

ISaGRAF使用说明书

概述

ISaGRAF 是一套完全符合 IEC61131-3 标准的可编程控制语言开发软件。其功能强大，使得开放式自动化控制成为可能。并已超越了传统工业控制设备、标准开放式硬件、操作系统和网络技术之间的界限。它提供的技术基础加快了应用/产品的开发和推向市场的进度。

ISaGRAF 提供 5 种符合 IEC61131-3 标准的编程语言：顺序功能图，梯形图，结构文本，指令表，功能块图。另外，ISaGRAF 还支持流程图编程。用户可以使用自己熟悉的编程方式编制程序控制支持 ISaGRAF 的所有控制器。

安装

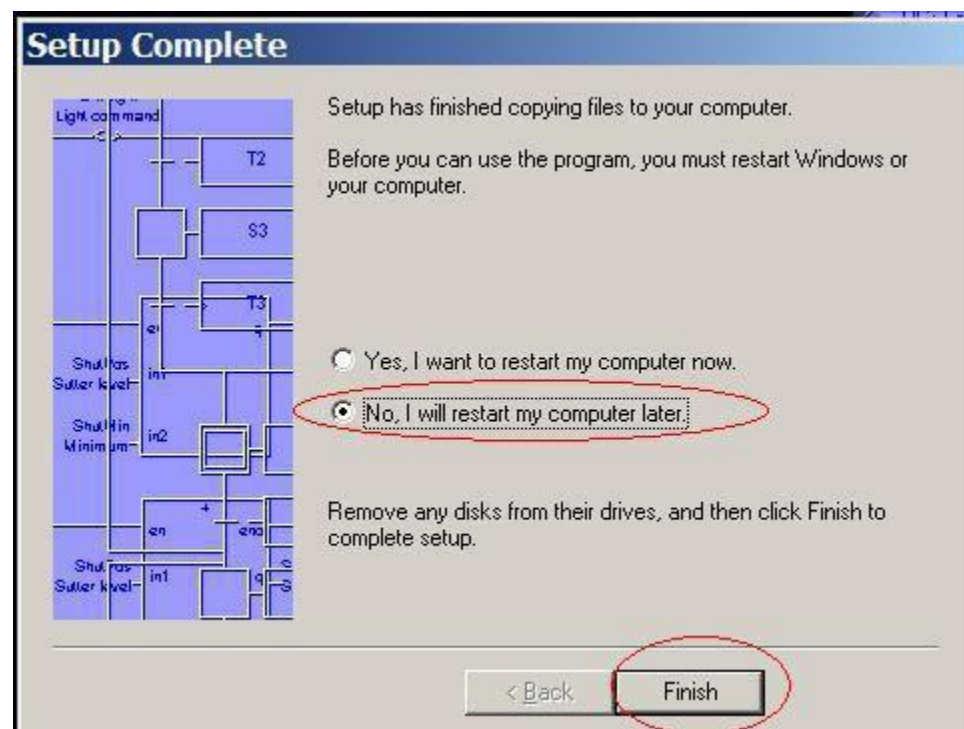
1. 将我们提供的安装文件解压缩，将自动生成一个 isagraf 目录。



2. 进入 isagraf 目录，再进入其中的 English\disk1 目录，双击运行 install.exe.



3. 连续点击 next 等待其安装完成。



4. 选择 “No, I will restart my computer later” ，然后单击 “Finish” 。
5. 关闭自动弹出的 “Readme” 。
6. 运行 isagraf 目录下的 IsagrafPatch.exe



7. 选择 isagraf 的安装路径





8. 重新启动计算机，安装完成。

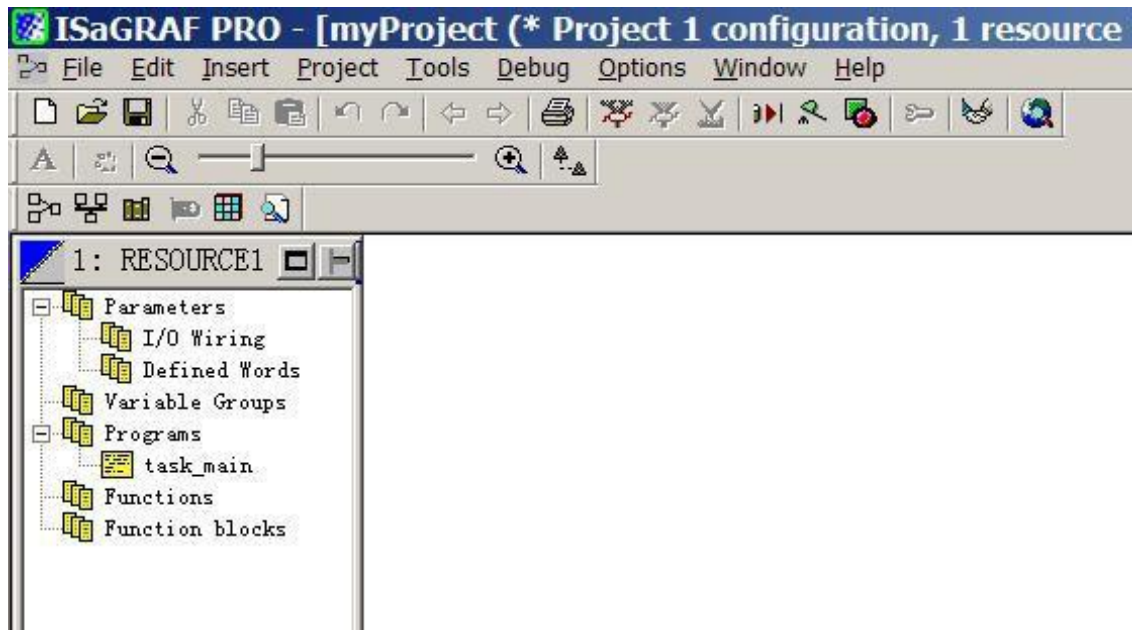
注意：如果运行出错，请安装同目录的Sentinel Protection Installer 7.6.8.exe

启动第一个项目

1. 将我们提供的工程模板 symcDemoV3.rar 解压缩，生成 symcDemoV3 目录。
2. 将该目录复制到您的工作目录，并且将目录名改成您需要的名字（ISaGRAF 根据目录名来区别不同的项目）。
3. 通过开始菜单启动 ISaGRAF PRO。
4. 通过“File”菜单的“Open Project/Library”打开您刚刚复制并改名的工程模板。
5. 由于更改过目录名，打开时 ISaGRAF 会提示更新工程，选择确定即可。



6. 启动后主界面如下：如果在主界面中双击程序名或者函数/功能块名，ISaGRAF 会另外打开一个窗口以便于用户对其进行编辑。新弹出窗口的标题栏开头为“DGE”，以后称之为编辑窗口。而将标题栏以“ISaGRAF PRO”开始的如下窗口称为主窗口。




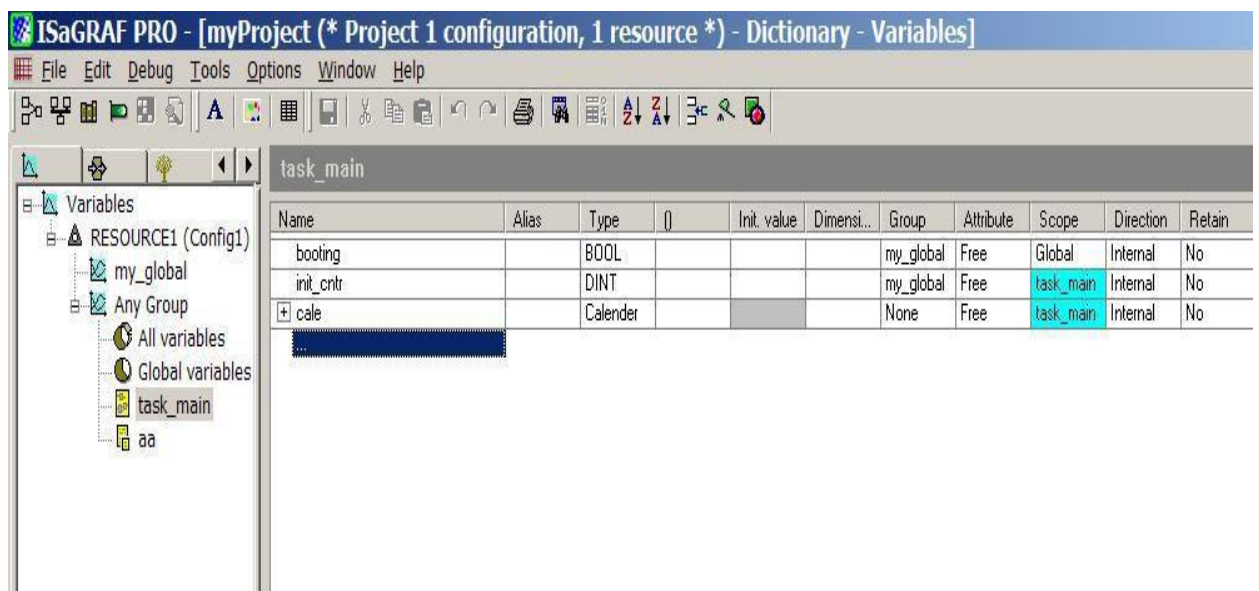
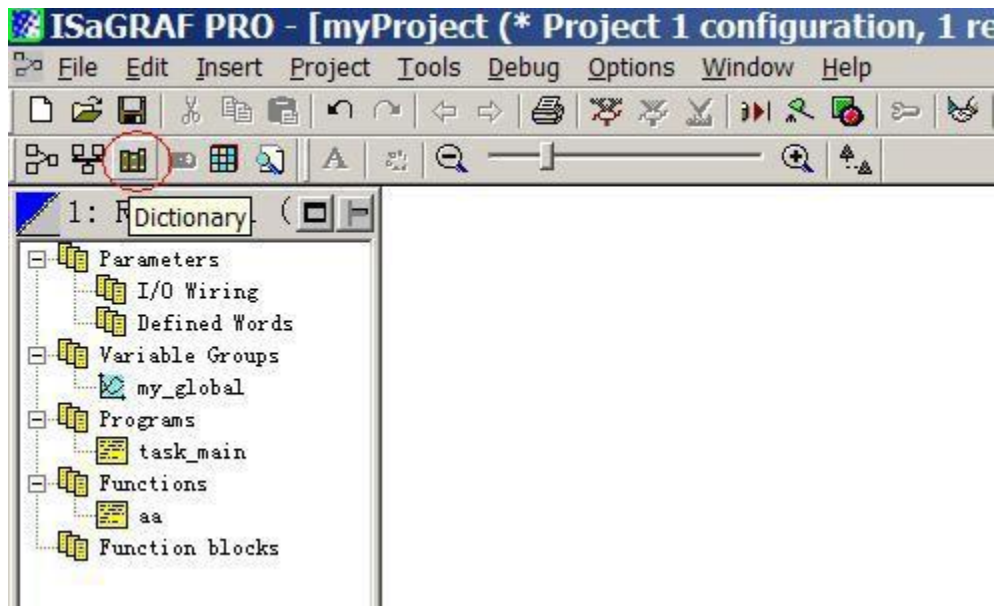
7. I/O wiring 定义 IO 口可变量之间的连接关系，双击打开。
8. Defined Words 用于常量定义，双击可以打开定义表。
9. Variable Groups 分组放置用户定义的变量，右键单击可以添加变量组。将变量分组是一个比较好的编程习惯，尤其是变量较多的时候。
10. Programs 中存放用户主程序，右键单击可以添加。每一个主程序都是一个任务，自动单独运行，不需要象其它 PLC 一样由统一的入口程序 PLC_PRG 调用。
11. Function 和 Function blocks 中可以添加用户自己的函数和功能块，在主程序中调用。
12. 双击主程序名如“task_main”新打开一个窗口编辑该主程序。



专题操作



变量定义

在 ISaGRAF 中，不管是全局变量、任务变量还是接口参数，都在一个数据字典中统一定义。在

主界面中单击  图标可以打开数据字典。



数据字典中的内容共分为五类，从左到右分别为： 用户变量、 函数（功能块）输入

输出参数、 用户定义类型（数组及结构）、 常量定义。

选择某一类后，可以在右侧的表中增加、删除和修改变量。

IO 口绑定

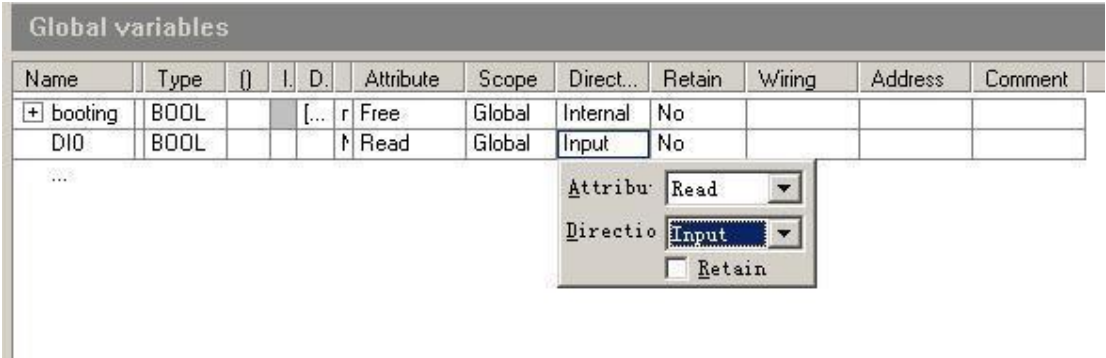
IO 口绑定是指将一个用户变量连接到一个硬件 IO 口，在程序中对该变量的所有操作都相当于直接操作其绑定的 IO 口。


IO 口绑定不是必须的，用户如果愿意，可以直接使用形如“%ID3.11”的 IEC61131-3 标号对所有的 IO 口直接进行操作。但是我们不推荐这种风格，主要原因有：

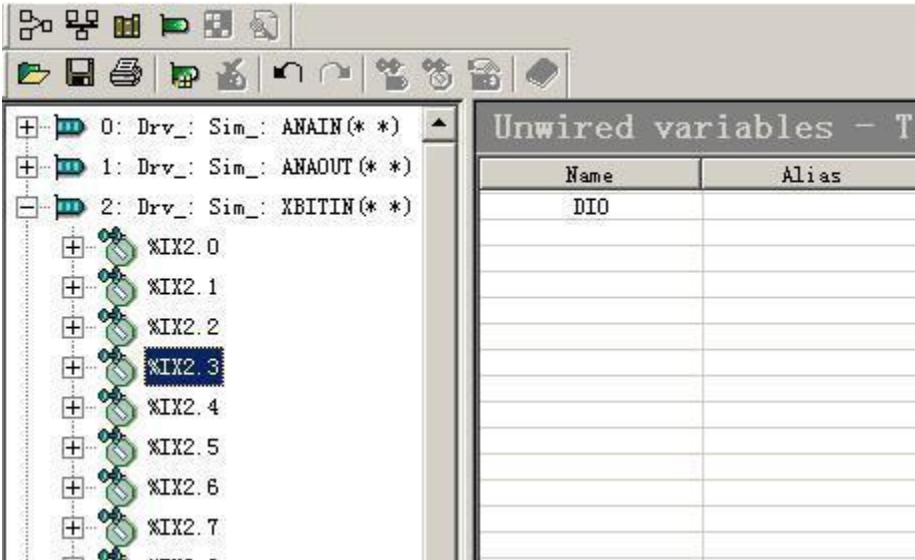
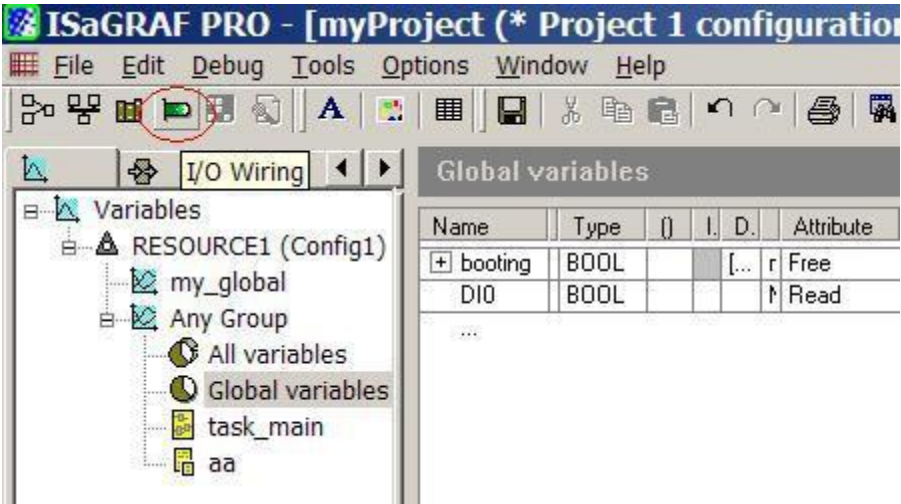
1. 不直观，程序不易理解；
2. 程序的可维护性和可移植性不好；

绑定方法：

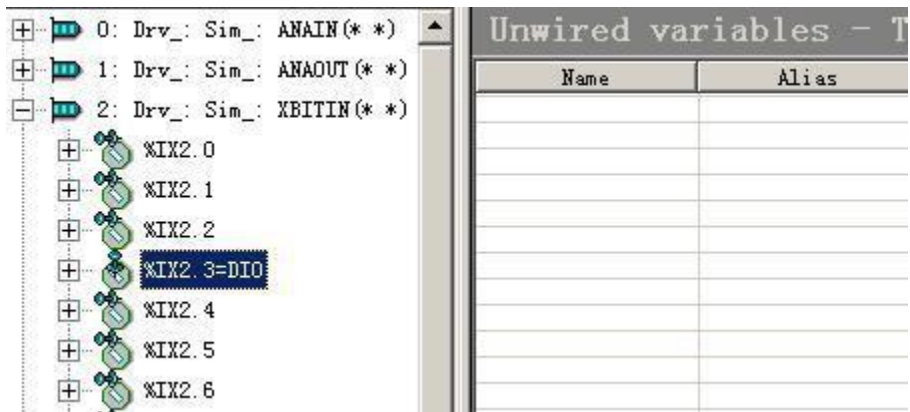
1. 首先需要在数据字典中定义 IO 变量，需要绑定到开关量的，数据类型必须为 BOOL；需要定义到模拟量或者脉冲及 PWM 的，数据类型必须为 DINT。如果需要绑定的 IO 口是输入口，则该变量需要定义为：input 和 read；如果要绑定的 IO 口是输出口，则需要将该变量定义为：output 和 write。新加变量的缺省属性是 Free 和 Internal，在其上双击即可打开一个下拉框对其进行修改，如下图。



如果想要把刚定义的 DIO 绑定到控制器的第三个开关量输入，需要在工具条上点击图标打开 IO 绑定界面。



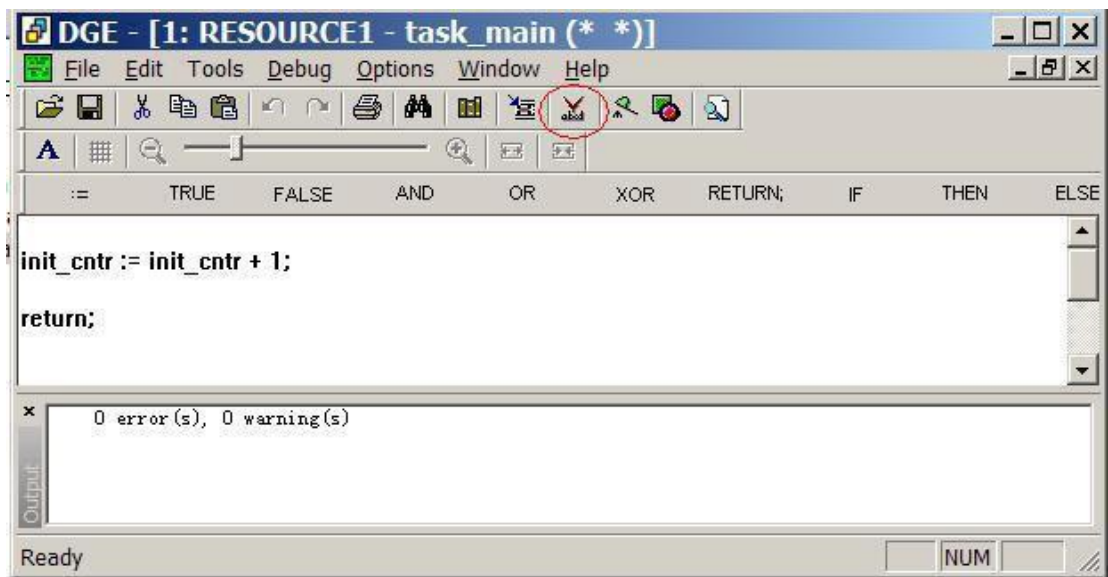
在左侧单击选中一个 IO 标号后，右侧 Unwired variables 窗口中会自动列出所有可以和该 IO 口匹配的尚未绑定的变量，上图中所示为之前定义的 DI0。在 DI0 上双击鼠标，即可将该变量绑定到左侧选中的 IO 口。




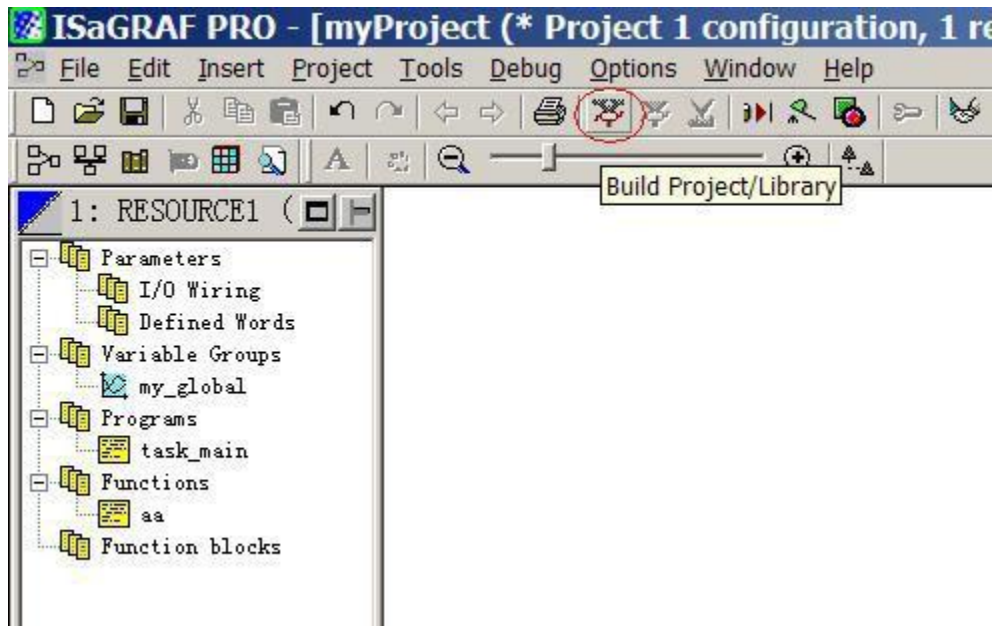
如果需要重新绑定，可以首先单击选中左侧已绑定的 IO 标号及变量，然后按下键盘上的删除（Delete）键，即可取消该绑定。

程序编译

在编辑一个程序的时候，可以直接工具栏上的  图标对当前程序进行编译。下面的 output 窗口将显示当前程序的错误数和警告数。



所有的程序（包括函数、功能块等）都分别编译通过后，如果想要运行、调试，还必须对整个工程进行一次编译。方法是在主窗口单击  图标。

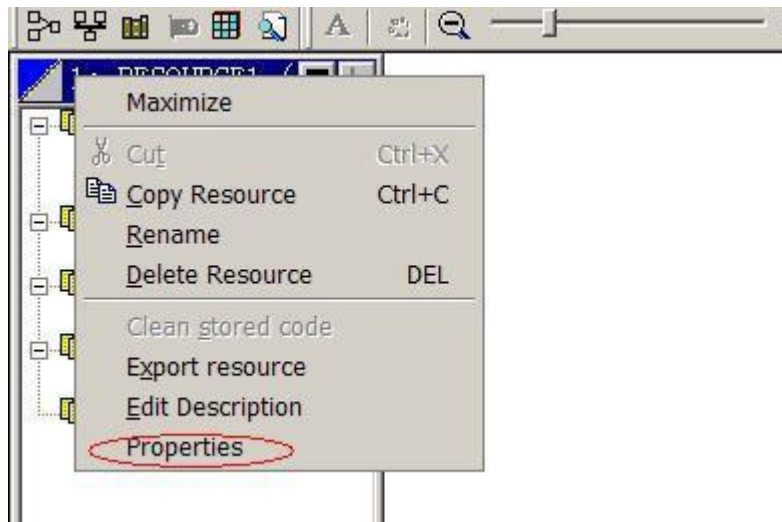


注意：对整个工程进行编译之前，必须保存所有编辑过的程序，并关闭数据字典。


软件仿真调试

ISaGRAF 可以提供一种功能非常强大的纯软件仿真功能，使您可以在不需要连接控制器的情况下手工设定任何输入端口的值或状态，以便于进行快速、方便的程序调试。

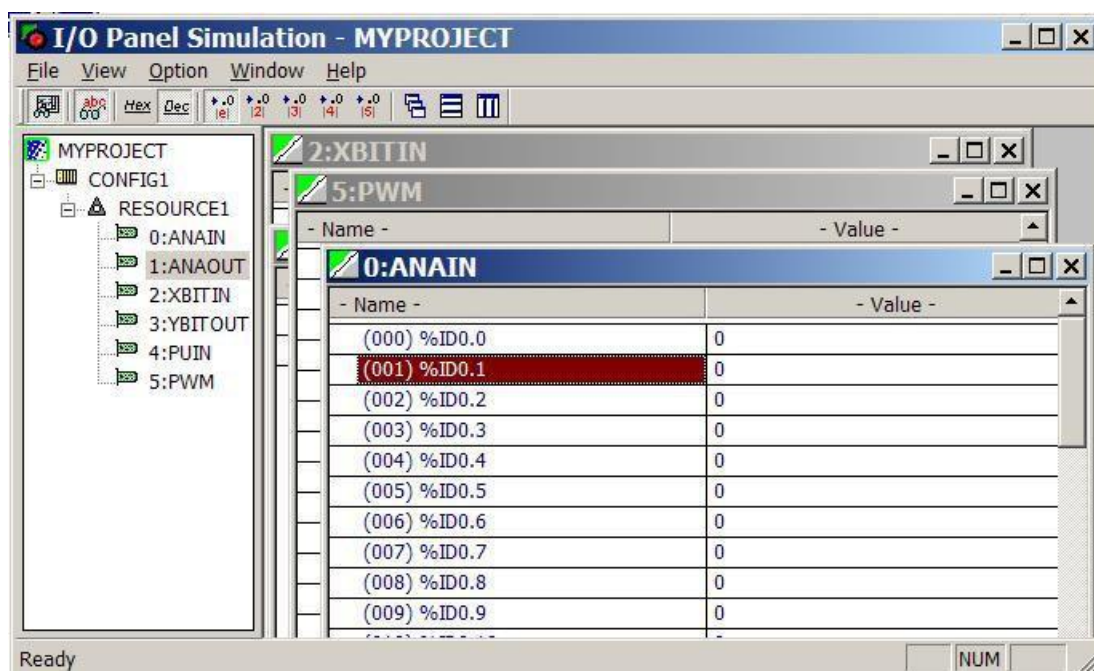
为了启动该项功能，首先需要编辑工程属性，打开编译器的仿真支持。方法：在主窗口中右键单击 Resource 子窗口的标题栏，在弹出的下拉菜单中选择“Properties”，然后在弹出的对话框中选择“target/code 标签”，选中“code for simulate”，单击“确定”退出。





然后保存并重新编译工程，单击主窗口工具栏上的  图标，即可进入软件仿真状态。进

入软件仿真状态后，ISaGRAF 会自动弹出一个“I/O Panel Simulation”窗口，您可以在其中改变所有模拟量输入口的值、开关量输入口的状态，也可以设置程序中没有处理的输出口的值和状态。



软件下装调试


准备工作：

1. 工程编译完成，软件中没有错误；
2. 连接您编程的计算机与 SYMC 控制器之间的网络电缆（参见 SYMC IO 定义说明书）。
3. SYMC 控制器的 IP 地址固定为 172.16.16.148，修改您的计算机的网络设置使其与控制器在同一个网段。比如，可以设置您的 IP 地址为：172.16.16.1，子网掩码为 255.255.255.0，

网关无需改动。(SYMC 可以自动匹配子网掩码, 假如您原来的 IP 地址为 172.16.123.222, 子网掩码为 255.255.255.0, 只需将子网掩码改为 255.255.0.0, 即可实现与 SYMC 控制器同网段。)

4. 控制器上电, 在您的开发计算机上运行命令 `ping 172.16.16.148`, 检验与 SYMC 控制器是否连通, 如果不通, 请重新检查网线和网络设置。

如果网络已经连通, 可以按以下步骤下装程序并进行联机调试:

1. 点击主窗口工具栏上的  图标下装程序到 SYMC 控制器。
2. 在弹出的 Download 对话框中单击 “Select All” 按钮, 然后单击 “Download” 按钮。如果控制器中已经有程序存在, ISaGRAF 会弹出一个询问对话框进行确认, 单击相应的按钮后程序开始下装, 下装过程中您可以看到进度百分比。下装完成后程序会自动开始运行。

您可以点击主窗口工具栏上的  图标进入联机调试状态。

常见问题汇总

1 为什么有时启动调试功能时 ISaGRAF 软件会没有响应？

由于 ISaGRAF 的下装和调试都需要启动两个底层通信处理进程（可以看到两个 DOS 窗口），并且每次完成后都需要完全退出，下次使用时再重新启动。所以在程序下装完成后，需要确认两个 DOS 窗口已经完全退出，然后再点击“debug”按钮进行调试。

2 为什么有时候无法编译工程？

可能是因为数据字典没有关闭。ISaGRAF 的很多操作都要求数据字典处于关闭状态，例如：工程编译、程序或者函数改名、添加程序/函数/功能块、增加变量组等。此时只需将数据字典窗口关闭即可。

3 函数或者功能块的内部变量如何定义？

在 ISaGRAF 中函数和功能块的内部变量分成两个部分：输入输出变量作为参数在数据字典的参数表中定义；局部变量与其它的全局变量和任务变量一样在数据字典的变量表中统一定义，只不过其 scope 为函数或功能块。在工程浏览窗口中右键单击一个函数或功能块名，弹出的下拉菜单中有 Parameters 和 Local Variables 两个选项，分别单击即可添加或者编辑输入输出参数和内部变量。

4 为什么有时编译程序时会出现某个函数“unexpected statement”错误？

ISaGRAF 对数据类型检查非常严格，如果一个函数定义了返回值，而用户在调用的时候没有将其返回值赋给一个变量，则会出现这种错误。只需定义一个和其返回值同类型的变量，接收该函数的返回值即可。

5 数据类型不匹配如何解决？

与上面所述一样，由于 ISaGRAF 的类型检查及其严格，不同的数据类型绝对不允许互相赋值或进行数据传递。不过 ISaGRAF 也提供了几个类型转换函数，可以比较方便的进行数据类型的强制转换。与其它开发环境略有不同的是，ISaGRAF 的类型转换函数不需要指明源数据类型，只需知道目的数据类型，例如不管任何一个数据类型想要转换成 SINT 类型，调用 ANY_TO_SINT() 函数即可。

另外需要注意的是，进行长类型到短类型的强制类型转换时，一定要注意数据的值是否会因为被截断而改变。

6 为什么有时候 “I/O Wiring” 菜单和按钮都无效？

由于 ISaGRAF 支持多个 resource，而 “I/O Wiring” 一次只能针对一个 resource 的 I/O 进行相关操作，所以即使当前用户只配置了一个 resource，如果该 resource 窗口没有被激活，“I/O Wiring” 按钮和菜单也会显示无效。用户只需在当前工程浏览窗口中单击 resource 窗口即可。