是否闪烁 属性为 Yes 时此 读取存储器地址+5 属性的值方能作用, 其内容值大于 1 时, 则动态线条元件呈现闪烁状态, 其内容值 0 时, 则不闪烁。 |
| | 请注意当您选择 位置可变 的属性为 No 时, 大小可变 的存储器参考存储器地址将自动往上补位(读取存储器地址 将变成为元件右下角水平(Right)位置, 读取存储器地址+1 将变成为元件右下角垂直(Bottom)位置, 读取存储器地址+2 将变成为矩形前景颜色值)。 | |
| 是否闪烁 | 以闪烁的显示方式提醒用户。 | |
| 元件前景颜色 | 可设定动态矩形的显示颜色。 | |
| 线条宽度 | 可设 1~8 Pixels 的动态矩形边线宽。 | |
| 数值格式 | 动态矩形元件提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 | |
| 透明色 | 仅以矩形边线显示, 中间将镂空, 元件前景颜色功能失效。 | |

| 动态矩形属性说明 | |
|----------|--|
| 圆角半径 | 提供 0~30 pixels 圆角半径可供选择。  |
| 位置可变 | (参阅动态矩形读取存储器地址栏位说明) |
| 大小可变 | (参阅动态矩形读取存储器地址栏位说明) |
| 颜色可变 | (参阅动态矩形读取存储器地址栏位说明) |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.9.5 动态椭圆形

| 动态椭圆形属性说明 | | |
|-----------------------------------|------------------------------|--|
| 控制所绘制的椭圆形能依据寄存器值作椭圆的延展与椭圆前景颜色的变化。 | | |
| | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明) | |
| 读取存储器地址 | 读取存储器地址 | 当 中心点可变 属性为 Yes 时此 读取存储器地址 属性的值方能作用, 其意义为元件中心点水平位置也就是屏幕第一点的 X 座标 (X_1)。 |
| | 读取存储器地址+1 | 当 中心点可变 属性为 Yes 时此 读取存储器地址+1 属性的值方能作用, 其意义为元件中心点垂直位置也就是屏幕第一点的 Y 座标 (Y_1)。 |
| | 读取存储器地址+2 | 当 半径可变 属性为 Yes 时此 读取存储器地址+2 属性的值方能作用, 其意义为元件水平半径。以第一点为基准向水平位置延伸, 也就是宽。 |
| | 读取存储器地址+3 | 当 半径可变 属性为 Yes 时此 读取存储器地址+3 属性的值方能作用, 其意义为元件垂直半径。以第一点为基准向垂直位置延伸, 也就是高。 |
| | 读取存储器地址+4 | 当 颜色可变 属性为 Yes 时此 读取存储器地址+4 属性的值方能作用, 其意义为椭圆形前景颜色值, 数值范围依照机型不同而不同, 16 色灰阶机种为 0~15, 256 色机种为 0~255, 65536 色机种为 0~65535。 |
| | 读取存储器地址+5 | 当 是否闪烁 属性为 Yes 时此 读取存储器地址+5 属性的值方能作用, 其内容值大于 1 时, 则动态线条元件呈现闪烁状态, 其内容值 0 时, 则不闪烁。 |

| 动态椭圆形属性说明 | |
|-----------|---|
| | 请注意当您选择 <u>中心点可变</u> 的属性为 No 时, <u>半径可变</u> 的存储器参考存储器地址将往上补位 (<u>读取存储器地址</u> 将变成元件水平半径, <u>读取存储器地址+1</u> 将变成元件垂直半径, <u>读取存储器地址+2</u> 将变成椭圆形前景颜色值)。 |
| 线条颜色 | 可设定动态椭圆形的边线颜色。 |
| 是否闪烁 | 以闪烁的显示方式提醒用户。 |
| 元件前景颜色 | 可设定动态椭圆形的显示颜色。 |
| 线条宽度 | 可设 1~8 Pixels 的动态椭圆形边线宽。 |
| 数值格式 | 动态椭圆形元件提供 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的存储器内容。 |
| 透明色 | 仅以椭圆边线显示, 中间将镂空, 元件前景颜色功能失效。 |
| 中心点可变 | (参阅动态椭圆形读取存储器地址栏位说明) |
| 半径可变 | (参阅动态椭圆形读取存储器地址栏位说明) |
| 颜色可变 | (参阅动态椭圆形读取存储器地址栏位说明) |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.9.6 即时图显示

| 即时图显示属性说明 | |
|--|--------------------------------|
| 1. 提供一个 D11 函数库, 让用户能够透过外部程序的函数呼叫, 以串行通讯界面, 将自订的图形即时地传到 HMI。 2. 一个画面上只能有一个即时图形元件。 | |
| 元件前景颜色 | 可设定即时图背景的显示颜色。 |
| 传输方式 | 通讯协议, RS-232/RS-485。 |
| 通讯端口 | 人机上的通讯串口, 使用上不能够与 PLC 通讯的串口共用。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

| 即时图显示属性说明 | |
|-----------|--|
| 范例说明 | <p>1. 载入 ImgTrans.dll (文件置于安装目录\Utility 下)</p> <p>2. 资料结构定义:</p> <pre style="text-align: center;">typedef struct _COMM_INFO { char szCOM[8]; } COMM_INFO;</pre> <p>, szCOM 为一字串, 传入值为 COM1, COM2, COM3, ...。</p> <p>3. 函数列表:</p> <p>(1) int hmOpen(const COMM_INFO* pCommInfo);</p> <p>(2) int hmSendImageFromFile(LPCTSTR szFileName);</p> |
| 范例说明 | <p>(3) HANDLE hmAsyncSendImageFromFile(LPCTSTR szFileName);</p> <p>(4) int hmSendImage(HBITMAP hbmp);</p> <p>(5) HANDLE hmAsyncSendImage(HBITMAP hbmp);</p> <p>(6) int hmAbortAction();</p> <p>(7) int hmClose();</p> |

| 即时图显示属性说明 | |
|-----------|--|
| | <p>4. 函数说明:</p> <p>(1) 函数: <code>int hmOpen(const COMM_INFO* pCommInfo);</code> 传入值: COMM_INFO 结构 回传值: 1: 成功, 0: 失败 说明: 开启人机的通讯, 传入 COM Port 的字串, 如 COM1, COM2 等</p> <p>(2) 函数: <code>int SendImageFromFile(LPCTSTR szFileName);</code> 传入值: 图形文件文件名 回传值: 1: 成功, 0: 失败 说明: 将传入的图形格式转换成人机上元件的宽高和 Bits 值后, 开始传输图形资料给人机, 此函数为同步函数, 待资料传输完成后, 此函数才会返回。</p> <p>(3) 函数: <code>HANDLE hmAsyncSendImageFromFile(LPCTSTR szFileName);</code> 传入值: 图形文件文件名 回传值: 0: 失败, 非 0 值为该 Thread Handle 说明: 将传入的图形格式转换成人机上元件的宽高和 Bits 值后, 开始传输图形资料给人机, 此函数为非同步函数, 用户可以取得目前传输图形资料的 Thread Handle 来执行相关的运算。</p> <p>(4) 函数: <code>int hmSendImage(HBITMAP hbmp);</code> 传入值: Window 中的 HBITMAP Handle 回传值: 1: 成功, 0: 失败 说明: 将传入的图形格式转换成人机上元件的宽高和 Bits 值后, 开始传输图形资料给人机, 此函数为同步函数, 待资料传输完成后, 此函数才会返回。</p> <p>(5) 函数: <code>HANDLE hmAsyncSendImage(HBITMAP hbmp);</code> 传入值: Window 中的 HBITMAP Handle 回传值: 0: 失败, 非 0 值为该 Thread Handle</p> |



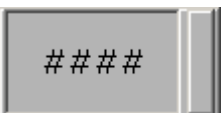
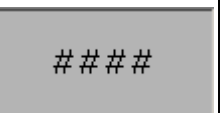
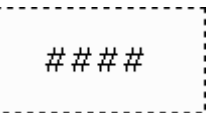

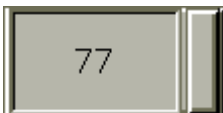
| 即时图显示属性说明 | |
|-----------|--|
| 范例说明 | <p>说明：将传入的图形格式转换成人机上元件的宽高和 Bits 值后，开始传输图形资料给人机，此函数为非同步函数，用户可以取得目前传输图形资料的 Thread Handle 来执行相关的运算。</p> <p>(6) 函数：int hmAbortAction(); 传入值：无 回传值：0:失败, 1:成功 说明：中止非同步函数的图形传输。</p> <p>(7) 函数：int hmClose(); 传入值：无 回传值：1: 成功, 0:失败 说明：关闭人机的通讯(D11 结束时，此函数会自动被叫起)</p> |

3.8.10 输入








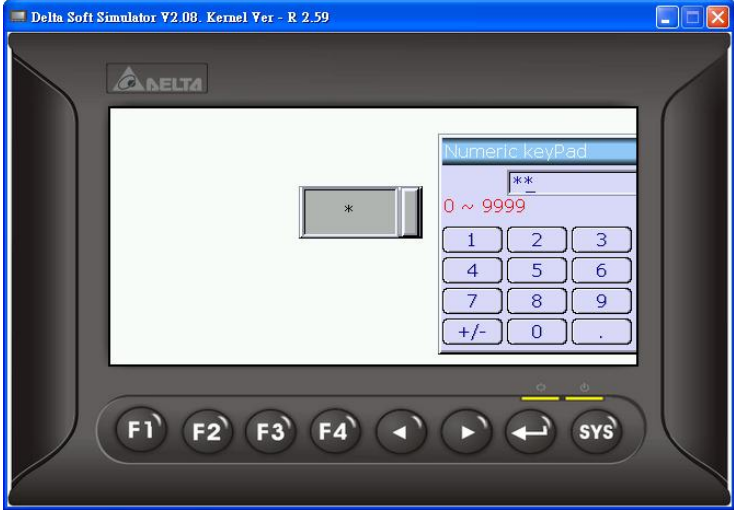


图 3-8-27 输入

3.8.10.1 数值输入


| 数值输入属性说明 | | | | |
|--|---|---|--|--|
| 触摸此数值输入元件，人机马上显示系统内建的 TEN-KEY 在屏幕上供用户输入数值。当按下 ENTER 时人机就送出数值给相对应的寄存器，且输入的最大值与最小值可由用户自行决定，同时还可以设定输入前或是输入后触发地址来触发指定的某一 Bit 地址。 | | | | |
| 写入存储器地址 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） 若只有设定写入存储器地址而没有设定读取存储器地址，人机会自动读取写入存储器地址的数值。 | | | |
| 文字大小 文字颜色 | 设定数值输入的文字大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 | | | |
| 外框颜色 元件背景颜色 | 当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为蓝色，外框颜色设为红色的 Raised 数值输入元件。  （型式为 Transparent，外框颜色与元件背景颜色是无作用） | | | |
| 元件型式 | Standard  | Raised  | Sunken  | Transparent  |
| 前位数值补零 | 下两张图为补零与不补零的变化。（请注意整数位数设为 4 位）   | | | |
| 触发 触发方式 | 用户可选择数值写入前后触发指定的 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 | | | |


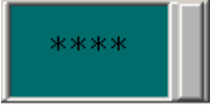


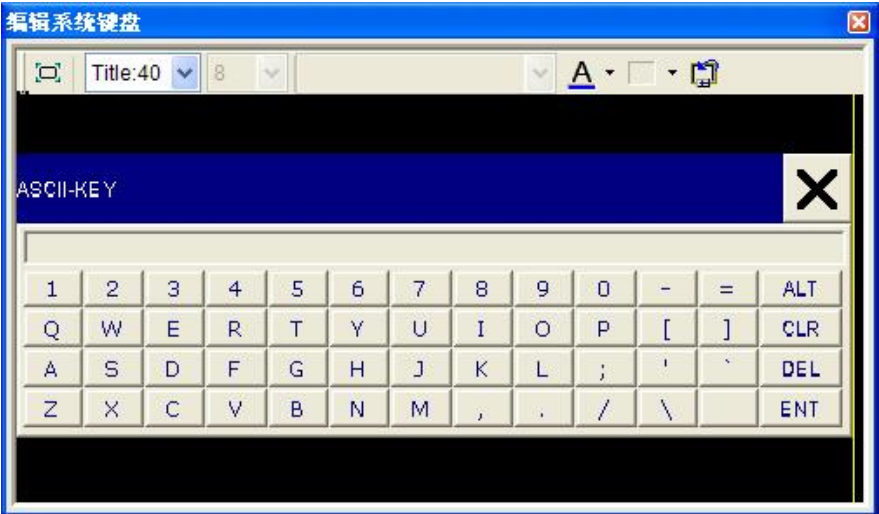



| 数值输入属性说明 | | | | | | | | | | | | | | | | | |
|---|--|-------------|-------------|--------|--------|---------------|---------------|-------------------|-------------------|---------------------|---------------------|--------|--------|-----------|-----------|--|-------------|
| | | | | | | | | | | | | | | | | | |
| 设定值 | <p>数值单位 提供 16bits Word 与 32bits Double Word。</p> | | | | | | | | | | | | | | | | |
| | <p>数值格式 选择不同的数值单位会有不同的数值格式可供选择如下：</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">Word</td> <td style="width: 50%;">Double Word</td> </tr> <tr> <td>1. BCD</td> <td>1. BCD</td> </tr> <tr> <td>2. Signed BCD</td> <td>2. Signed BCD</td> </tr> <tr> <td>3. Signed Decimal</td> <td>3. Signed Decimal</td> </tr> <tr> <td>4. Unsigned Decimal</td> <td>4. Unsigned Decimal</td> </tr> <tr> <td>5. Hex</td> <td>5. Hex</td> </tr> <tr> <td>6. Binary</td> <td>6. Binary</td> </tr> <tr> <td></td> <td>7. Floating</td> </tr> </table> | Word | Double Word | 1. BCD | 1. BCD | 2. Signed BCD | 2. Signed BCD | 3. Signed Decimal | 3. Signed Decimal | 4. Unsigned Decimal | 4. Unsigned Decimal | 5. Hex | 5. Hex | 6. Binary | 6. Binary | | 7. Floating |
| | Word | Double Word | | | | | | | | | | | | | | | |
| 1. BCD | 1. BCD | | | | | | | | | | | | | | | | |
| 2. Signed BCD | 2. Signed BCD | | | | | | | | | | | | | | | | |
| 3. Signed Decimal | 3. Signed Decimal | | | | | | | | | | | | | | | | |
| 4. Unsigned Decimal | 4. Unsigned Decimal | | | | | | | | | | | | | | | | |
| 5. Hex | 5. Hex | | | | | | | | | | | | | | | | |
| 6. Binary | 6. Binary | | | | | | | | | | | | | | | | |
| | 7. Floating | | | | | | | | | | | | | | | | |
| <p>编辑系统键盘 当输入方式设定为触控弹跳时，可自订系统键盘的格式。包括标题、字体、字体大小、字体颜色、键盘颜色等。系统键盘无法指定显示的位置，一律置于屏幕中央。</p> | | | | | | | | | | | | | | | | | |

| 数值输入属性说明 | | |
|------------|--|--|
| 编辑系统 键盘 |  可调整系统键盘的大小 | |
| | Title:40  设定标题栏高度 | |
| | 10  设定字体大小 | |
| | Arial  设定字体 | |
| |  设定字体颜色 | |
| |  设定背景颜色 | |
| |  预设大小 | |
| 设定值 | 最小值 最大值 用户可设定输入值之最小与最大值, 限制输入数值之范围。 | |
| | 整数位数 小数位数 决定输入的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值, 只是显示样式, 只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。 | |
| | 若勾选最大值与最小值为变量, 最小值由 <u>读取存储器地址+1</u> 、最大值由 <u>读取存储器地址+2</u> 来决定。 | |
| | 增益 (a) 偏移 (b) 运算后 四舍五入 套用 $y=(a)x+(b)$ 公式决定显示的变化。例如增益 (a) 我们设 2、偏移 (b) 设 3, 当读取存储器地址值为 3 时, 显示的数值将变为 $9 = (2) \times 3 + (3)$ 。 若设定的增益或偏移数值为小数时, 请将数值格式设定为 Floating。 | |
| | 您输入的最小值与最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式、整数位数与小数位数作数值范围的检查。 | |
| 启动输入方式 | 分为触控弹跳、主动非弹跳、触控非弹跳。预设值为触控弹跳启动输入。主动非弹跳、触控非弹跳功能将在键盘元件里详述。 | |
| 用户权限 | 利用此功能可设定该元件按压动作的权限, 只有高于或等于设定权限才能使用。 | |
| 显示为 * 号 | 选择 Yes, 当您在输入数值时, 会呈现如下图所示:  | |

| 数值输入属性说明 | |
|--------------|--|
| 显示快速更新 | 针对元件显示数值作即时的更新。 请注意，此功能是针对设定元件作较频繁的通讯动作，以达到快速更新数值的效果，所以一个画面最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能，会拖慢人机正常执行速率，所以如非特殊需要，建议不要开启此功能。 您可以从菜单里的【画面】>>【画面属性】设定快速更新频率，分为高、中、低三个等级。 |
| 输入后设为最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低，可预防 Operator 的误触。 |
| 生效位元 生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 |
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 |
| 超过范围信息提示 | 启用此功能时，当输入超过所设定的范围值时 HMI 会出现提示信息视窗。 |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.10.2 文数值输入

| 文数字输入属性说明 | |
|--|---|
| 设定写入的地址与读取的地址。由于文数字输入元件可以显示字元，所以只支持 ASCII 的输入与显示。输入跟显示的地址可以是不相同的。（字符串长度限制最长 28 个字） | |
| 写入存储器地址 读取存储器地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） 若只有设定写入存储器地址而没有设定读取存储器地址，人机会自动读取写入存储器地址的数值。 |
| 文字大小 文字颜色 | 设定文数字输入的文字大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 |
| 外框颜色 元件背景颜色 | 当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为蓝色，外框颜色设为灰色的 Raised 文数字输入元件。 <div style="text-align: center;">  </div> (型式为 Transparent，外框颜色与元件背景颜色是无作用) |

| 文数字输入属性说明 | | | | |
|---|---|---|--|--|
| 元件型式 | Standard  | Raised  | Sunken  | Transparent  |
| 字符串长度 | 可显示的字符串长度范围为 1~28 个字。预设值为 4 个字。 | | | |
| 触发 触发方式 | 用户可选择文数字写入前后触发指定的控制器 Bit 地址为 ON。注意:因触发功能仅将该控制器地址设为 ON, 如需再触发时则用户需自行将该地址清为 OFF。 | | | |
| 编辑系统键盘 | 当输入方式设定为触控弹跳时, 可自订系统键盘的格式。包括标题、字体、字体大小、字体颜色、键盘颜色等。 | | | |
| |  | | | |
| |  | 可选择系统键盘的大小 | | |
| | Title:40 | 设定标题列高度 | | |
| | 10 | 设定字体大小 | | |
| | Arial | 设定字体 | | |
| | A | 设定字体颜色 | | |
| |  | 设定背景颜色 | | |
|  | 预设大小 | | | |
| 启动输入方式 | 分为触控弹跳、主动非弹跳、触控非弹跳。预设值为触控弹跳启动输入。主动非弹跳、触控非弹跳功能将在键盘元件里详述。 | | | |
| 用户权限 | 利用此功能可设定该元件按压动作的权限, 只有高于或等于设定权限才能使用。 | | | |


| 文数字输入属性说明 | |
|-----------|---|
| 显示为 * 号 | <p>选择 Yes, 当您在输入文数字时, 会呈现如下图所示:</p>  |
| 显示快速更新 | <p>针对元件显示数值作即时的更新。请注意, 此功能是针对设定元件作较频繁的通讯动作, 以达到快速更新数值的效果, 所以一个画面最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能, 会拖慢人机正常执行速率, 所以如非特殊需要, 建议不要开启此功能。您可以从菜单里的【画面】>>【画面属性】设定快速更新频率, 分为高、中、低三个等级。</p> |
| 输入后设为最低权限 | <p>强制在执行按压动作之后将目前的用户权限设为最低, 可预防 Operator 的误触。</p> |
| 生效位元生效准位 | <p>当设定的生效位元为 ON 或是 OFF 时 (由生效准位决定), 此按键才能操作。</p> |
| 执行前宏 | <p>执行按钮动作之前, 会先执行此宏。</p> |
| 执行后宏 | <p>执行按钮动作之后, 再执行此宏。</p> |
| 不可见位元 | <p>当此位元被设为 On 时, 按钮元件将被隐藏。</p> |
| X | <p>元件左上角 X 座标</p> |
| Y | <p>元件左上角 Y 座标</p> |
| 宽 | <p>元件宽度</p> |
| 高 | <p>元件高度</p> |


3.8.10.3 Barcode 输入

| Barcode 输入属性说明 | |
|--|-------------------------------------|
| <p>设定写入的地址与读取的地址。由于文数字输入元件可以显示字元, 所以只支持 ASCII 的输入与显示。输入跟显示的地址可以是不相同的。(字串长度限制最长 127 个字)</p> | |
| 写入存储器地址 | <p>可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明)</p> |

| Barcode 输入属性说明 | |
|----------------|---|
| 读取存储器地址 | 若只有设定写入存储器地址而没有设定读取存储器地址，人机会自动读取写入存储器地址的数值。 |

| Barcode 输入属性说明 | | | | | | | | | | | | | |
|---|---|---|---|----------|-------------|---|---|--|---|---|--------|---|--------|
| 文字大小 文字颜色 | 设定文数字输入的文字大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 | | | | | | | | | | | | |
| 外框颜色 元件背景颜色 | <p>当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为蓝色，外框颜色设为灰色的 Raised 文数字输入元件。</p>  <p>(型式为 Transparent，外框颜色与元件背景颜色是无作用)</p> | | | | | | | | | | | | |
| 元件型式 | <table border="1"> <thead> <tr> <th>Standard</th> <th>Raised</th> <th>Sunken</th> <th>Transparent</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | Standard | Raised | Sunken | Transparent |  |  |  |  | | | | |
| Standard | Raised | Sunken | Transparent | | | | | | | | | | |
|  |  |  |  | | | | | | | | | | |
| 字串长度 | 可显示的字串长度范围为 1~127 个字。预设值为 4 个字。 | | | | | | | | | | | | |
| 触发 触发方式 | 用户可选择文数字写入前后触发指定的控制器 Bit 地址为 ON。注意：因触发功能仅将该控制器地址设为 ON，如需再触发时则用户需自行将该地址清为 OFF。 | | | | | | | | | | | | |
| 编辑系统键盘 | <p>当输入方式设定为触控弹跳时，可自订系统键盘的格式。包括标题、字体、字体大小、字体颜色、键盘颜色等。</p>  <table border="1"> <tbody> <tr> <td></td> <td>可选择系统键盘的大小</td> </tr> <tr> <td>Title:40</td> <td>设定标题栏高度</td> </tr> <tr> <td>10</td> <td>设定字体大小</td> </tr> <tr> <td>Arial</td> <td>设定字体</td> </tr> <tr> <td>A</td> <td>设定字体颜色</td> </tr> <tr> <td></td> <td>设定背景颜色</td> </tr> </tbody> </table> |  | 可选择系统键盘的大小 | Title:40 | 设定标题栏高度 | 10 | 设定字体大小 | Arial | 设定字体 | A | 设定字体颜色 |  | 设定背景颜色 |
|  | 可选择系统键盘的大小 | | | | | | | | | | | | |
| Title:40 | 设定标题栏高度 | | | | | | | | | | | | |
| 10 | 设定字体大小 | | | | | | | | | | | | |
| Arial | 设定字体 | | | | | | | | | | | | |
| A | 设定字体颜色 | | | | | | | | | | | | |
|  | 设定背景颜色 | | | | | | | | | | | | |

| Barcode 输入属性说明 | |
|----------------|--|
| |  预设大小 |
| 启动输入方式 | 分为触控弹跳、主动非弹跳、触控非弹跳。预设值为触控弹跳启动输入。主动非弹跳、触控非弹跳功能将在键盘元件里详述。 |

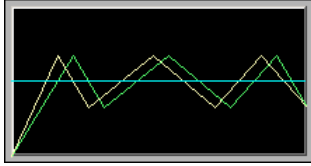
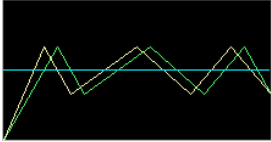
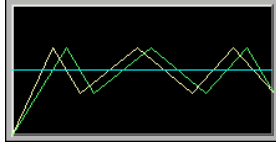
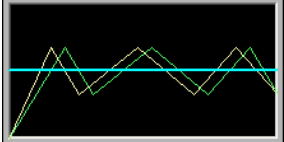

| Barcode 输入属性说明 | |
|----------------|---|
| 用户权限 | 利用此功能可设定该元件按压动作的权限，只有高于或等于设定权限才能使用。 |
| 显示为 * 号 | 选择 Yes，当您在输入文数字时，会呈现如下图所示：  |
| 显示快速更新 | 针对元件显示数值作即时的更新。请注意，此功能是针对设定元件作较频繁的通讯动作，以达到快速更新数值的效果，所以一个画面最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能，会拖慢人机正常执行速率，所以如非特殊需要，建议不要开启此功能。您可以从菜单里的【画面】>>【画面属性】设定快速更新频率，分为高、中、低三个等级。 |
| 输入后设为最低权限 | 强制在执行按压动作之后将目前的用户权限设为最低，可预防 Operator 的误触。 |
| 生效位元生效准位 | 当设定的生效位元为 ON 或是 OFF 时（由生效准位决定），此按键才能操作。 |
| 执行前宏 | 执行按钮动作之前，会先执行此宏。 |
| 执行后宏 | 执行按钮动作之后，再执行此宏。 |
| 不可见位元 | 当此位元被设为 On 时，按钮元件将被隐藏。 |
| 弹跳启动位元 | 此位元被设为 On 时，可使用手动输入。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |


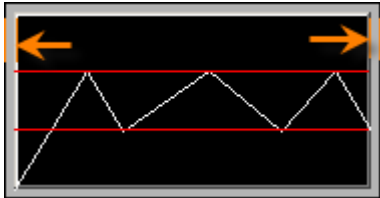
3.8.11 曲线图

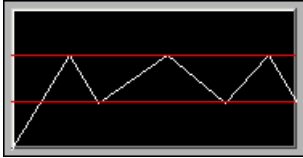


图 3-8-28 曲线图

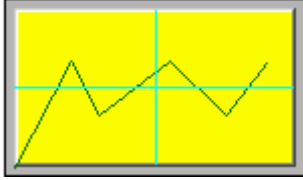
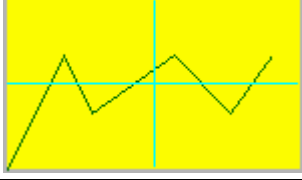
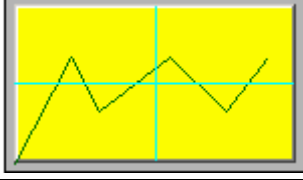
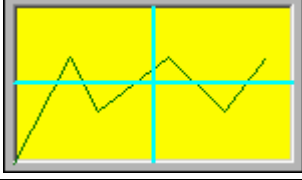
3.8.11.1 一般曲线图

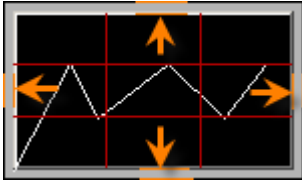
| 一般曲线图属性说明 | |
|---|--|
| <p>一般曲线图可在属性的曲线栏位数上设定总共需要几条曲线，并在设定值的属性栏位里的将读取位置设定存储器地址与读取的存储器格式，之后在曲线选项中设定线条宽度与线条颜色。一个一般曲线图元件最多只可同时显示四条曲线。</p> <p>一般曲线图元件会从用户设定的读取地址开始，连续读取” 取样点数” 个寄存器，转换为曲线图后显示于人机屏幕上。此元件功能为将存储器地址数值读取并静态的显示在画面上，需要通过设定控制区的标志位来触发曲线图的描绘与清除，有关控制区的设定请参考第三章 3.4 节控制区及状态区说明。</p> | |
| <p>外框颜色 元件背景颜色</p> | <p>当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为黑色，外框颜色设为灰色的 Raised 一般曲线图元件。</p>  |
| <p>元件型式</p> | <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Standard</p>  </div> <div style="text-align: center;"> <p>Raised</p>  </div> <div style="text-align: center;"> <p>Sunken</p>  </div> </div> |
| <p>曲线数目</p> | <p>可设定 1~4 条显示曲线。</p> |
| <p>设定值</p> |  |

| 一般曲线图属性说明 | | |
|-----------|------------------------|---|
| 设定值 | <p>取样点数 最大显示点数</p> | <p>可设定为常数，也可设定为变数。 <u>取样点数为常数时：</u> 取样点数代表一条曲线将由几笔寄存器资料构成。当取样点数超出元件可绘制的宽度时，会出现如下图之警告视窗。</p>  <p>也就是说，当所建立的一般曲线图宽度只能容纳 10 的取样点，但若用户设定取样点数超过 10 时，软件就会出现警告。</p>  <p>当取样点数为常数时最大显示点数栏位是无法使用的。</p> <p><u>取样点数为变数时：</u> 当取样点数设定为寄存器时，此时取样点数即为变数，用户可通过设定寄存器数值，变更取样点数数量。当取样点数为变数时，必须设定最大显示点数，若设定的取样点数值大于所设定的最大显示点数值时，系统只会取样最大显示点数值的点数。</p> |
| | 数值格式 | <p>Word</p> <ol style="list-style-type: none"> 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex |
| | 读取地址 | <p>可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） 一般曲线图元件会从用户设定的读取地址开始，连续读取 ”取样点数” 个寄存器，转换为曲线图后显示于人机屏幕上。</p> |

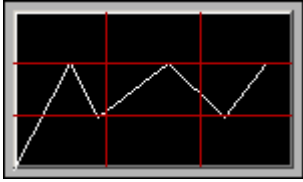
| 一般曲线图属性说明 | | |
|---------------|--|--|
| 设定值 | 取样标志 | 设定触发以及清除的标志，当取样标志被触发时，才会开始读取数据资料并绘出曲线图。此取样标志位于控制区编号 2 的寄存器里，有关控制区的设定请参考第三章 3.4 节控制区及状态区说明。 |
| | 最小值 最大值 | 设定资料显示的最大值与最小值，也是 Y 轴的最大值与最小值。如果读取的资料高于或低于此值，则显示此值。 |
| | 线条宽度 | 设定曲线线条显示的宽度，范围从 1~8，单位为 Pixel。 |
| | 线条颜色 | 设定曲线线条显示的颜色。 |
| 格子颜色 横向格子数 | 请参考以下设定。我们将格子颜色设为红色，横向格子数设为 3 条。  | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.11.2 X-Y 曲线图

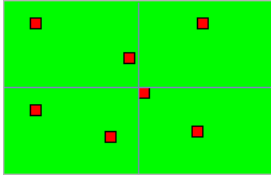
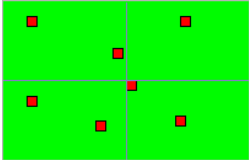
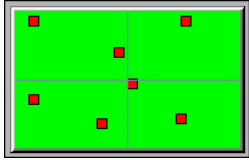
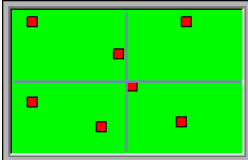

| X-Y 曲线图属性说明 | | | |
|--|---|--|---|
| 人机读取设定的存储器地址的连续数据，转换为 X-Y 曲线图后显示在人机屏幕上。此元件功能为将存储器地址数值读取并静态的显示在画面上，需要通过设定控制区的标志位来触发曲线图的描绘与清除，有关控制区的设定请参考第三章 3.4 节控制区及状态区说明。 | | | |
| 外框颜色 元件背景颜色 | 当元件型式为 Raised、Sunken 外框颜色才有作用。下图为元件背景颜色设为黄色，外框颜色设为灰色的 Raised X-Y 曲线图元件。  | | |
| 元件型式 | Standard  | Raised  | Sunken  |
| 连接相邻两点 | 若选择 Yes，在绘制 X-Y 数据资料时，将点与点之间以线条连接起来。 | | |

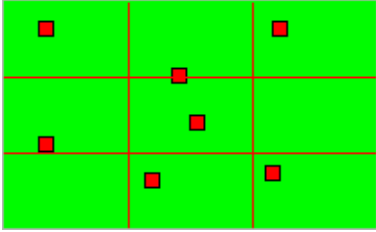
| X-Y 曲线图属性说明 | |
|-------------|---|
| 曲线数目 | 可设定 1~4 条 X-Y 显示曲线。 |
| 设定值 | <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">  </div> <p>可设定为常数，也可设定为变数。 <u>取样点数为常数时：</u> 取样点数代表一条曲线将由几笔寄存器资料构成。 当取样点数超出元件可绘制的宽度时，会出现如下图之警告视窗。</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">  </div> <p style="text-align: center;">取样点数 最大显示点数</p> <p>也就是说，当所建立的 X-Y 曲线图宽度只能容纳 10 的取样点，但若用户设定取样点数超过 10 时，软件就会出现警告。</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-bottom: 10px;">  </div> <p>当取样点数为常数时，<u>最大显示点数</u>栏位是无法使用的。</p> |

| X-Y 曲线图属性说明 | | |
|-------------|----------------|--|
| 设定值 | 取样点数 最大显示点数 | 取样点数为变数时： 当取样点数设定为寄存器时，此时取样点数即为变数，用户可通过设定寄存器数值，变更取样点数数量。当取样点数为变数时，必须设定 最大显示点数 ，若设定的取样点数值大于所设定的 最大显示点数值 时，系统只会取样 最大显示点数值 的点数。 |
| | 数值格式 | Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex |
| | 水平读取地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） X-Y 曲线图元件会从用户设定的水平读取地址开始，连续读取”取样点数”个寄存器，转换为 X 轴点数后显示于人机屏幕上。 |
| | 垂直读取地址 | 可选择内部存储器或控制器寄存器地址。（参阅一般按钮说明） X-Y 曲线图元件会从用户设定的水平读取地址开始，连续读取”取样点数”个寄存器，转换为 Y 轴点数后显示于人机屏幕上。 |
| | 取样标志 | 设定触发以及清除的标志，当取样标志被触发时，才会开始读取数据资料并绘出 X-Y 曲线图。此取样标志位于控制区编号 2 的寄存器里，有关控制区的设定请参考第三章 3.4 节控制区及状态区说明。 |
| | 水平最小值 水平最大值 | 设定水平资料显示的最大值与最小值，也是 X 轴的最大值与最小值。如果读取的资料高于或低于此值，则显示此值。 |
| | 垂直最小值 垂直最大值 | 设定垂直资料显示的最大值与最小值，也是 Y 轴的最大值与最小值。如果读取的资料高于或低于此值，则显示此值。 |
| | 线条宽度 | 设定 X-Y 曲线线条显示的宽度，范围从 1~8，单位为 Pixel。 |
| | 线条颜色 | 设定 X-Y 曲线线条显示的颜色。 |

| X-Y 曲线图属性说明 | |
|------------------------|---|
| 格子颜色 水平线总数 垂直线总数 | 请参考以下设定。我们将格子颜色设为红色，水平线与垂直线皆设为2条。  |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.11.3 X-Y 即时分布图

| X-Y 即时分布图属性说明 | |
|--|--|
| 人机读取设定的存储器地址的连续或不连续数据，转换为 X-Y 即时分布图后显示于人机屏幕上。此元件功能为将存储器地址数值读取并静态的显示在画面上，需要通过设定标志位来触发曲线图的描绘与清除。 | |
| 外框颜色 元件背景颜色 | 下图为元件背景颜色设为绿色，外框颜色设为灰色的 Standard XY 即时分布图元件。  |
| 元件型式 | <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Standard</p>  </div> <div style="text-align: center;"> <p>Raised</p>  </div> <div style="text-align: center;"> <p>Sunken</p>  </div> </div> |
| 取样数目 | 可设定 1~4 条 XY 显示点。 |
| 设定值 |  |

| X-Y 即时分布图属性说明 | | |
|------------------------|--|---|
| 设定值 | 控制地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明), 可以控制是否取样或清除。值的定义为 BIT 0 触发 (OFF→ON) 表示读取取样点。 BIT 1 触发 (OFF→ON) 表示清除取样点。 |
| | 取样后自动清除标志 | 勾选后, 读取取样点后, 会自动将控制地址标志设为 OFF |
| | 是否连续地址 | 勾选后, 所有参数读取地址会以 X 读取地址为起始地址, 自动指定连续地址为其他参数读取地址, 可以减少与控制器的通讯时间。例如当 X 设为 1@D301, 勾选连续地址, 人机会以 1@D302 当 Y 读取地址, 1@D303 当颜色地址, 1@D304 当连接地址。若没有选取, 则每个参数可以个别指定不同地址。 |
| | X 读取地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明), 数据读取回来会转换成 X 轴点座标后显示到人机屏幕上。 |
| | Y 读取地址 | 可选择内部存储器或控制器寄存器地址。(参阅一般按钮说明), 数据读取回来会转换成 Y 轴点座标后显示到人机屏幕上。 |
| | 颜色 | 取样点的颜色, 可选择内部存储器、控制器寄存器地址或常数, 数值的范围 0~65535。 |
| | 连接 | 与前一个点是否做连接线, 可选择内部存储器、控制器寄存器地址或常数, 0 表示连续两点间不连线, 1 表示连接。 |
| | X 最小值 X 最大值 | 设定 X 轴的最大值与最小值。如果读取的数据高于或低于此值, 则以此设定值显示。 |
| | Y 最小值 Y 最大值 | 设定 Y 轴的最大值与最小值。如果读取的数据高于或低于此值, 则以此设定值显示。 |
| 格子颜色 水平线总数 垂直线总数 | 将格子颜色设为红色, 水平线与垂直线皆设为 2 条。  | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.12 取样功能

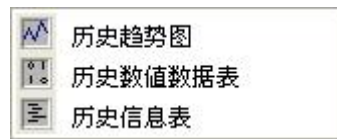

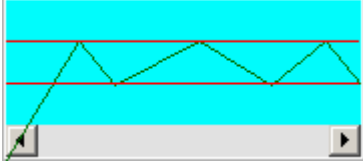


图 3-8-29 取样功能

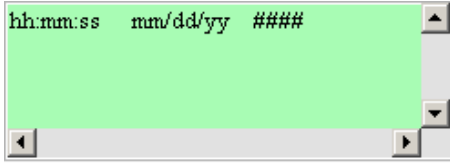
3.8.12.1 历史趋势图

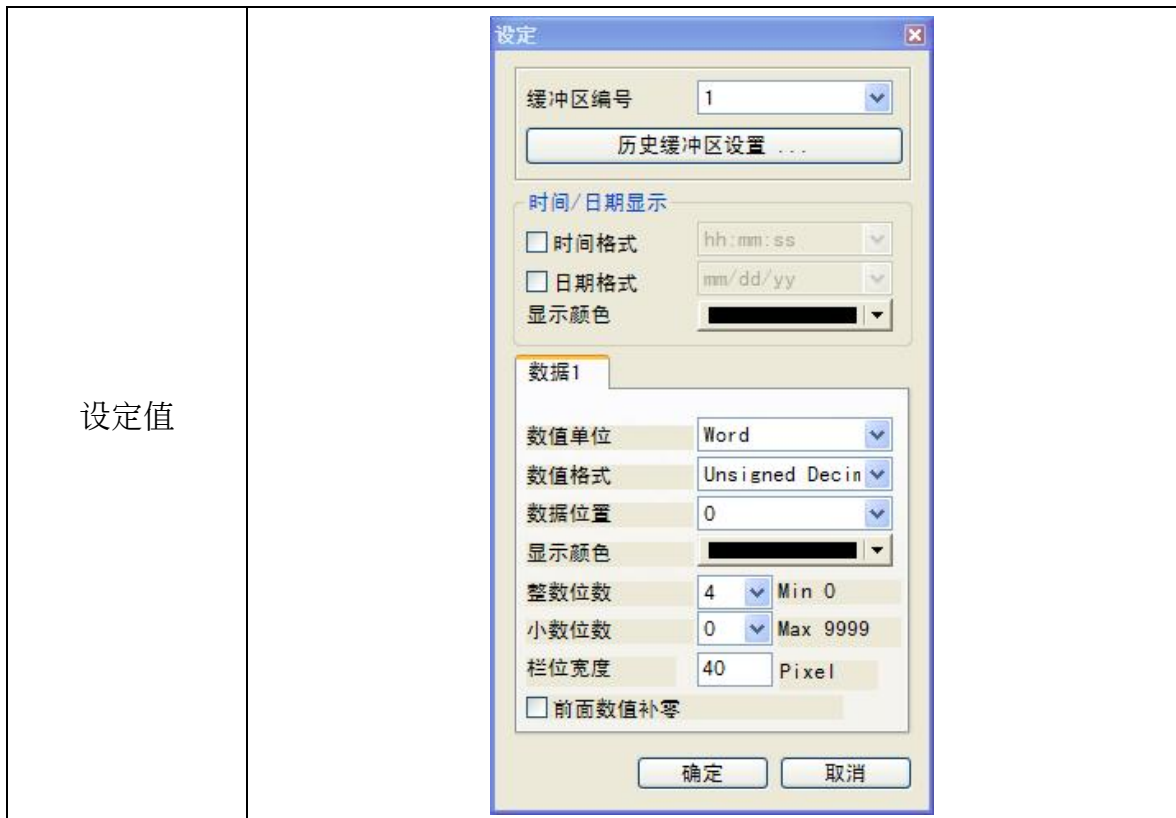
| 历史趋势图属性说明 | |
|-----------------------|--|
| 将历史资料转换为连续曲线显示于人机屏幕上。 | |
| 外框颜色 元件背景颜色 | <p>下图为元件背景颜色设为黑色，外框颜色设为灰色的历史趋势图元件。</p> |
| 曲线数目 | 可设定 1~16 条显示曲线。 |
| 设定值 | <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid blue; padding: 5px;"> <p>设定</p> <p>缓冲区编号: 1</p> <p>数值格式: Unsigned Deci</p> <p>整数位数: 4 (Min 0)</p> <p>小数位数: 0 (Max 9999)</p> <p><input checked="" type="checkbox"/> 全域范围限定</p> <p>最小值: 0</p> <p>最大值: 100</p> <p>历史缓冲区设置 ...</p> <p>时间/日期显示</p> <p><input checked="" type="checkbox"/> 时间格式: hh:mm:ss</p> <p><input checked="" type="checkbox"/> 日期格式: mm/dd/yy</p> <p>显示颜色: [Color Picker]</p> <p>曲线1 曲线2 曲线3 曲线4 曲线 < ></p> <p><input checked="" type="checkbox"/> 启动</p> <p>数据位置: 0</p> <p>线条宽度: 1</p> <p>线条颜色: [Color Picker]</p> <p>最小值: 0</p> <p>最大值: 100</p> <p>确定 取消</p> </div> <div style="border: 1px solid red; padding: 5px;"> <p>设定</p> <p>缓冲区编号: 1</p> <p>数值格式: Unsigned Deci</p> <p>整数位数: 4 (Min 0)</p> <p>小数位数: 0 (Max 9999)</p> <p><input type="checkbox"/> 全域范围限定</p> <p>最小值: 0</p> <p>最大值: 100</p> <p>历史缓冲区设置 ...</p> <p>时间/日期显示</p> <p><input checked="" type="checkbox"/> 时间格式: hh:mm:ss</p> <p><input checked="" type="checkbox"/> 日期格式: mm/dd/yy</p> <p>显示颜色: [Color Picker]</p> <p>曲线1 曲线2 曲线3 曲线4 曲线 < ></p> <p><input checked="" type="checkbox"/> 启动</p> <p>数据位置: 0</p> <p>线条宽度: 1</p> <p>线条颜色: [Color Picker]</p> <p>最小值: 0</p> <p>最大值: 100</p> <p>确定 取消</p> </div> </div> |
| 缓冲区编号 | 设定历史缓冲区编号 (No. 1~No. 12) 读取控制器资料对应的地址，您可以按下【历史缓冲区设定】按钮或是于菜单里的【选项】中【历史缓冲区设定】菜单设定控制器资料对应的地址。 |

| 历史趋势图属性说明 | | |
|-----------|--------------|--|
| 设定值 | 数值格式 | Word 1. BCD 2. Signed BCD 3. Signed Decimal 4. Unsigned Decimal 5. Hex 6. Floating 请注意当您选择数值格式为 Floating 时，历史缓冲区设定里取样的 <u>数值单位</u> 必需要大于等于 2 Words 以上，否则或出现以下的警告对话框。 |
| | |  |
| | 整数位数 小数位数 | 决定读取的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示样式，只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。 |
| | 全域范围 限定 | 开启此功能可设定此缓冲区内资料的最大值最小值范围设定，相反就根据各曲线自行定义最大最小值。 |
| | 最小值 最大值 | 设定资料显示的最大值与最小值，也是 Y 轴的最大值与最小值。如果读取的资料高于或低于此值，则显示此值。 |
| | 时间/日期显示 | |
| | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 |
| | 启动 | 若勾选，所设定的曲线资料才会显示。 |
| | 资料位置 | 每次触发时，所要读取的 Word 资料位置。例如我们在历史缓冲区设定里取样的 <u>数值单位</u> 设 3 Words，所以在 <u>资料位置</u> 的属性我们就可以选 0~2 个位置。 请注意当您选择数值格式为 Floating 时，且历史缓冲区设定里取样的 <u>数值单位</u> 为奇数个 Words 时，在选择 <u>资料位置</u> 上要特别小心，Floating 的资料位置必为偶数位。 |
| | 线条宽度 | 设定曲线线条显示的宽度，范围从 1~8，单位为 Pixel。 |
| 线条颜色 | 设定曲线线条显示的颜色。 | |

| 历史趋势图属性说明 | |
|---------------|---|
| 最小值 最大值 | 各自曲线设定资料显示的最大值与最小值, 如果此缓冲区有有使用 8 个曲线, 系统就会分别有 8 个最大值与最小值提供设定。 |
| 格子颜色 横向格子数 | 请参考以下设定。我们将格子颜色设为红色, 横向格子数设为 3 条。  |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.12.2 历史数值数据表

| 历史数值数据表属性说明 | |
|--|---|
| 将历史资料转换为数值数据, 以表格方式显示于人机屏幕上。在历史资料缓冲区里面设定读取地址, 并且在数值单位里面设定需要几个 Word, 范围为 1~16。而设定值里面的资料位置, 会相对应于在历史缓冲区中所选的数值单位。 | |
| 外框颜色 元件背景颜色 | 下图为元件背景颜色为青绿色, 外框颜色为灰色的历史数值数据表元件。  |
| 资料栏位总数 | 可设定 1~16 个数据栏位。 |

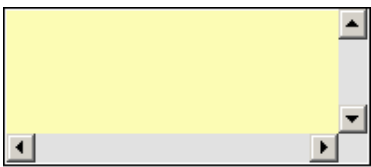



历史数值数据表属性说明

| | | | | | | | | | | | | | | | | | | |
|---------------------|---|---|------|-------------|--------|--------|---------------|---------------|-------------------|-------------------|---------------------|---------------------|--------|--------|-----------|-----------|--|-------------|
| 设定值 | 缓冲区编号 | 设定历史缓冲区编号 (No.1~No.12) 读取控制器资料对应的地址，您可以按下【历史缓冲区设定】按钮或是于菜单里的【选项】中【历史缓冲区设定】菜单设定控制器资料对应的地址。 | | | | | | | | | | | | | | | | |
| | 时间/日期显示 | | | | | | | | | | | | | | | | | |
| | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | | | | | | | | | | | | | | | | |
| | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。 | | | | | | | | | | | | | | | | |
| | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 | | | | | | | | | | | | | | | | |
| | 数值单位 | 提供 16bits Word、与 32bits Double Word。 | | | | | | | | | | | | | | | | |
| | 数值格式 | 选择不同的数值单位会有不同的数值格式可供选择如下表 <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Word</td> <td style="width: 50%;">Double Word</td> </tr> <tr> <td>1. BCD</td> <td>1. BCD</td> </tr> <tr> <td>2. Signed BCD</td> <td>2. Signed BCD</td> </tr> <tr> <td>3. Signed Decimal</td> <td>3. Signed Decimal</td> </tr> <tr> <td>4. Unsigned Decimal</td> <td>4. Unsigned Decimal</td> </tr> <tr> <td>5. Hex</td> <td>5. Hex</td> </tr> <tr> <td>6. Binary</td> <td>6. Binary</td> </tr> <tr> <td></td> <td>7. Floating</td> </tr> </table> | Word | Double Word | 1. BCD | 1. BCD | 2. Signed BCD | 2. Signed BCD | 3. Signed Decimal | 3. Signed Decimal | 4. Unsigned Decimal | 4. Unsigned Decimal | 5. Hex | 5. Hex | 6. Binary | 6. Binary | | 7. Floating |
| | Word | Double Word | | | | | | | | | | | | | | | | |
| 1. BCD | 1. BCD | | | | | | | | | | | | | | | | | |
| 2. Signed BCD | 2. Signed BCD | | | | | | | | | | | | | | | | | |
| 3. Signed Decimal | 3. Signed Decimal | | | | | | | | | | | | | | | | | |
| 4. Unsigned Decimal | 4. Unsigned Decimal | | | | | | | | | | | | | | | | | |
| 5. Hex | 5. Hex | | | | | | | | | | | | | | | | | |
| 6. Binary | 6. Binary | | | | | | | | | | | | | | | | | |
| | 7. Floating | | | | | | | | | | | | | | | | | |
| 资料位置 | 每次触发时，所要读取的 Word 资料位置。例如我们在历史缓冲区设定里取样的数值单位设 3 Words，所以在 | | | | | | | | | | | | | | | | | |

| | | |
|---|--------------|--|
| | | <p>资料位置的属性我们就可以选 0~2 个位置。 请注意当您选择数值单位为 Double Word 且数值格式为 Floating 时，另外在历史缓冲区设定里设定取样的数值单位为奇数个 Words 时，在选择资料位置上要特别小心，Floating 的资料位置必为偶数位。</p> |
| | 显示颜色 | 设定资料栏位上的数据颜色。 |
| | 整数位数 小数位数 | 决定读取的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示样式，只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。 |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.12.3 历史信息表

| 历史信息表属性说明 | | |
|---|--|---------------------|
| 从历史缓冲区取样到的资料，根据设定将相对应的”文字信息”方式显示于人机屏幕上。 | | |
| 文字/文字大小 字体/文字颜色 | 用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。(参阅一般按钮说明) 每一个状态所输入的文字信息。人机会将此状态值下所输入的文字信息列表于历史信息表元件上。若无输入文字，则会显示”?”。 | |
| 外框颜色 元件背景颜色 | 下图为元件背景颜色为米黄色，外框颜色为黑色的历史信息表元件。  | |
| 数值单位 | Word | 历史信息表元件可以有 256 个状态。 |
| | LSB | 历史信息表元件可以有 16 个状态。 |
| 数值格式 | 当您选择 数值单位 为 Word 时此栏位属性才可以设定。历史信息表提供了 BCD、Signed Decimal、Unsigned Decimal、Hex 等四种数值格式来解释读取到的缓冲区所设定读取存储器的内容。 | |
| 新增删除状态数 | 设定历史信息表之状态总数。如果数值单位为 Word，则可以设定 1-256 个状态；LSB 就可以设定 16 个状态。 | |

| | | |
|-----------|--|---|
| 设定值 |  | |
| | 缓冲区编号 | 设定历史缓冲区编号 (No. 1~No. X) 读取控制器资料对应的地址，您可以按下【历史缓冲区设定】按钮或是于菜单里的【选项】中【历史缓冲区设定】菜单设定控制器资料对应的地址。 |
| | 资料位置 | 每次触发时，所要读取的 Word 资料位置。例如我们在历史缓冲区设定里取样的数值单位设 3 Words，所以在资料位置的属性我们就可以选 0~2 个位置。 |
| 历史信息表属性说明 | | |
| 设定值 | 时间/日期显示 | |
| | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 |
| | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。 |
| | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.13 警报显示

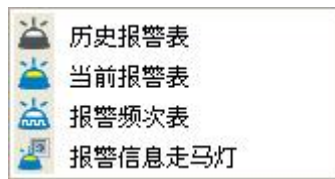



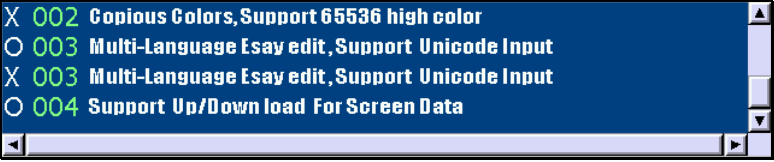


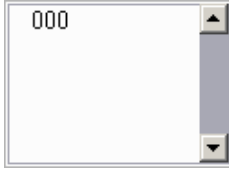


图 3-8-30 警报显示

3.8.13.1 历史报警表

| 历史报警表属性说明 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|----------------|------|--------------|------|------|---|---------|----------------|----|--------------|---|---------|----------------|----|------|---|---------|----------------|----|------|---|--|--------------|----|------|---|--|--------------|----|------|---|--|--------------|----|------|---|--|--------------|----|------|---|--|--------------|----|------|---|--|--------------|----|------|----|--|--------------|----|------|----|--|--------------|----|------|----|--|--------------|----|------|
| 人机会自动在固定的时间内，监看在警报设定里所设定的读取地址，依照所设定的警报属性，若设定的地址其中某一个 Bit 接点为 ON 或是 OFF 时，会将警报信息文字以历史报警表元件显示在人机屏幕上。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 元件背景颜色 | <p>下两张图为元件背景颜色分别被设为白色与被设为墨绿色时的情况。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>背景颜色为白色</p> </div> <div style="text-align: center;">  <p>背景颜色为墨绿色</p> </div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 设定值 | <div style="text-align: center;">  </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <p>时间格式 提供有 HH:MM:SS、HH:MM 两种显示格式。</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <p>日期格式 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 警报编号 | <p>若勾选，在警报发生时会在警报信息文字前加入在【警报设定】里所指定的警报编号，请参阅下图的设定与执行结果。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>编号</th> <th>信息内容</th> <th>文字颜色</th> <th>警报属性</th> <th>警报画面</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Alarm 1</td> <td>RGB(252. 0. 0)</td> <td>On</td> <td>1 - Screen_1</td> </tr> <tr> <td>2</td> <td>Alarm 2</td> <td>RGB(0. 0. 252)</td> <td>On</td> <td>None</td> </tr> <tr> <td>3</td> <td>Alarm 3</td> <td>RGB(0. 252. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>4</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>5</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>6</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>7</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>8</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>9</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>10</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>11</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> <tr> <td>12</td> <td></td> <td>RGB(0. 0. 0)</td> <td>On</td> <td>None</td> </tr> </tbody> </table> | 编号 | 信息内容 | 文字颜色 | 警报属性 | 警报画面 | 1 | Alarm 1 | RGB(252. 0. 0) | On | 1 - Screen_1 | 2 | Alarm 2 | RGB(0. 0. 252) | On | None | 3 | Alarm 3 | RGB(0. 252. 0) | On | None | 4 | | RGB(0. 0. 0) | On | None | 5 | | RGB(0. 0. 0) | On | None | 6 | | RGB(0. 0. 0) | On | None | 7 | | RGB(0. 0. 0) | On | None | 8 | | RGB(0. 0. 0) | On | None | 9 | | RGB(0. 0. 0) | On | None | 10 | | RGB(0. 0. 0) | On | None | 11 | | RGB(0. 0. 0) | On | None | 12 | | RGB(0. 0. 0) | On | None |
| 编号 | 信息内容 | 文字颜色 | 警报属性 | 警报画面 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Alarm 1 | RGB(252. 0. 0) | On | 1 - Screen_1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Alarm 2 | RGB(0. 0. 252) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Alarm 3 | RGB(0. 252. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | RGB(0. 0. 0) | On | None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |




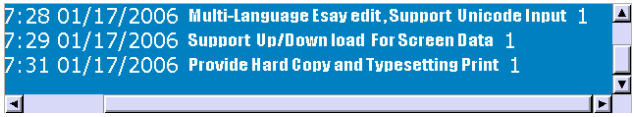
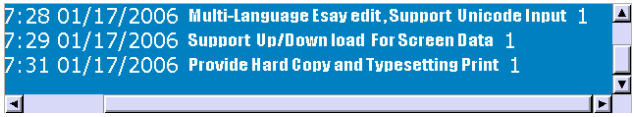
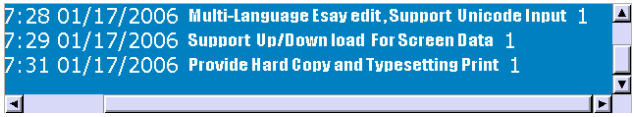
| 历史警报表属性说明 | | |
|-----------|------------|--|
| 设定值 | 警报编号 |  |
| | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.13.2 当前警报表




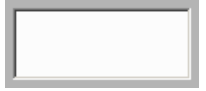



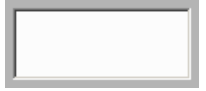



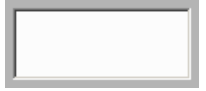



| 当前警报表属性说明 | | | | | | | | | |
|------------------------------|--|--|----------------------------|------|--|------|---------------------------------|------|--------------------------|
| 将目前正在发生的警报，以当前警报表元件显示在人机屏幕上。 | | | | | | | | | |
| 元件背景颜色 | <p>下两张图为元件背景颜色分别被设为白色与被设为黄色时的情况。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>背景颜色为白色</p> </div> <div style="text-align: center;">  <p>背景颜色为黄色</p> </div> </div> | | | | | | | | |
| | 设定值 |  | | | | | | | |
| | <table border="1"> <tr> <td>时间格式</td> <td>提供有 HH:MM:SS、HH:MM 两种显示格式。</td> </tr> <tr> <td>日期格式</td> <td>提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。</td> </tr> <tr> <td>警报编号</td> <td>当前警报表元件警报编号是永远需要显示的。请参阅下图的执行结果。</td> </tr> <tr> <td>显示颜色</td> <td>当您勾选显示时间与日期时，您可以指定显示的颜色。</td> </tr> </table> | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。 | 警报编号 | 当前警报表元件警报编号是永远需要显示的。请参阅下图的执行结果。 | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 |
| 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | | | | | | | | |
| 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY.MM.DD、YY/MM/DD、MM.DD、MM/DD 七种显示格式。 | | | | | | | | |
| 警报编号 | 当前警报表元件警报编号是永远需要显示的。请参阅下图的执行结果。 | | | | | | | | |
| 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | |

| 当前警报表属性说明 | |
|-----------|------------|
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.13.3 警报频次表

| 警报频次表属性说明 | | | | | | | | | | | |
|---|---|---|----------------------------|------|--|------|---|-----------|---|------|--------------------------|
| <p>人机会自动在固定的时间内，监看在警报设定里所设定的读取地址，依照所设定的警报属性，若设定的地址其中某一个 Bit 接点为 ON 或是 OFF 时，会记录该接点的发生次数，并将结果以警报频次表元件显示于人机屏幕上。</p> | | | | | | | | | | | |
| 元件背景颜色 | <p>下两张图为元件背景颜色分别被设为白色与被设为黄色时的情况。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>背景颜色为白色</p> </div> <div style="text-align: center;">  <p>背景颜色为黄色</p> </div> </div> | | | | | | | | | | |
| 设定值 |  | | | | | | | | | | |
| 设定值 | <table border="1"> <tr> <td>时间格式</td> <td>提供有 HH:MM:SS、HH:MM 两种显示格式。</td> </tr> <tr> <td>日期格式</td> <td>提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY 三种显示格式。</td> </tr> <tr> <td>警报编号</td> <td> 请参考历史警报表元件警报编号属性的设定。下图为执行结果。  </td> </tr> <tr> <td>计数为零时是否显示</td> <td>当统计警报监视的各点其警报发生的累计次数为 0 时是否要显示其信息于警报频次表元件上。</td> </tr> <tr> <td>显示颜色</td> <td>当您勾选显示时间与日期时，您可以指定显示的颜色。</td> </tr> </table> | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY 三种显示格式。 | 警报编号 | 请参考历史警报表元件 警报编号 属性的设定。下图为执行结果。  | 计数为零时是否显示 | 当统计警报监视的各点其警报发生的累计次数为 0 时是否要显示其信息于警报频次表元件上。 | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 |
| | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | | | | | | | | | |
| | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY 三种显示格式。 | | | | | | | | | |
| | 警报编号 | 请参考历史警报表元件 警报编号 属性的设定。下图为执行结果。  | | | | | | | | | |
| | 计数为零时是否显示 | 当统计警报监视的各点其警报发生的累计次数为 0 时是否要显示其信息于警报频次表元件上。 | | | | | | | | | |
| 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 | | | | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | | | |
| Y | 元件左上角 Y 座标 | | | | | | | | | | |
| 宽 | 元件宽度 | | | | | | | | | | |
| 高 | 元件高度 | | | | | | | | | | |

3.8.13.4 警报信息走马灯

| 警报信息走马灯属性说明 | | | | | | | | | |
|--|---|---|---|--------|-------------|---|---|--|---|
| 人机会自动在固定的时间内，监看在警报设定里所设定的读取地址，依照所设定的警报属性，若设定的地址其中某一个 Bit 接点为 ON 或是 OFF 时，将警报信息文字以走马灯步进的方式显示于人机屏幕上。 | | | | | | | | | |
| 外框颜色 元件背景颜色 | <p>下图为元件背景颜色为青绿色，外框颜色为红色的警报信息走马灯元件。</p>  | | | | | | | | |
| 元件型式 | <table border="1"> <tr> <td>Standard</td> <td>Raised</td> <td>Sunken</td> <td>Transparent</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> | Standard | Raised | Sunken | Transparent |  |  |  |  |
| Standard | Raised | Sunken | Transparent | | | | | | |
|  |  |  |  | | | | | | |
| 每次移动点数 | 文字移动的量，单位为 Pixel，范围从 1~50 Pixels。 | | | | | | | | |
| 间隔时间 (毫秒) | 文字移动与下次文字移动时的间隔时间，单位为 ms，范围从 50~3000 ms。 | | | | | | | | |
| 设定值 |  | | | | | | | | |
| | 时间格式 | 提供有 HH:MM:SS、HH:MM 两种显示格式。 | | | | | | | |
| | 日期格式 | 提供有 MM/DD/YY、DD/MM/YY、DD.MM.YY 三种显示格式。 | | | | | | | |
| | 警报编号 | 请参考历史警报表元件 警报编号 属性的设定。下图为执行结果。  | | | | | | | |
| | 显示颜色 | 当您勾选显示时间与日期时，您可以指定显示的颜色。 | | | | | | | |
| X | 元件左上角 X 座标 | | | | | | | | |
| Y | 元件左上角 Y 座标 | | | | | | | | |
| 宽 | 元件宽度 | | | | | | | | |
| 高 | 元件高度 | | | | | | | | |

3.8.14 键盘



图 3-8-31 键盘

键盘(1)/键盘(2)/键盘(3)属性说明

预设三种键盘，提供用户十进制，十六进制及文数字的输入。

| | | | |
|-----|---|---|-------|
| 1 | 2 | 3 | CLR |
| 4 | 5 | 6 | DEL |
| 7 | 8 | 9 | Enter |
| +/- | 0 | . | |

十进制输入键盘

| | | | | |
|---|---|---|---|-------|
| 0 | 1 | 2 | 3 | CLR |
| 4 | 5 | 6 | 7 | DEL |
| 8 | 9 | A | B | Enter |
| C | D | E | F | |

十六进制输入键盘


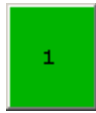

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | - | = | CLR |
| Q | W | E | R | T | Y | U | I | O | P | [|] | DEL |
| A | S | D | F | G | H | J | K | L | ; | , | ` | Enter |
| Z | X | C | V | B | N | M | , | . | / | \ | | |


文数字输入键盘

每种预设键盘都可以重新定义各个按钮的显示文字，另有<ESC>（跳出），<ENT>（输入），<CLR>（清除），（删字），<ASCII>（输入字元）等弹性化的设定，如下图：



当字符设定为“1”时，该按钮便会输入1的字码；设为“2”时，便输入2的字码；设为“A”时，输入为A的字码；设为“%”时，输入为%的字码；... 依此类推。键盘一开始建立在画面时是一个组合（Group），可以利用取消组合（Ungroup）的功能将键盘内的各个按钮打散，以便对按钮做个别移动或调整大小。

| 键盘(1)/键盘(2)/键盘(3)属性说明 | |
|---|--|
| <p>键盘必须搭配<u>数值输入</u>元件或<u>文数字输入</u>元件使用，其中在这两个元件上都必须在<u>启动输入方式</u>的属性栏位里选择<u>主动非弹跳式</u>或<u>触控非弹跳式</u>的属性，同时设定<u>生效位元</u>，有关其属性设定请参阅<u>数值/文数字输入</u>元件说明。（<u>主动非弹跳式</u>与<u>触控非弹跳式</u>此两者属性差别在于当生效位元被启动时设为<u>主动非弹跳式</u>的输入元件会直接以闪烁表示，而设为<u>触控非弹跳式</u>属性则要当用户触碰输入元件后才会闪烁。</p> | |
| <p>文字 文字大小 字体 文字颜色</p> | <p>用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。其中点选字体属性设定时会出现如下的对话框</p> <div style="text-align: center;">  </div> <p>在字体属性设定对话框里您不但可以设定上述的功能，还可以指定字体的缩放比例，经由<u>预览</u>可以看到字体的变化。请注意在上图里由于我们设定了多国语言的编辑，所以您可以用点选页签的方式分别作各国语言字体的设定。</p> |
| <p>图形库名称 图形名称</p> | (参阅一般按钮说明) |
| <p>图形背景是否 透明 指定图形透明色</p> | (参阅一般按钮说明) |
| <p>元件前景颜色 元件型式</p> | <p>元件型式仅分为 Standard 与 Raised 两种样式，底下分别配合元件型式的设定并修改前景颜色所得到的结果。</p> <div style="text-align: center;">  <p>(元件型式为 Standard, 前景颜色为绿色)</p>  <p>(元件型式为 Raised, 前景颜色为红色)</p> </div> |

| 键盘(1)/键盘(2)/键盘(3)属性说明 | |
|-----------------------|--|
| 设定值 | <p>您可以各别指定每个键盘按键的定义值</p>  |
| | <p>显示 键盘按键的显示用的符号。</p> <p>模式 分为下列几种模式：</p> <ol style="list-style-type: none"> 1. <ESC> 取消输入。(若将键盘放在子视窗上，此按键另外会执行关闭子视窗的动作。) 2. <ENT> 决定输入。 3. <CLR> 清除整列输入。 4. 清除单一字符输入。 5. <ASCII> 由用户所指定的按键字码。 |

3.8.15 绘图




图 3-8-32 绘图

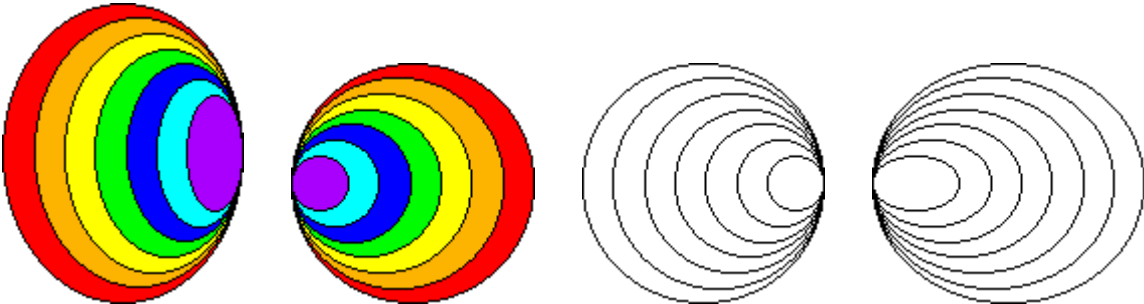
3.8.15.1 线

| 线条属性说明 | |
|--|----------------------|
| <p>按住鼠标的左键决定直线起点，拖曳鼠标到所需要的长度，放开左键便可建立一条直线。而当您点选此直线时，会出现一个矩形的范围，这是为了方便用户去移动图形，并且藉著调整此矩形的范围来再一次修正此线条，而线条的宽度跟颜色，用户可以自行更改。而直线以外的地方，则以透明效果处理。</p> | |
| <p>线条的方向 线条宽度由 1 到 8</p> | |
| 线条颜色 | 可设定线条的显示颜色。 |
| 线条宽度 | 可设 1~8 Pixels 的线宽。 |
| 线条型式 | <p>提供以下的线条型式可供选择</p> |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.15.2 矩形

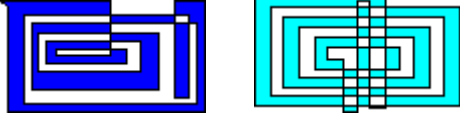
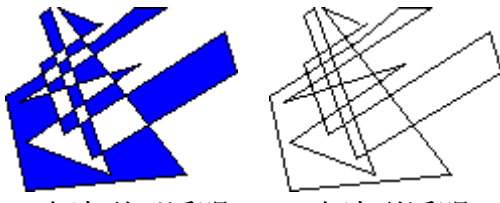
| 矩形属性说明 | |
|--|--|
| 按住鼠标的左键然后拖曳出一个范围，一个矩形就形成了。可从图形库中载入图形，并且决定矩形外框的线条长度以及前景的颜色，如此一张矩形图便形成了。 | |
| 线条颜色 | 可设定矩形的边线颜色。 |
| 图形库名称 图形名称 | (参阅一般按钮说明) |
| 图形背景是否透明 指定图形透明色 | (参阅一般按钮说明) |
| 元件前景颜色 | 可设定动态矩形的显示颜色。 |
| 线条宽度 | 可设 1~8 Pixels 的矩形边线宽。 |
| 透明色 | 仅以矩形边线显示，中间将镂空，元件前景颜色功能失效。 |
| 圆角半径 | 提供 0~38 pixels 圆角半径可供选择。  |
| X | 元件左上角 X 坐标 |
| Y | 元件左上角 Y 坐标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.15.3 圆形

| 圆形属性说明 | |
|--|--------------------------|
| 按住鼠标的左键然后拖曳出一个范围，一个圆形就形成了。如果是长等于宽，那么图形就会变成一个圆形。如果是不相等，则会变成一个椭圆形，图形的颜色可以改变。同样的，修改矩形的大小，就可以改变图形的形状及大小，而图形以外矩形以外的其他部分，在人机端将会以透明效果处理。而属性中有一项属性是“透明色”，一旦选择 YES 图形将只会剩下框框而已，如果底下有其他元件就会显示出来。 | |
|  | |
| 线条颜色 | 可设定(椭)圆形的边线颜色。 |
| 元件前景颜色 | 可设定(椭)圆形的显示颜色。 |
| 线条宽度 | 可设 1~8 Pixels 的(椭)圆形边线宽。 |

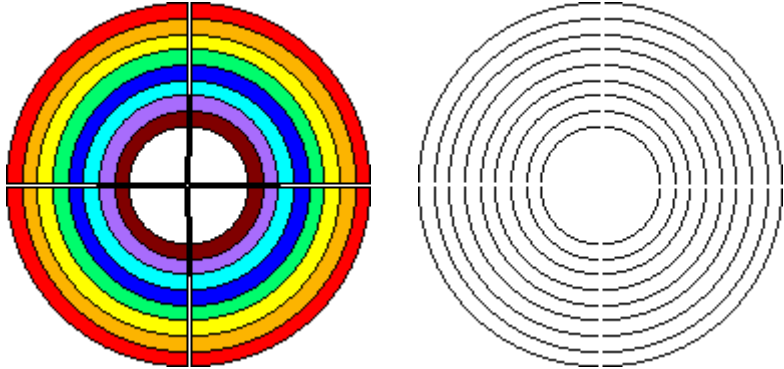
| 圆形属性说明 | |
|--------|--------------------------------|
| 透明色 | 仅以(椭)圆边线显示, 中间将镂空, 元件前景颜色功能失效。 |

3.8.15.4 多边形


| 多边形属性说明 | |
|--|---|
| 利用按鼠标左键的方式, 来决定多边形的每一个点。当所有的点都设好之后, 按鼠标右键, 软件便能自动帮您组成一个多边形, 而且此图形可让用户决定颜色。同样的, 修改整个矩形的大小, 可以改变等比例缩放多边形大小, 而图形以外矩形以外的其他部分, 在人机端将会以透明图处理。而属性中有一项属性是“透明色”, 一旦选择 YES 图形将只会剩下框框而已, 如果底下有其他元件就会显示出来。 | |
| 线条颜色 | 可设定多边形的边线颜色。 |
| 元件前景颜色 | 可设定多边形的显示颜色。如下图所示: <div style="text-align: center;">  <p>前景颜色为蓝色 前景颜色为青绿色</p> </div> |
| 线条宽度 | 可设 1~8 Pixels 的多边形边线宽。 |
| 透明色 | 仅以多边形边线显示, 中间将镂空, 元件前景颜色功能失效。如下图所示: <div style="text-align: center;">  <p>多边形不透明 多边形透明</p> </div> |


3.8.15.5 弧

| 弧形属性说明 | |
|--|--|
| 当选择弧形之后, 第一步先按住鼠标左键, 拖曳出一个所需范围, 放开鼠标左键时的位置, 一个弧形就形成了。如果属性表里面透明色的选项为 Yes, 那么图形为弧形; 反之如果是 No, 图形就是扇形。同样的, 修改矩形的大小, 可以改变等比例缩放扇形的形状及圆弧的大小, 而图形以外矩形以外的其他部分, 在人机端将会以透明图处理。 | |

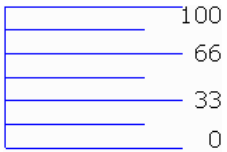
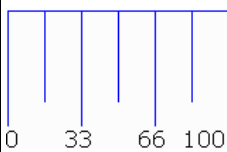
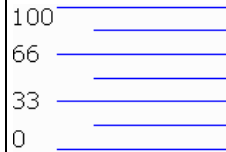
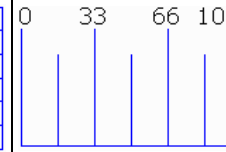
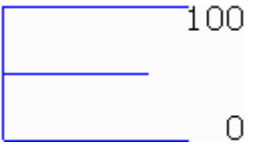
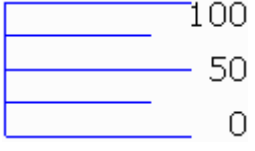
| 弧形属性说明 | |
|--|----------------------------|
|  | |
| 线条颜色 | 可设定弧形的边线颜色。 |
| 元件前景颜色 | 可设定弧形的显示颜色。 |
| 线条宽度 | 可设 1~8 Pixels 的弧形边线宽。 |
| 透明色 | 仅以弧形边线显示，中间将镂空，元件前景颜色功能失效。 |

3.8.15.6 静态文字

| 静态文字属性说明 | |
|---|--|
| <p>在画面上规划出一个矩形，可以在里面输入您所需要的文字，而元件前景颜色的设定，会使得整个矩形变成所设定的颜色。</p> | |
| <p>文字 文字大小 字体 文字颜色</p> | <p>用户可依 Windows®所提供之文字大小、字体与颜色功能，设定该元件文字显示型态。其中点选字体属性设定时会出现如下的框</p>  <p>在字体属性设定对话框里您不但可以设定上述的功能，还可以指定字体的缩放比例，经由<u>预览</u>可以看到字体的变化。请注意在上图里由于我们设定了多国语言的编辑，所以您可以用点选页签的方式分别作各国语系字体的设定。</p> |
| 元件前景颜色 | 可设定静态文字的显示颜色，下图中我们把元件前景颜色设为 |

| | |
|-----|---|
| | 蓝色。  |
| 透明色 | 仅描绘出文字的外貌，元件前景颜色功能失效。 |
| X | 元件左上角 X 座标 |
| Y | 元件左上角 Y 座标 |
| 宽 | 元件宽度 |
| 高 | 元件高度 |

3.8.15.7 刻度

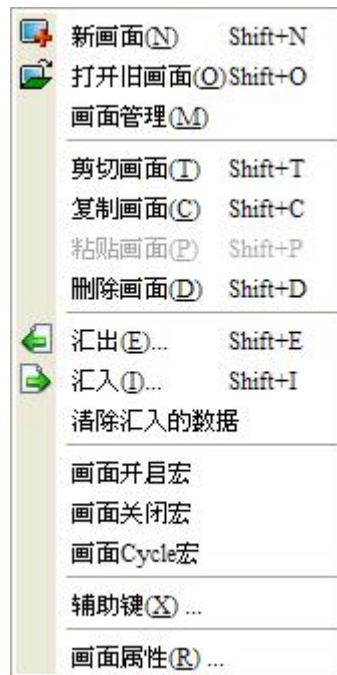
| 刻度属性说明 | | | | |
|--|---|---|--|---|
| 可以利用元件型式的选项，来改变刻度的方向或是属性主次刻度数目，来改变主次刻度的数目，并利用颜色的改变来创造独特的刻度。而显示标记可以选择要不要显示主刻度所代表的数值大小，数值的最大最小值是从设定值里面设定的。 | | | | |
| 文字大小 文字颜色 | 设定显示文数字的大小与颜色，这里所使用的字体是采用人机系统里的预设字体，人机系统提供了 8~64 的系统字体大小可供选择。 | | | |
| 元件型式 | Standard | Rotation 90 | Rotation 180 | Rotation 270 |
| |  |  |  |  |
| 主刻度数目 | 请参照下图：   主刻度数目为 2 时 主刻度数目为 3 时 | | | |

| 刻度属性说明 | | |
|--------|---|--|
| 设定值 | | |
| | 数值单位 | 提供 16bits Word、与 32bits Double Word。 |
| | 数值格式 | 提供以下的数值格式可供选择： Word/Double Word 1. BCD 2. Signed Decimal 3. Unsigned Decimal |
| | 最小值 最大值 | 用户可设定输入值的最小与最大值，限制输入数值的范围。 |
| | 整数位数 小数位数 | 决定输入的整数位数与小数位数各有几位。这里的小数位数并不是真的小数值，只是显示样式，只有在您数值格式选择 Floating 时小数位数的设定才是真正的小数。 |
| 设定值 | 您输入的最小值与最大值在按下确定按钮后程序会参照您选择的数值单位、数值格式、整数位数与小数位数作数值范围的检查。 | |
| 次刻度数目 | 主刻度数目为 3 时次刻度数目的变化请参照下图： <p style="text-align: center;">次刻度数目为 1 时 次刻度数目为 2 时</p> | |
| X | 元件左上角 X 坐标 | |
| Y | 元件左上角 Y 坐标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.8.15.8 表格

| 表格属性说明 | | |
|---|------------------|--------------------------|
| 可以利用元件型式的选项，来改变表格格子数的大小、外观以及格子的颜色也可以更改。如果搭配其他元件使用，那麼每个元件将会整齐的呈现在用户面前。 | | |
| 元件背景颜色 | 可设定表格每一个栏位的显示颜色。 | |
| 设定值 | | |
| | 列表头 | 设定第一列（列表头）的颜色，并决定是否使用。 |
| | 行表头 | 设定第一行（行表头）的颜色，并决定是否使用。 |
| | 列交织 | 设定列交织的颜色，并决定是否使用。 |
| | 行交织 | 设定行交织的颜色，并决定是否使用。 |
| | 列表头交织 | 设定列交织的颜色，由列表头开始，并决定是否使用。 |
| | 行表头交织 | 设定行交织的颜色，由行表头开始，并决定是否使用。 |
| | 等列高间距 | 每列的间距相等。 |
| | 等行高间距 | 每行的间距相等。 |
| 外框颜色 | 设定表格外框边线颜色。 | |
| 格子颜色 | 设定每个栏位与栏位间的线条颜色。 | |
| 列数 | 设定表格列数范围从 1~99。 | |
| 行数 | 设定表格行数范围从 1~99。 | |
| X | 元件左上角 X 座标 | |
| Y | 元件左上角 Y 座标 | |
| 宽 | 元件宽度 | |
| 高 | 元件高度 | |

3.9 功能菜单—画面



3.9.1 新画面



新增一编辑画面，画面名称由用户输入决定或采用系统内定。可以点选画面里的选项或是按下工具栏中的图标，也可使用系统内定的热键 Shift + N 之后，会出现新增画面对话框（图 3-9-1）。



图 3-9-1 新增画面对话框范例

3.9.2 打开旧画面

开启被隐藏的画面视窗，选择画面里面的开启旧画面选项或是按下工具栏中  的图标，也可使用系统内定的热键 Shift + O。为了方便用户选取画面，在选取画面时，可以从画面预览中知道其画面的编辑图案（图 3-9-2）。

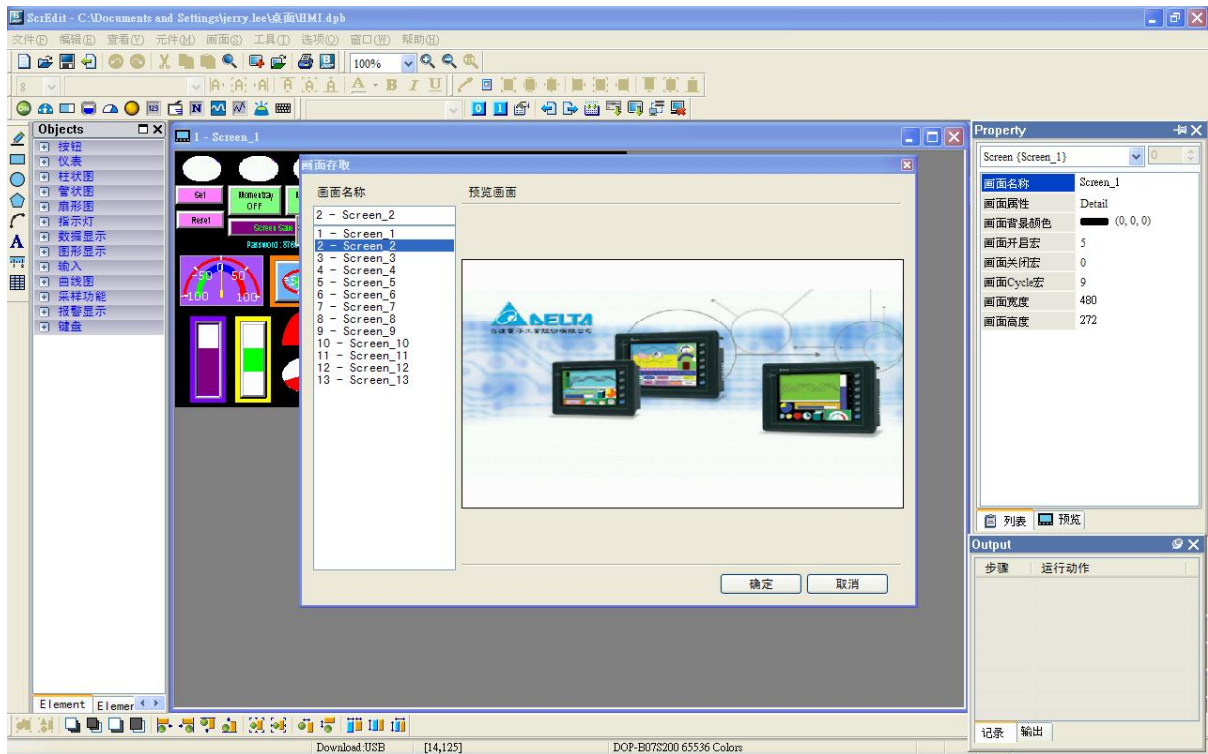


图 3-9-2 开启旧画面对话框

3.9.3 画面管理

启动画面管理系统（图 3-9-3），您可以在该系统下以 Windows®【档案总管】的操作方式以鼠标圈选的方式复制画面、贴上画面、剪切画面。您也可以在此系统下按下鼠标右键以点取选项的方式来管理编辑所有的画面（图 3-9-4）。但是需要注意的是”剪切”功能，剪下画面如同删除画面一样，无法使用复原恢复。不同的是，剪下画面可用贴上的方式得到相同的画面（图 3-9-5）；另外在画面管理中也提供有【启动屏幕保护编辑】的选项，点选进入后，您可以用托曳的方式决定屏幕保护启动时所展示的画面（图 3-9-6）。关于如何让人机启动屏幕保护请参考【设置模块参数】中的【其它】页签里将有完整的描述。

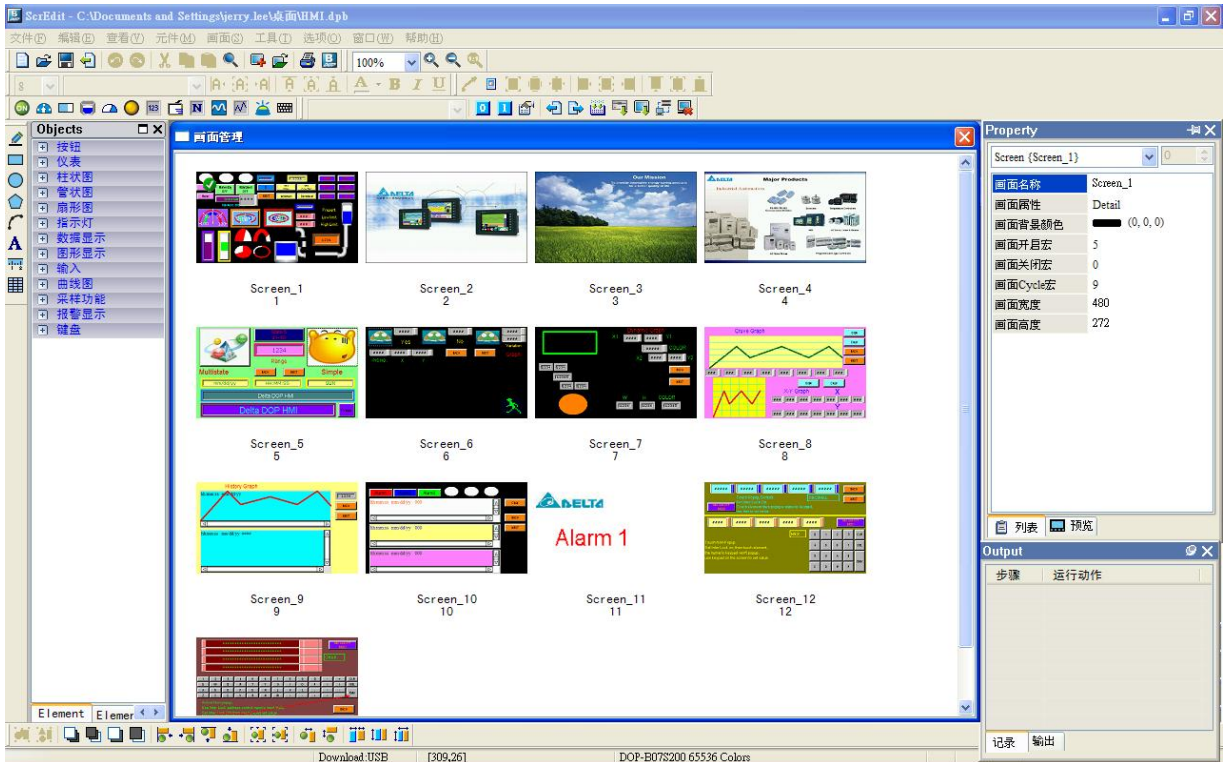


图 3-9-3 画面管理系统

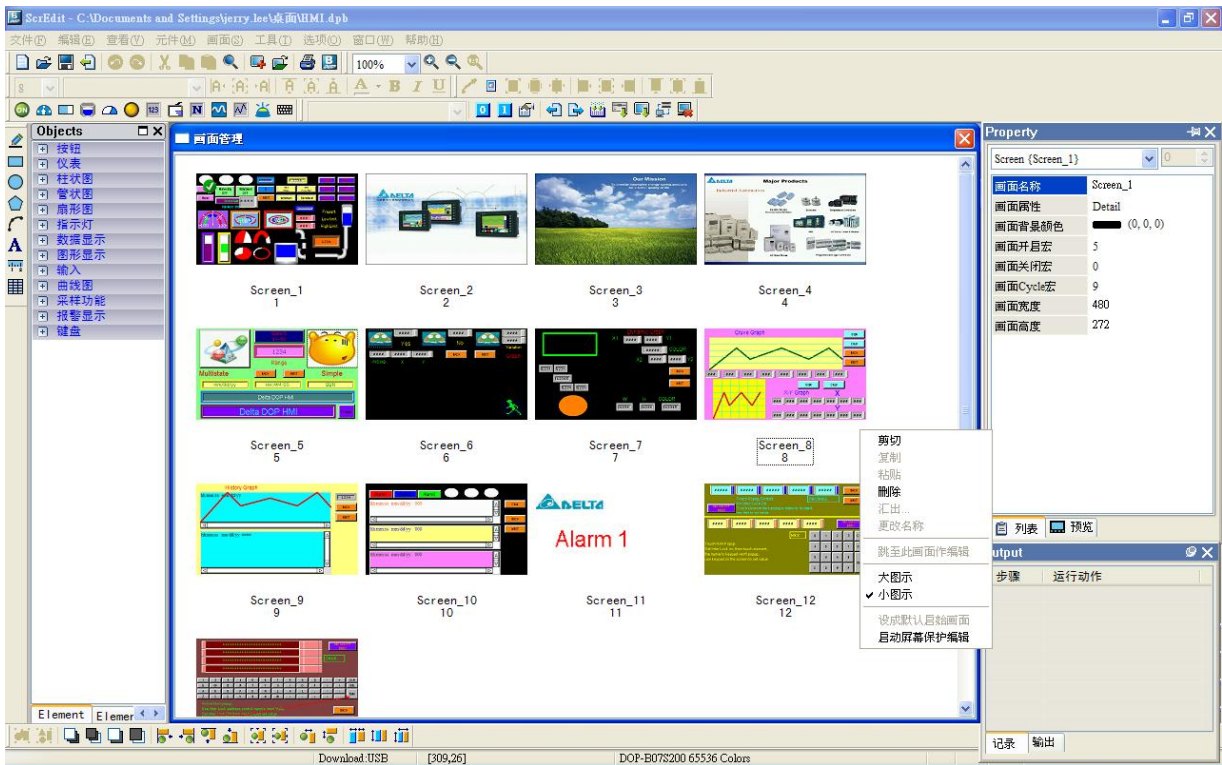


图 3-9-4 以菜单选取的方式管理画面

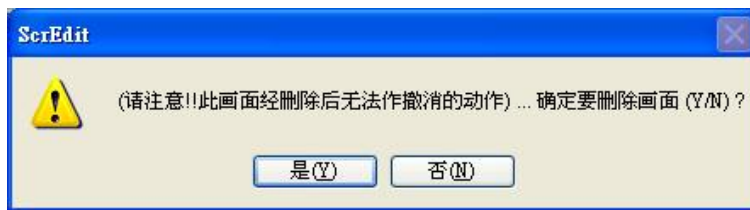


图 3-9-5 剪切画面时的信息

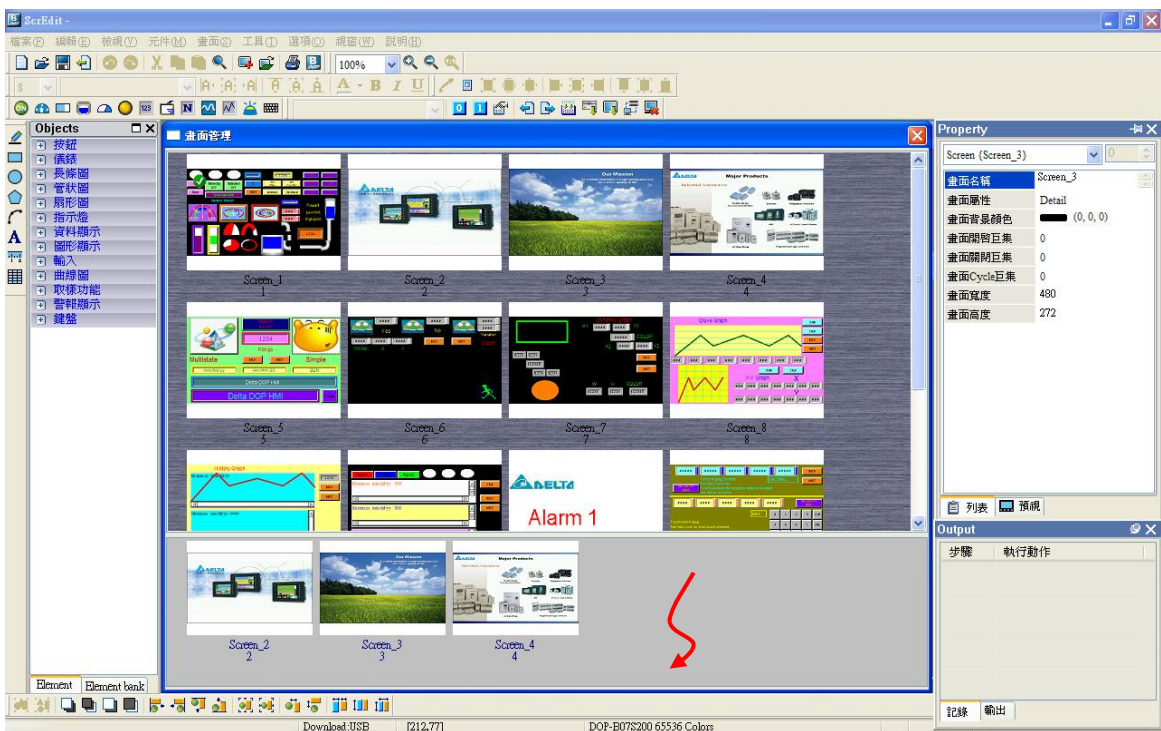


图 3-9-6 以托曳的方式决定屏幕保护画面

3.9.4 剪切画面

将整个画面剪下。如果此时再点选画面贴上的话，就会将原本的画面贴上，类似文字编辑的剪下。（注意！剪下画面如同删除画面一样，无法使用复原恢复。不同的是，剪下画面可用贴上的方式得到相同的画面）；可以点选画面里面的剪下画面选项或是使用系统内定的热键。请参考（图 3-9-7）。



图 3-9-7 剪切画面时的信息

3.9.5 复制画面

复制整个画面。如果此时再点选画面贴上的话，就会将原本的画面贴上，类似文字编辑的复制。可以点选画面里面的复制画面选项或是使用系统内定的热键 Shift + C。

3.9.6 贴上画面

将剪下的画面或是复制的画面贴上。可以点选画面里面的贴上画面选项或是使用系统内定的热键 Shift + P。贴上后其画面各项设定都不改变，只是 Screen Editor 会自动指定编号。

3.9.7 删除画面

删除目前编辑的画面视窗及元件（注意：当您执行这项动作后，被删除掉的画面是无法回复的）。您可以选择点选画面选项里的删除画面或是使用系统内定的热键 Shift + D。

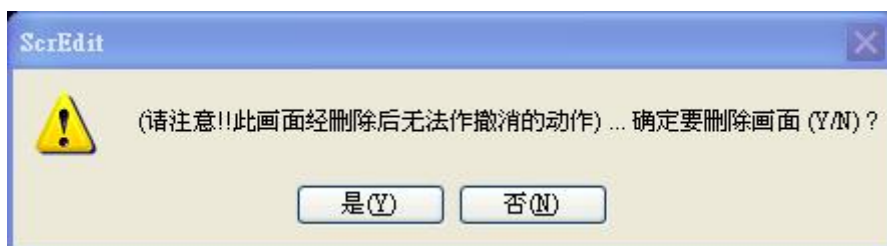



图 3-9-8 删除画面时所出现的信息

3.9.8 汇出

将目前编辑的画面资料以图形格式（bmp 档）储存至磁盘，可点取画面中的选项或是工具中的图标，或是使用系统所设定的热键 Shift + E。

3.9.9 汇入

从文件载入一张图形当作此编辑画面的底图，请注意这与基底画面意义不尽相同。所汇入的图形资料不具任何元件型式而存在，但基底画面会在编译后以元件的型式存于该编辑的画面中（基底画面的定义及使用方式会在”画面属性”中作详细的介绍）。所支持的汇入图形格式包含 BMP、JPG、GIF 等。可点取画面中的选项，或是使用系统所设定的热键 Shift + I。

3.9.10 清除汇入的数据

清除目前编辑画面中有使用到汇入外部图形时所占用的空间。您可以点取画面中的清除汇入的数据选项。

3.9.11 画面开启宏

该编辑画面开启时会自动被执行的宏程序。开启宏编辑视窗的方式除了可以由选项中直接选取，也可由画面右侧的属性表中点选（图 3-9-9）。

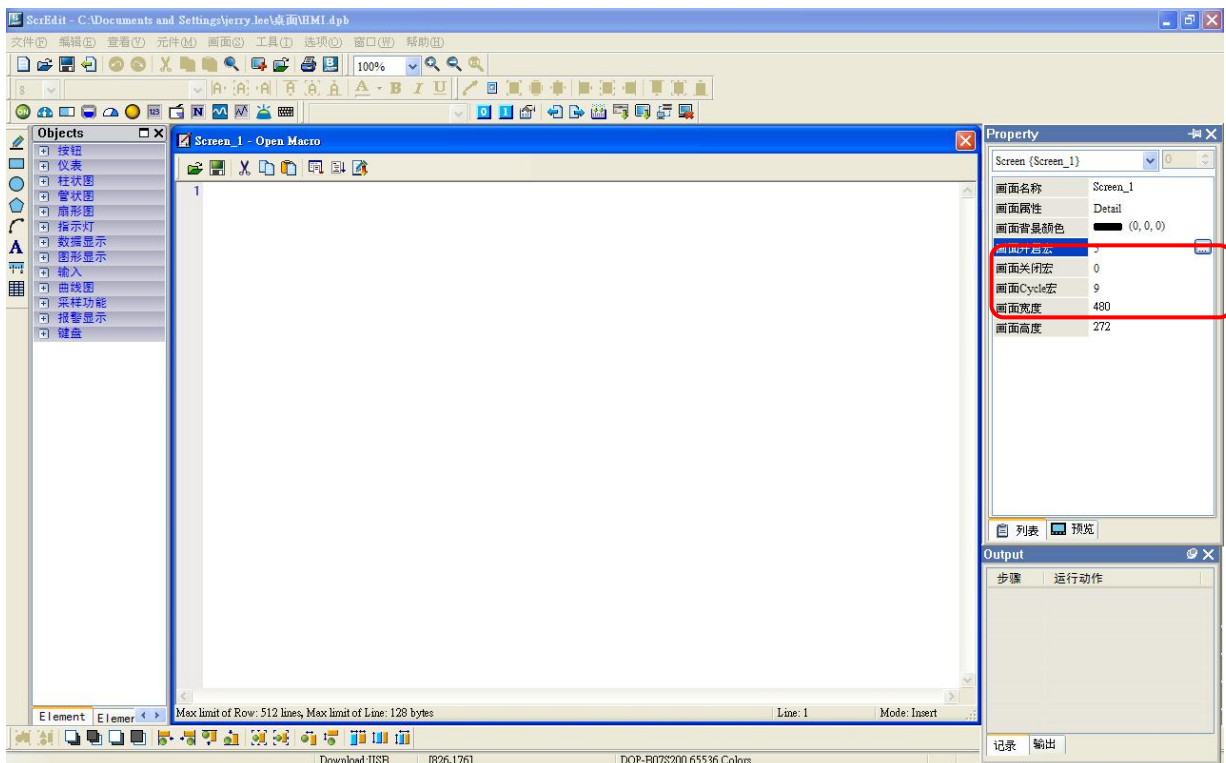


图 3-9-9 宏的编辑视窗

3.9.12 画面关闭宏

该编辑画面关闭时会自动被执行的宏程序。

3.9.13 画面 Cycle 宏

当该编辑画面显示时会不断地依 Cycle time 所设定的时间自动被执行的宏程序。

3.9.14 辅助键

DOP-B 系列提供每一个画面均能个别定义辅助键功能。辅助键定义如（图 3-9-10）所示。

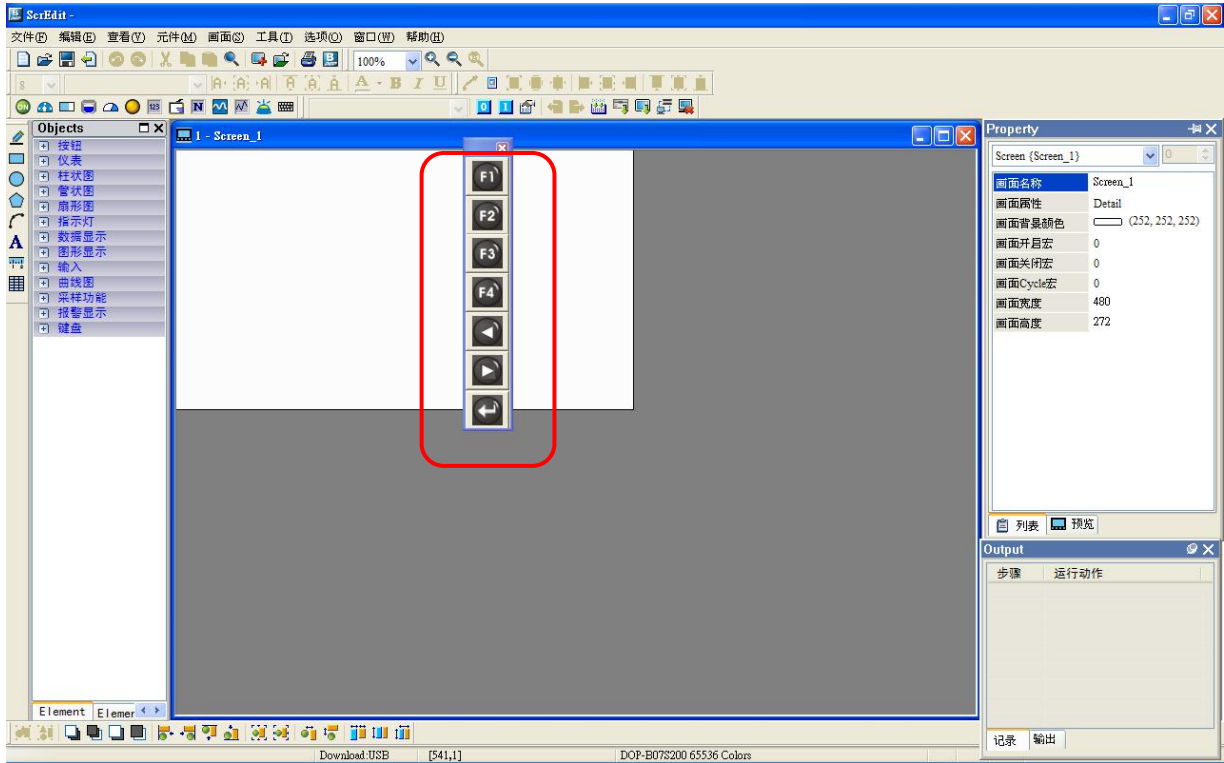


图 3-9-10 辅助键设定功能视窗

3.9.15 画面属性

此选项在设定此画面的属性。画面属性可直接由画面里面的选项选择或是点选画面后，再选择属性表中的画面属性（图 3-9-11）。至于画面属性的设定请参考（图 3-9-12）。画面属性功能繁杂，请参考以下叙述。

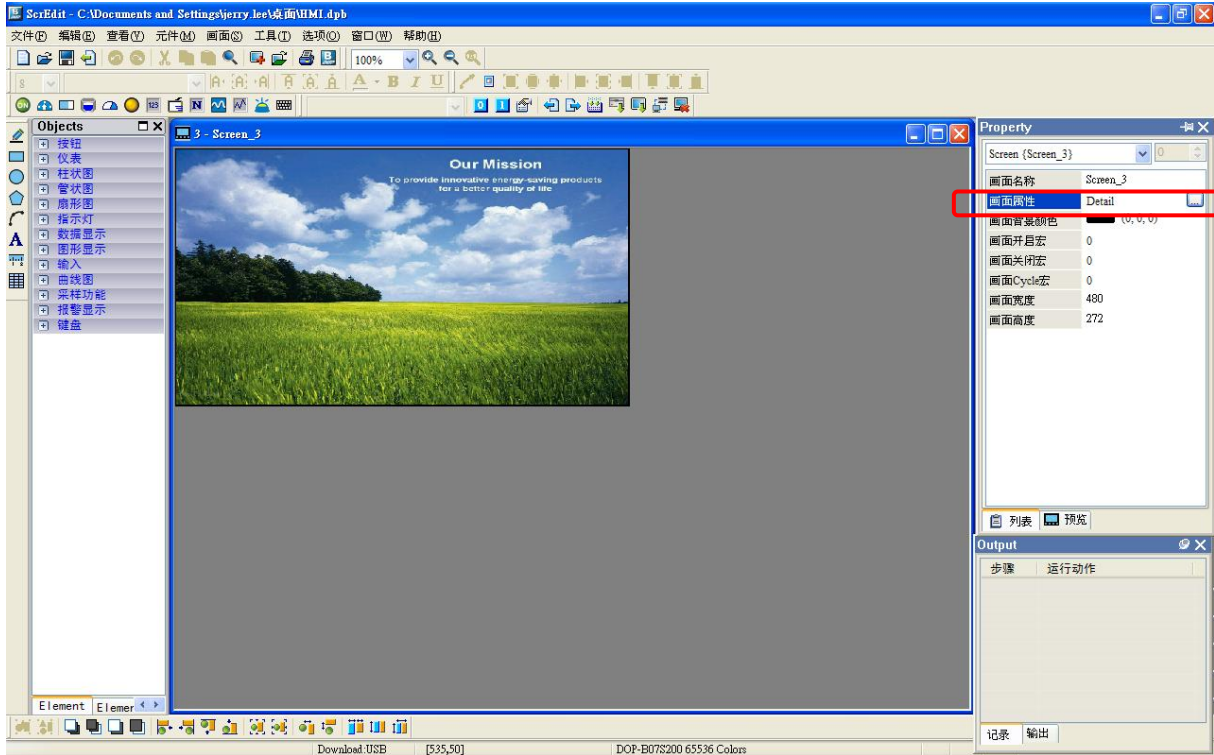


图 3-9-11 画面属性的点选

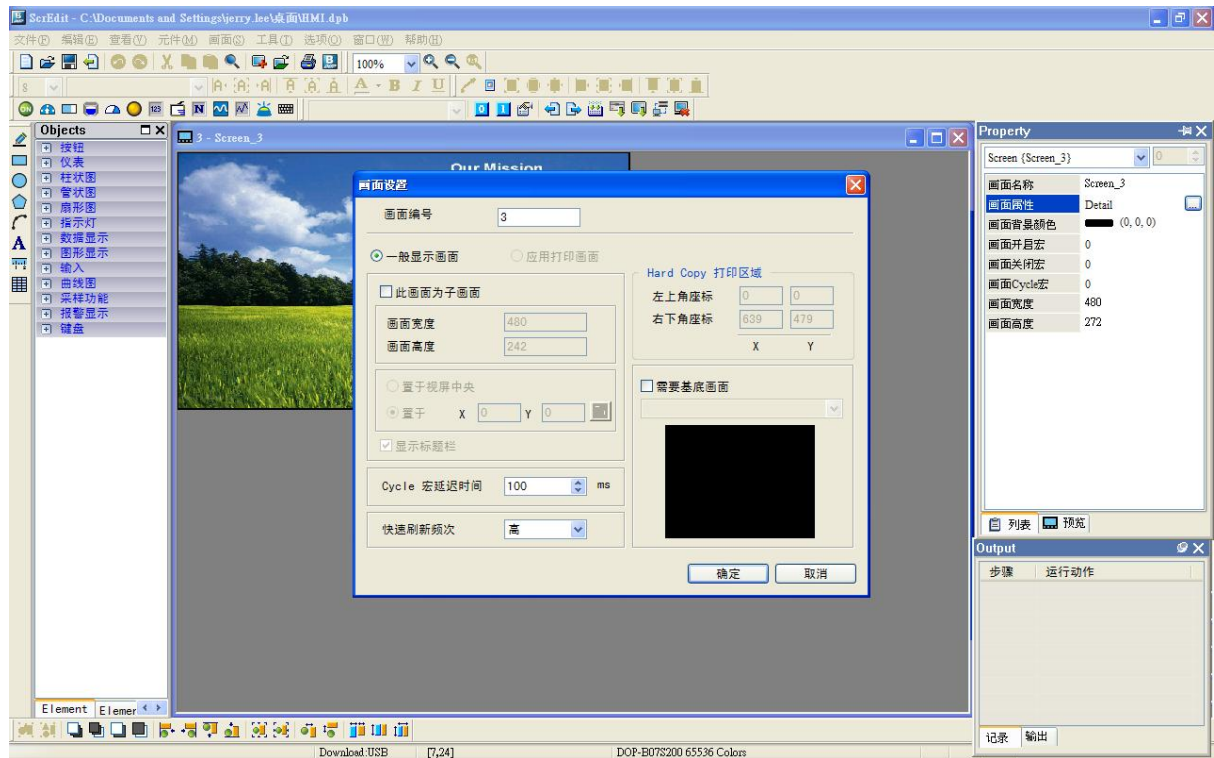
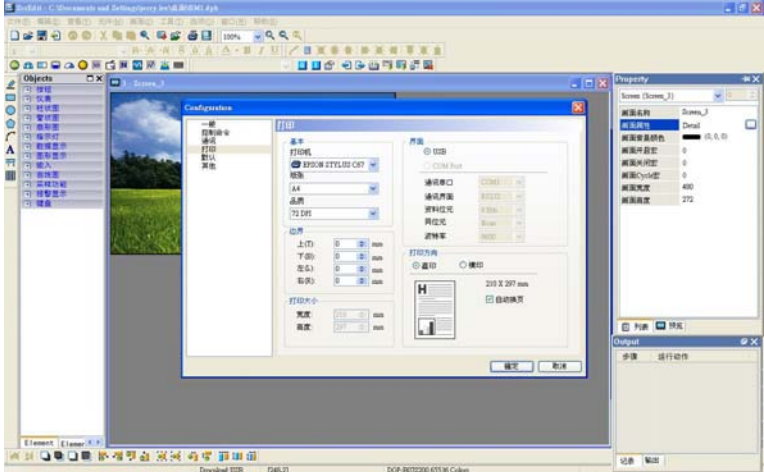
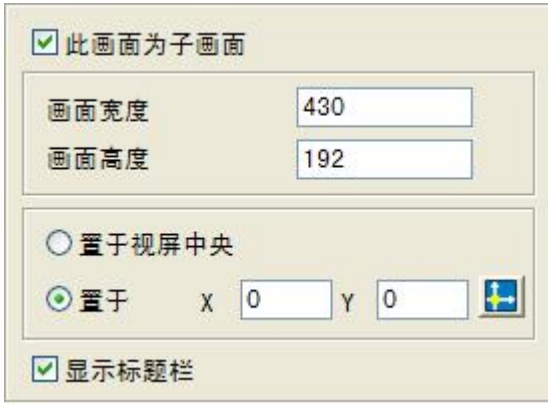


图 3-9-12 画面属性的设定

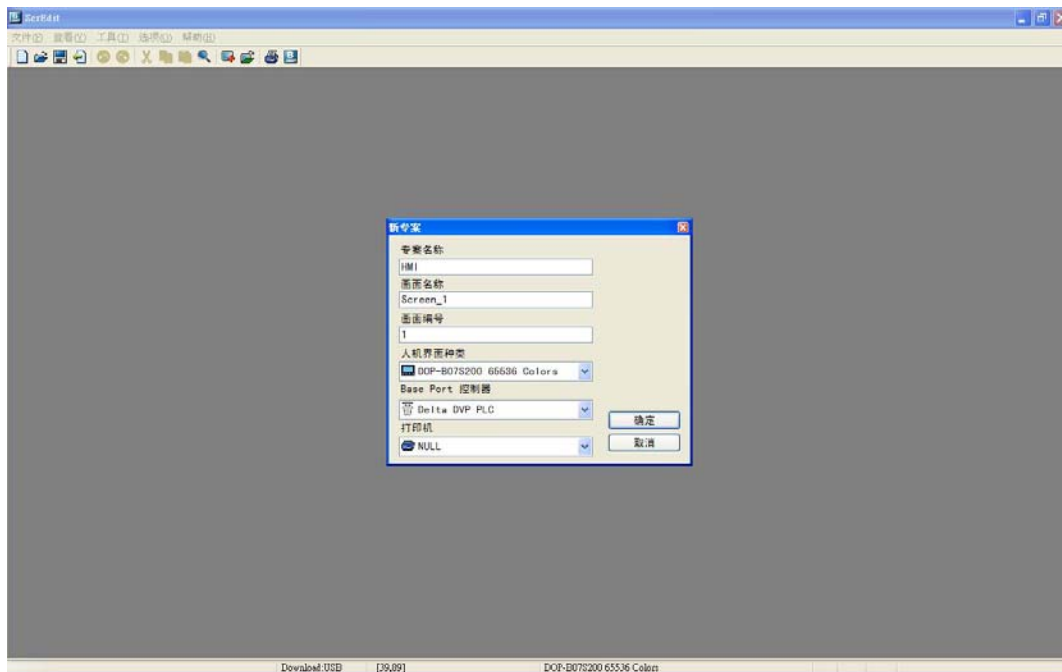
| 项目 | | 说明 |
|-------|--------|---|
| 画面编号 | | 画面 ID 值为 1~65535，不同画面的 ID 不可重覆。 |
| 画面应用 | 一般显示画面 | 作为一般的显示画面。 |
| | 套用打印画面 | <p>1. 将画面指定为打印画面。此选项只有在您选择有支持打印机的人机机型时方能使用。</p> <p>2. 被指定为打印画面属性的编辑画面，其编辑区的范围会被系统转换为输出到打印机纸张的真实大小，即所见即所得，通常套用打印画面是被用来作排版打印。</p> <p>上述两点您可以在【选项】的菜单里【设置模块参数】中的【打印】页面作设定，如下图所示：</p>  |
| 子画面设定 | 勾选 | <p>只有<u>一般显示画面</u>才能设定为子画面。</p>  |
| | 画面宽度 | 设定子画面宽度，单位为 Pixel。 |
| | 画面高度 | 设定子画面高度，单位为 Pixel。 |

| 项目 | | 说明 |
|-------------|---------|---|
| 子画面设定 | 子画面显示位置 | <p>您可以将子画面设定为开启时自动置于视屏中央，或者由您指定其开启时的位置。您可以直接输入座标值或是按下  用托曳的方式决定其位置，如下图所示：</p>  |
| | 显示标题栏 | 子画面被开启时是否要有标题栏。 |
| Cycle 宏延迟时间 | | 执行画面 Cycle 宏的间隔时间，范围为 100ms ~ 5s。 |
| 快速更新频率 | | 分为高、中、低三个等级。快速更新的主要用途是针对元件显示数值作即时的更新。 请注意，此功能是针对设定元件作较频繁的通讯动作，以达到快速更新数值的效果，所以一个画面中最多只能有四个元件能作即时更新的动作。若有过多元件开启此功能，会拖慢人机正常执行速率，所以如非特殊需要，建议不要开启此功能。 |
| HardCopy | 设定 | 此设定只有在您选择有支持打印机的人机机型时方能使用。在画面上规划输出报表按钮元件，将指定的输出装置设定为打印机，当用户按下输出报表按钮元件时，人机会依照 HardCopy 设定中的打印区域将人机屏幕上的画面输出到打印机作打印。 |
| | 左上角座标 | 设定要打印的区域大小范围，单位为 Pixel。 |
| | 右下角座标 | |
| 基底画面 | 勾选 | 每一个画面皆可以指定任何一个编辑画面作为基底画面，基底画面被设定后会以背景图的方式置于编辑区的最底层，人机会依照基底画面上的所有元件设定执行动作。 |

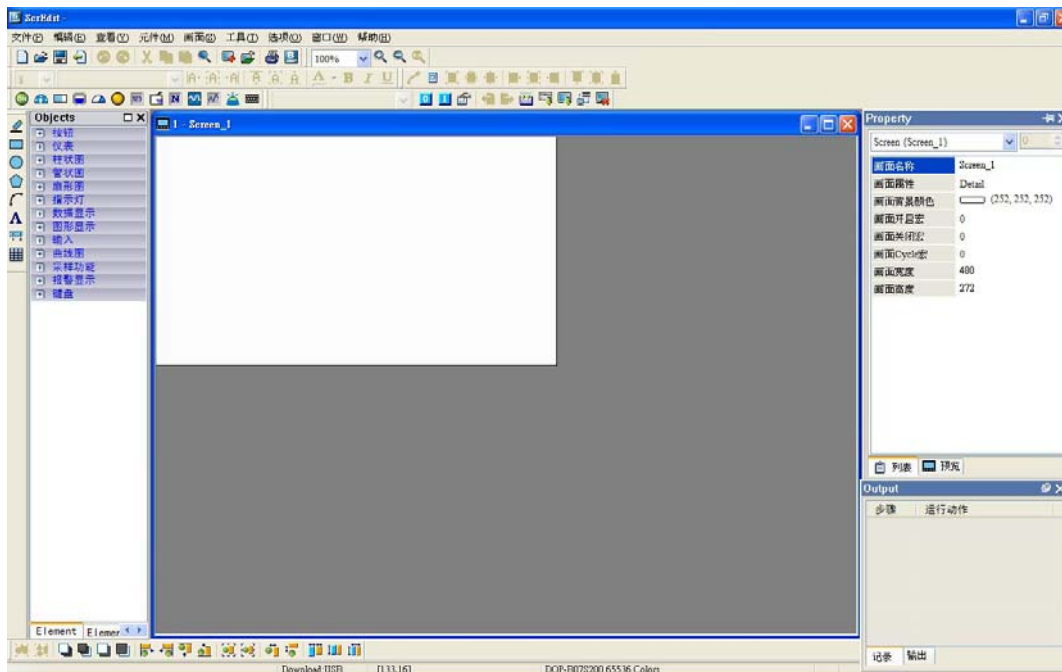
第四章 Screen Editor 使用范例说明

4.1 16 位配方使用范例

首先建立一新专案，如下图所示：

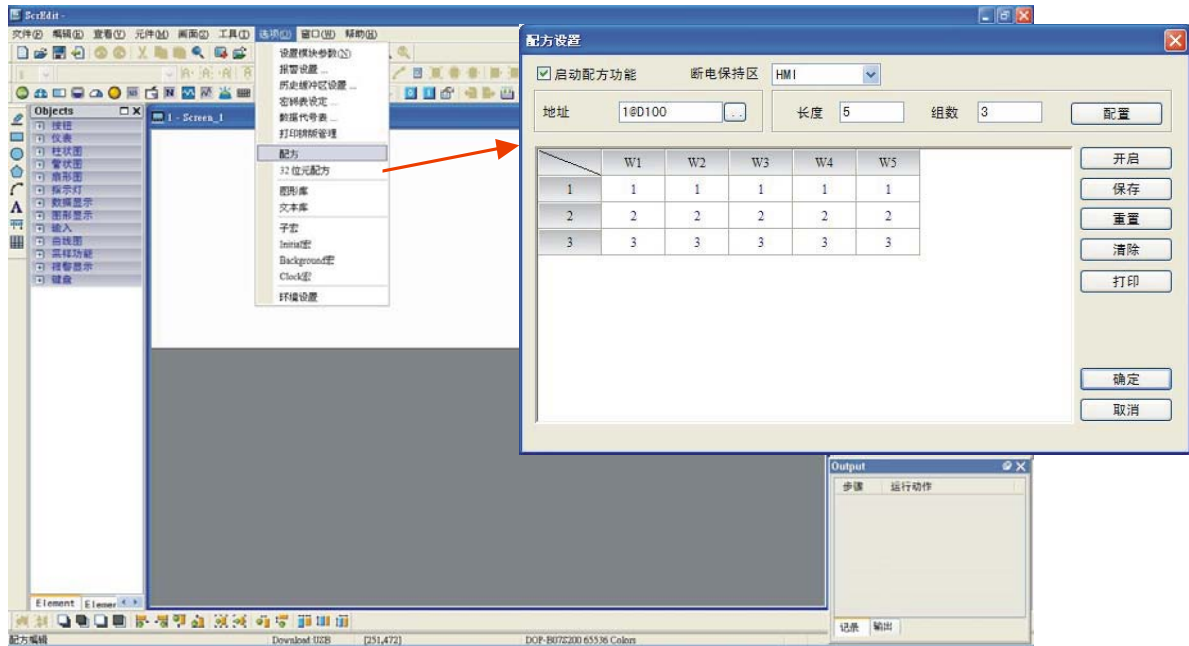


按下确定后，即进入到人机编辑画面，如下图所示：



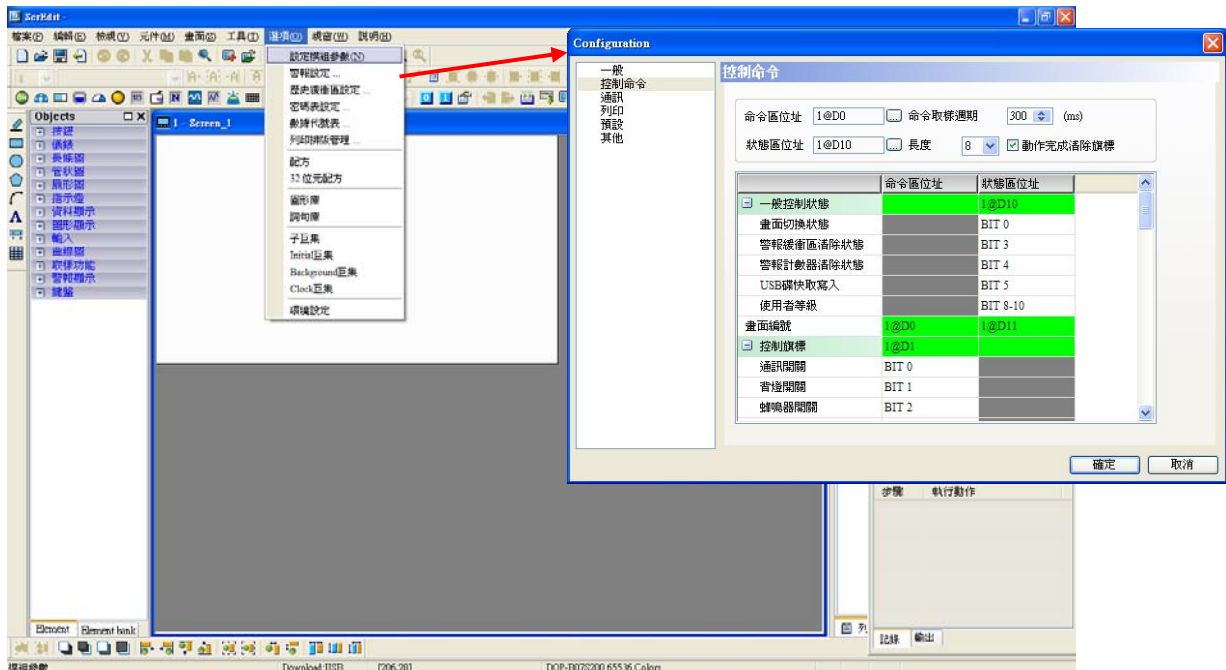
选择功能菜单中的选项/配方来编辑 16 位配方表，在此设定一长度乘上组数为 5×3 的配方，写入位置为 1@D100，断电保持区设在 HMI，按下配置后，配方表即建立完成，如下

图所示：



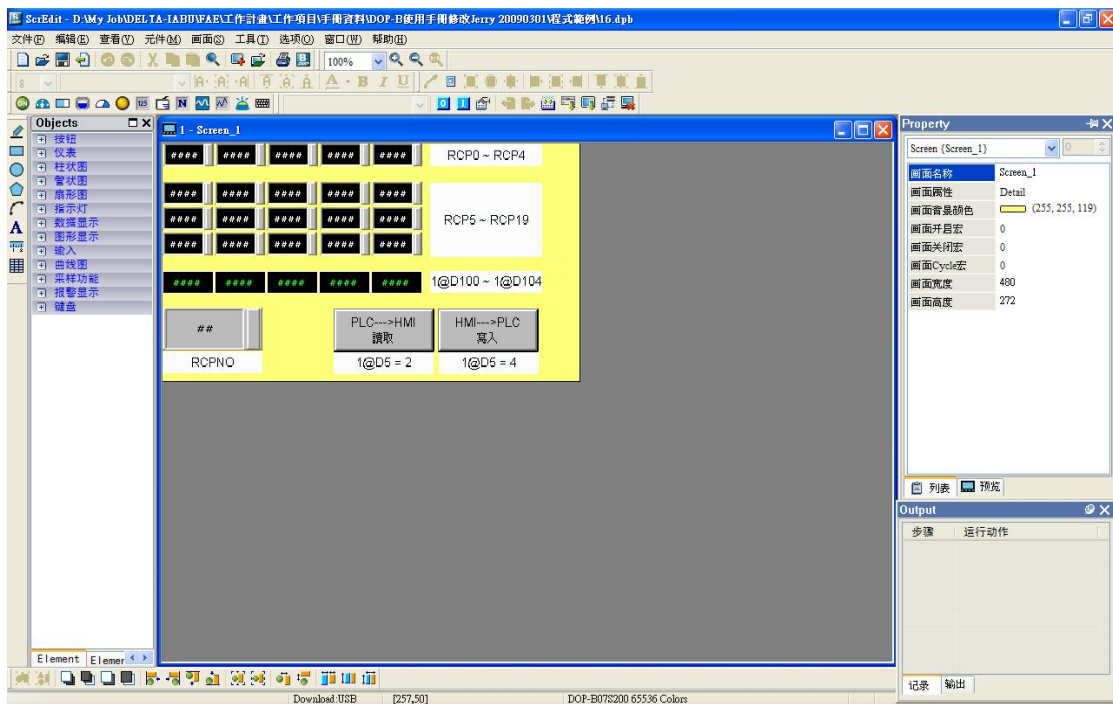
按下确定，完成配方设定。

选择功能菜单中的选项/设置模块参数的[控制命令]来设定控制区，以便控制配方的读取与写入，在此我们设定控制区的位址为 1@D0，长度为 8，如下图所示：



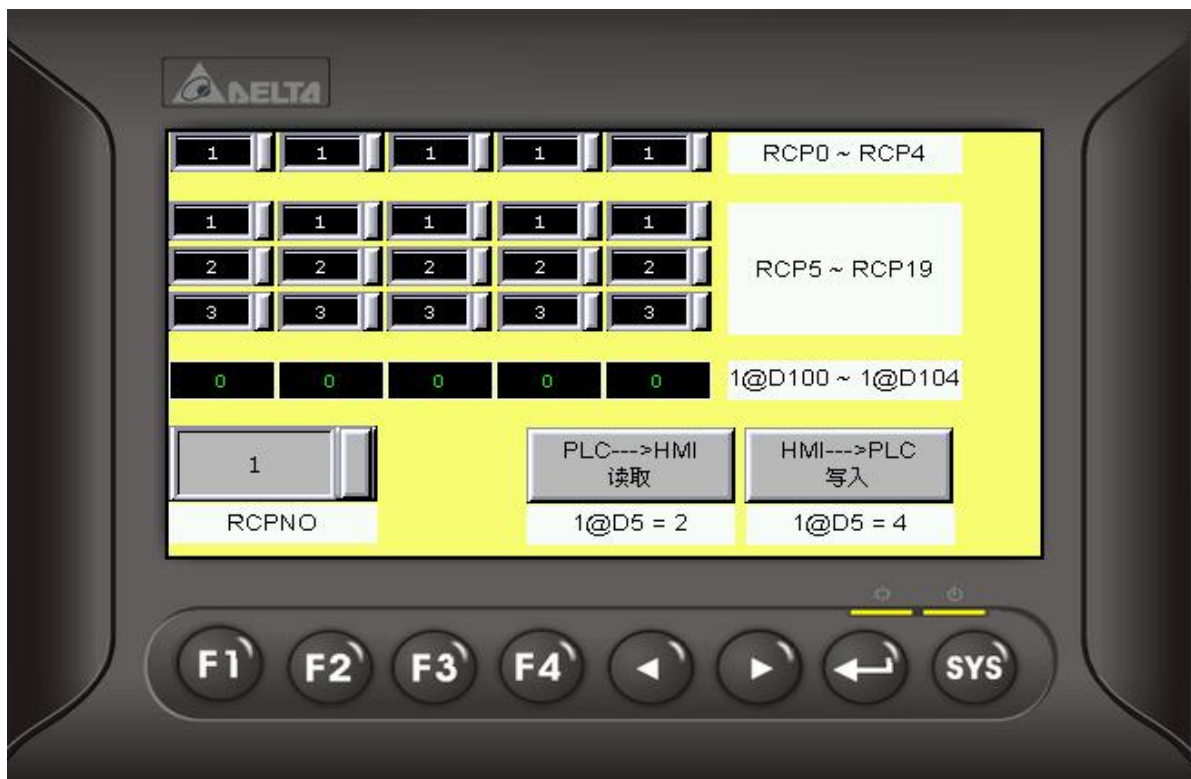
按下确定，完成控制区设定。

在人机编辑画面中，我们将配方缓冲区 (RCP0~RCP4)、配方表 (RCP5~RCP19) 及配方写入位址 (1@D100~1@D104) 以数值输入元件列出；另外也建立配方组别指定寄存器 (RCPNO) 的数值输入元件，以及配方读取/写入元件 (1@D5)，如下图所示：

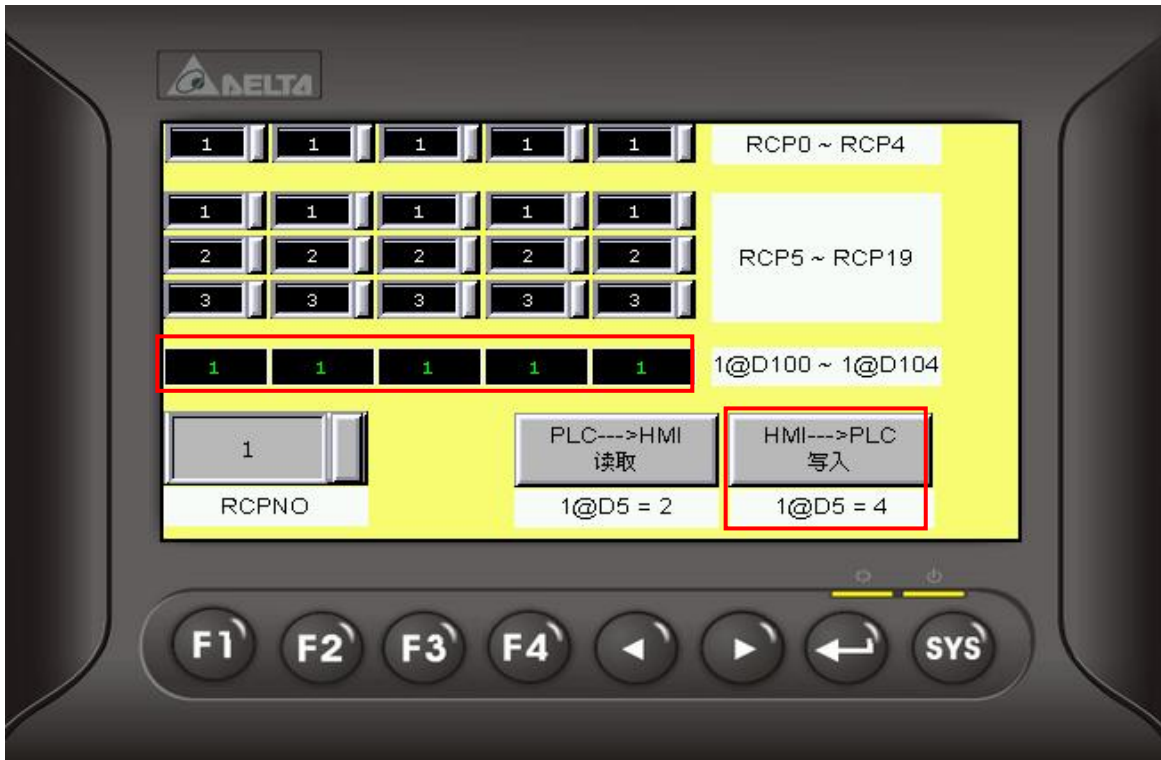


建立完成后进行编译动作，编译完成即可下载至人机进行操作。

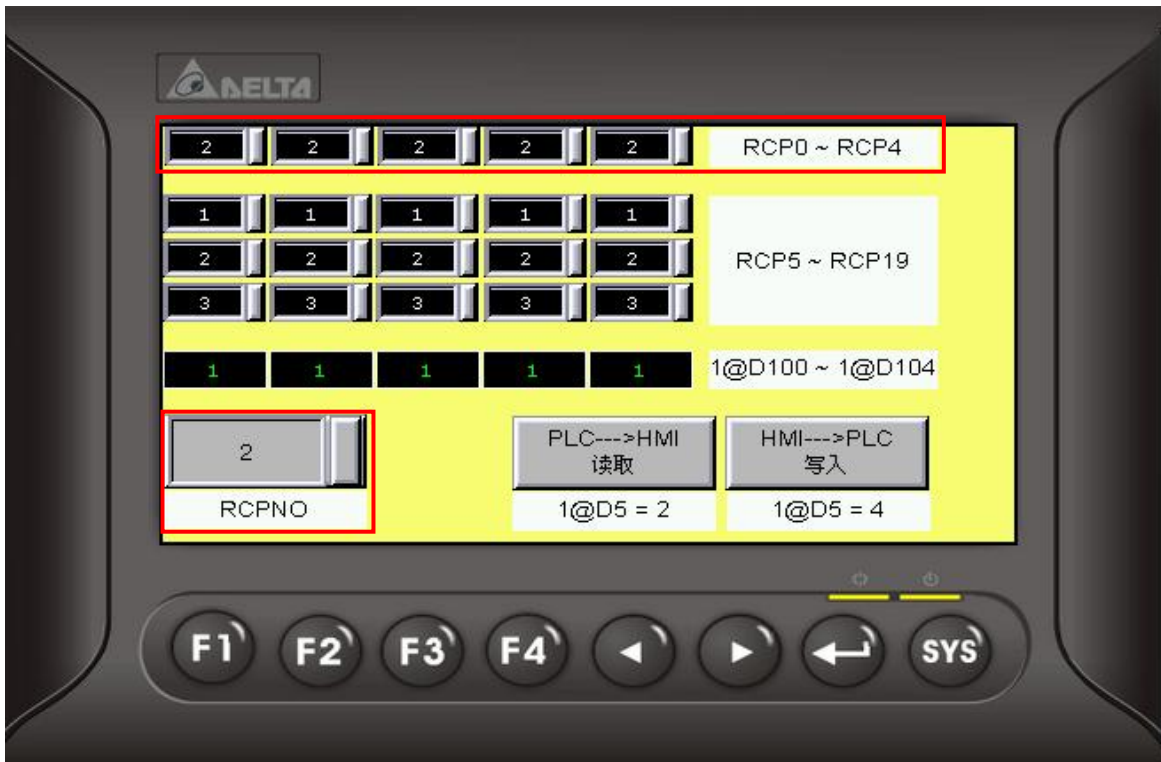
程式执行画面如下，进入程式画面后，RCPNO 默认为 1，因此缓冲区 RCP0~RCP4 会显示第一组配方，此时配方尚未写入 PLC 位址 1@D100~1@D104，因此显示数值为 0，如下图所示：



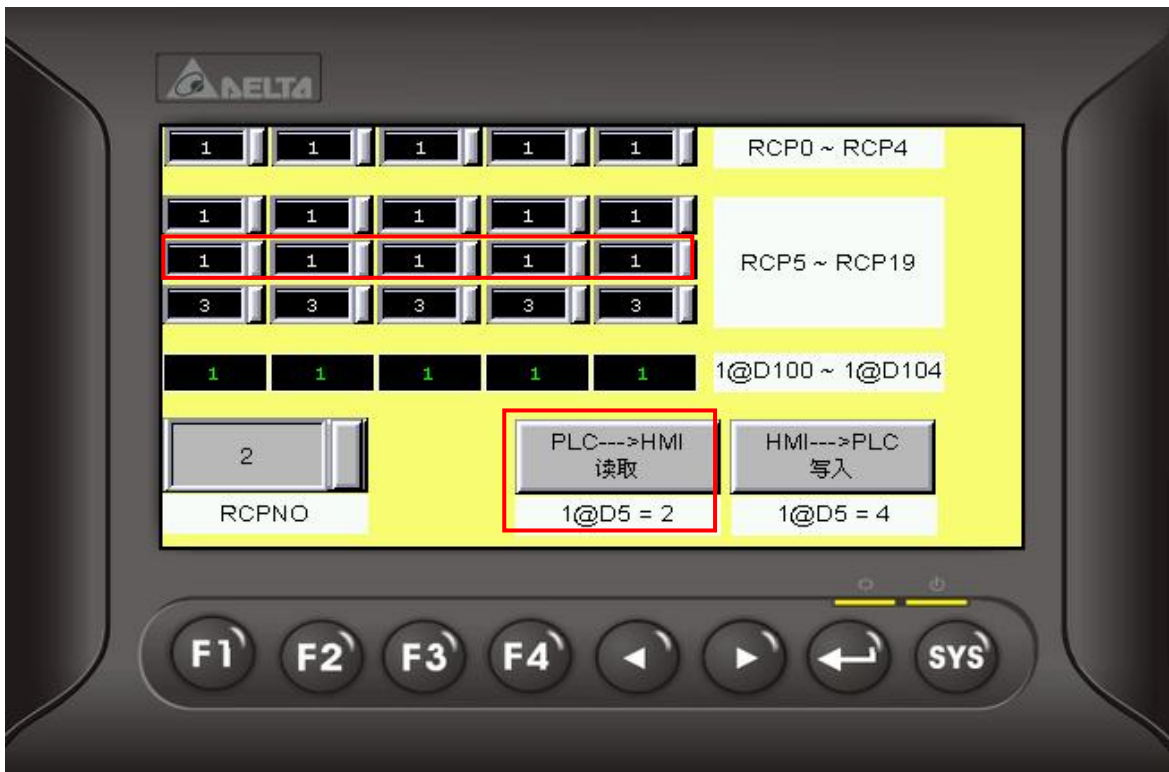
按下配方写入按钮，人机立即将第一组配方写入 PLC 位址 1@D100~1@D104，此时显示数值为 1，如下图所示：



选择第二组配方，即 RCPNO = 2，则 RCP0~RCP4 即显示第二组配方，如下图所示：



此时若按下配方读取按钮，则人机立即将 PLC 内的配方读回并存入第二组配方中，如下图所示：

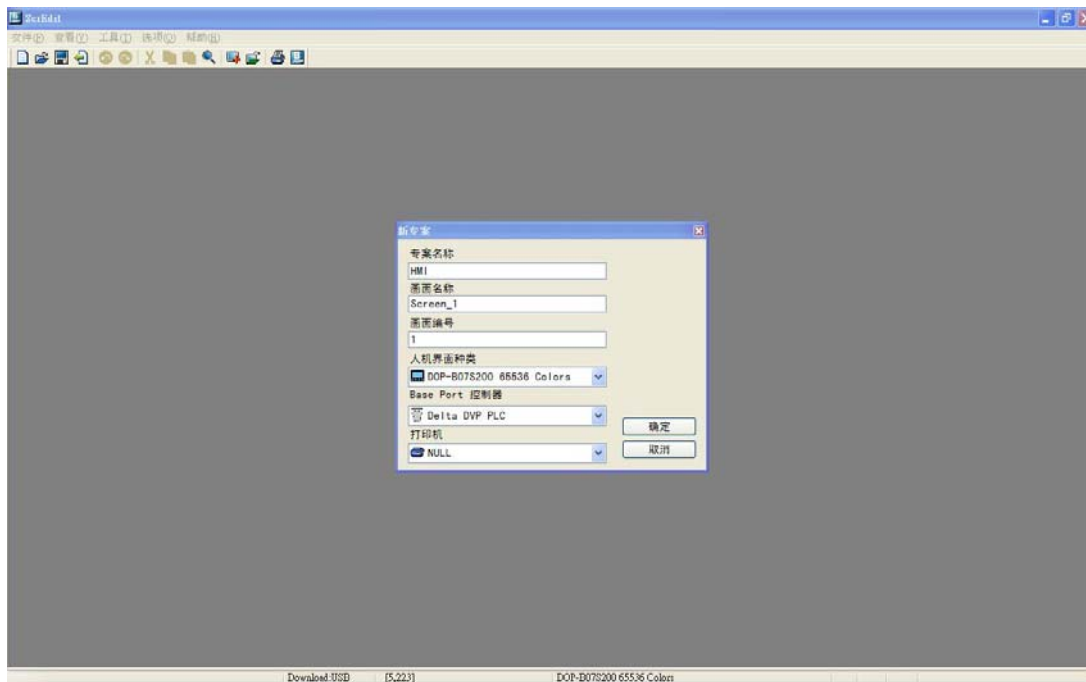


此时人机第二组配方已被更改。

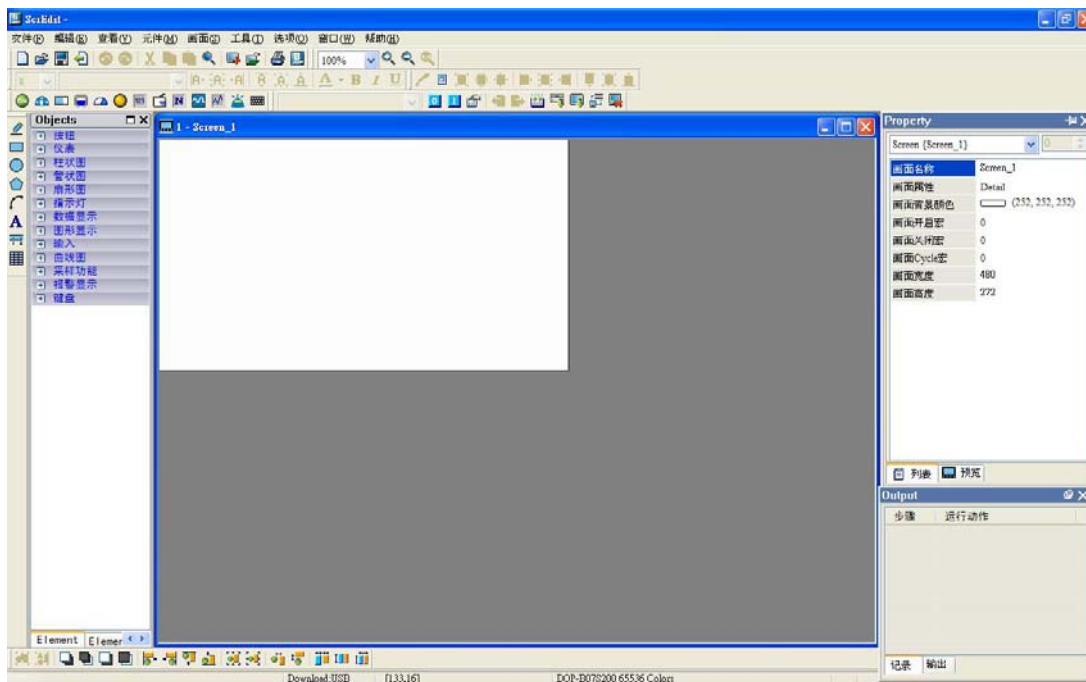
以上即为 16 位配方使用范例。

4.2 32 位配方使用范例

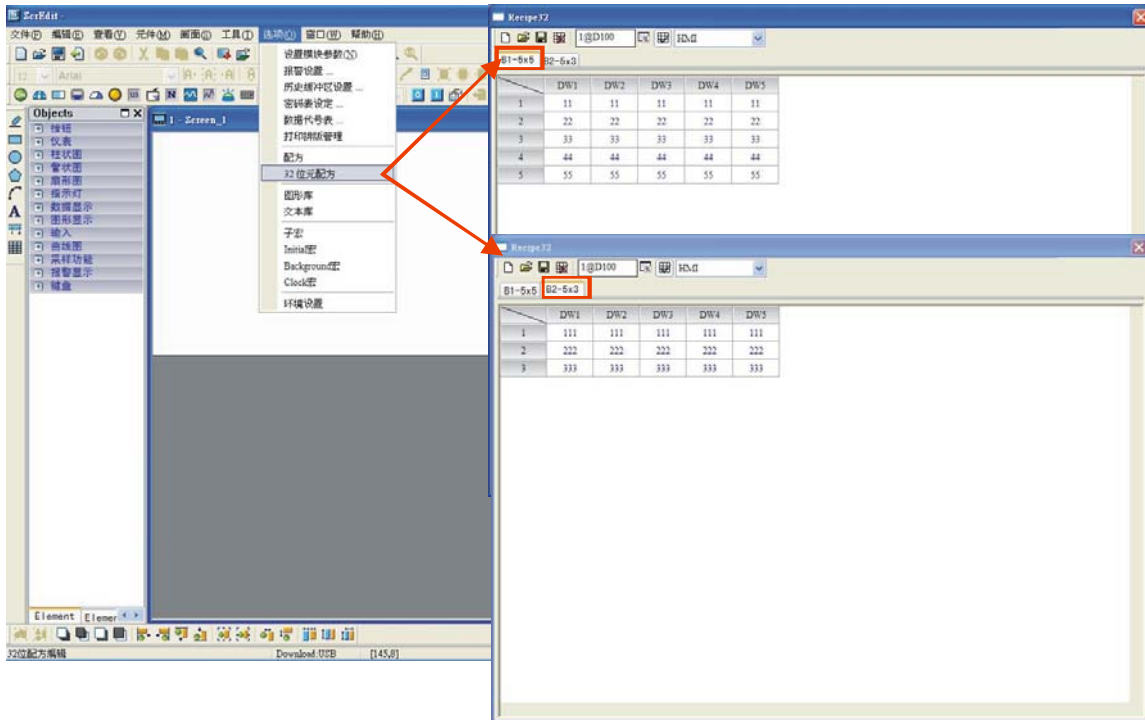
首先建立一新专案，如下图所示：



按下确定后，即进入到人机编辑画面，如下图所示：

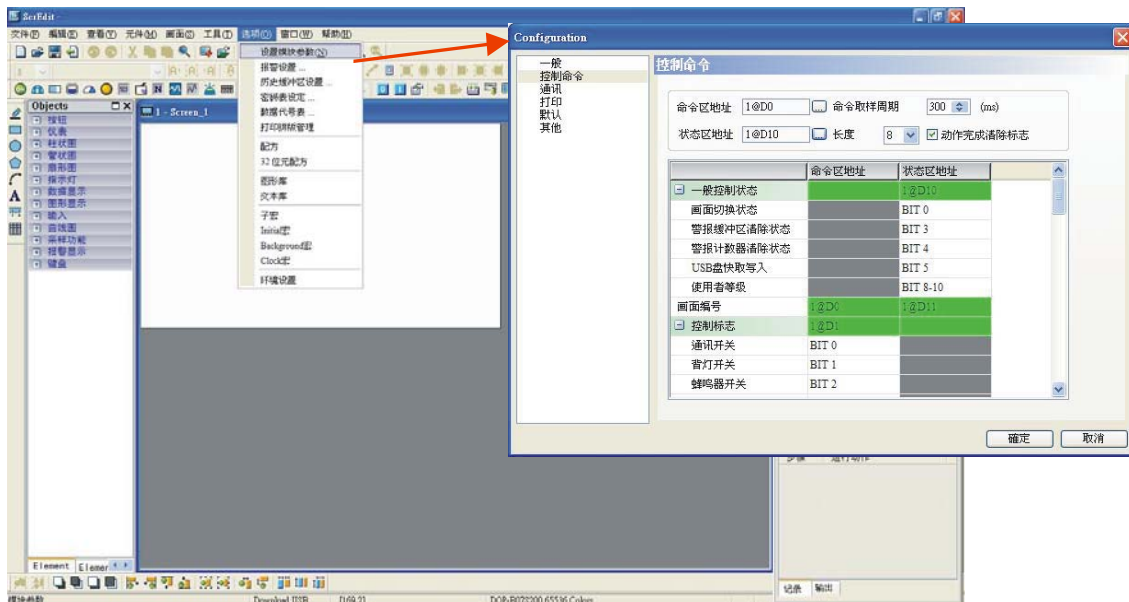


选择功能菜单中的选项/32 位配方来编辑 32 位配方表，在此设定两个组合配方表，写入位置为 1@D100，第一组合配方表为一长度乘上组数为 5*5 的配方，断电保持区设在 HMI；第二组合配方表为一长度乘上组数为 5*3 的配方，断电保持区设在 HMI，建立完成如下图所示：



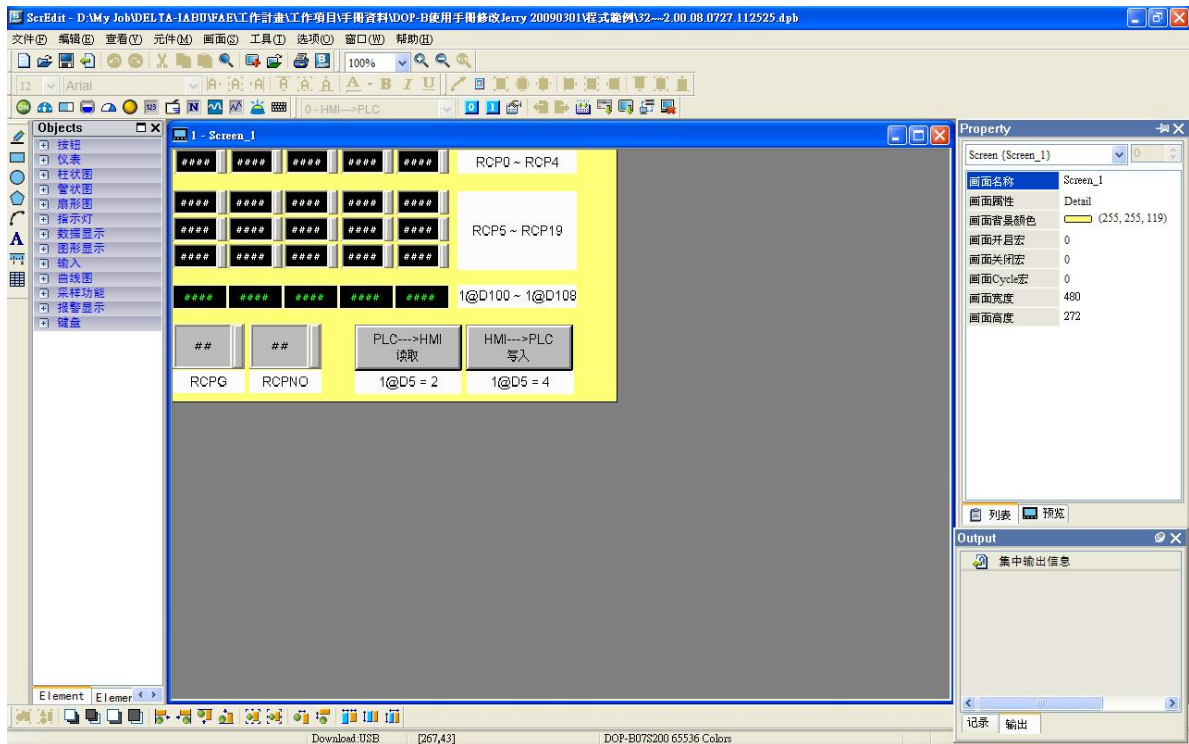
按下确定，完成配方设定。

选择功能菜单中的选项/设置模块参数的[控制命令]来设定控制区，以便控制配方的读取与写入，在此我们设定控制区的位址为 1@D0，长度为 8，如下图所示：



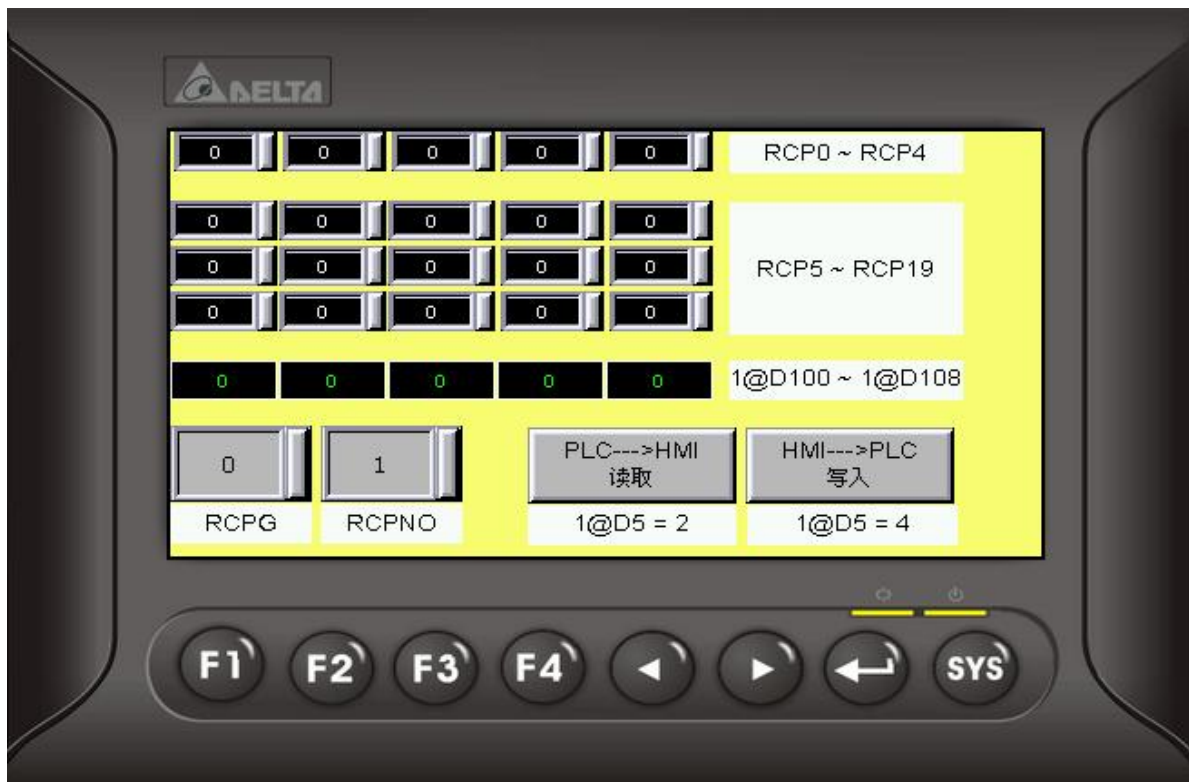
按下确定，完成控制区设定。

在人机编辑画面中，我们将配方缓冲区 (RCP0~RCP4)、配方表 (RCP5~RCP19) 及配方写入位址 (1@D100~1@D108) 以数值输入元件列出；另外也建立配方组合别指定寄存器 (RCPG) 及配方组别指定寄存器 (RCPNO) 的数值输入元件，以及配方读取/写入元件 (1@D5)，如下图所示：

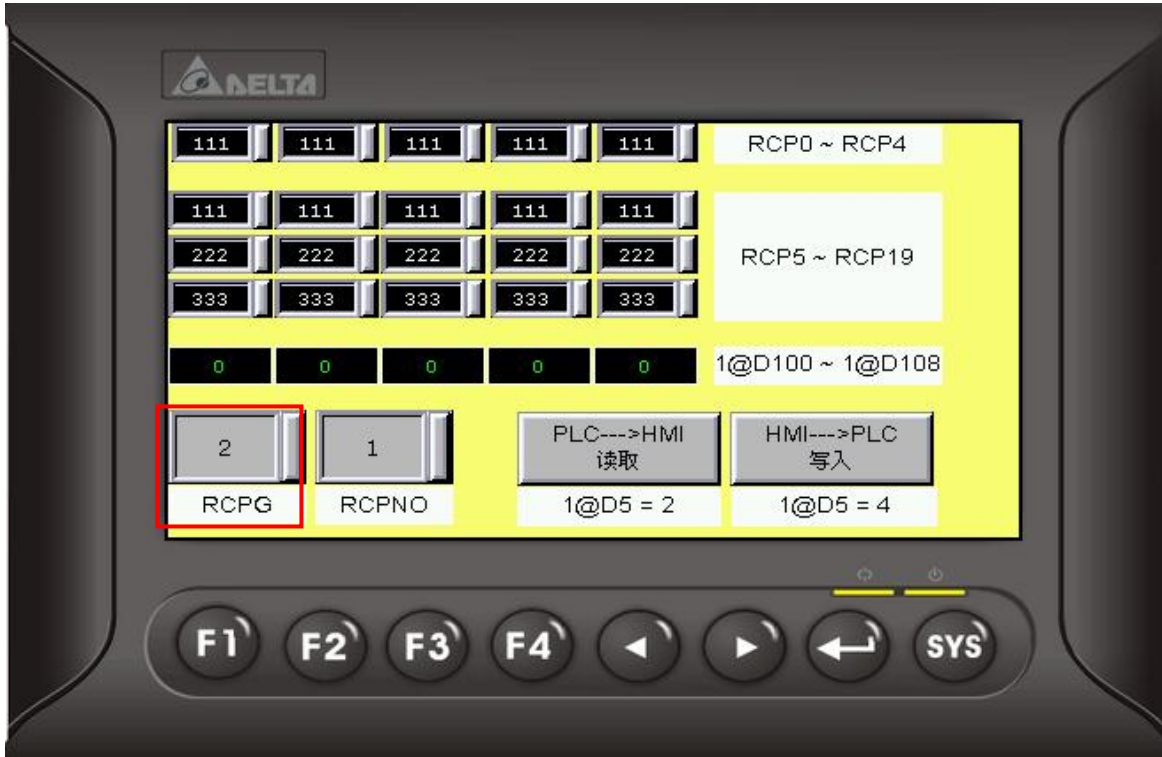


建立完成后进行编译动作，编译完成即可下载至人机进行操作。

程式执行画面如下，进入程式画面后，RCPG 默认为 0，RCPNO 默认为 1，因此时尚未选择配方组合别，因此缓冲区 RCP0~RCP4 以及配方表 RCP5~RCP19 显示为 0。此时配方尚未写入 PLC 位址 1@D100~1@D108，因此显示数值为 0，如下图所示：



若 RCPG 设定为 2，则配方表 RCP5~RCP19 立即显示第二组合配方，缓冲区 RCP0~RCP4 将显示第一组配方，如下图所示：

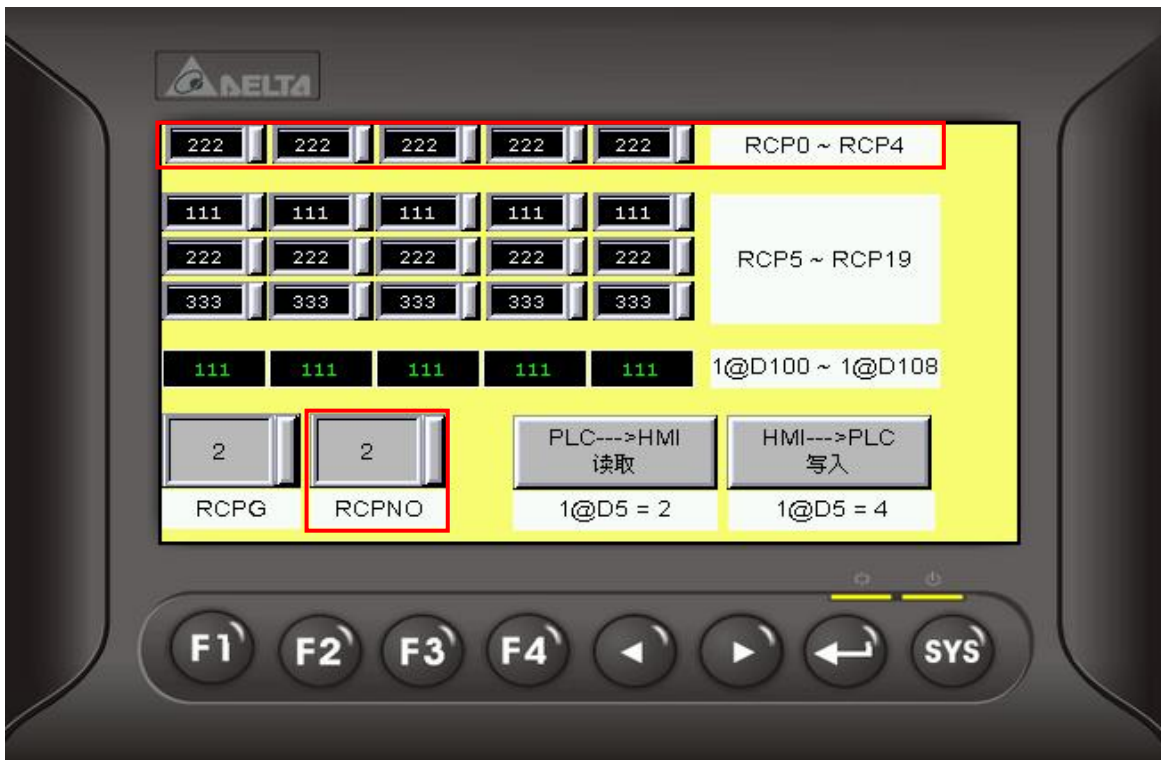


此时配方尚未写入 PLC 位址 1@D100~1@D108，因此显示数值依旧为 0。

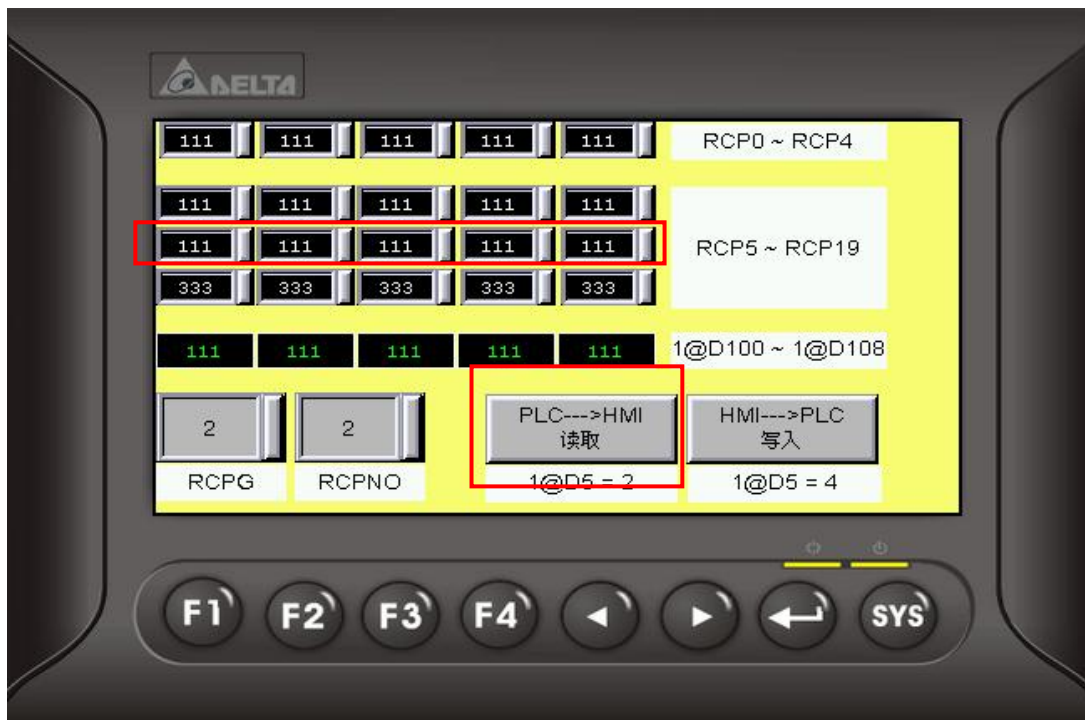
按下配方写入按钮，人机立即将第一组配方写入 PLC 位址 1@D100~1@D108，此时显示数值为 111，如下图所示：



选择第二组配方，即 RCPNO = 2，则 RCP0~RCP4 即显示第二组配方，如下图所示：



此时若按下配方读取按钮，则人机立即将 PLC 内的配方读回并存入第二组配方中，如下图所示：



此时人机第二组配方已被更改。

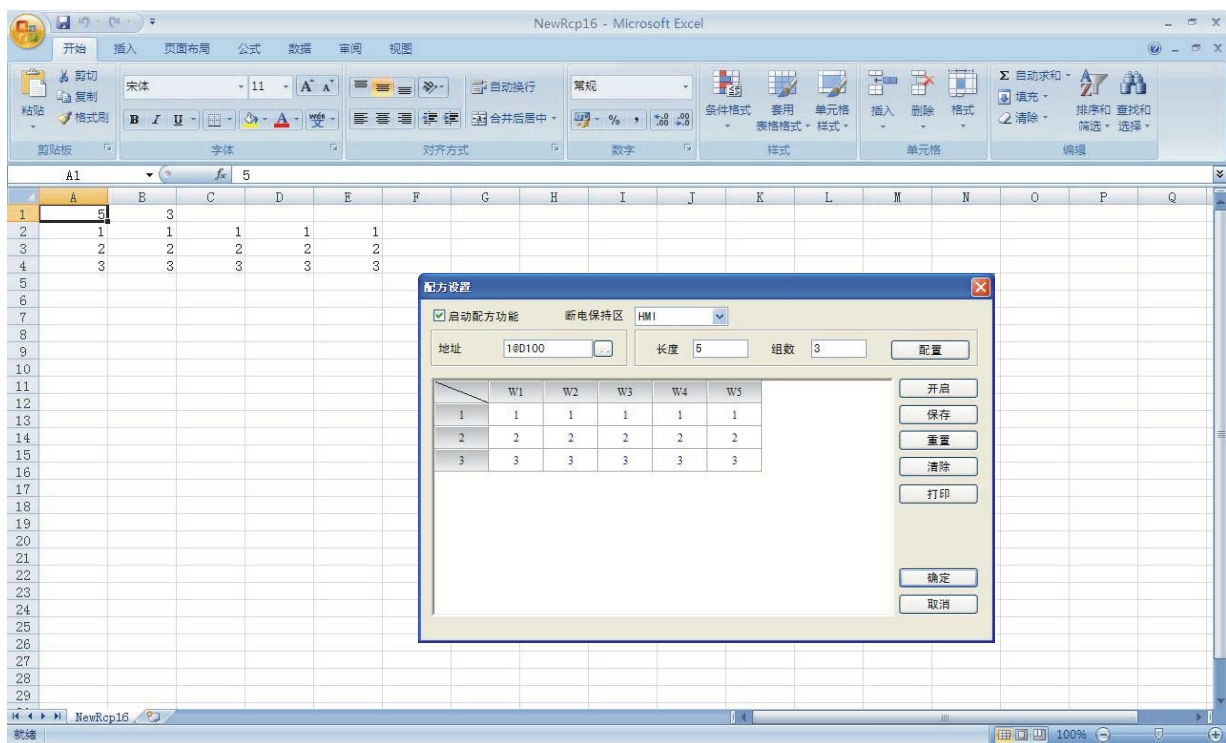
以上即为 32 位配方使用范例。

4.3 配方 CSV 使用范例

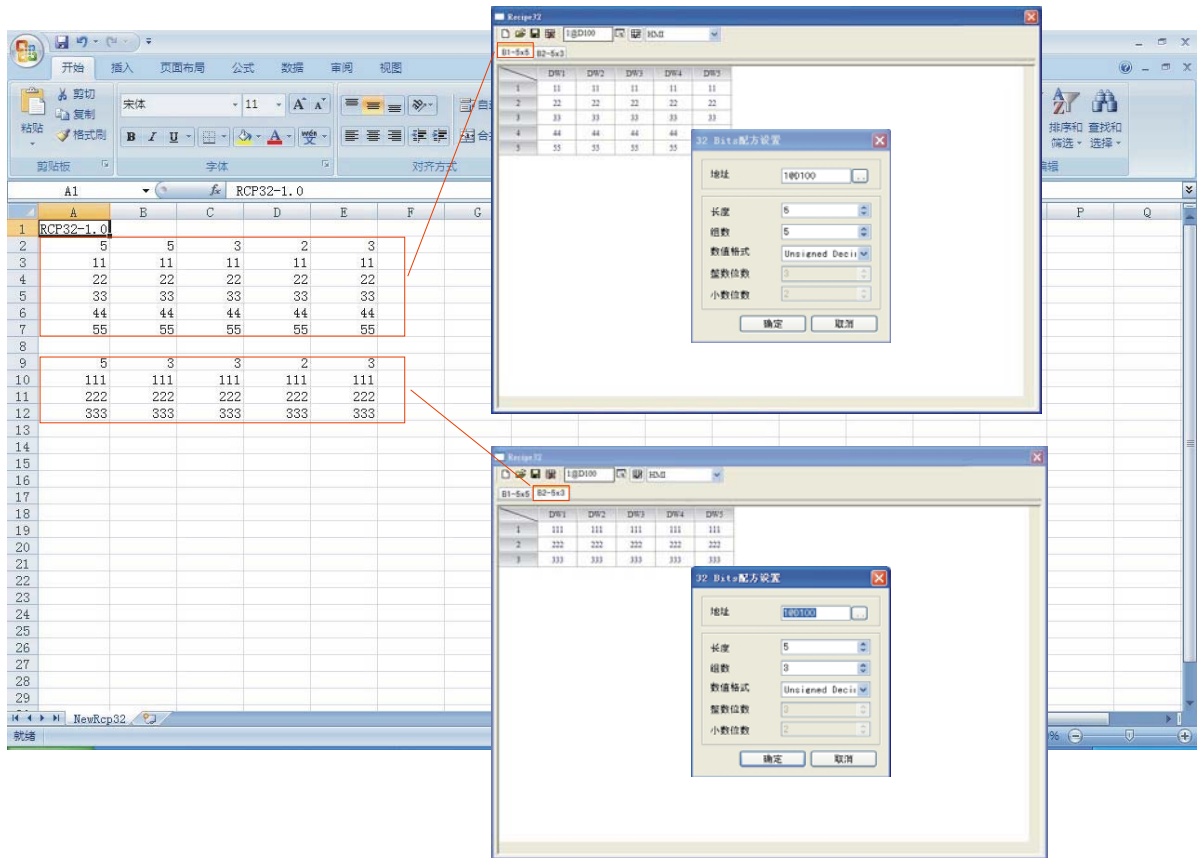
目前提供的配方编辑方式有两种，一种为 RCP 格式只能在 Screen Editor 中编辑，另外一种则是 CSV 文件，可在 Windows® Excel 中编辑。RCP 格式的编辑方式则如前面章节所提到的方式，以下将介绍以 Windows® Excel 编辑的方式。

以上述两个范例为例，在 Screen Editor 编辑好配方后，可储存文件为 CSV 文件，使用 Windows® Excel 开启储存好的 CSV 文件。而配方与 32 位配方的 CSV 文件定义有些不同，以下将会分开解释。

以配方范例的设定为例，其配方 CSV 文件如下图所示，第一行为配方的长度与组数，其他则是实际的配方数值。



以 32 位配方范例的设定为例，其配方 CSV 文件如下图所示，第一列的 RCP32-1.0 为 32 位配方定义版本，不可去修改。



如上图所示，右边上下两小图为 Screen Editor 配方编辑视窗，左边为 Excel 编辑视窗，上面红色框为第一个配方组合的资料，下面红色框为第二个配方组合的资料，其中 Excel 第二列的数据则是表示配方组合的设定，对应关系如下表。

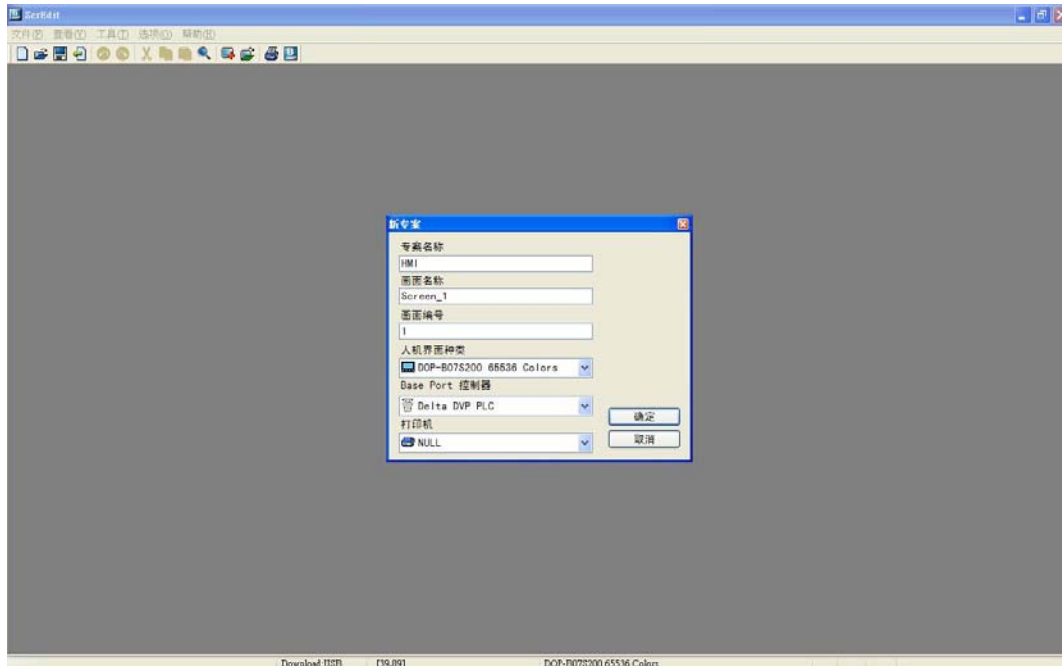
| Excel 栏位 | 配方组合设定值 |
|----------|--|
| A | 长度 |
| B | 组数 |
| C | 整数位数(数值格式为 Floating 才可设定) |
| D | 小数位数(数值格式为 Floating 才可设定) |
| E | 数值格式: 2: Signed Decimal 3: Unsigned Decimal 6: Floating |

依序空一列之后则是下一个配方组合的设定，其设定方式相同。

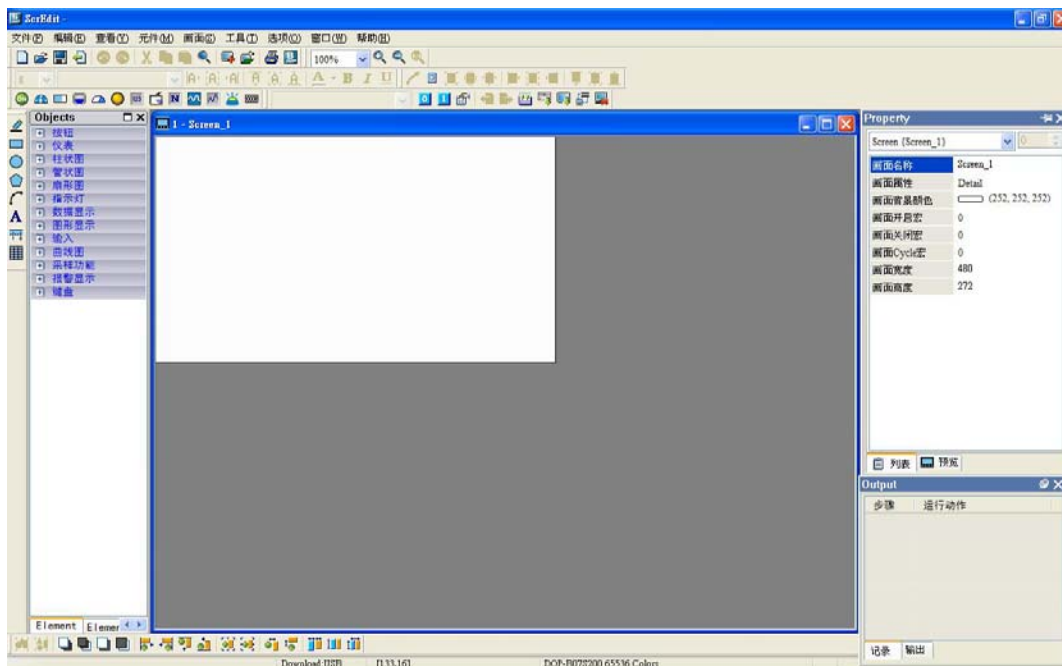
以上即为配方 CSV 使用范例。

4.4 多国语言使用范例

首先建立一新专案，如下图所示：



按下确定后，即进入到人机编辑画面，如下图所示：

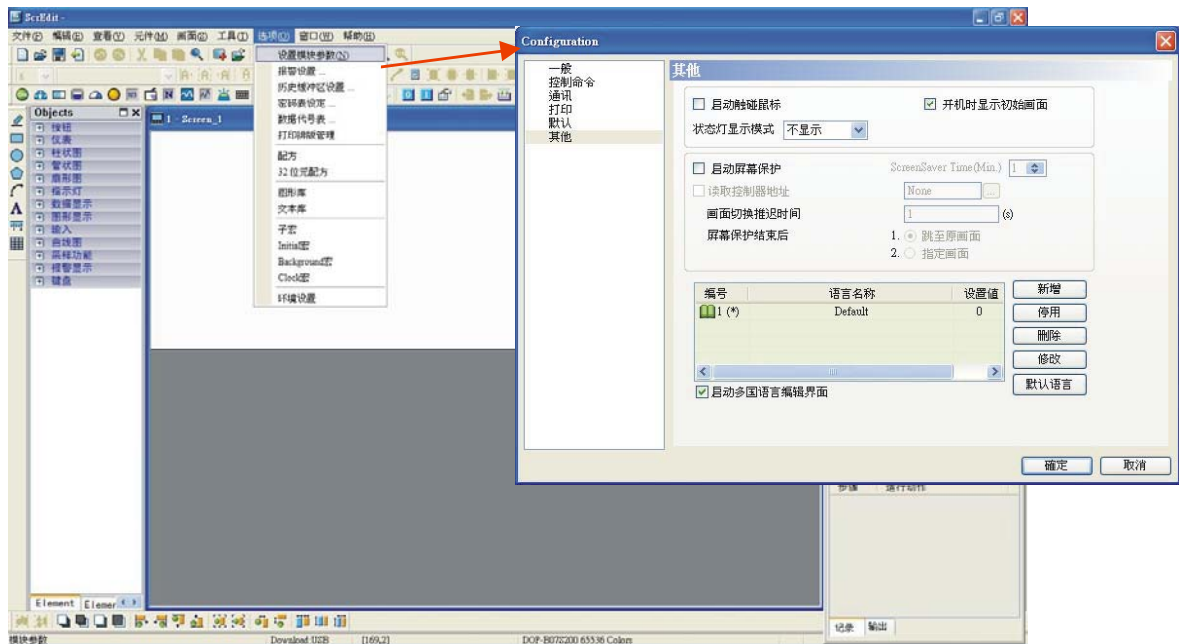


选择功能菜单中的选项/设置模块参数的[控制命令]来设定控制区，以便控制多国语言的选择，在此我们设定控制区的位址为\$0，长度为8，如下图所示：

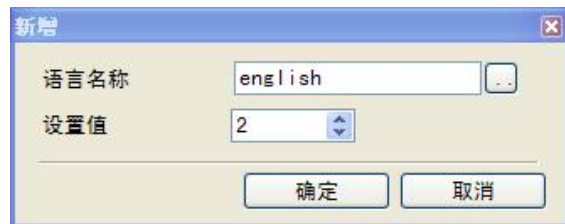


按下确定，完成控制区设定。

切换至设定模组参数中的[其他]页面，如下图所示：

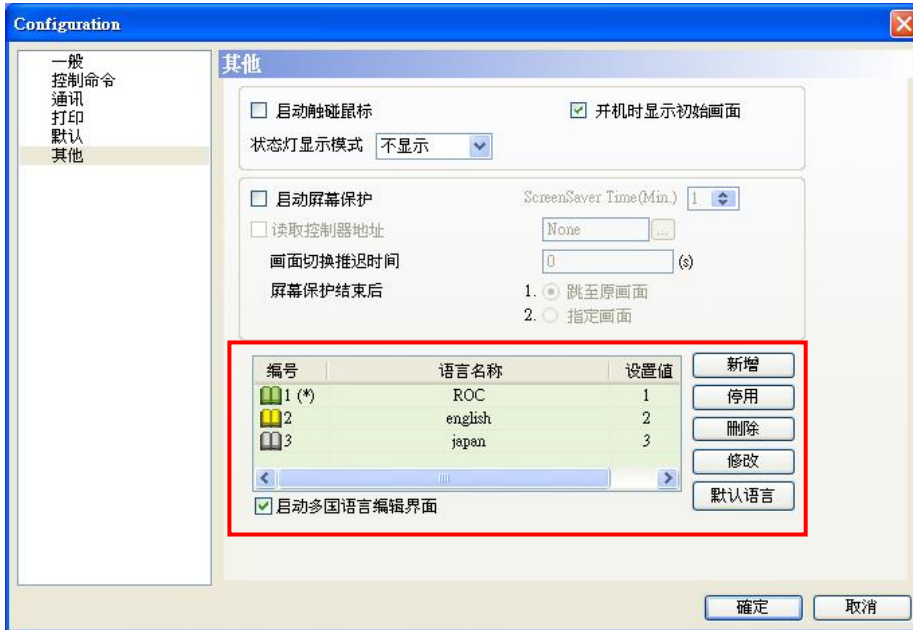


按下新增按钮，来设定多国语言，在此我们设定中文语言的设定值为 1，英文语言的设定值为 2，日文语言的设定值为 3，如下图所示：





在此设定中文为开机默认语言(*), 设定完成后, 视窗会显示出用户所设定的所有语言, 如下图所示:



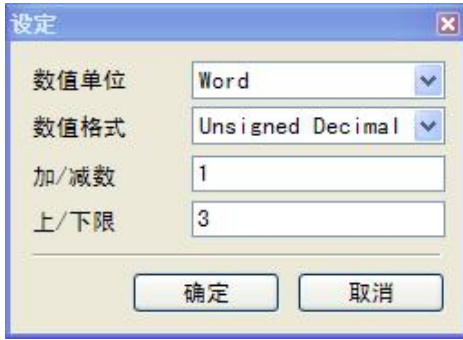
按下确定结束设定。

在画面中建立一静态文字元件, 在属性表里可以输入各语言文字, 如下图所示:



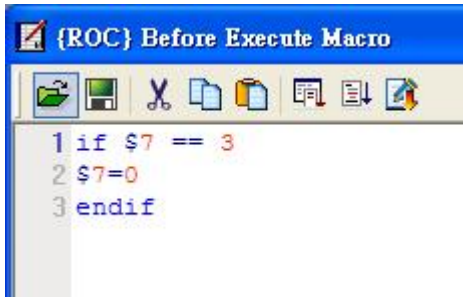
用 可自行设定文字大小、字型与文字颜色等。

此外，语言切换需要使用到控制区，因此我们在画面中设定一增加值按钮(\$7)，设定值范围如下图所示：

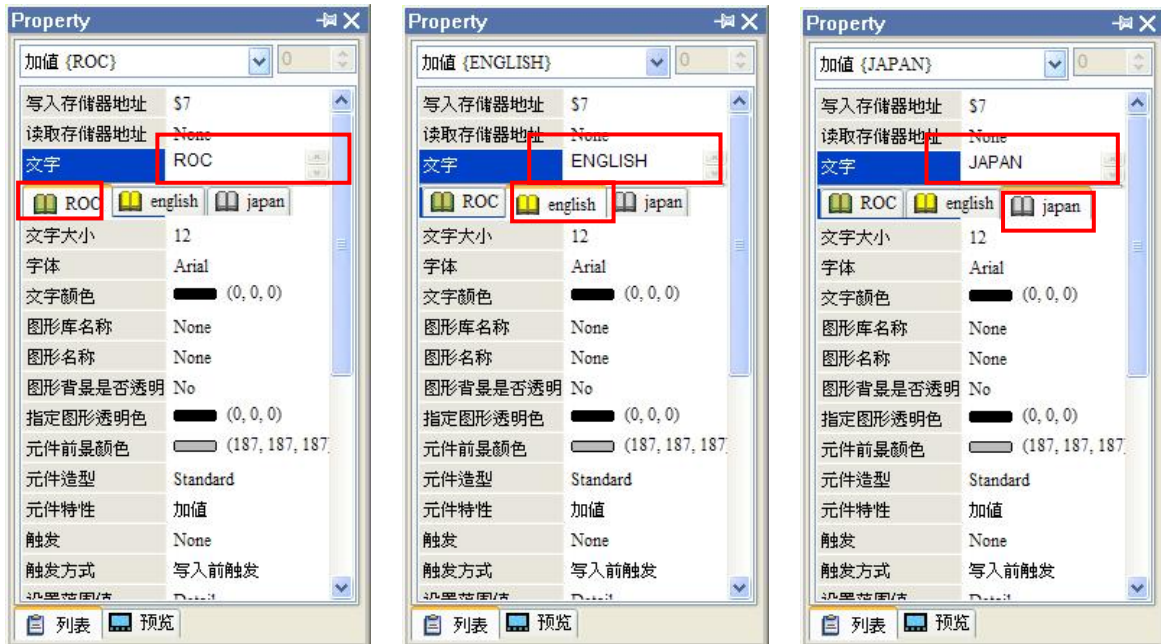


因此范例使用到三种语言，所以上/下限为 3。

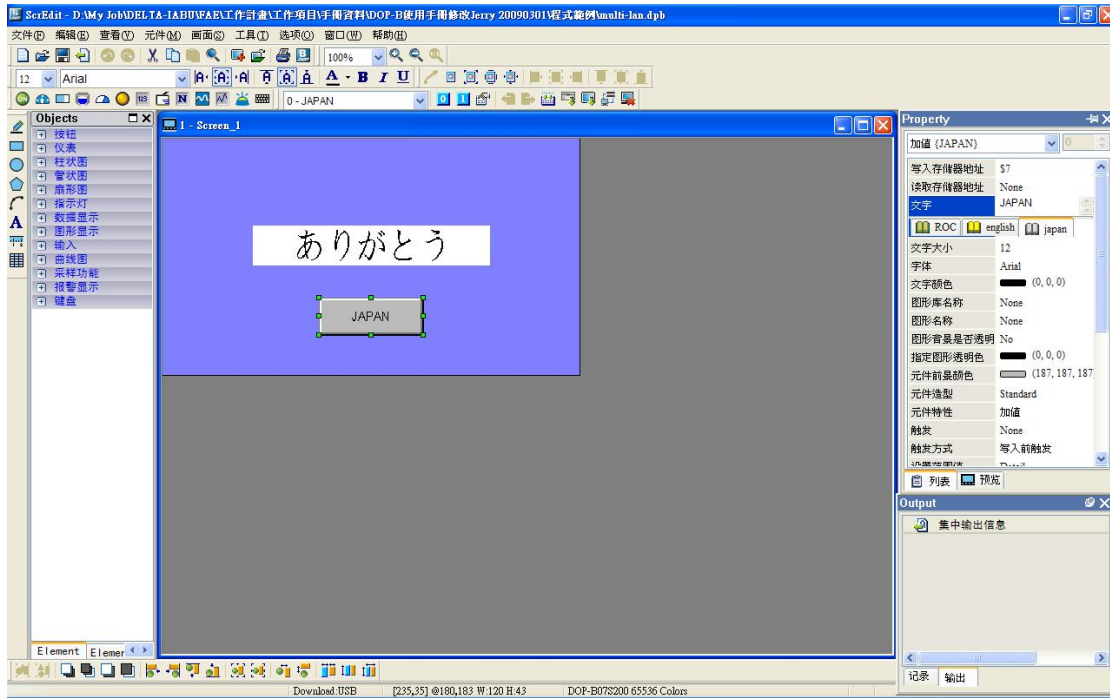
在增加值按钮的执行前宏写入如下的宏指令，代表切换到最后一个语言后，会回到第一个语言：



我们也在增加值按钮上做多国语言的切换设定：

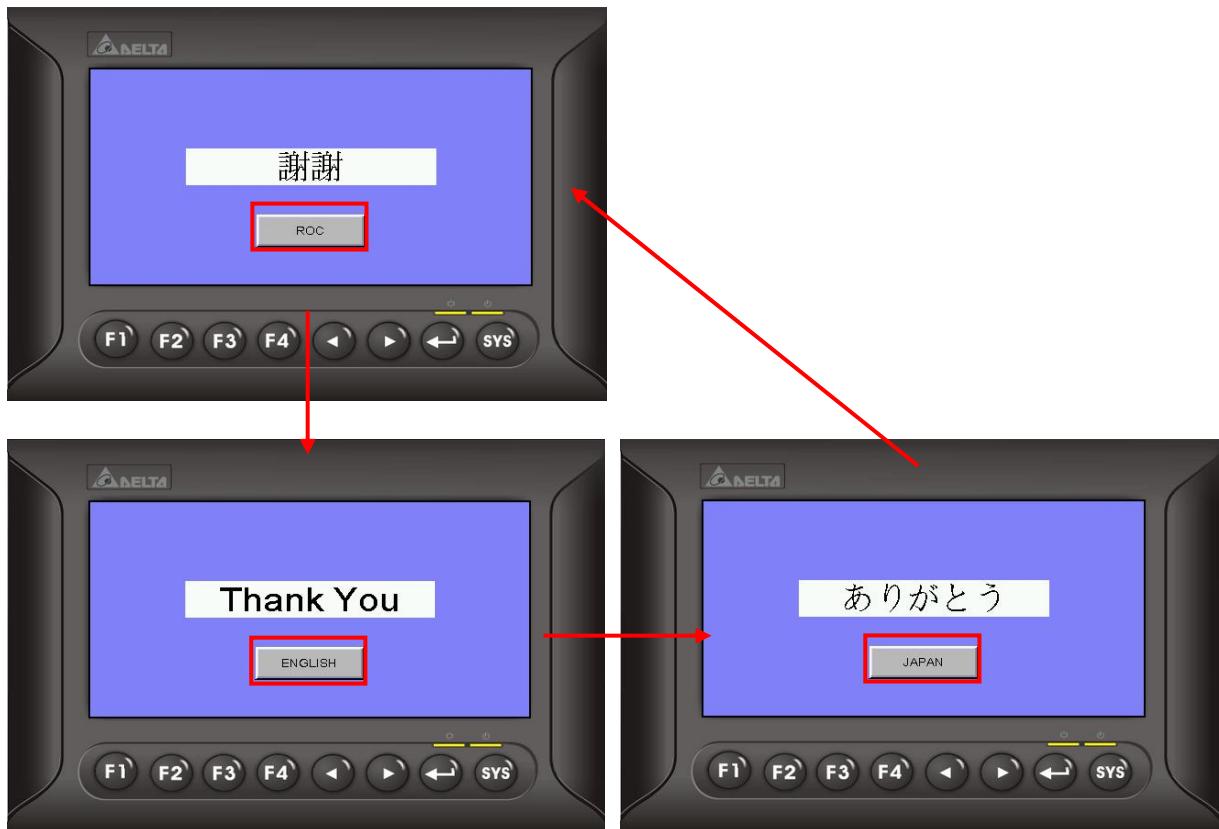


建立完成后如下图所示：



建立完成后进行编译动作，编译完成即可下载至人机进行操作。

程式执行画面如下，默认语言为中文，若要切换语言，按下加值键即可任意切换语言：

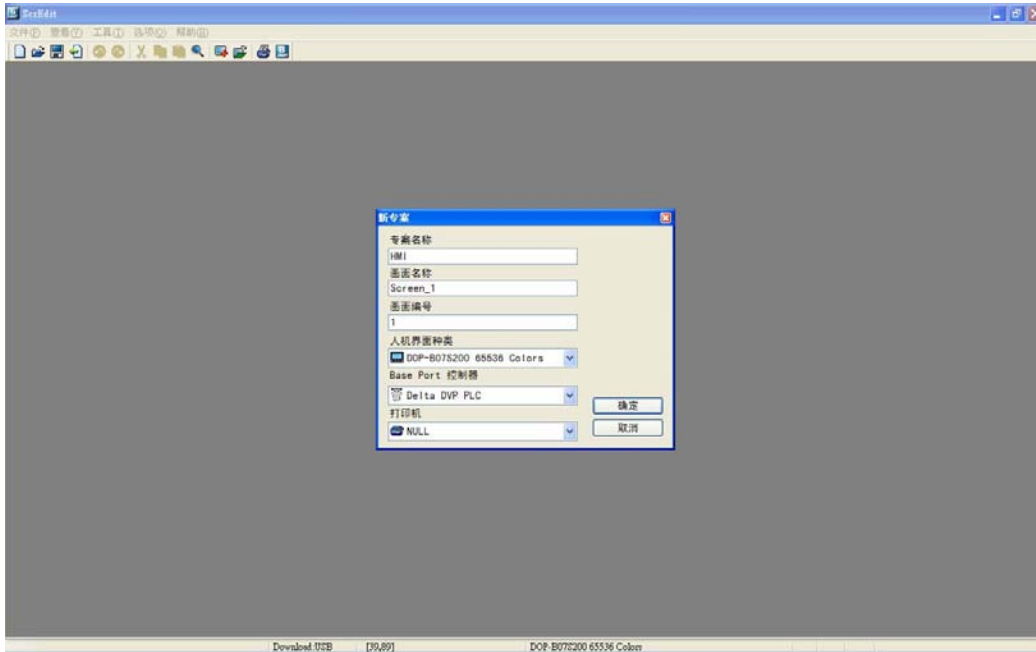


以上即为多国语言使用范例。

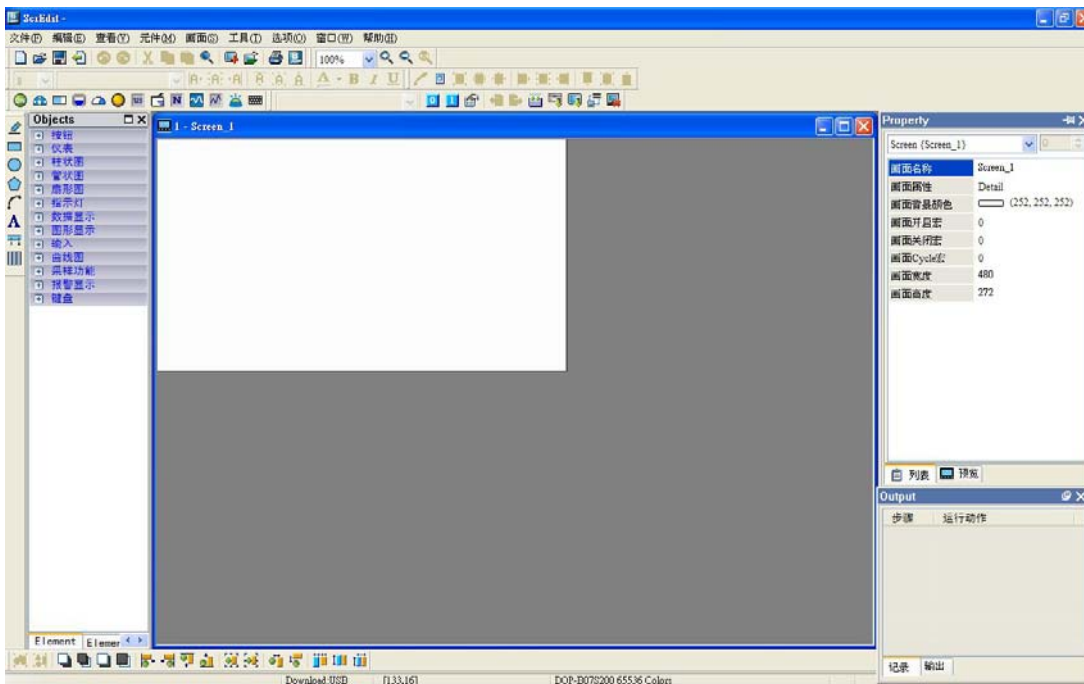
4.5 人机辅助软件 Flash Transfer 使用范例

人机辅助软件 Flash Transfer 可读取历史趋势图、历史数值数据表及警报等，在此我们以历史趋势图及警报来做范例说明。

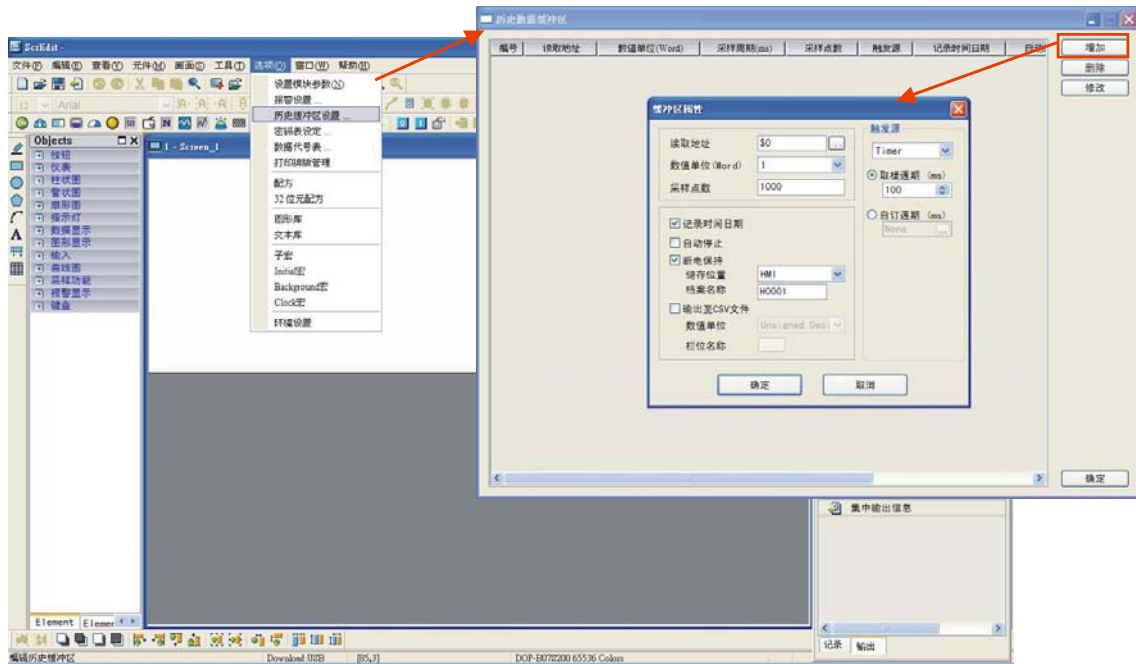
首先建立一新专案，如下图所示：



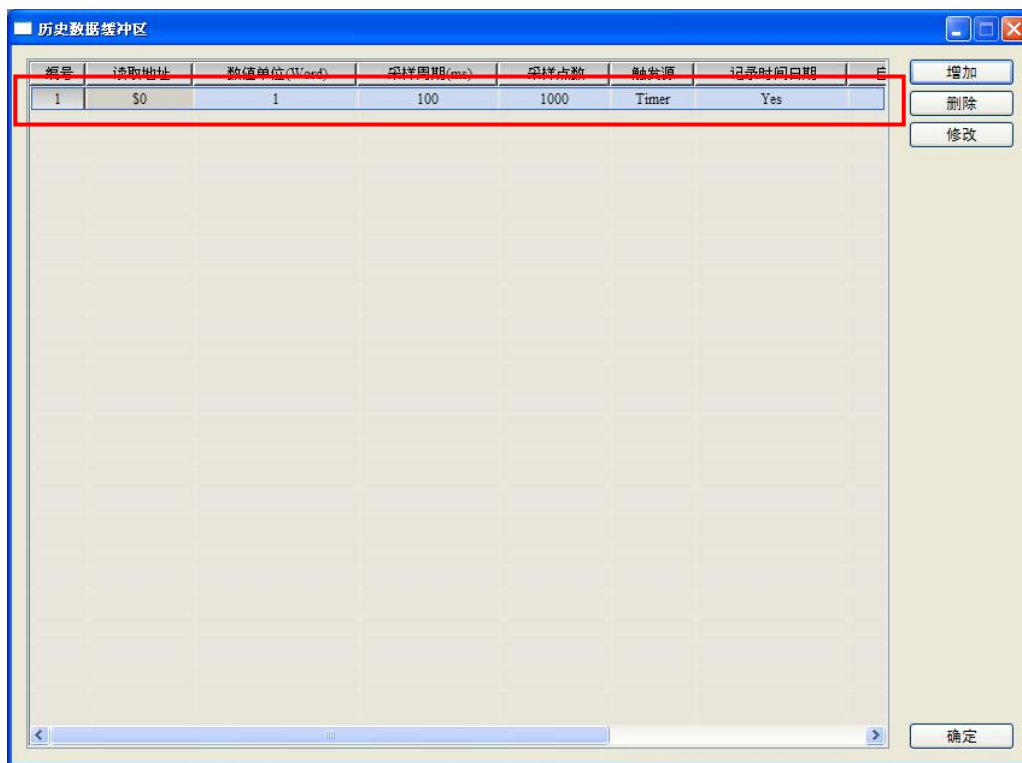
按下确定后，即进入到人机编辑画面，如下图所示：



选择功能菜单中的选项/历史缓冲区设定来设定历史缓冲区，在此设定一组历史缓冲区，写入位置为人机内部存储区\$0，断电保持区设在 HMI，如下图所示：

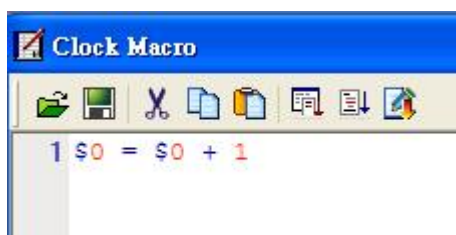


按下确定后，历史缓冲区即建立完成，如下图所示：

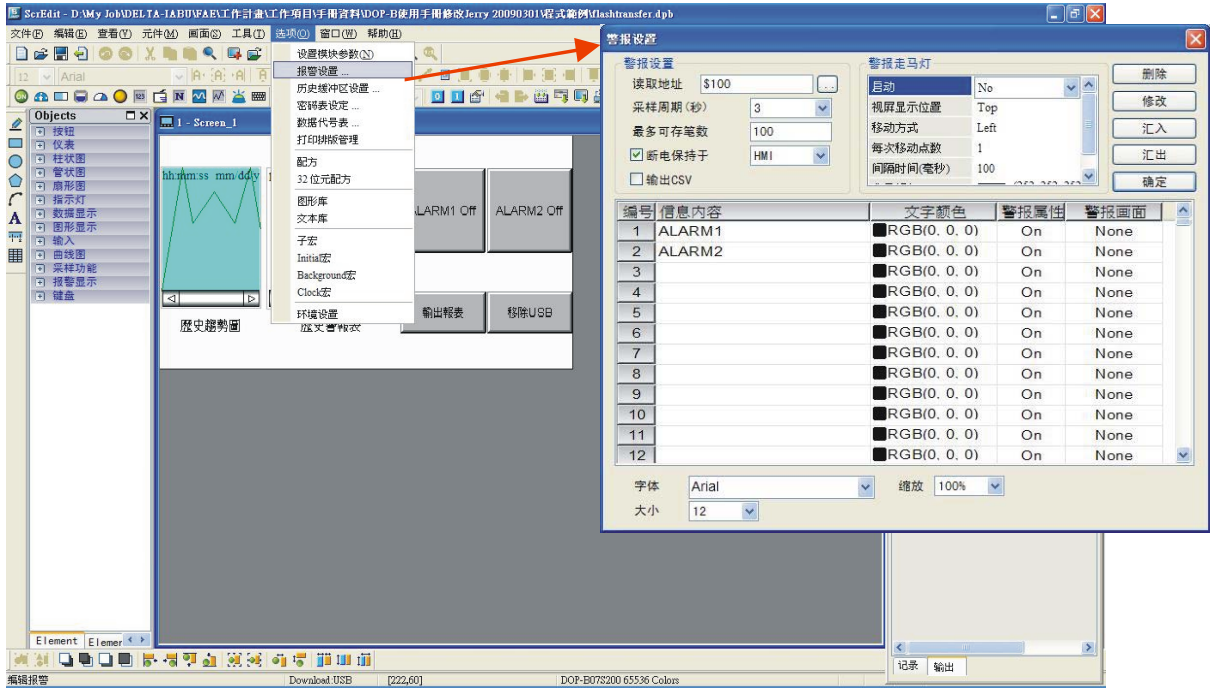


再按下确定，完成历史缓冲区设定。

在此，我们在 Clock 宏里写入累加的宏指令，以便让历史缓冲区\$0 的数值有变化：

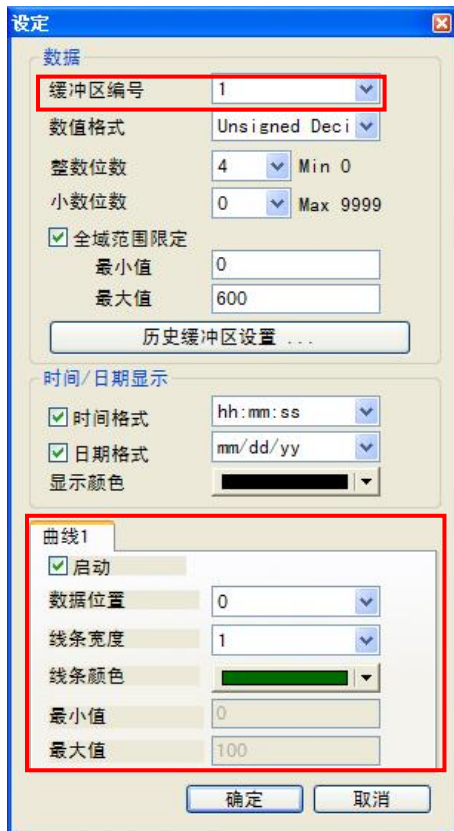


选择功能菜单中的选项/警报设定来设定警报，在此设定两组警报，触发位置为人机内部寄存器\$100.0 及\$100.1，断电保持区设在 HMI，如下图所示：

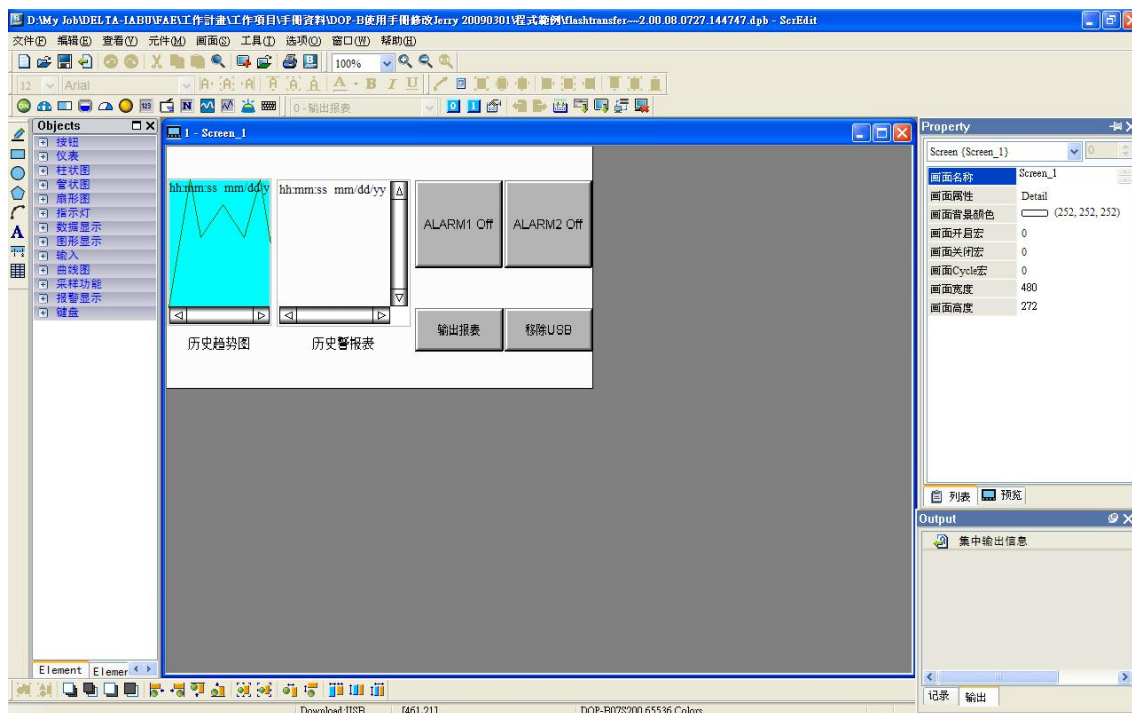


按下确定，完成警报设定。

在人机画面中，建立历史趋势图以及历史警报表，历史趋势图的设定如下，务必确认是否有选择缓冲区编号：



建立两个交替型按钮，用以触发警报 1 与警报 2，寄存器位置分别设定为\$100.0 及 \$100.1；另外再建立输出报表与移除 USB 按钮，如下图所示：



建立完成后进行编译动作，编译完成即可下载至人机进行操作。

操作画面如下所示，此时可发现历史趋势图有变化，这是因为 Clock 宏内进行了累加的动作，即 $S0 = S0 + 1$ ：



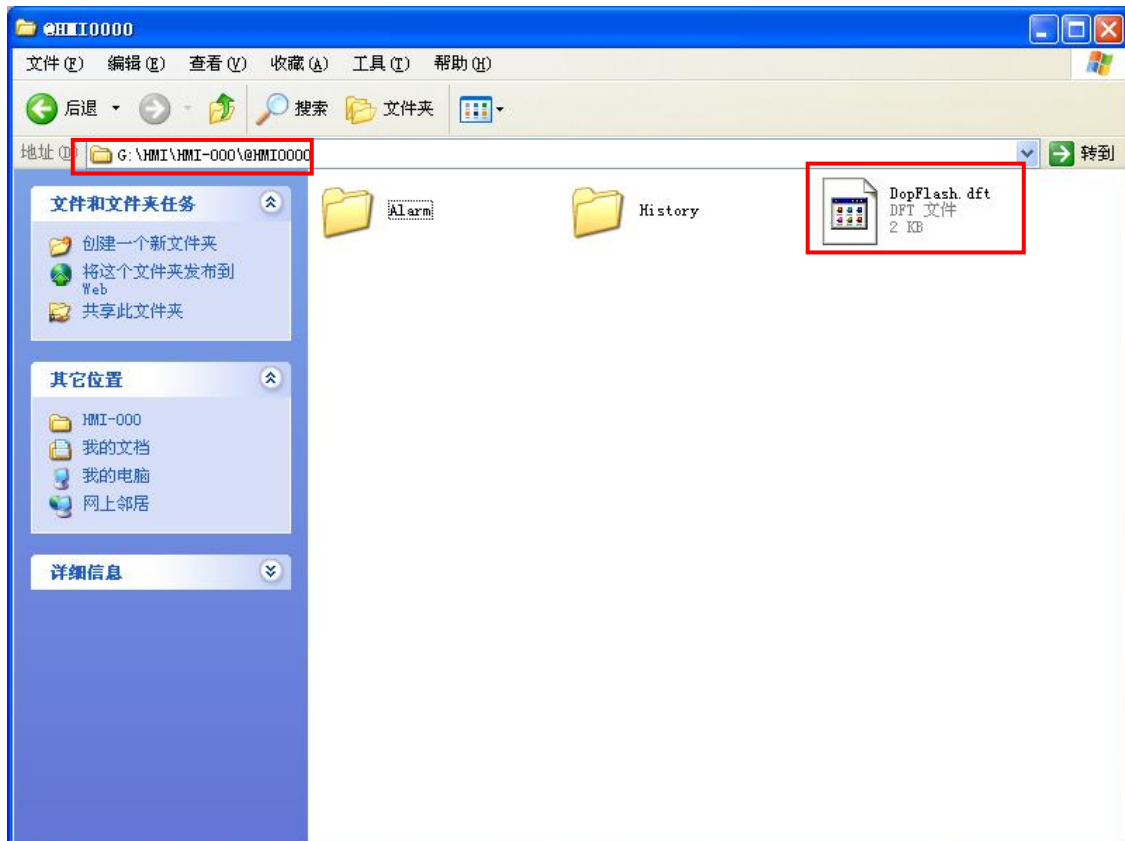
另外，用 可藉由按压两个交替型按钮来触发或关闭警报，警报会纪录在历史警报表中，

如下图所示：

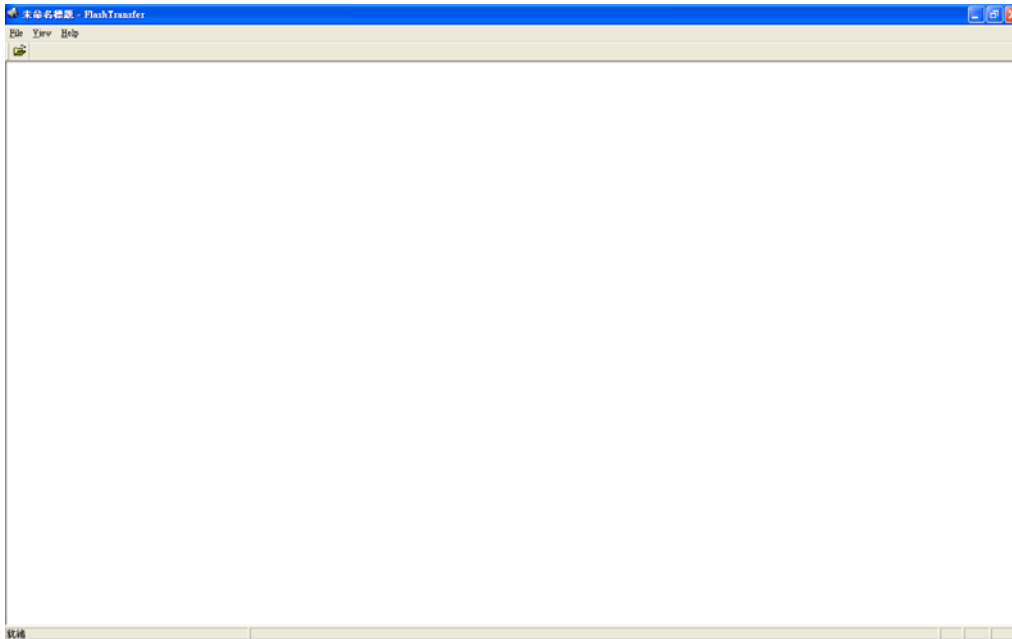



按下输出报表，将断电保持区内的历史缓冲区及警报资料输出至 U 盘，输出完成后，按下移除 USB 钮即可将 U 盘移除。

将 U 盘插上电脑并打开，可发现 DopFlash.dft 文件，Flash Transfer 必须打开包含此文件的文件夹方能读取人机所汇出的资料，如下图所示：



执行 Flash Transfer, 其位置在开始功能表/所有程序/Delta Industrial Automation/HMI/Screen Editor 2.00.18/Flash Transfer, 执行视窗如下图所示:



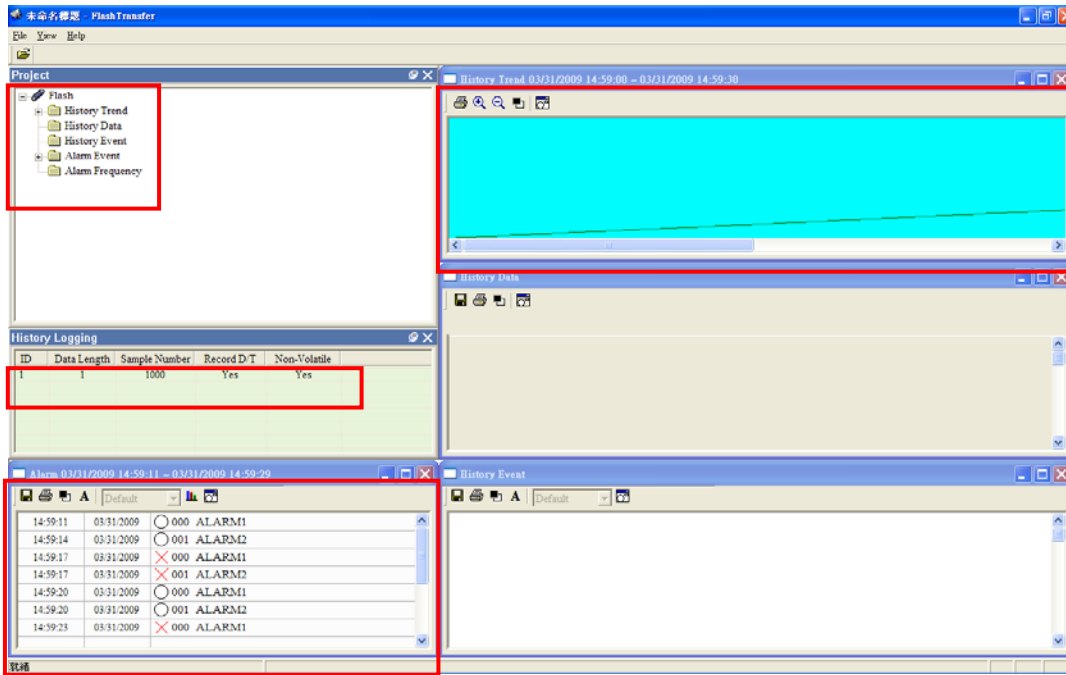
按下  Open 钮, 会出现以下视窗:



选择 U 盘的路径以及包含 DopFlash.dft 文件的文件夹:



按下确定键打开，打开之后的画面如下：

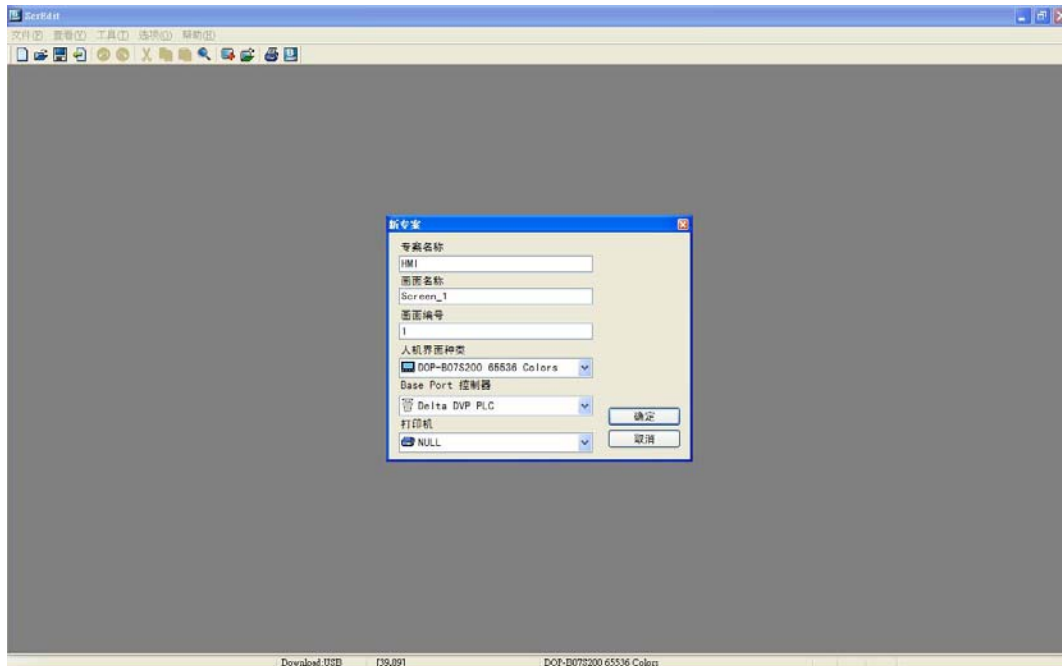


用 即可从 Flash Transfer 来读取人机所汇出的资料。

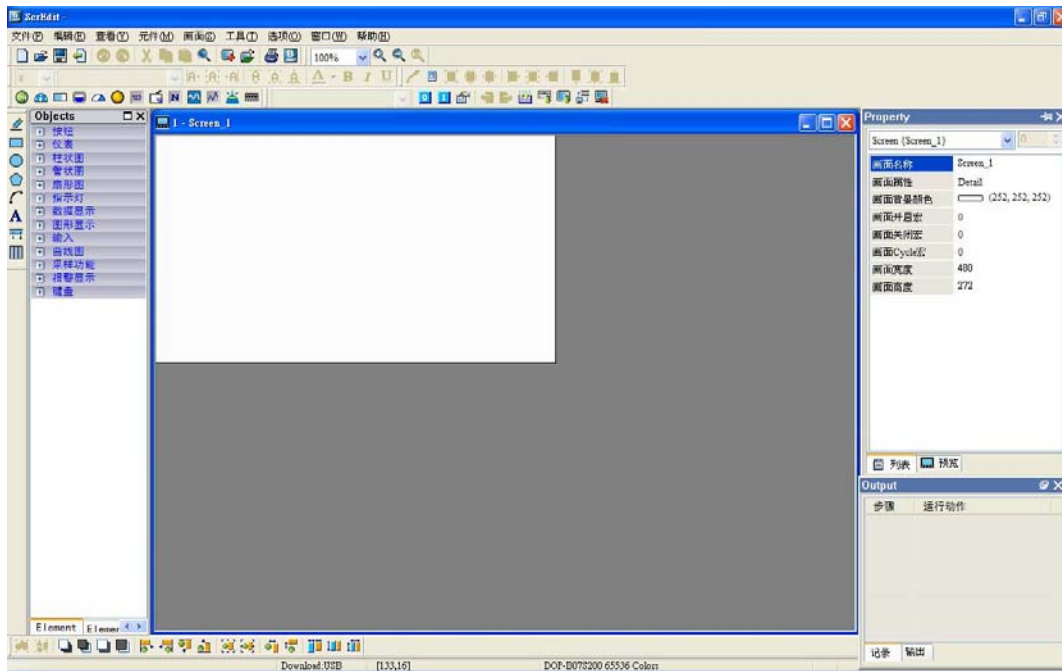
以上即为人机辅助软件 Flash Transfer 使用范例。

4.6 即时图显示使用范例

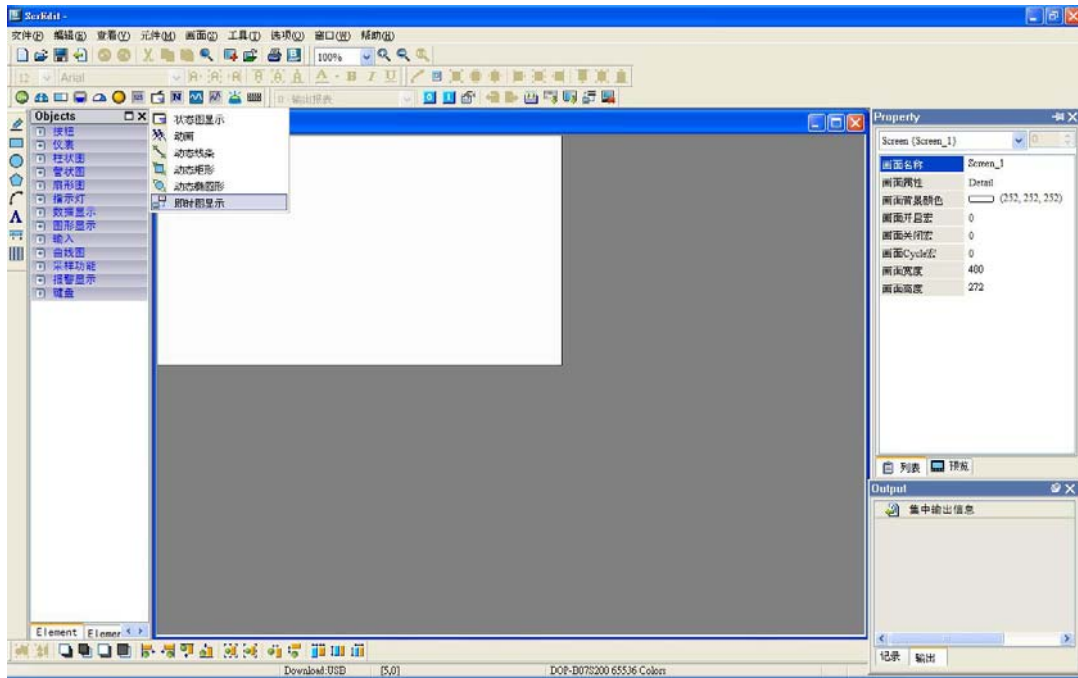
首先建立一新专案，如下图所示：



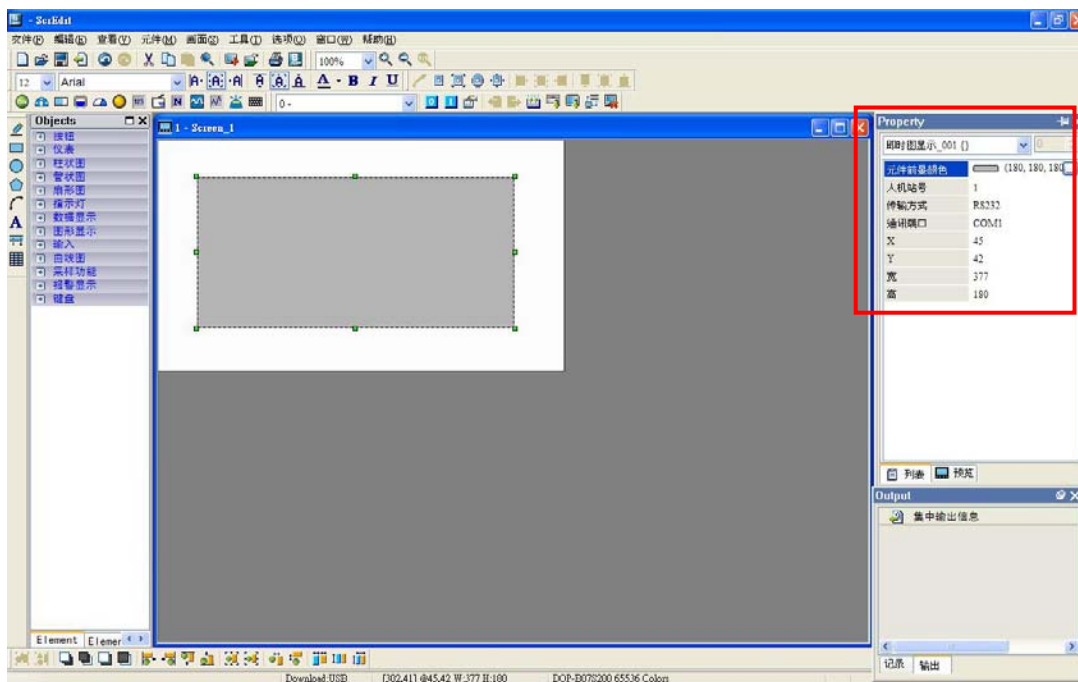
按下确定后，即进入到人机编辑画面，如下图所示：



建立一即时图显示元件，如下图所示：



在属性表中可设定传输方式，在此设定传输方式为 Com1、RS-232，如下图所示：



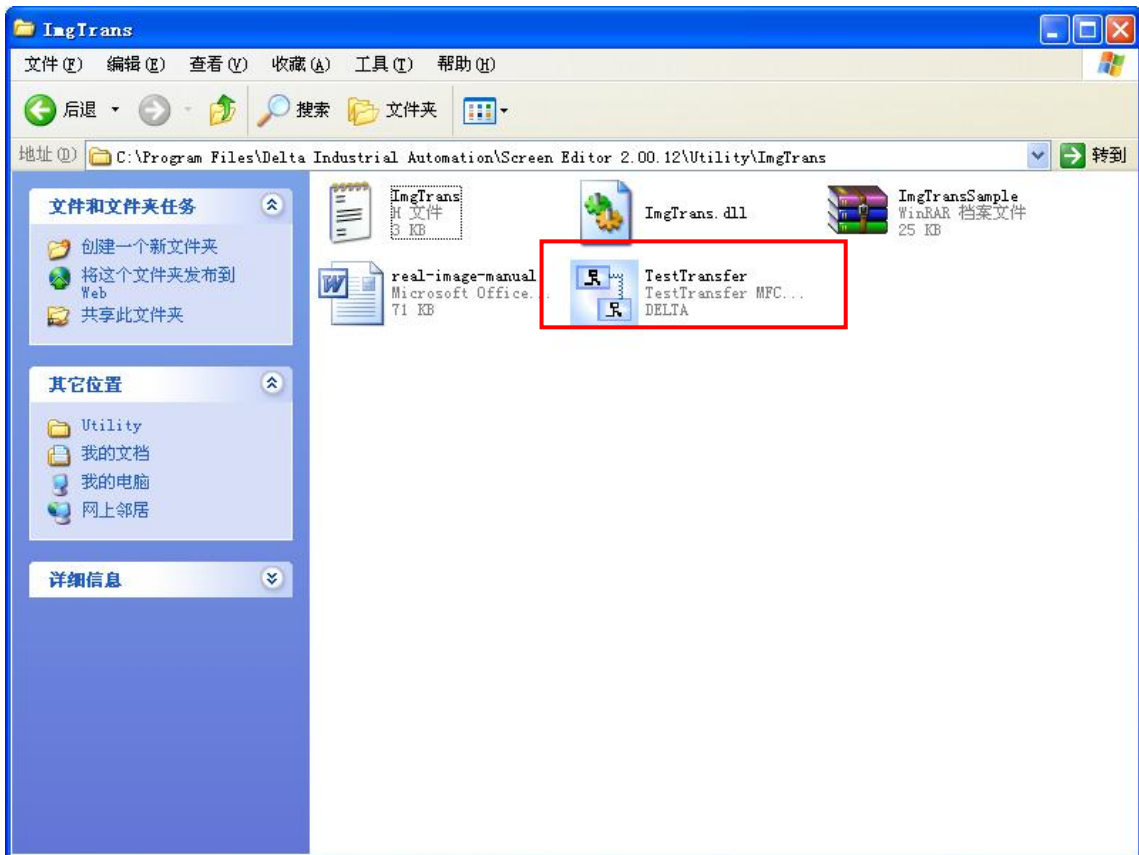
建立完成后进行编译动作，编译完成即可下载至人机进行操作。

选定要传输的图片，如下图所示：

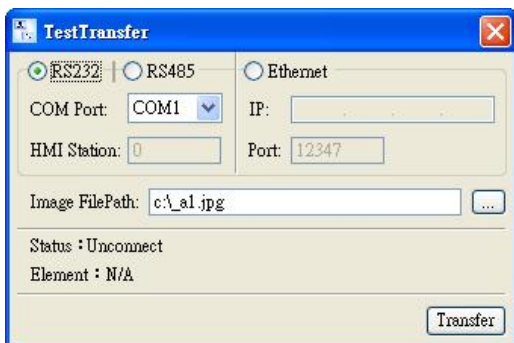


要传输图文件到人机，需使用特定软件 TestTransfer.exe，其位置在 Screen Editor 软件安装文件夹中：

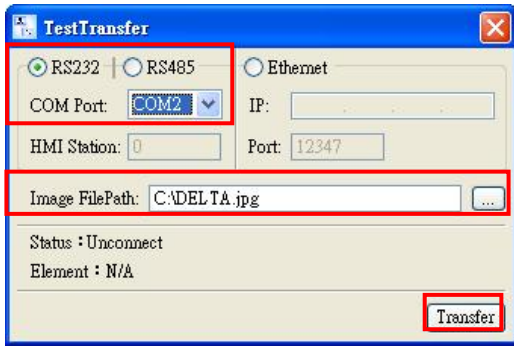
C:\Program Files\Delta Industrial Automation\Screen Editor
2.00.07\Utility\ImgTrans



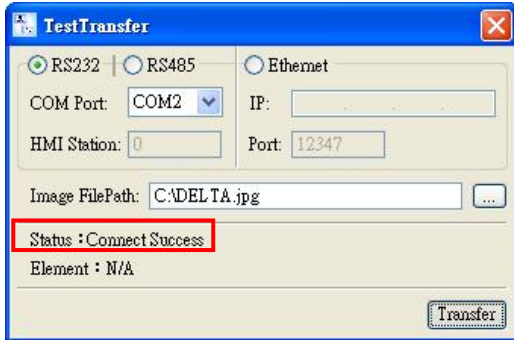
执行 TestTransfer.exe，画面如下所示：



选定传输方式以及图文件后，按下 Transfer 即可，如下图所示：



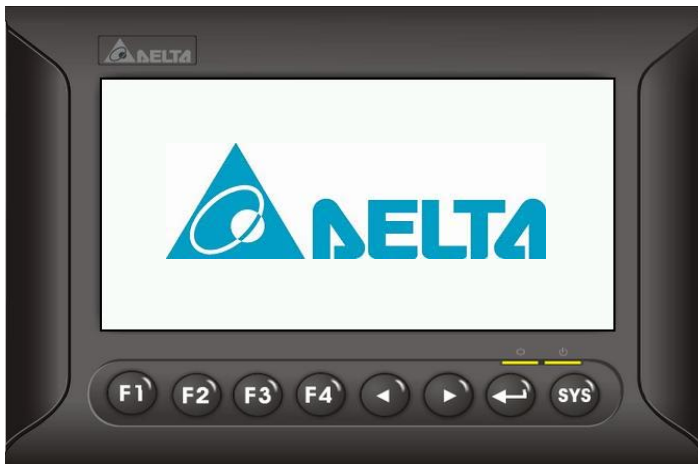
若传输成功，状态会显示 Connect Success:



若传输失败，会出现以下信息，此时请检查设定或接线是否有问题:



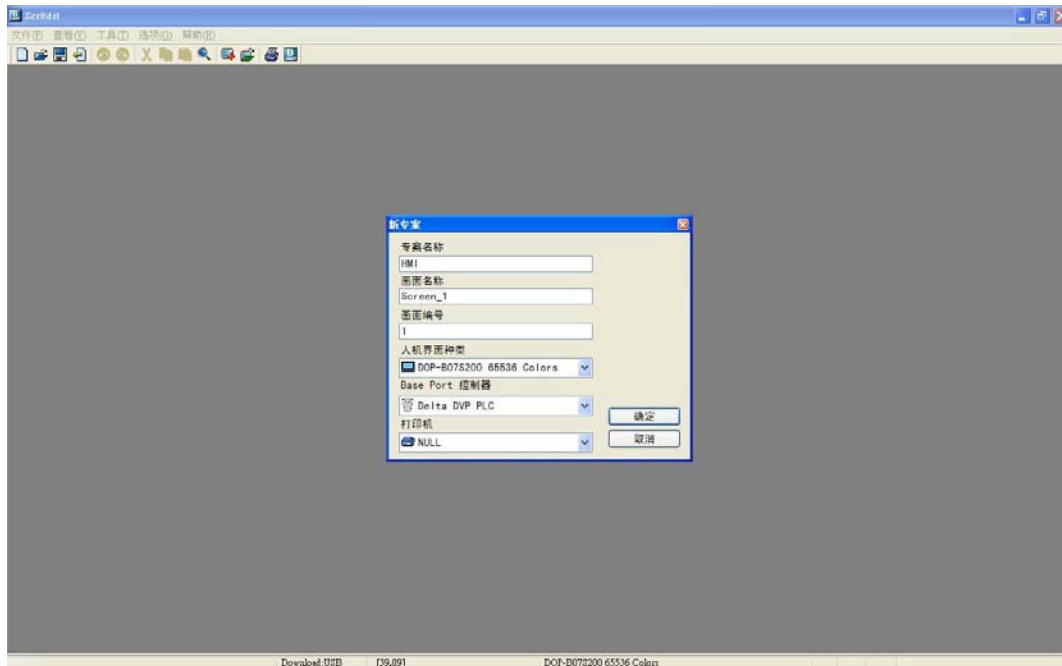
传输成功，画面如下所示:



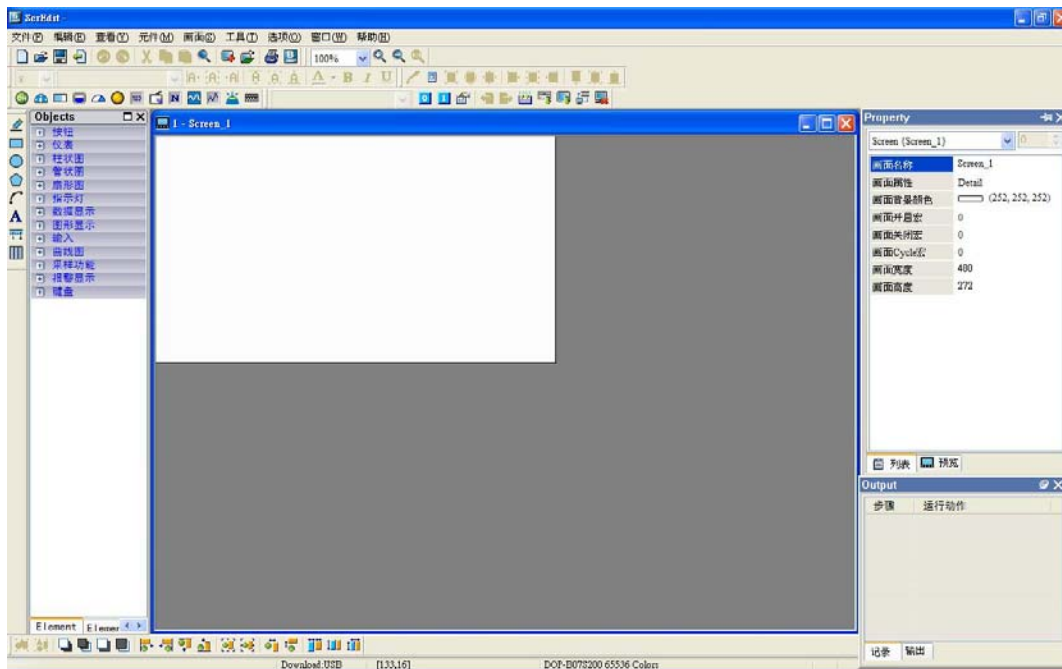
以上即为即时图显示使用范例。

4.7 一般曲线图使用范例

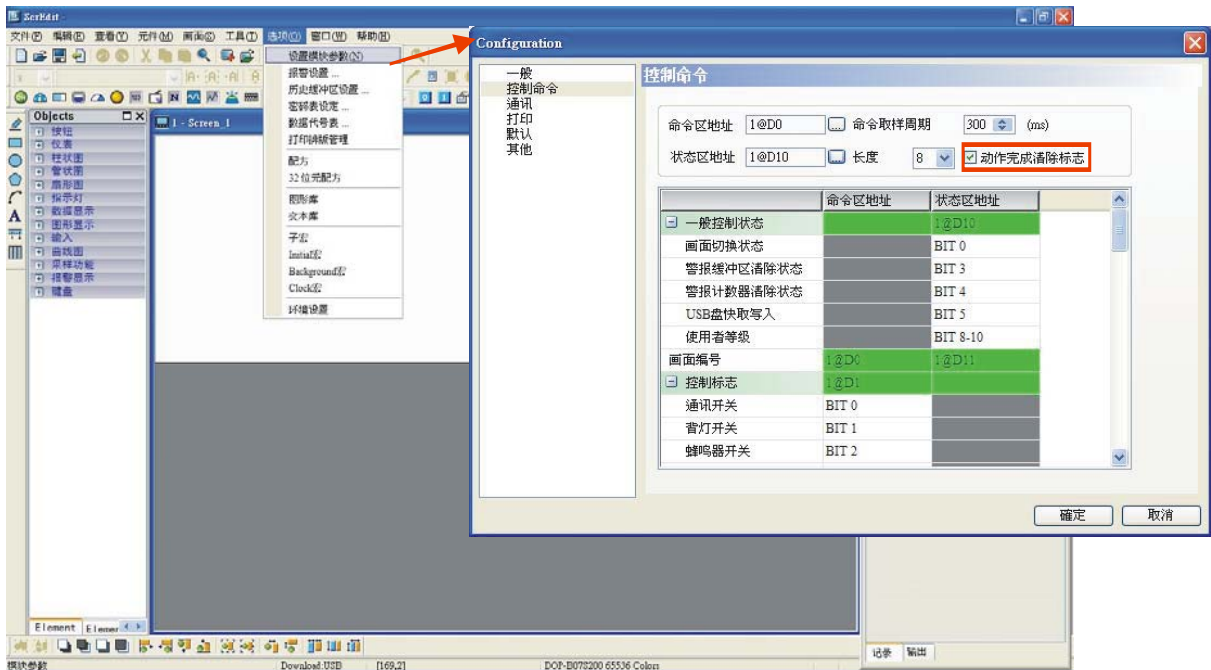
首先建立一新专案，如下图所示：



按下确定后，即进入到人机编辑画面，如下图所示：



选择功能菜单中的选项/设置模块参数的[控制命令]来设定控制区，如此即可使用曲线取样标志的功能，在此我们设定控制区的位址为\$0，长度为8，如下图所示：



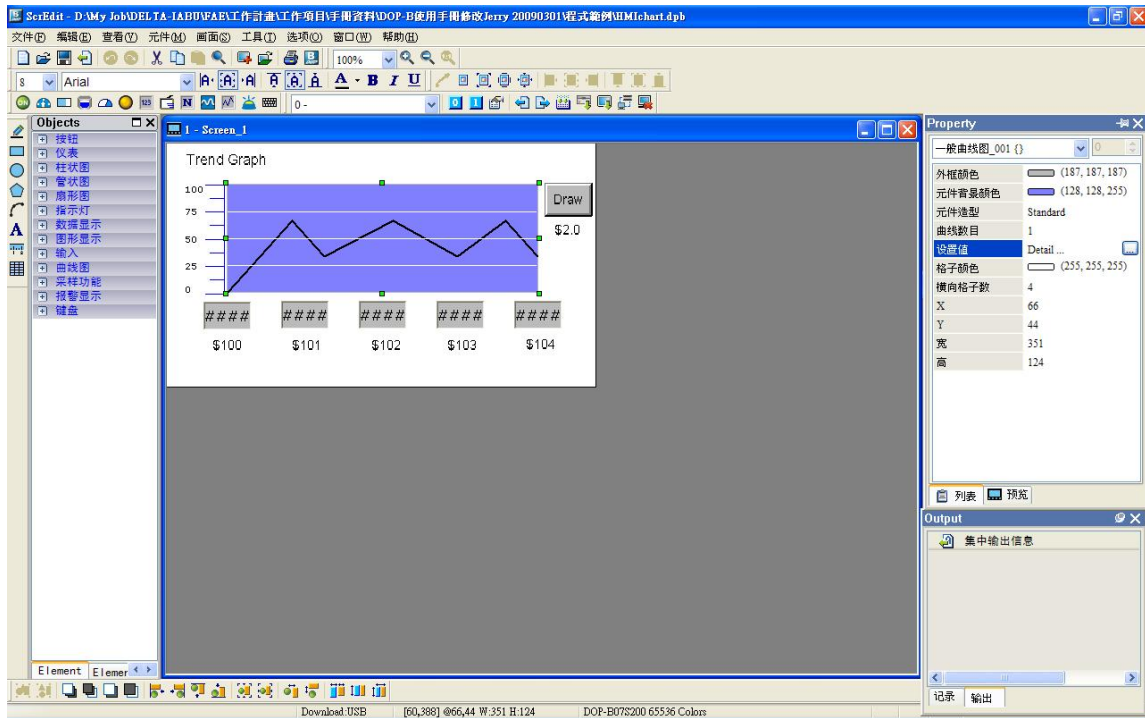
动作完成清除标志务必选取，因曲线取样标志再次触发前，必须将该标志清除为 Off。

人机画面中建立一一般曲线图元件，设定方式如下：



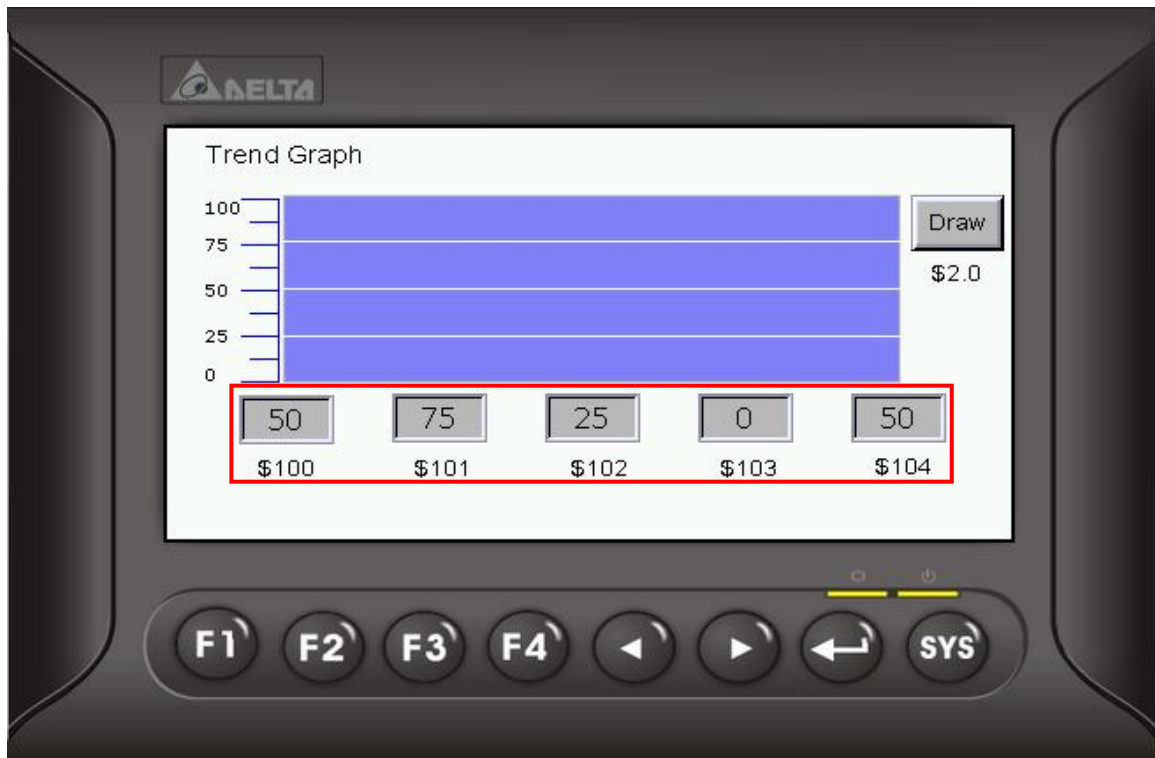
取样点数设为五点，读取位置设为人机内部寄存器\$100，因此取样点的寄存器位置分别是\$100、\$101、\$102、\$103 及\$104；取样标志设定为 1，代表由控制区\$2.0 去作触发取样。

此外，人机画面中需再建立五个数值输入元件，其功能是用来给定\$100~\$104 数值；建立一个 On 元件，用来触发曲线取样其标，如下图所示：

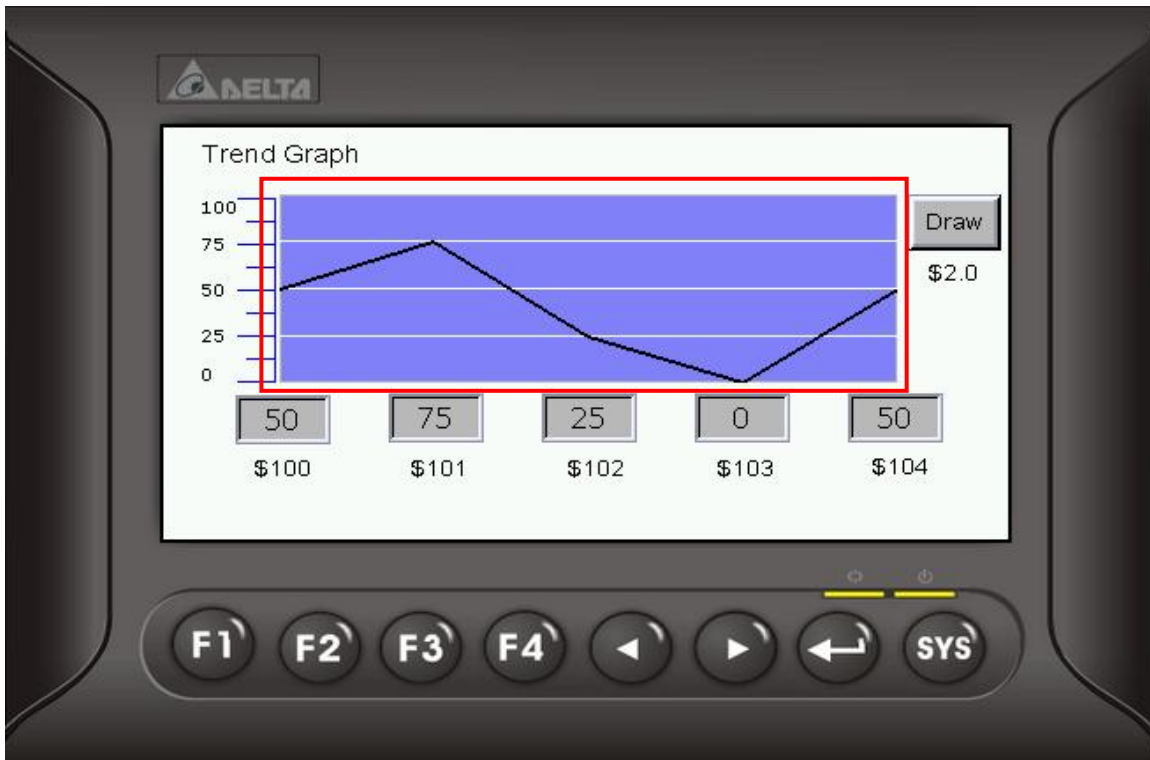


建立完成后进行编译动作，编译完成即可下载至人机进行操作。

程式执行画面如下，此时给定\$100~\$104 数值，如下图所示：



按下 Draw 来取样，结果如下图所示：



以上即为一般曲线图使用范例。

(此页有意留为空白)

第五章 DOP-B 系列系统画面相关说明

5.1 人机系统画面简介

➤ 进入系统目录

1. 按压 **SYS** 键约 2 秒 (系统键状态处于未被设置且有效的情况下)
2. 按压画面元件 - 系统目录按钮

进入后画面如下图:








➤ 退出系统目录

按压 **SYS** 键约 2 秒，即可退出。





















➤ 系统目录主画面操作方式






















1. 直接触碰屏幕上图示进入该项功能选项。
2. 按压实体键方式进入


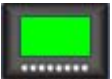







实体按键   可左右移动屏幕上选定图示功能的选项， 即进入选定的选项。

选定区底色如为白色  即代表未被选定。选定区底色如为黄色  代表已被选定。

➤ 系统目录架构展开图表

| | | | | | |
|--|--|---|---|--------------|--|
|  <p>System Setting 系统设置</p> |  <p>Touch Panel 触控面板</p> |  <p>TP Delay</p> | 设置触碰面板延迟时间 | | |
| | |  <p>TP Force</p> | 设置触碰面板按压力道 | | |
| | |  <p>TP Calibrate</p> | 触碰面板校正 | | |
| |  <p>Date/Time 日期/时间设置</p> |  <p>Date</p> | 设置日期 | | |
| | |  <p>Time</p> | 设置时间 | | |
| | |  <p>Alarm Clock</p> | 闹钟设置 (保留) | | |
| |  <p>Display LCD 显示设置</p> |  <p>Contrast</p> | 对比调整 (保留) | | |
| | |  <p>Brightness</p> | 亮度调整 (保留) | | |
| | |  <p>Gamma</p> | LCD Gamma 值调整 | | |
| |  <p>File Manager 文件管理</p> |  <p>Formatting 格式化</p> |  <p>HMI</p> | 人机格式化 | |
| | | |  <p>USB Disk</p> | U 盘格式化 (保留) | |
| | | |  <p>CF Card</p> | CF 卡格式化 (保留) | |
| | | |  <p>SD Card</p> | SD 卡格式化 (保留) | |
| | |  <p>Copy File</p> | 画面档复制 | | |

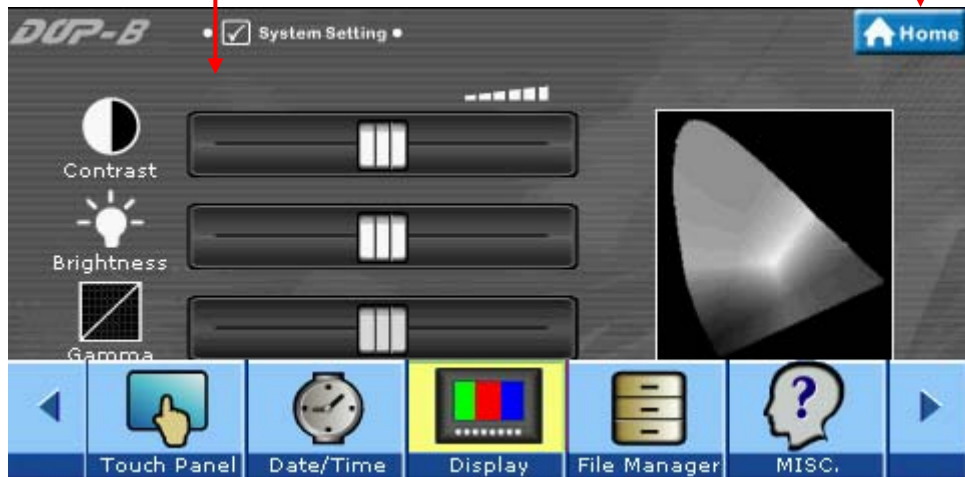
| | | | | |
|---|--|---|--------------------|--|
|  System Setting 系统设置 |  File Manager 文件管理 |  Multi-Screen File | 多重开机画面设置 | |
| |  MISC. | 杂项设置 | | |
| |  Security | 密码表设置 | | |
| |  Audio 音效设置 |  Volume | 音量调整设置（保留） | |
| | |  Buzzer | 蜂鸣器开关设置 | |
| |  COM Port | 通讯口设置 | | |
|  Up/Download 上/下载 |  Standard Mode 标准模式 |  COM1 | COM1 上/下载 | |
| |  Bypass Mode 旁通模式 |  COM2 | COM2 上/下载 | |
| | | Mode 1 | COM1 → COM2 Bypass | |
| | | Mode 2 | COM1 → COM3 Bypass | |
| | Mode 3 | COM2 → COM1 Bypass | | |
|  System Info | 系统信息 |  | 下一页 | |
| | |  | 上一页 | |
|  HMI Doctor 人机医生 |  | White Screen Test 白画面检测 | | |
| |  | Black Screen Test 黑画面检测 | | |
| |  | Red Screen Test 红画面检测 | | |

| | | |
|--|---|----------------------------|
|  <p>HMI Doctor 人机医生</p> |  | Green Screen Test 绿画面检测 |
| |  | Blue Screen Test 蓝画面检测 |
| |  | Draw Line Test 触碰面板画线检测 |
| |  | Buzzer/LED Test 蜂鸣器/LED 检测 |
| |  | ADC Test ADC 检测 |
| |  | USB Test USB 检测 |
| |  | Key Test 实体键检测 |
| |  | Color Screen Test 色阶饱和度检测 |

5.2 系统设置(System Setting)画面介绍

底图呈现灰阶部分为预览窗口，显示所选定项目的设置画面。直接点选灰色部分可直接进入设置。点选成功即变为蓝色画面。

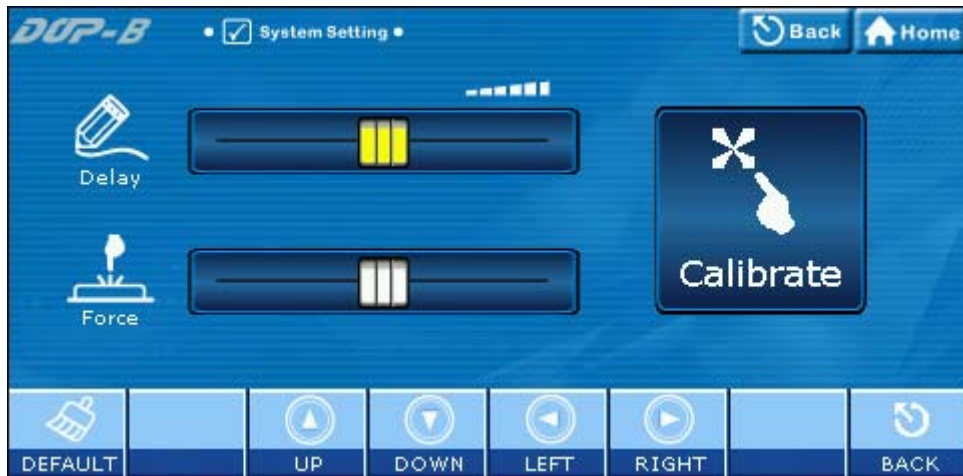
返回主系统目录画面









选定功能区域，彩色部分为设置选项列，按键   可移动选项选定。当底图呈现  黄色代表选定，再次点选该选项即可进入设置。当底图呈现  蓝色代表未选定。

1. Touch Panel (触碰面板)

- 触碰面板相关设置，包含延迟、力道、校正，画面如下：



此为功能列， 代表恢复为默认值；  代表变更设置选项；
  分别代表设置选项变更设置值； 代表退出此设置页。

- 实体键/功能列操作对应表

| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| DEFAULT | UP | DOWN | LEFT | RIGHT | BACK |
|  |  |  |  |  |  |
| F1 | F3 | F4 | | | SYS |


1.1 TP Delay – 触碰延迟

设置人机处理触碰移动信息的延迟时间。往右设置延迟缩小，代表速度加快。往左设置延迟变大，代表速度变慢。

1.2 TP Force – 触碰力道

设置人机触碰的力道大小。往左设置力道变小，代表人机认定触发力道较小因此变为较易触碰。往右设置力道变大，代表较难触碰。

1.3 Calibrate – 触碰校正

人机触碰面板 3 点校正，请依照指示触碰  中心点。

NOTE

若用户触碰校正位置远离屏幕上中心点的位置（如下图，中心点的位置），会引起触碰面板较大的误差偏移，因此建议用户使用触控专用笔来进行屏幕位置校正。



画面如下：






2. Date/Time（日期/时间设置）

- 设置人机日期/时间，设置选项有年/月/日/时/分/秒，所选定的设置以红色字体表示，画面如下：



此为功能列，  代表移动年/月/日/时/分/秒设置选项；


  分别代表设置选项变更设置值； 代表取消设置；

 代表退出此设置页。


➤ 实体键/功能列操作对应表

| | | | | | |
|---|---|---|---|--|---|
|  CANCEL |  UP |  DOWN |  LEFT |  RIGHT |  BACK |
|  |  |  |  |  |  |


2.1  Date – 日期设置

 依序为年/月/日设置，触碰上/下键修改设置值。

2.2  Time – 时间设置







 依序为时/分/秒设置，触碰上/下键修改设置值。

2.3  Alarm Clock – 闹钟设置（保留）

3.  Display (LCD 显示设置)


➤ 设置人机 LCD 相关设置值，当设置项处于黑色，代表此设置项无效/保留，画面如下：



此为功能列， 代表恢复为 LCD 默认值； 代表变更设置选项； 分别代表设置选项变更设置值； 代表退出此设置页。

➤ 实体键/功能列操作对应表

| | | | | | |
|---|--|--|--|---|--|
|  DEFAULT |  UP |  DOWN |  LEFT |  RIGHT |  BACK |
|  F1 |  F3 |  F4 |  |  |  SYS |

3.1  Contrast – 对比调整（保留）






3.2  Brightness – 亮度调整（保留）

3.3  Gamma 调整

4.  File Manager（文件管理）

➤ 人机内部文件磁盘管理，画面如下：



此为功能列,   代表变更设置选项;  等同点选的动作;
 代表退出此设置页;  黄色代表选定。

➤ 实体键/功能列操作对应表







| | | | |
|---|--|--|---|
|  LEFT |  RIGHT |  CLICK |  BACK |
|  |  |  |  |

4.1  Formatting – 格式化




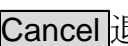
此功能为人机对内部/外部磁盘做格式化，若磁盘图示设置选项处于黑色，代表此设置项无效或保留，画面如下：




➤ 实体键/功能列操作对应表

| | | | | |
|--|--|---|---|---|
|  HMI |  USB |  CF |  SD |  BACK |
|  |  |  |  |  |

4.1.1  HMI -人机格式化

点选  或按下 ，人机会显示确认窗口，若确定格式化，点选  直接进行人机内部磁盘格式化，点选  退出。

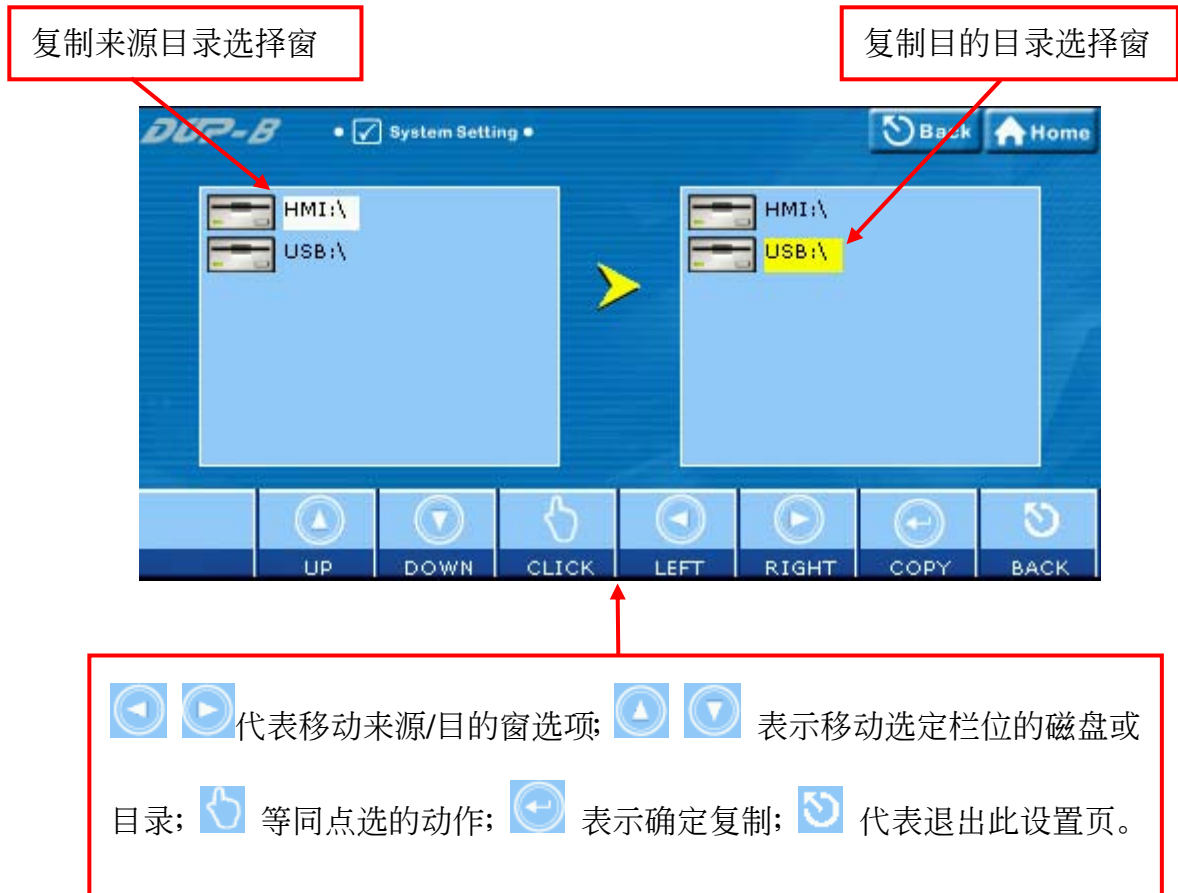
4.1.2  USB Disk –U 盘格式化（保留）

4.1.3  CF Card -CF 卡格式化（保留）

4.1.4  SD Card -SD 卡格式化（保留）

4.2 Copy File – 书面文件复制

将人机画面档做内/外磁盘之间的复制，进入时需输入最高权限密码，画面如下：



➤ 实体键/功能列操作对应表

| | | | | | | |
|---|---|---|---|--|---|---|
|  |  |  |  |  |  |  |
| UP | DOWN | CLICK | LEFT | RIGHT | COPY | BACK |
|  |  |  |  |  |  |  |
| F2 | F3 | F4 | | | | SYS |

➤ 注意事项

- 人机不支持磁盘直接复制。
- 人机只支持固定目录 HMI-000 ~ HMI255。
- 目的目录会先移除后，再将来源目录复制过去。
- 若目的目录选择为 **New...**，则人机会先从 HMI-000 至 HMI-255，找出不存在的目录，建立为目的目录。
- 若来源目录的画面文件有做密码保护，人机会出现密码输入，请输入来源目录画面文件的最高权限，方可复制。

4.3 Multi-Screen File –多重开机画面设置

设置人机内/外磁盘内的画面文件为开机画面文件，画面如下：



➤ 实体键/功能列操作对应表

| | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| CLICK | UP | DOWN | ENTER | BACK |
|  |  |  |  |  |
| F4 | ← | → | ↵ | SYS |

➤ 注意事项

- 人机只支持固定目录 HMI-000 ~ HMI255。
- 若开机时发现开机目录不存在，人机会直接开启内部画面文件，不会修改开机目录的路径，因此待下次开机时若发现开机目录又存在了，就会以它为开机目录。
- 若开机目录为外部磁区，则此画面的断电保持数据不论是否设为内部或外部磁区，一律会自动存于此开机目录内。

5. MISC (杂项设置)

➤ 杂项设置，包含：

- 触碰光标
- 屏幕保护时间
- 开机延迟时间
- 默认语言选择
- 蓝光 LED 闪烁功能选择
- 默认开机画面

画面如下：

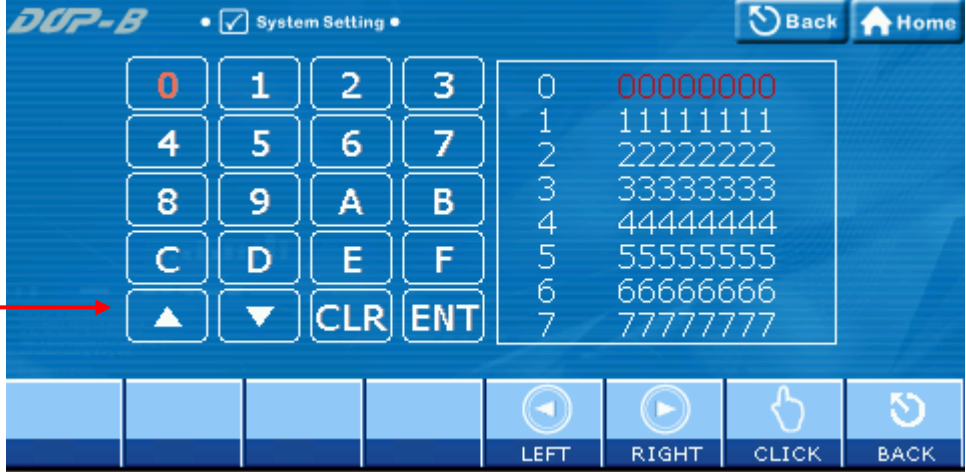




➤ 实体键/功能列操作对应表





| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| DEFAULT | UP | DOWN | LEFT | RIGHT | BACK |
|  |  |  |  |  |  |
| F1 | F3 | F4 | | | SYS |

6. Security (密码表设置)

- 设置 0~7 等级密码，进入此设置会显示密码输入，系统会依照所输入的密码等级，只开放所输入等级的密码修改权限。若输入为最高等级密码则开放 0~7 所有等级密码的修改权限。画面如下：



  表示移动所要设置的等级密码

  代表移动选定的按钮；
 等同按下选定的按钮；  代表退出此设置页。

- 实体键/功能列操作对应表

| | | | |
|---|---|---|---|
|  |  |  |  |
| LEFT | RIGHT | CLICK | BACK |
|  |  |  |  |

7. Audio (音效设置)

➤ 人机 Audio 相关设置，画面如下：



此为功能列， 代表恢复 Audio 默认值；  代表变更设置选项；  分别代表设置选项变更设置值； 代表退出此设置页。

➤ 实体键/功能列操作对应表

| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| DEFAULT | UP | DOWN | LEFT | RIGHT | BACK |
|  |  |  |  |  |  |
| F1 | F3 | F4 | | | SYS |

7.1 Volume – 音量调整设置

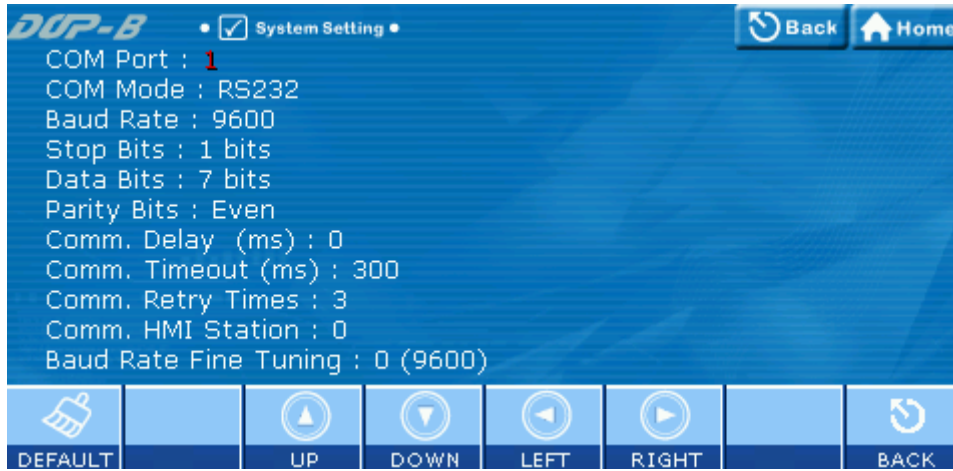
往左设置音量越小，往右设置音量越大







7.2 Buzzer – 蜂鸣器开关设置

往左设置音量越小，往右设置音量越大

8. COM Port (通讯口设置)

➤ 人机通讯口协议等相关设置，画面设置如下：



此为功能列， 代表恢复此通讯口为默认值；  代表变更设置选项；  分别代表设置选项变更设置值； 代表退出此设置页。

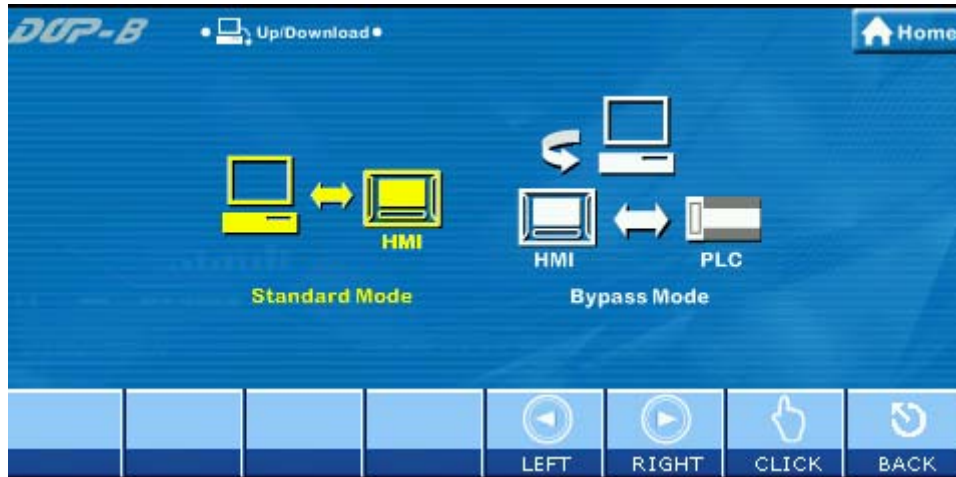
➤ 实体键/功能列操作对应表

| | | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| DEFAULT | UP | DOWN | LEFT | RIGHT | BACK |
|  |  |  |  |  |  |
| F1 | F3 | F4 | | | SYS |

5.3 上/下载(Upload/Download)画面介绍

- 设置 **COM Port** 作为与 **PC** 沟通协议用，共有两种沟通协议模式：
 1. 标准上/下载模式
 2. 旁通模式

设置画面如下：

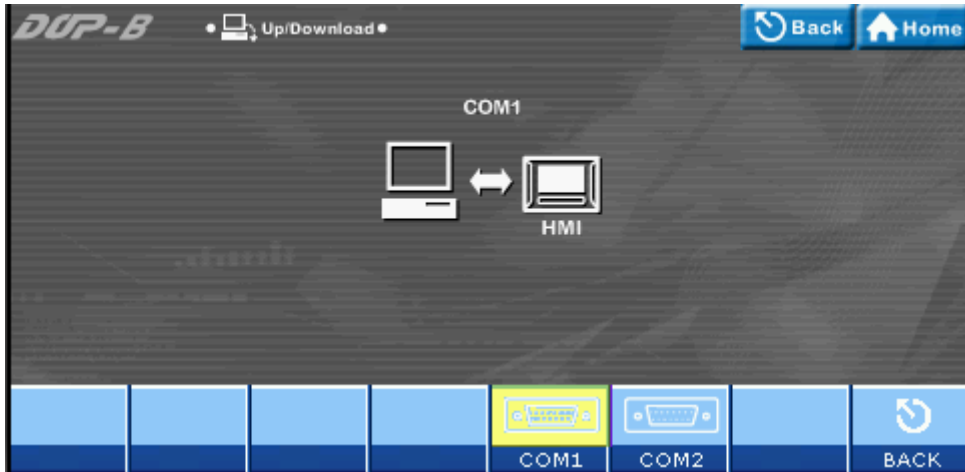


- 实体键/功能列操作对应表







| | | | |
|--|---|---|--|
|  LEFT |  RIGHT |  CLICK |  BACK |
|  |  |  |  |

1.  **Standard Mode (标准上/下载模式)**

- 将通讯口设为与 **ScrEditor** 沟通协议的设置, 并且等待 **ScrEditor** 送出动作指令及数据封包, 进行上/下载。共有 **COM1/COM2** 两个通讯口可做选择, 设置画面如下:



- 实体键/功能列操作对应表

| | | |
|--|--|--|
|  COM1 |  COM2 |  BACK |
|  F3 |  F4 |  SYS |

1.1  **COM1 (COM1 上/下载)**

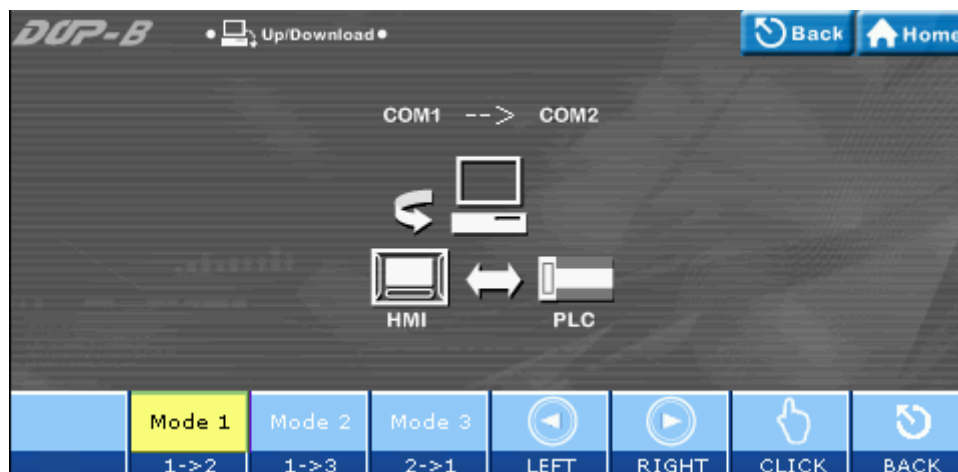
将 COM1 通讯口设置为收送 ScrEditor 的上/下载指令数据及数据封包的模式。

1.2  **COM2 (COM2 上/下载)**



将 COM2 通讯口设置为收送 ScrEditor 的上/下载指令数据及数据封包的模式。

2. Bypass Mode -旁通模式

- 将人机作为中介者，人机会将来源通讯口所收到的数据往通讯目的口送出。



- 实体键/功能列操作对应表

| | | | | | | |
|--|--|--|---|---|--|---|
|  Mode 1 1 → 2 |  Mode 2 1 → 3 |  Mode 3 2 → 1 |  LEFT |  RIGHT |  ENTER |  BACK |
|  |  |  |  |  |  |  |

2.1 Mode 1 – COM1 → COM2 旁通模式

COM1 为来源通讯口，COM2 为通讯目的口，将 COM1 所收到的数据以 COM2 通讯口的协议送出。

2.2 Mode 2 – COM1 → COM3 旁通模式

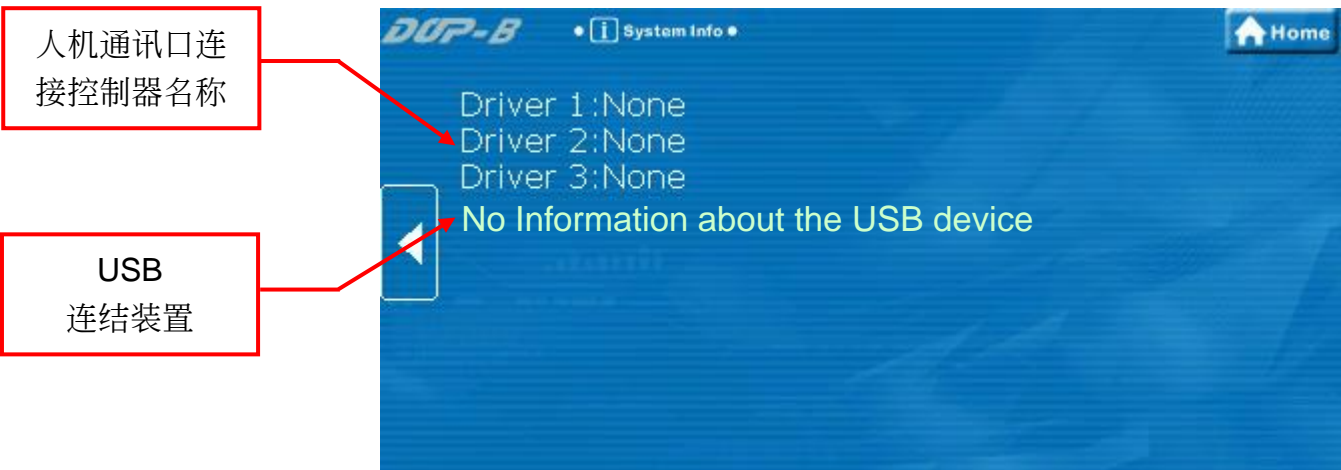
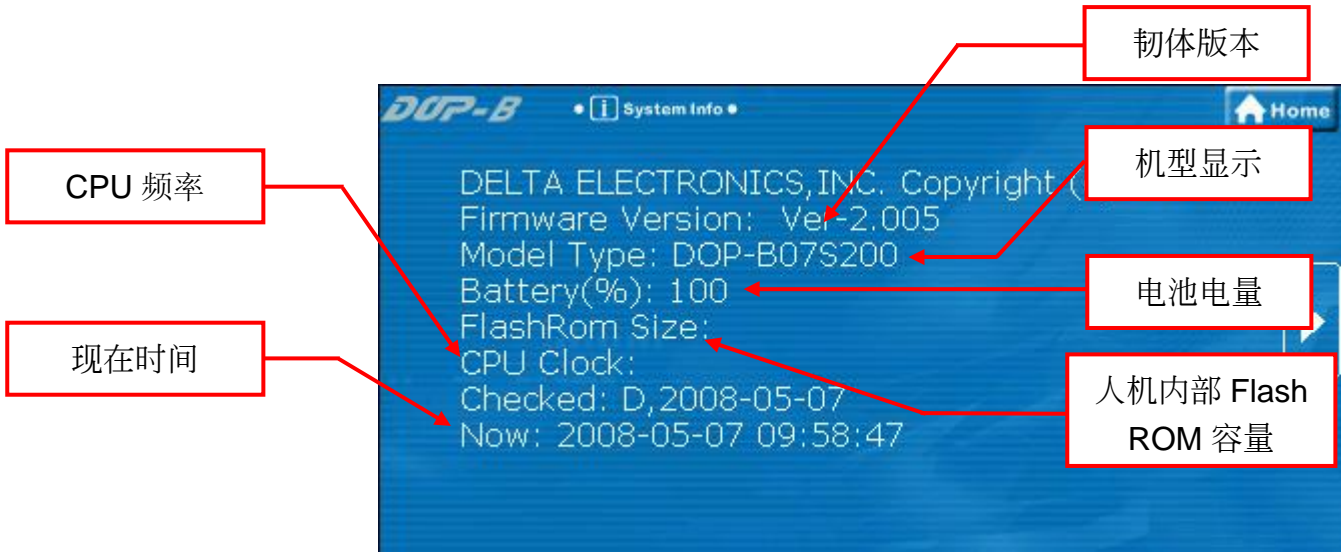
COM1 为来源通讯口，COM3 为通讯目的口，将 COM1 所收到的数据以 COM3 通讯口的协议送出。

2.3 Mode 3 – COM2 → COM1 旁通模式

COM2 为来源通讯口，COM1 为通讯目的口，将 COM2 所收到的数据以 COM1 通讯口的协议送出。

5.4 系统新息(System Info)画面介绍

➤ 显示人机相关信息，画面如下：



➤ 实体键/功能列操作对应表

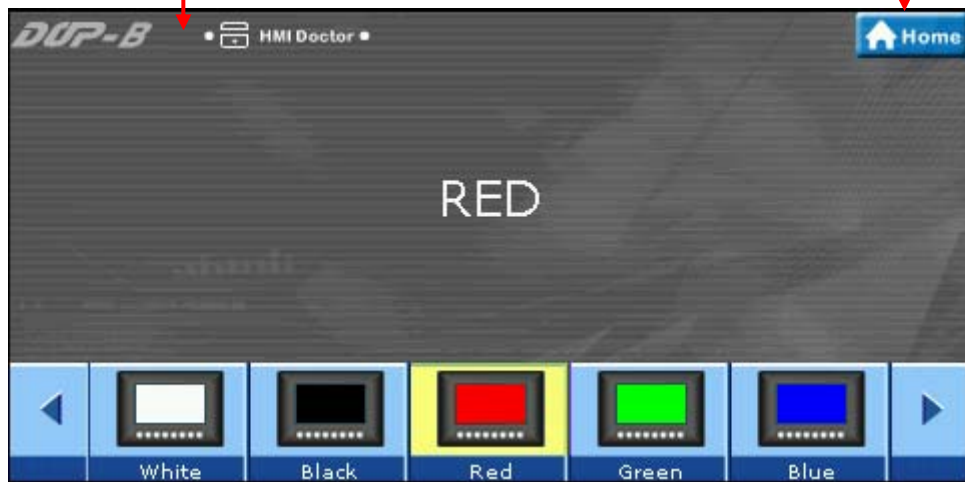
| | | |
|---|---|---|
| 上一页 | 下一页 | 退出 |
|  |  |  |

5.5 人机医生(HMI Doctor)画面介绍


人机医生为提供使用者测试硬件界面的简易测试程式，目前提供有 LCD、Touch Panel、Buzzer、LED、USB、ADC Channel、Hot Key 等测试选项。画面如下：

底图呈现灰阶部分为预览窗口，显示所选定项目的设置画面。直接点选此部分可直接进入设置。

返回主系统目录画面



选定功能区域，彩色部分为设置选项列。

当底图呈现  黄色代表选定，再次点选该选项即可进入设置。

当底图呈现  蓝色代表未选定，  可移动选项选定。

➤ 实体键操作:

  代表可左右移动选项， 即为进入选定的选项， 代表人机医生系统设置。

1.  **White Screen Test (白画面检测)**

- 检测 LCD 显示是否有暗点及类似表面不洁的污点产生。

2.  **Black Screen Test (黑画面检测)**

- 检测 LCD 显示是否有红色、蓝色、绿色或白色的亮点

3.  **Red Screen Test (红画面检测)**

- 检测 LCD 是否有红色不亮点及类似表面不洁的污点产生。

4.  **Green Screen Test (绿画面检测)**

- 检测 LCD 是否有绿色不亮点及类似表面不洁的污点产生。

5.  **Blue Screen Test (蓝画面检测)**

- 检测 LCD 是否有蓝色不亮点及类似表面不洁的污点产生。

6.  **Draw Line Test (触碰面板划线检测)**

- 检测下笔画线的位置是否与屏幕显示的位置有偏差，若偏差太大，请重新校正触碰面板。

7.  **Buzzer/LED Test (蜂鸣器/LED 检测)**

- 检测蜂鸣器是否鸣叫。
- 检测红/蓝/绿三色 LED 是否点亮。

8.  **ADC Test -ADC 检测**

- 检测 ADC 各通道转换是否正确，包含按压力道、触碰 XY、系统电压、温度等检测。

9.  **USB Test (USB 检测)**

- 检测 USB 是否正确。测试时请将 USB Device 及 USB Host 相互对接。

10. Key Test (实体键检测)

- 检测硬件键是否可正常按压。进入此测试请直接按压实体键  ~  则画面上相对应的 KEY 就会显示由 UP → DOWN，若按压无反应及代表此硬件键损毁。

11. Color Screen Test (色阶饱和度检测)

- 色阶及饱和度检测

(此页有意留为空白)

附录 A 新增功能说明

A.1 增加宏指令 IMRCP16/32、EXRCP16/32 配方汇入汇出功能

| | |
|----------------------------|--|
| Var1 = EXRCP16(Var2, Var3) | 汇出 16 位配方 |
| Var1 = IMRCP16(Var2, Var3) | 汇入 16 位配方 |
| Var1 = EXRCP32(Var2, Var3) | 汇出 32 位配方 |
| Var1 = IMRCP32(Var2, Var3) | 汇入 32 位配方 |
| 参数说明 | |
| Var1 | 动作回传值，动作成功为 1，动作失败为 0； |
| Var2 | 汇入或汇出的文件名称； |
| Var3 | CSV 文件汇入或汇出的位置： 2 为 U 盘 3 为 SD 卡 |

以下举汇出 16 位配方 (EXRCP16) 为例，首先建立 16 位配方表，如下图所示：

配方设置

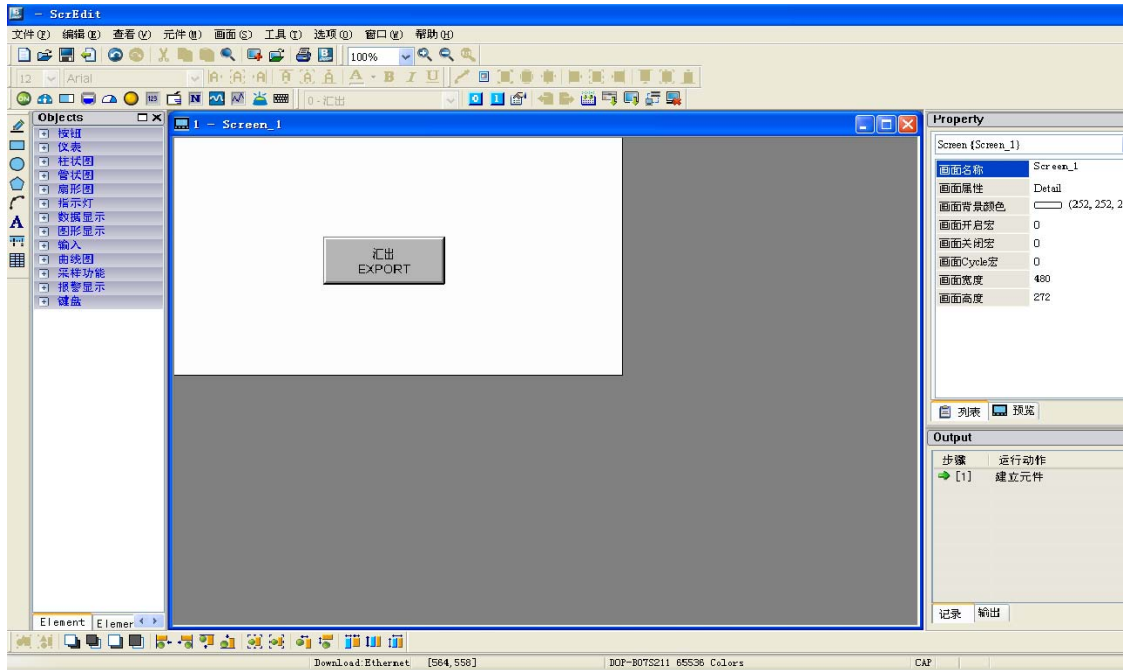
启动配方功能 配方断电保持区 HMI

地址 \$500 长度 2 组数 2 配置

| | W1 | W2 |
|---|----|----|
| 1 | 1 | 2 |
| 2 | 3 | 4 |

开启
保存
重置
清除
打印
确定
取消

画面中建立一交替型按钮，如下图所示：



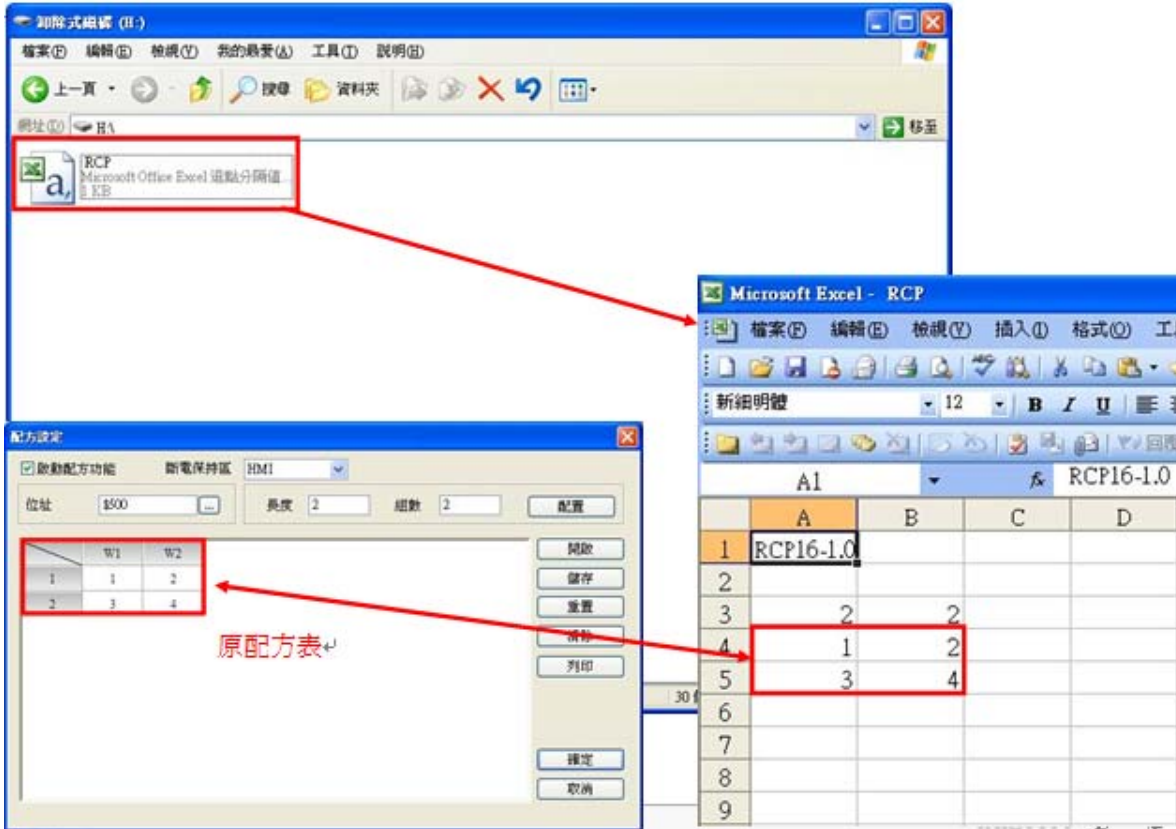
在按钮 On 宏编辑以下宏

| | |
|----------------------------|-----------------------|
| FILLASC(\$I00, "rcp") | 设定汇出配方文件名称为 rcp |
| \$6789 = EXRCP16(\$I00, 2) | 将配方汇出到 U 盘，汇出文件名为 rcp |

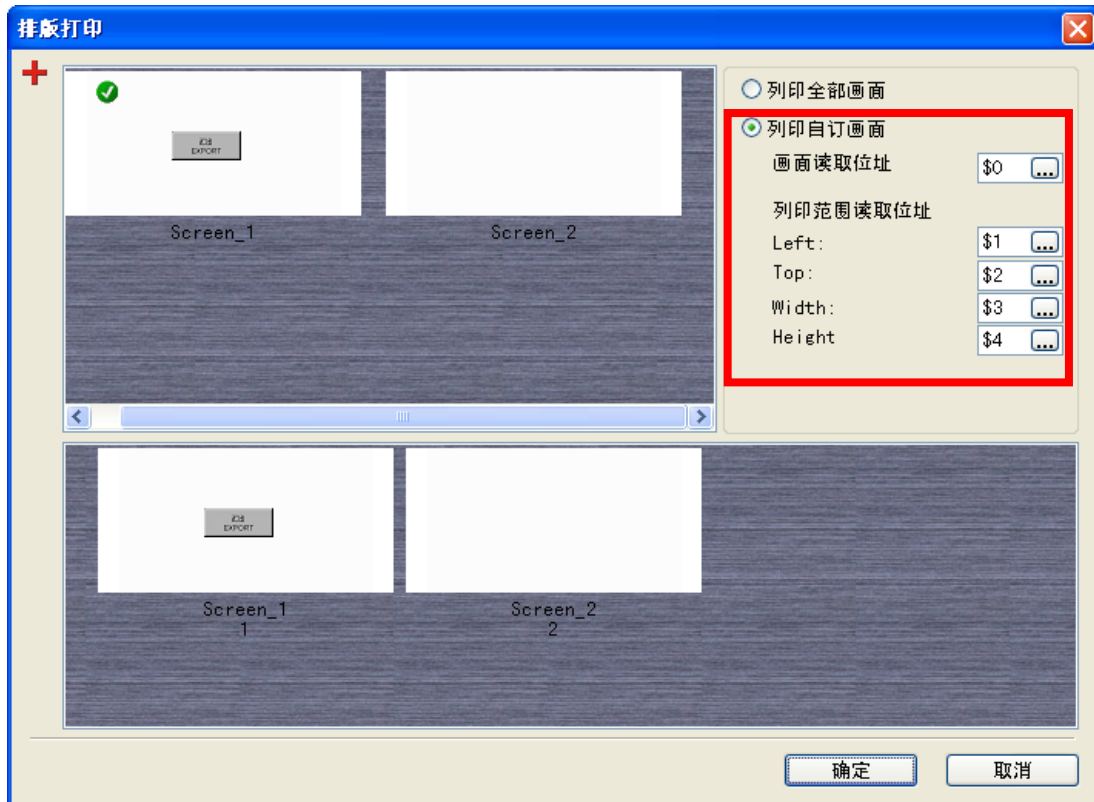
执行结果如下，若按下此交替型按钮，配方表将立即汇出至 U 盘：



汇出在 U 盘的配方文件：



A.2 打印排版管理—新增可指定打印画面页面及打印范围功能



以上图为例，各寄存器地址对应功能如下表：

| | | |
|-----|------------------|---|
| \$0 | 指定打印画面页码 | 可各别指定设定于”排版打印”中的单一页面，当输入数值为 0 时，则会打印设定在”排版打印”中的全部画面 |
| \$1 | 打印范围起点 X 座标 | 设定要打印的范围起点 X 座标 |
| \$2 | 打印范围起点 Y 座标 | 设定要打印的范围起点 Y 座标 |
| \$3 | 打印范围宽度(单位:pixel) | 设定打印范围宽度 |
| \$4 | 打印范围高度(单位:pixel) | 设定打印范围高度 |

本新增功能可搭配控制区打印标志位、输出报表按钮元件、输出报表宏使用。

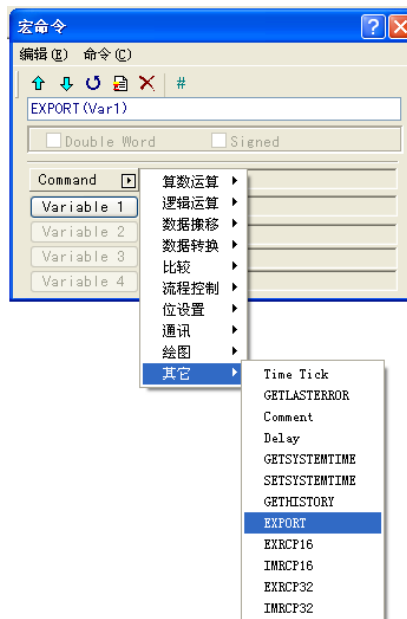
A.3 新增触碰校正按钮元件

通过此元件不需进入系统菜单，即可直接执行触控校正功能



A.4 新增 EXPORT 宏指令

EXPORT 宏指令与输出报表按钮元件功能相同，以宏方式执行，可搭配宏条件判断进行更弹性的应用。使用方式如下：



| | | |
|---------------|-----------|-------------------|
| EXPORT (Var1) | | Var1 可为寄存器地址或是常数值 |
| Var1 = 0 | 输出数据至 SD | |
| Var1 = 1 | 输出数据至 U 盘 | |
| Var1 = 2 | 打印资料到打印机 | |

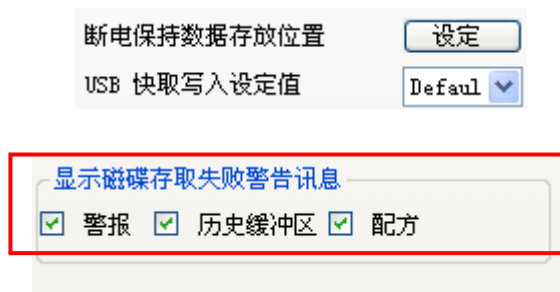
A.5 新增 STATIONON/STATIONOFF 宏指令

通过 STATIONON/STATIONOFF 宏指令可各别开启/关闭指定站号的通讯。使用方式如下：

| | | |
|-------------------------|----------------------|-------------------------|
| STATIONON (Var1, Var2) | | Var1, Var2 可为寄存器地址或是常数值 |
| STATIONOFF (Var1, Var2) | | |
| Var1 | 要开启或关闭通讯的 COM Port: | |
| | COM1 = 0 | |
| | COM2 = 1 COM3 = 2 | |
| Var2 | 要开启或关闭通讯的站号 | |

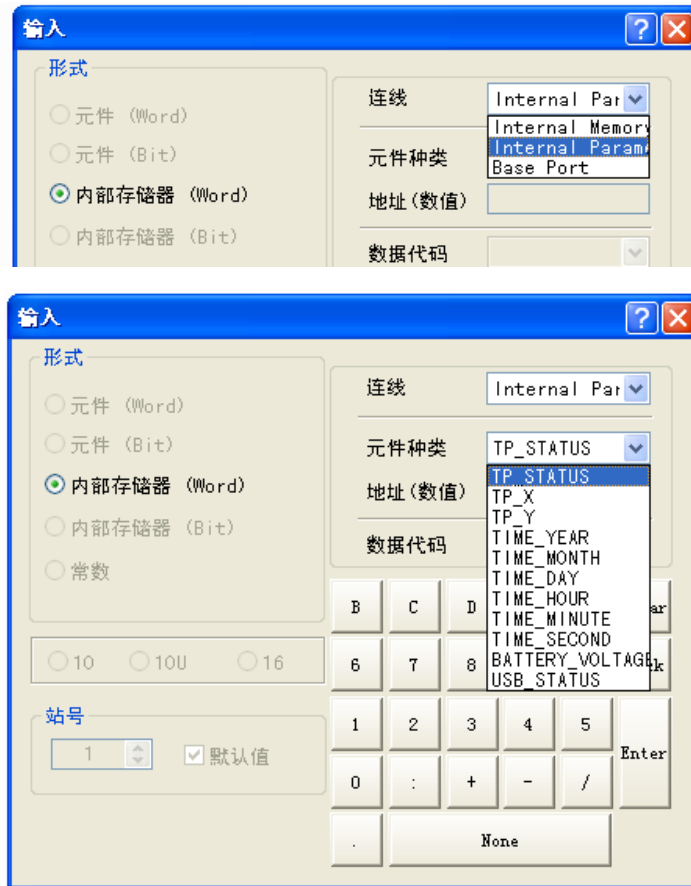
A.6 新增 U 盘报警选项

使用 U 盘储存数据(警报、历史缓冲区、配方)，若 HMI 运行中，当 U 盘被移除时，HMI 会跳出存取失败的信息。通过这个选项，可以取消显示存取失败的信息。



A.7 新增 HMI 内部系统参数

详细内部系统参数说明如下：



| 参数名称 | 参数说明 | 参数属性 |
|-----------------|--|--------------|
| TP_Status | 触碰状态 | Read only |
| TP_X | 触碰 X 坐标 | Read only |
| TP_Y | 触碰 Y 坐标 | Read only |
| Time_Year | 年 | Read / Write |
| Time_Month | 月 | Read / Write |
| Time_Day | 日 | Read / Write |
| Time_Hour | 时 | Read / Write |
| Time_Minute | 分 | Read / Write |
| Time_Second | 秒 | Read / Write |
| Battery_Voltage | 内存备份电池剩余电压百分比 | Read only |
| NET_IP1 | HMI IP 地址 | Read only |
| NET_IP2 | 以 IP 地址 192.168.0.1 为例： NET_IP1 = 192 | Read only |
| NET_IP3 | NET_IP2 = 168 | Read only |

| | | |
|---------|----------------------------|-----------|
| NET_IP4 | NET_IP3 = 0 NET_IP4 = 1 | Read only |
|---------|----------------------------|-----------|

A.8 新增不可见位属性

使用此位可控制元件是否隐藏。支持的元件如下：

| | |
|--|---|
| | <p>按钮元件</p> <p>状态指示灯</p> <p>数值输入</p> <p>文数字输入</p> <p>Barcode 输入</p> |
|--|---|

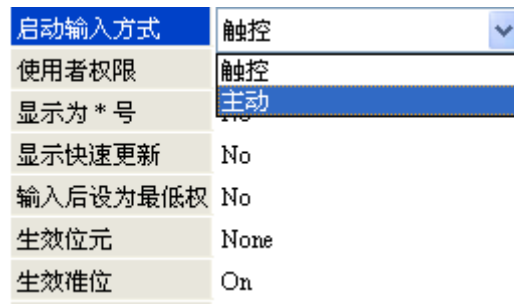
A.9 新增 Barcode 输入元件

新增 Barcode 输入元件，可透过 USB 界面接收 Barcode Reader 所扫描的 Barcode。Barcode 输入元件主要分为「触控」、「主动」两种输入方式：

| | |
|----------|------|
| 启动输入方式 | 触控 |
| 使用者权限 | 触控 |
| 显示为 * 号 | 主动 |
| 显示快速更新 | No |
| 输入后设为最低权 | No |
| 生效位元 | None |
| 生效准位 | On |

使用「触控」输入方式：需先触控画面上的 Barcode 输入元件，Barcode 输入元件就会进入等待 Barcode 输入的状态，再触碰一次画面上的 Barcode 输入元件即可解除等待 Barcode 输入的状态。当画面上有超过一个以上的 Barcode 输入元件，当第一个 Barcode 输入元件接收到 Barcode 后，并不会自动跳到下一个 Barcode 输入元件，需要用户自行触碰另外一个 Barcode 输入元件。

而当需要手动修正 Barcode 数据时，则需要设定弹跳启动位，当需要手动修正 Barcode 输入元件数据时，则需要触发弹跳启动位，画面上就会跳出文数字输入键盘供用户手动输入。

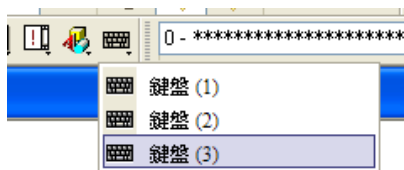


使用「主动」输入方式：

需先触发 Barcode 输入元件的生效位，Barcode 输入元件就会进入等待 Barcode 输入的状态，当画面上有超过一个以上的 Barcode 输入元件，当第一个 Barcode 输入元件接收到 Barcode 后，会自动跳到下一个使用相同生效位的 Barcode 输入元件等待 Barcode 输入。重置 Barcode 输入元件的生效位即可解除等待 Barcode 输入的状态。

使用「主动」输入方式时，如需要手动修正 Barcode 资料时，则需要先在画面上制作一个文数字键盘元件，当触发 Barcode 输入元件的生效位时，就可同时透过文数字键盘元件输入资料。

而当使用「主动」输入方式时，弹跳启动位无效的！



A.10 新增韧体更新后 HMI 自动重开机功能

勾选此选项，则更新韧体后将自动重开机而不显示确认视窗。

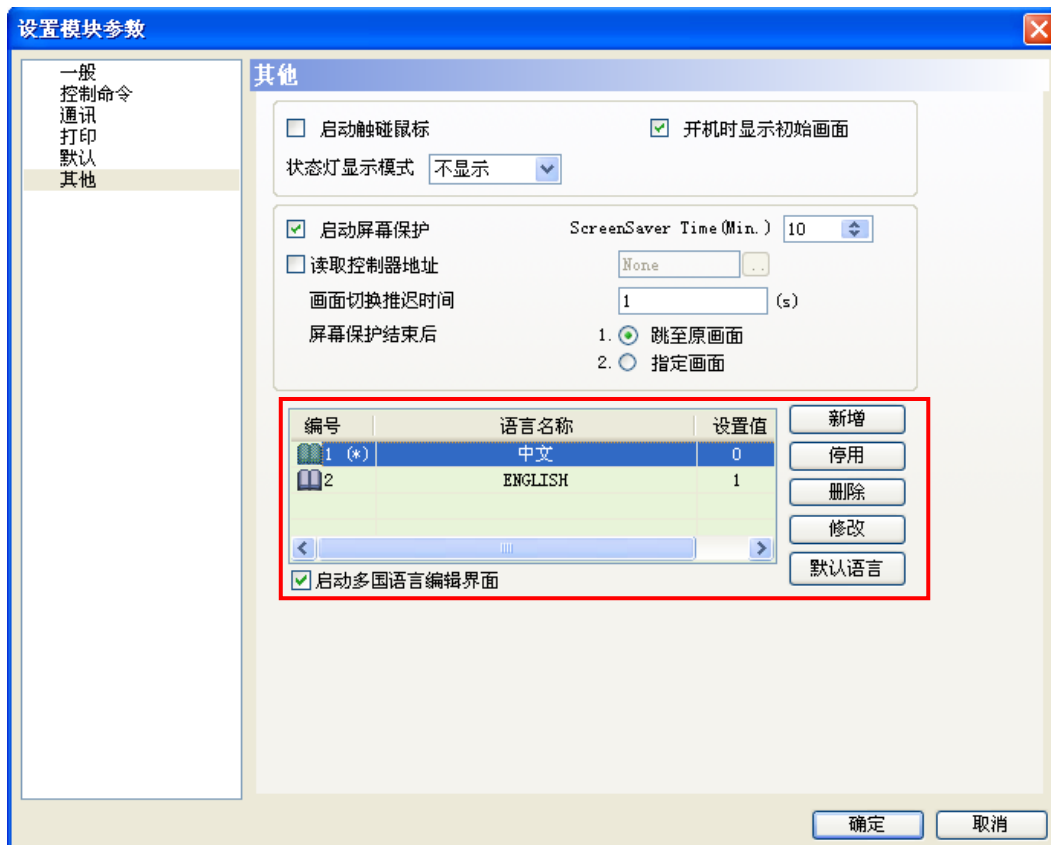


A.11 新增语系切换元件

使用此语系切换元件即可省去设定控制区相关步骤。



首先在[选项]/[设置模块参数]/[其他]设定多国语言，如下图所示：

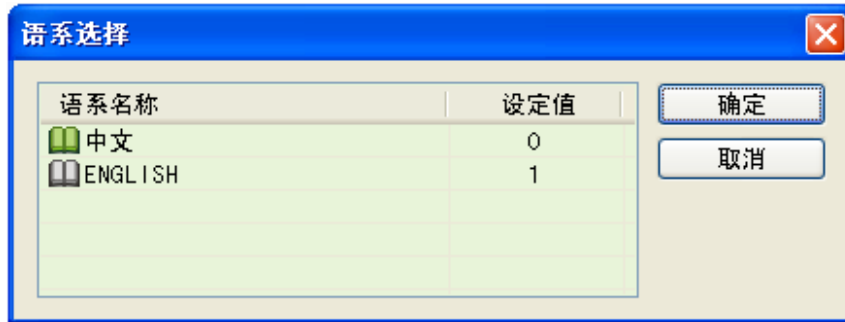


设定完成后，在人机画面上制作三元件：

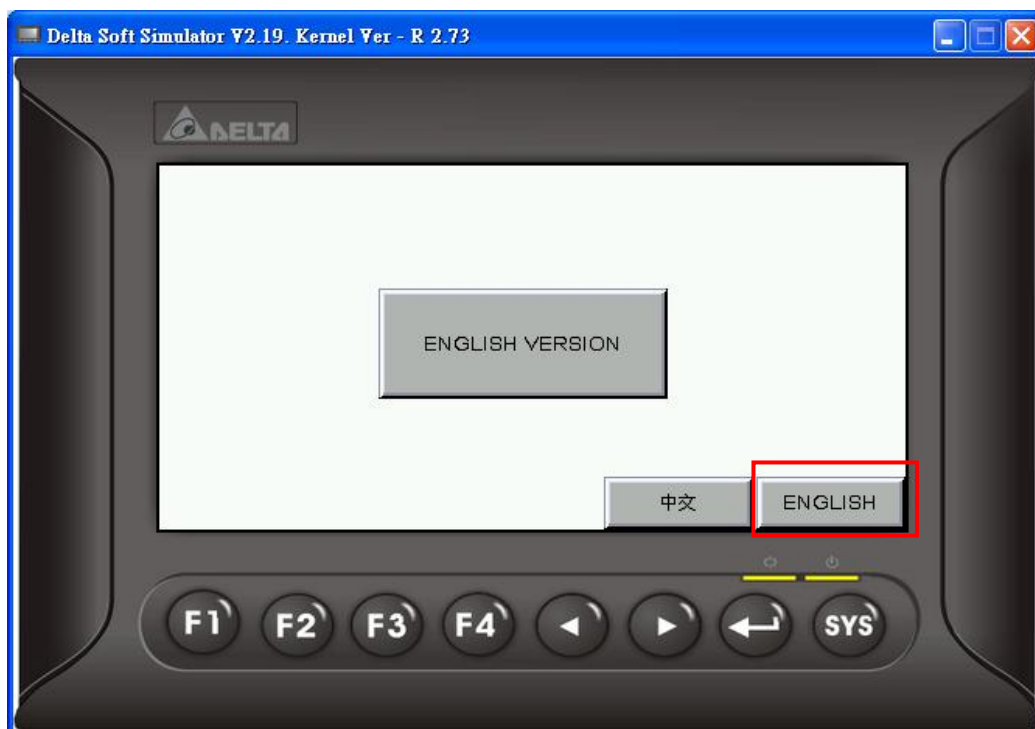
元件一：新增一交替型按钮，在属性表文字上分别输入中文版以及 ENGLISH VERSION



元件二：新增一语系切换元件，语系名称设定为中文



元件三：新增一语系切换元件，同上方式，语系名称设定为 ENGLISH
以上即完成设定，执行结果如下表所示：



按下 ENGLISH 键

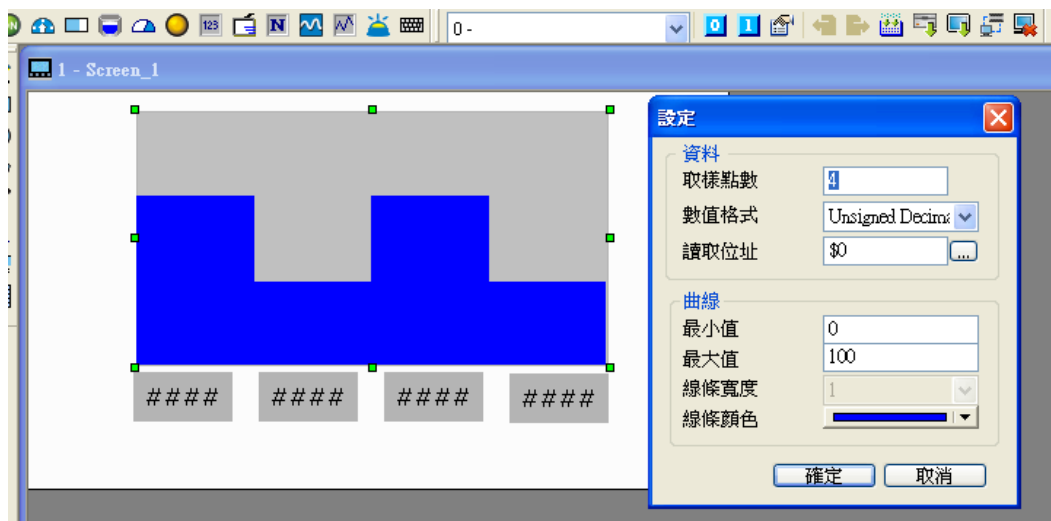


按下中文键

A.12 新增图表输入元件

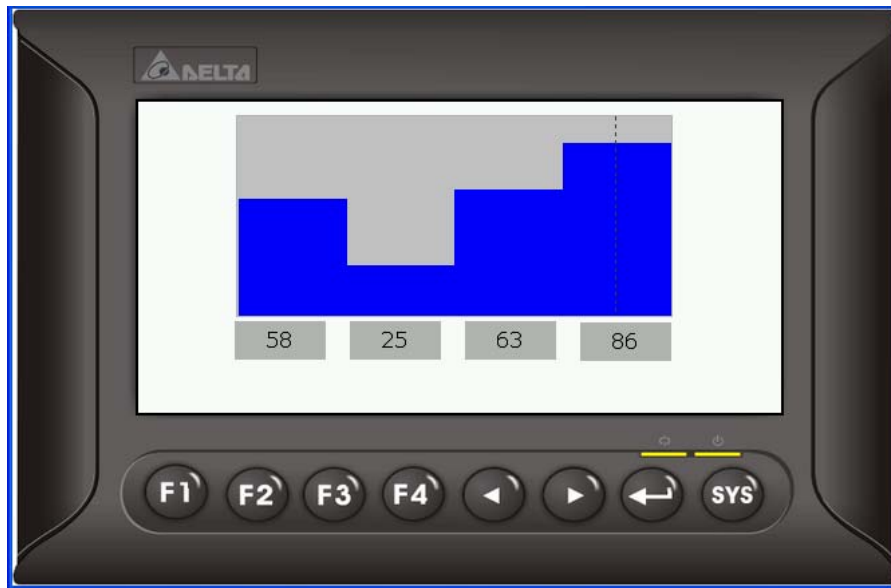


在画面中建立一图表输入元件，取样点数设为 4 点，读取位置为 \$0，则取样点数的位置分别为 \$0、\$1、\$2 及 \$3：



在人机画面中建立四个数值输入元件，内存位置分别为 \$0、\$1、\$2 及 \$3，如上图所示。如此即完成设定，编译完即可下载至人机。

用户可在图表上直接点选所需数值，也可使用数值输入来输入数值。

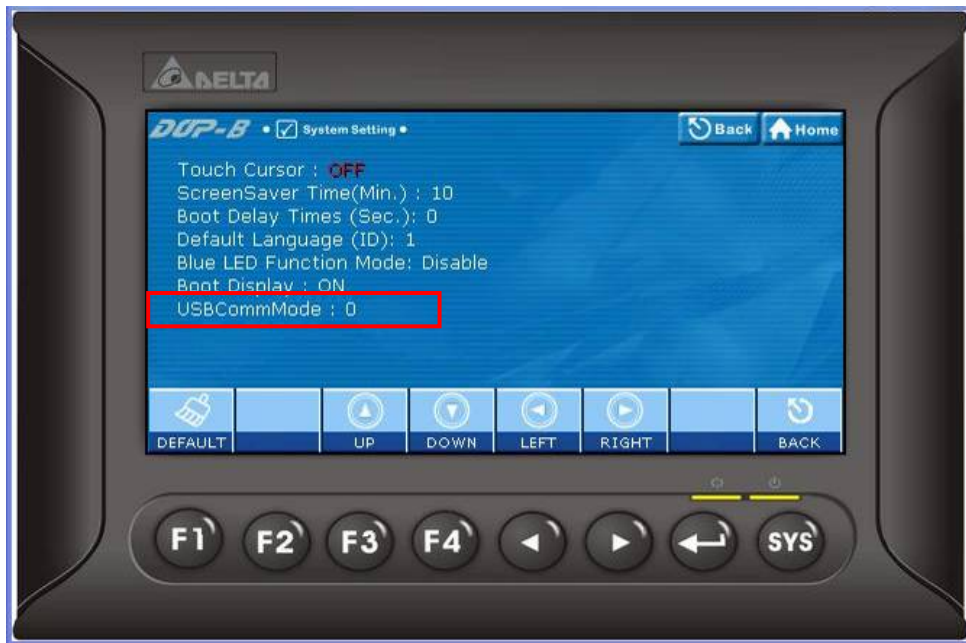


A.13 新增 U 盘上下下载传输模式

新增 U 盘上下下载传输模式，更新韧体后，进入人机系统画面后，进入 Setting 设定，将 USBCommMode 设定值更改为 1，之后再 HMI 重新连接到电脑的 USB 端口。



选择 MISC.



选择 USBCommMode

其中 USBCommMode 为 0 时为 USB DEVICE 传统上下载模式，USBCommMode 为 1 时为 U 盘上下载模式。之后再将 HMI 重新连接到电脑的 USB 端口。在电脑端“我的电脑”中就可以看到名为 DELTA 的”移动式存储设备”，此时即可在 Screen Editor 中下载。切换为 U 盘模式可兼容 Windows XP / Windows Vista / Windows 7 操作系统。




附錄B DVP PLC 程序上下載功能說明

B.1 新增 DVP PLC 程序上下載功能

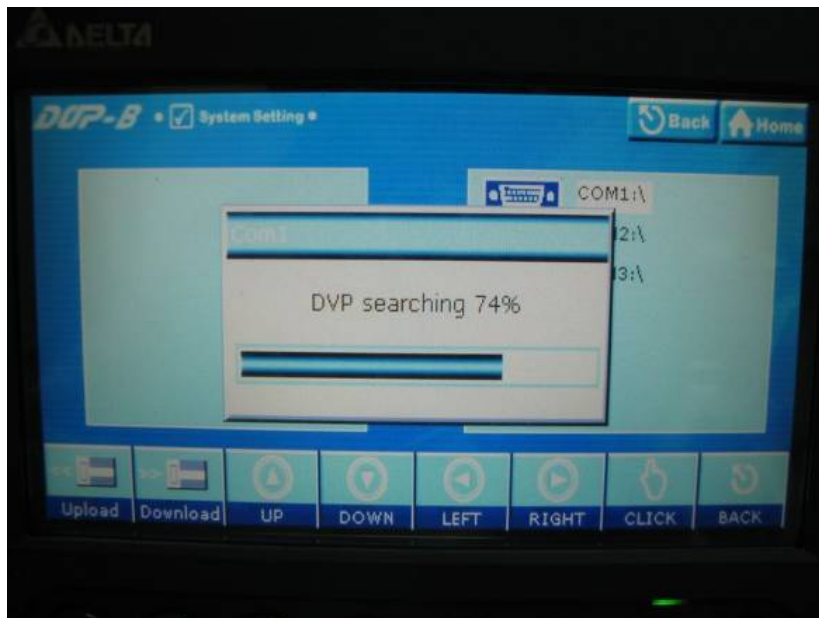
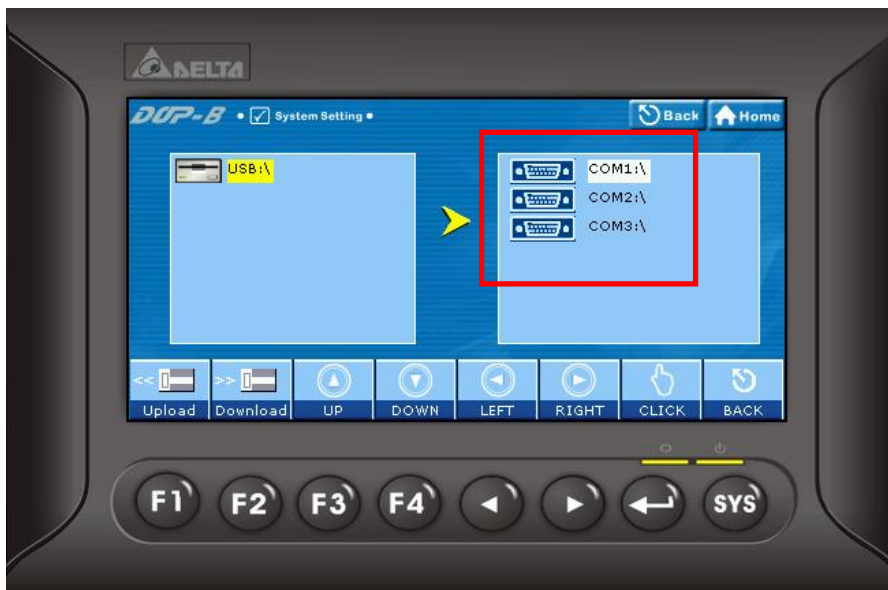
利用此功能可直接使用 HMI 對 DVP PLC 作程序的上下載。操作步驟如下：
長按 SYS 鍵進入系統菜單，選擇 Up/Download，



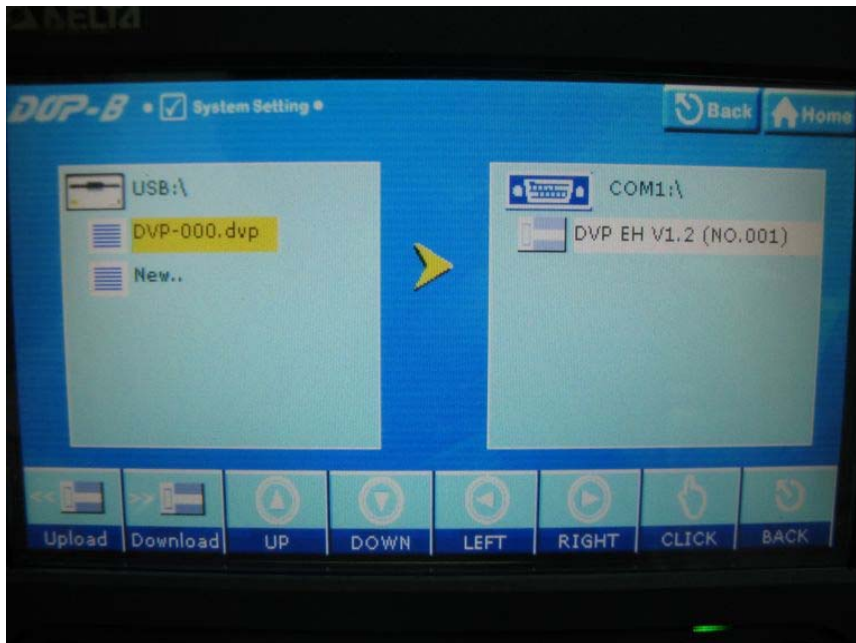
利用右鍵 () 切換到 Transfer Mode，



如下圖，用戶可直接由此畫面先選擇 PLC 通訊 COM Port，選擇 COM Port 後，人機就會自動搜尋 PLC。



PLC 搜尋結束後，即可在左邊視窗選擇要下載或是上傳的 PLC 文件，右邊視窗選擇 PLC，再利用畫面左下角的 Upload 或 Download 按鈕執行。



注意事項：

1. 使用此功能的 PLC 文件，文件名必須為 DVP-xxx.dvp (xxx 為 000 到 999 的數字)。
2. 若是要上傳 PLC 文件為新文件，則在左邊窗口選擇「New..」即可，再按 Upload 即可。
3. 詳細支持的功能版本如下：

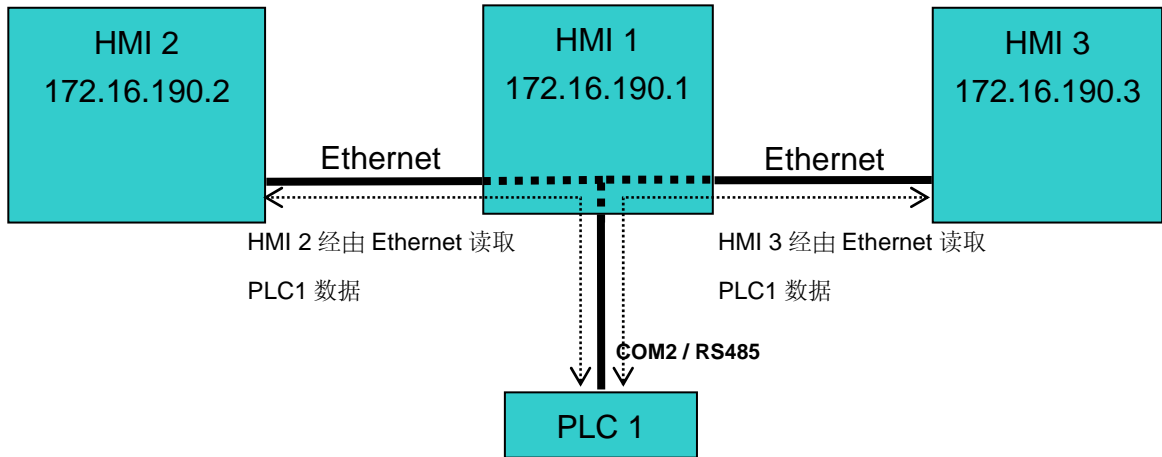
| PLC 機種 | 支援密碼檢查次數限制的版本 | 支援 PLC 識別碼限制的版本 |
|--------|---------------|-----------------|
| ES | X | X |
| ES2 | V1.0 以上 | V1.0 以上 |
| EX | X | X |
| EC | X | X |
| SS | X | X |
| SA | V 1.7 以上 | V 1.7 以上 |
| SX | V 1.7 以上 | V 1.7 以上 |
| SC | V 1.5 以上 | V 1.5 以上 |
| SV | V 1.2 以上 | V1.3 以上 |
| EH | X | X |
| EH2 | V 1.1 以上 | V1.3 以上 |
| EH2-L | V1.0 以上 | V1.0 以上 |

4. HMI Transfer Mode 只支持 *.dvp 程序上/下載，不支持子程序/Ladder 圖形碼/SFC 圖形 /裝置名稱注解/列注解/斷電保持數據/Label 結構/Symbol 結構...等資料上/下載。

(此頁有意留為空白)

附录 C 网络多主机连线说明

网络多主机功能可快速架构多台 HMI 连线一台(或多台)PLC 的通讯架构, 用户只需简单的设定即可, 多主机架构如下所示:



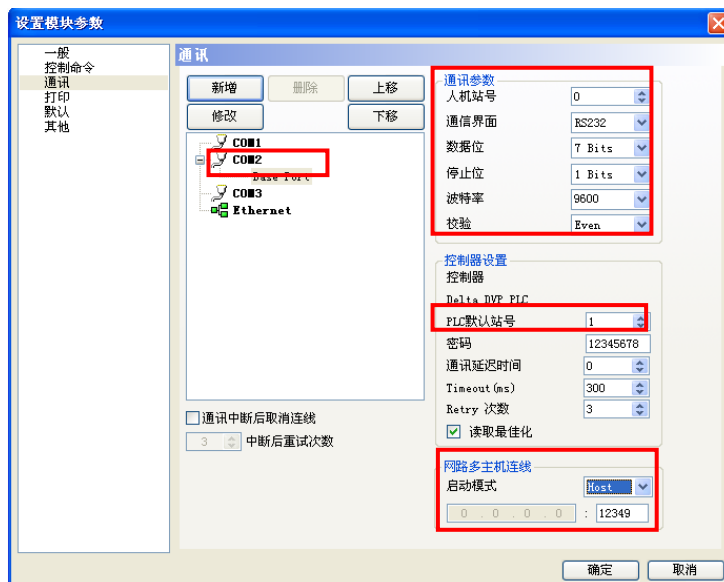
如上图所示, 三台人机 HMI 1、HMI 2、HMI 3 以网络线相连接, 其中仅 HMI 1 以实体 COM 2 RS-485 连接 PLC 1, HMI 2、HMI 3 未连接任何控制器。经由多主机架构的设定, HMI 2 与 HMI 3 可通过 Ethernet 读写 PLC1 内的寄存器。

多主机架构详细设定步骤如下所示:

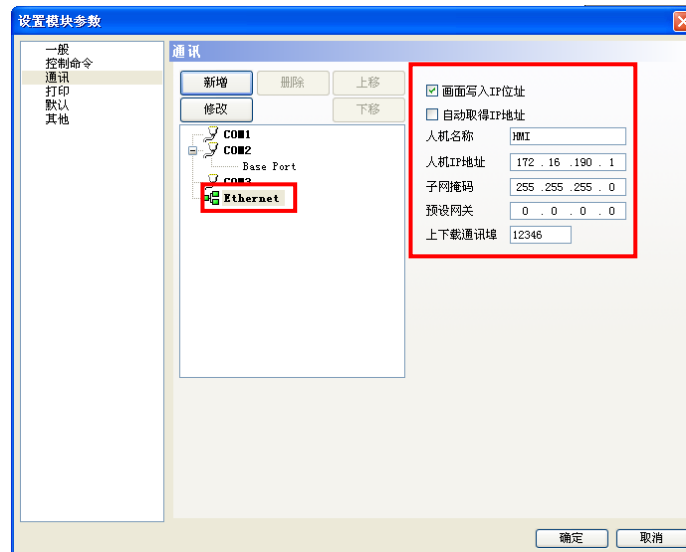
HMI 1 设定步骤:

在「选项」→「设置模块参数」→「通讯」, 设定如下信息:

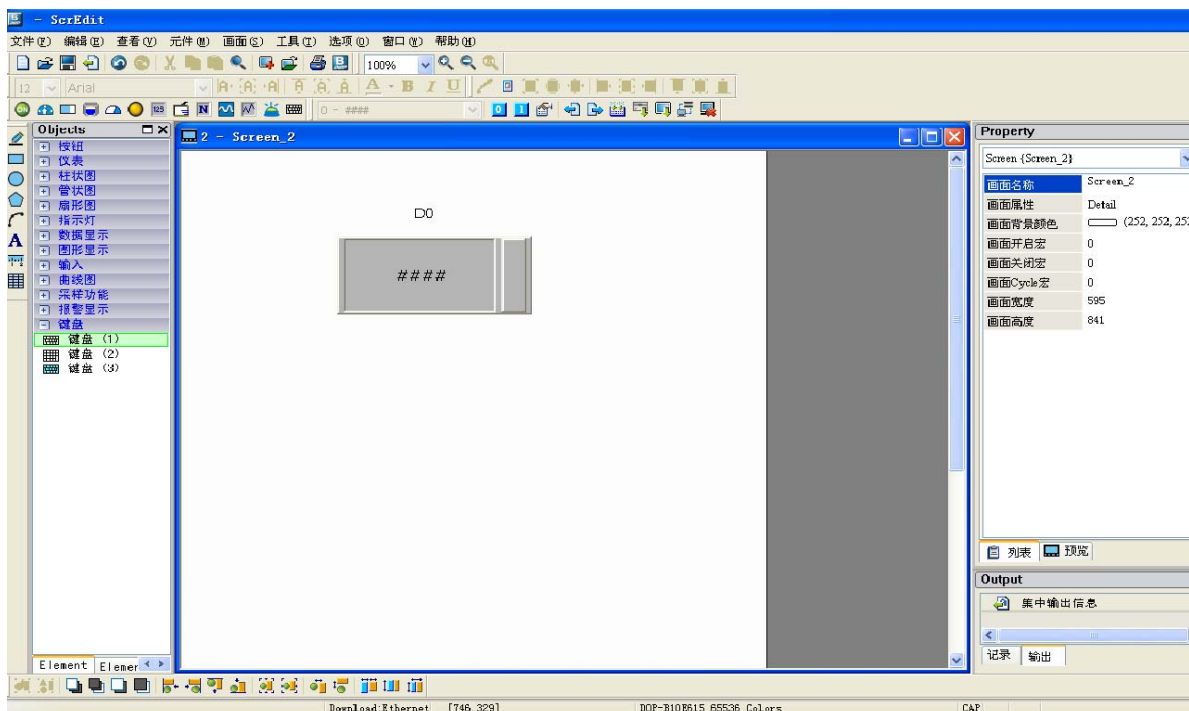
COM2 Base Port 实接一台 PLC, 通讯格式设定为 RS-485、[9600, 7, Even, 1], 站号为 1, 在网络多主机连线部分, 选择 HOST, 代表此 COM 口为实体 PLC。



再设定人机 IP 位址(172.16.190.1)，按下确定：



在人机画面中建立一数值输入元件，以读取 PLC 1 内部寄存器 D0：



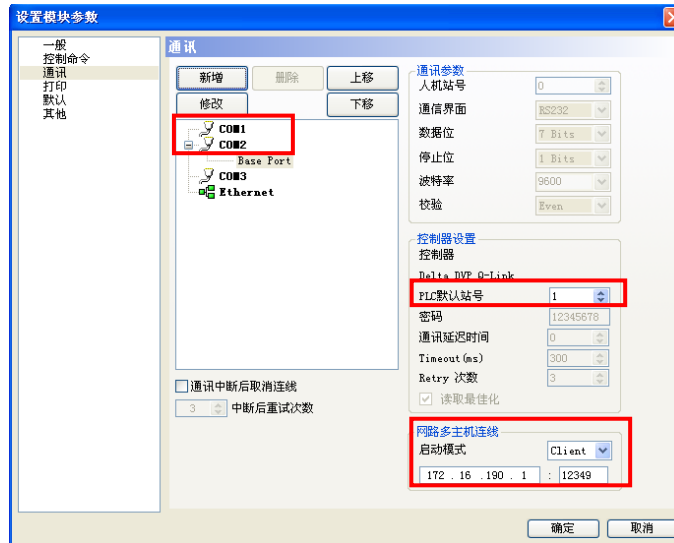
如此即完成 HMI 1 设定，编译后即可下载至 HMI 1。

HMI 2 设定步骤：

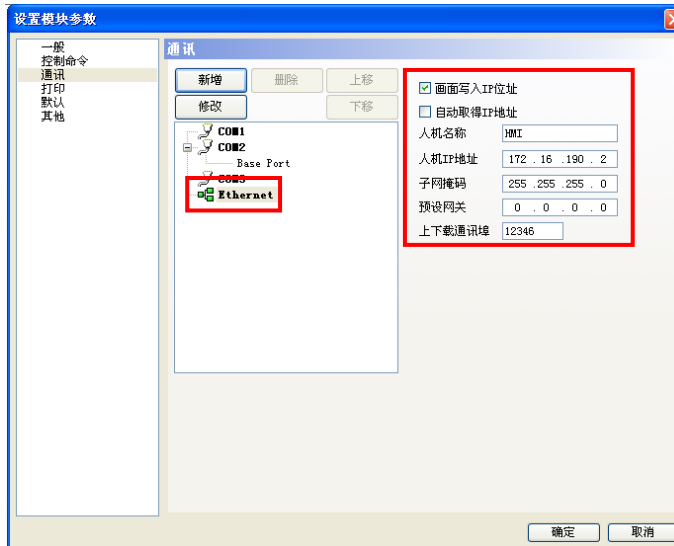
在「选项」→「设置模块参数」→「通讯」，设定如下信息：

COM1 Base port 先设定好 HMI 1 所连接的实体 PLC 站号后，在网络多主机连线选项由 Disable

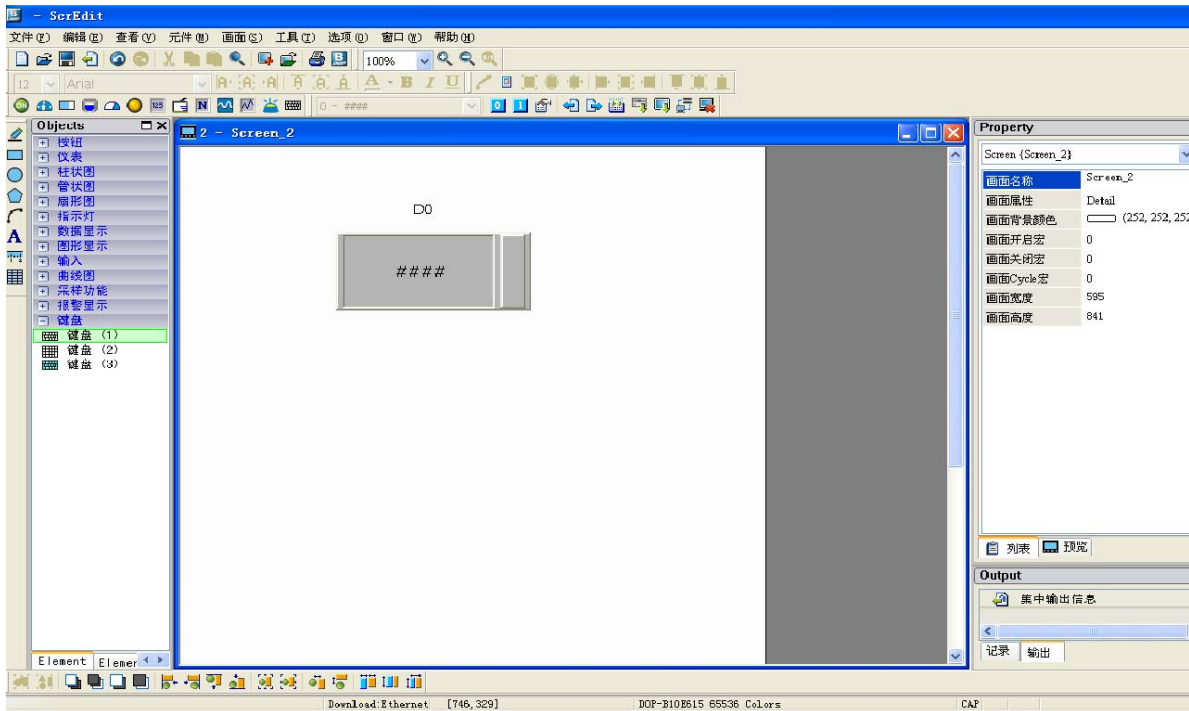
改成 Client, 再将 IP 改成 HMI 1 的 IP(172.16.190.1):



再设定人机 IP 位址(172.16.190.2), 按下确定:



在人机画面中建立一数值输入元件，以读取 PLC 1 内部寄存器 D0:

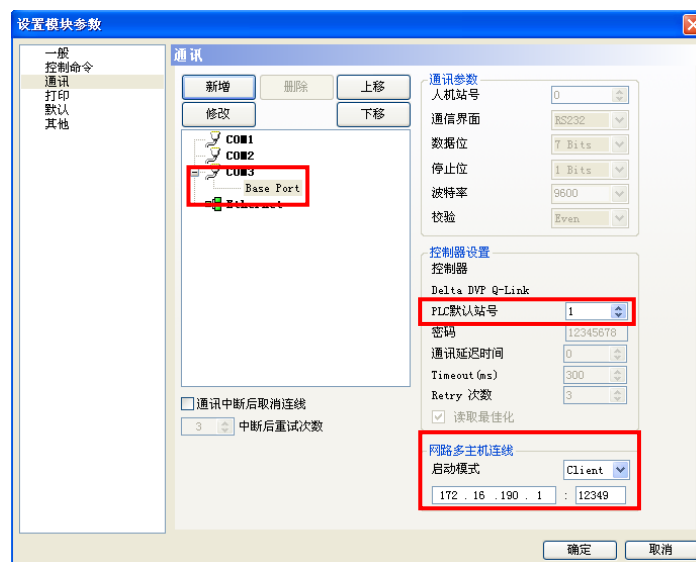


如此即完成 HMI 2 设定，编译后即可下载至 HMI 2。

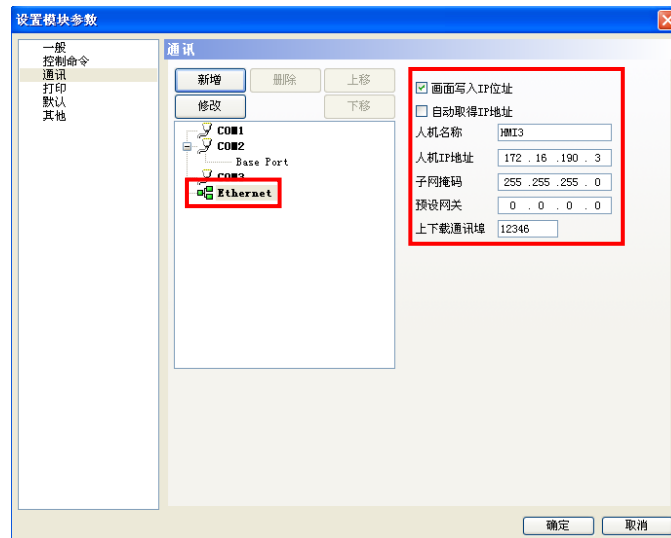
HMI 3 设定步骤:

在「选项」→「设置模块参数」→「通讯」，设定如下信息:

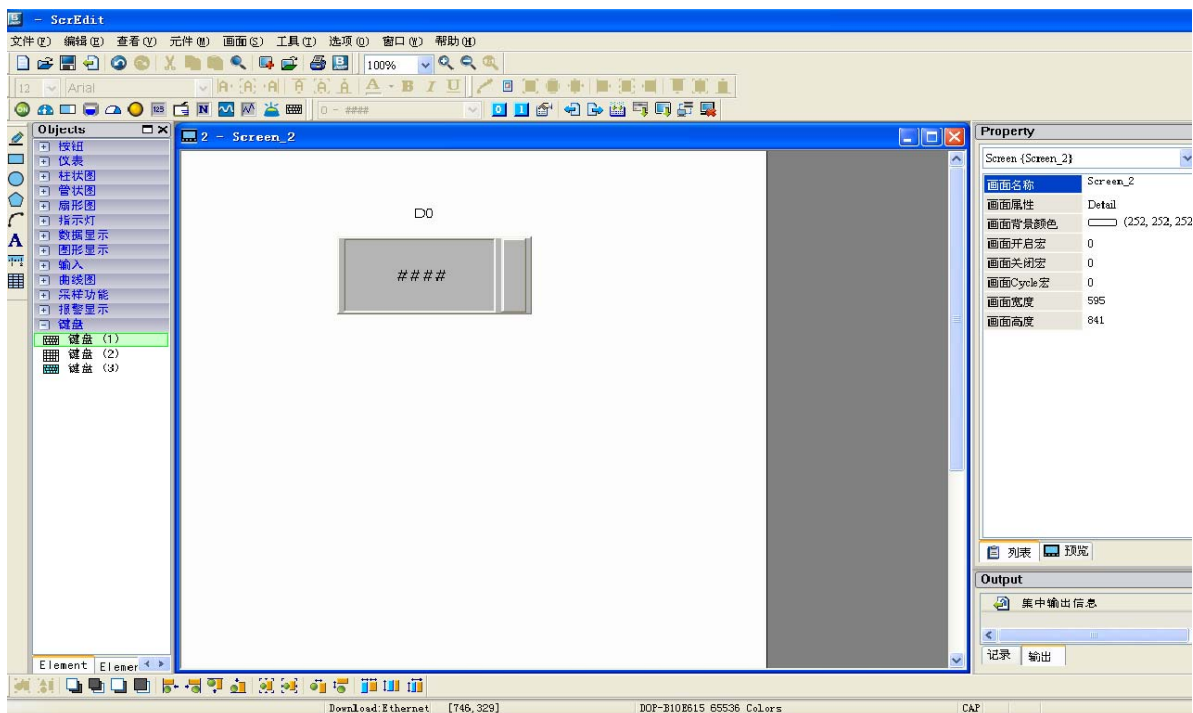
COM3 Base port 先设定好 HMI 1 所连接的实体 PLC 站号后，在网络多主机连线选项由 Disable 改成 Client，再将 IP 改成 HMI 1 的 IP(172.16.190.1):



再设定人机位址(172.16.190.3)，按下确定:



在人机画面中建立一数值输入元件，以读取 PLC 1 内部寄存器 D0：



如此即完成 HMI 3 设定，编译后即可下载至 HMI 3，再将其它装置接妥，即可使用多主机功能。

(此页有意留为空白)