



模块 23

Scada Basic - 初级

内容

1	介绍	4
1.1	PcVue 版本	4
1.2	本章学习内容	4
1.3	本章所用的文件	4
1.4	本章所使用的第三方软件	4
2	PcVue 架构中的该功能	5
3	SCADA Basic 与 VBA 的比较	6
4	SCADA Basic 初级	8
4.1	程序	8
4.1.1	SCADA basic 工作区	8
4.1.2	创建一个程序	10
4.1.3	管理程序	11
4.1.4	自动装载全局程序	12
4.2	函数	13
4.2.1	概述	13
4.2.2	语法	13
4.2.3	主函数	13
4.2.4	执行一个函数	15
4.2.5	传递参量	17
4.2.6	返回结果	18
4.2.7	通过动画触发一个函数	19
4.3	变量	20
4.3.1	内部变量	20
4.3.2	变量树变量	23
4.4	内置指令	24
4.4.1	查找正确指令	24
4.4.2	指令的使用	24
4.5	自定义窗口	26
4.6	触发函数的不同方式	28

4.6.1	事件	28
4.6.2	按键组合	31
4.6.3	周期循环	32
4.6.4	时间表	34
4.6.5	从另一个程序调用函数	35
5	总结	37

1 介绍

1.1 PcVue 版本

本章使用的 PcVue 版本为 12

1.2 本章学习内容

- ★ SCADA Basic 程序结构
- ★ SCADA Basic 语法
- ★ 如何创建一个简单脚本
- ★ 如何执行一个脚本
- ★ 常见错误
- ★ 如何通过点击按钮来执行脚本
- ★ 如何使用帮助文档

1.3 本章所用的文件

PcVue 项目使用模块 21 初级结束时的项目。

1.4 本章所使用的第三方软件

无

2 PcVue 架构中的该功能

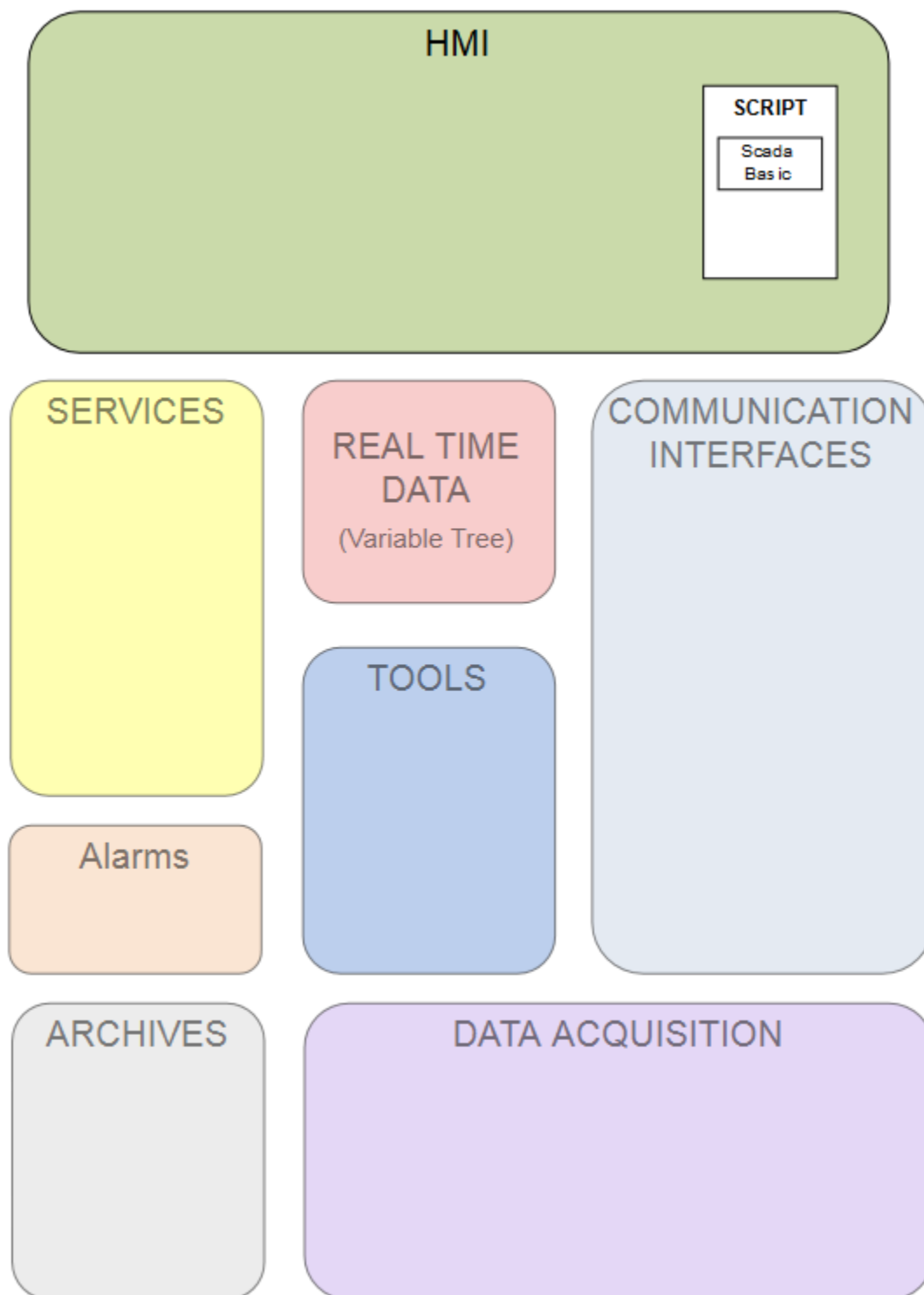


图 1

3 SCADA Basic 与 VBA 的比较

PcVue 支持两种不同的编程语言：SCADA Basic 和 VBA。

这两种语言不是“竞争者”，相反，他们是“互补的”。

如果不知道这两种语言各自的特点，对于项目开发人员来说在选择何种编程语言的时候就会出现困难。

以下图表将可以帮助用户理解这两种语言的使用范围。不同的绘线指代不同的可用语言：

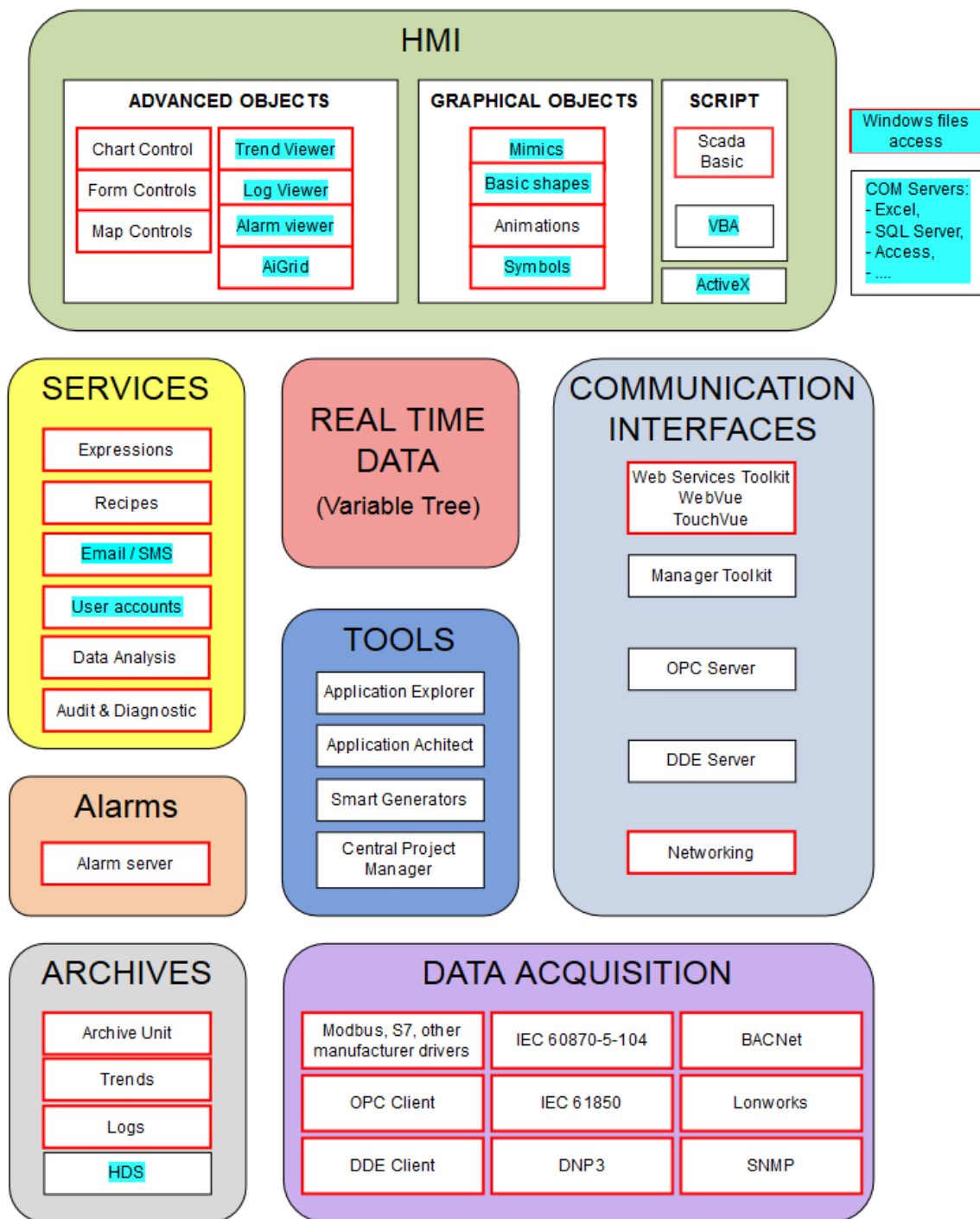


图 2

从以上图表可以看出 SCADA Basic 语言在访问 PcVue 对象的时候比 VBA 更适用。

当访问 VBA 和 SCADA Basic 都支持的 PcVue 对象时，用户可以任选其一。最好的方式是选择用户最擅长的语言。



如果需要处理 HMI 对象以及 Windows 文件访问，那么我们建议用户使用 VBA，因为它更适合处理此类。

4 SCADA Basic 初级

4.1 程序

脚本由一组程序组成。如果你熟悉 Visual Basic 语言，那么可以把程序比作模块/组件。

一个程序是一个包含脚本的文本文件。可以使用 PcVue 内置的编辑器或者任意文本编辑器来编辑该文件。



SCADA Basic 程序没有特殊的后缀名，但是我们建议使用后缀名来命名，我们比较喜欢使用这两个后缀名：.scb 或 .svb。



在程序执行之前，它必须预先装载好。

有一个特殊的程序叫**全局程序**。这个独一无二的程序包含 PcVue 所有内置的 SCADA Basic 指令。因此，如果需要使用一个或多个这些指令（在没有任何东西的情况下开发一个程序是有一定难度的），你必须创建并加载该全局程序。

4.1.1 SCADA basic 工作区

SCADA Basic 工作区是用来管理程序的。用户可以从工作区创建，修改，删除，装载，不装载和执行任意程序。

要打开 SCADA Basic 工作区，可以选择命令 组态 / 操作 / 程序... 或者使用功能键 **F9**。

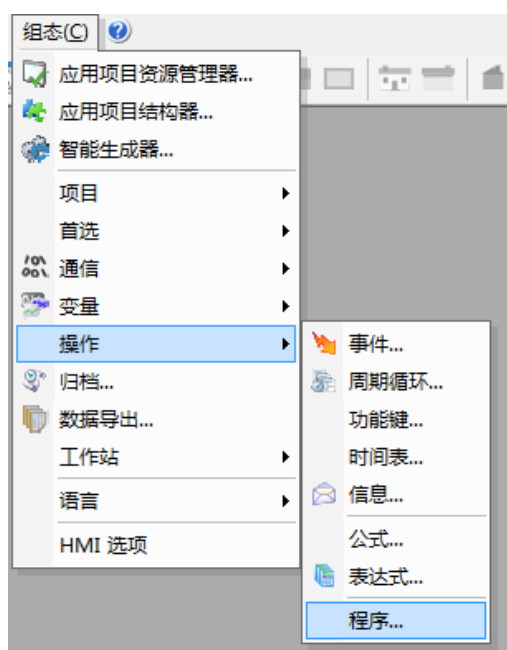


图 3



图 4

- [A]: 显示结果 或 清除结果 将结果显示到输出区域。
- [B]: 输出区域。用于显示错误信息以及内置 Print 函数的输出，可用于调试。
- [C]: 管理 全局程序。
- [D]: 管理任意 程序。
- [E]: 列出已装载的程序。
- [F]: 操作按钮。

4.1.2 创建一个程序

创建一个程序：

步骤 1. 打开 SCADA Basic 工作区。

步骤 2. 在程序栏中输入程序名称（全局程序或普通程序）。

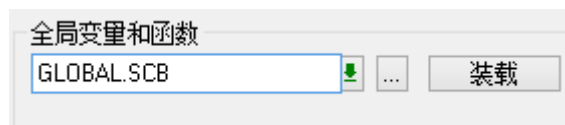


图 5

步骤 3. 点击省略号按钮 来编辑程序，PcVue 打开 SCADA Basic 编辑器。

步骤 4. 输入脚本。选择命令 文件 / 退出 或者 退出 按钮。点击 是 按钮来保存脚本。

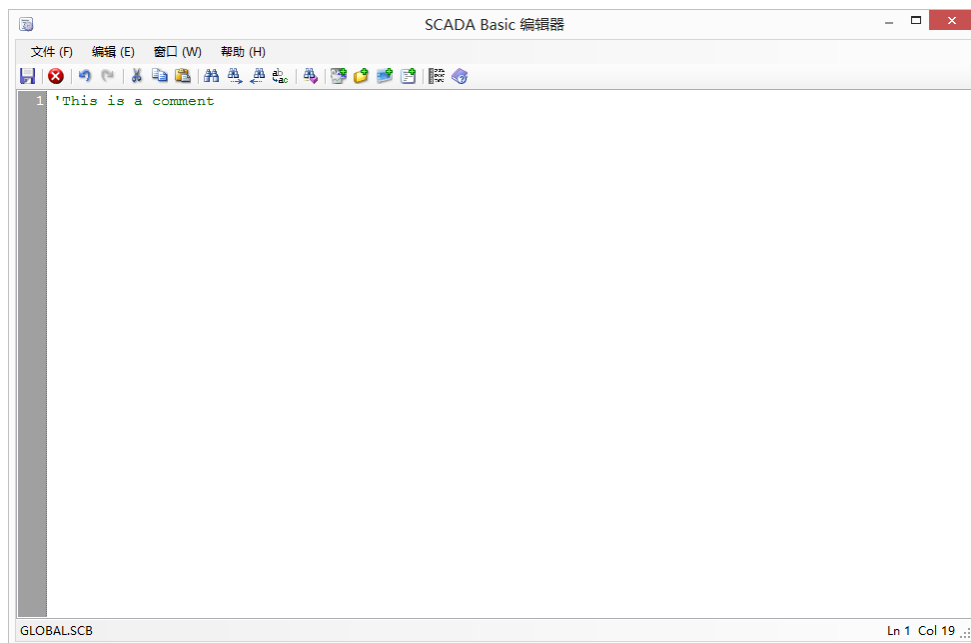


图 6



使用菜单命令窗口/选项可以在 SCADA Basic 编辑器里改变字体

4.1.3 管理程序

删除程序：

- 步骤 1.** 打开 SCADA Basic 工作区。
- 步骤 2.** 选择按钮  来打开程序选择器。
- 步骤 3.** 选择程序。
- 步骤 4.** 点击删除按钮。

装载程序：

- 步骤 1.** 打开 SCADA Basic 工作区。
- 步骤 2.** 选择按钮  来打开程序选择器。
- 步骤 3.** 选择程序。点击 确定 按钮。
- 步骤 4.** 点击 装载 按钮。在已装载的程序列表中，PcVue 增加了一行消息。

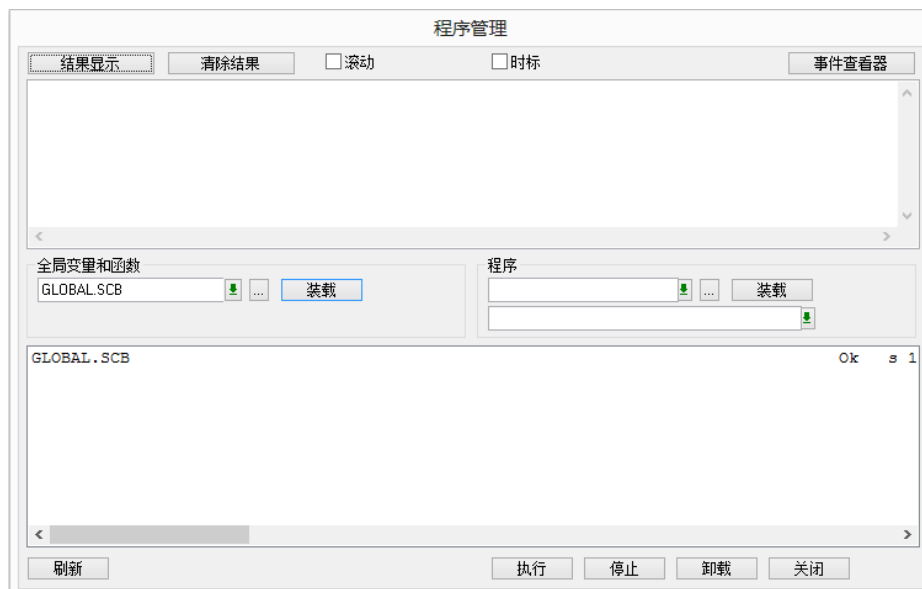


图 7

- 步骤 5.** 点击 结果显示 按钮来显示所有语法错误信息。（默认情况下，输出区域不会自动更新）。

4.1.4 自动装载全局程序

在 PcVue 启动时自动装载全局程序：

步骤 1. 选择命令 组态 / 项目 / 配置 然后打开 站启动 选项卡。

步骤 2. 选择选项 运行用户程序。

步骤 3. 在 声明 栏中输入(或选择)全局程序 。点击 确定 按钮。

4.2 函数

4.2.1 概述

程序是函数的集合体。每个函数都包含需要执行的脚本。

在各式的编程语言中，有两种不同类型的“脚本块”：一种返回结果（函数），一种无返回值（过程）。在 SCADA Basic 中，我们没有这样的区分，只有函数。SCADA Basic 函数可以返回一个结果，也可以不返回结果。

程序中没有函数个数的限制。

4.2.2 语法

声明函数的语法如下：

Sub <函数名称> ([参数 1 [, 参数 2,]])

---脚本-----

End Sub



一些规则：

- ★ SCADA Basic 语言不区分大小写。
- ★ 函数名称只能含有字母、数字以及下划线。不允许空格字符。
- ★ 在程序中，函数名称必须唯一。
- ★ 参数最大个数为 12。不需要声明参数的类型。（参数在其他章阐述）
- ★ 除 Sub, End Sub, If, For, Next, While and Wend 指令外，每条指令需要有“；”来结尾。

4.2.3 主函数

有一个特别的函数叫主函数。如果不指定其他函数，那么将执行主函数。主函数是唯一能够直接从 SCADA Basic 工作区执行的函数。

在 SCADA Basic 工作区执行主函数：

步骤 1. 编写以下主函数

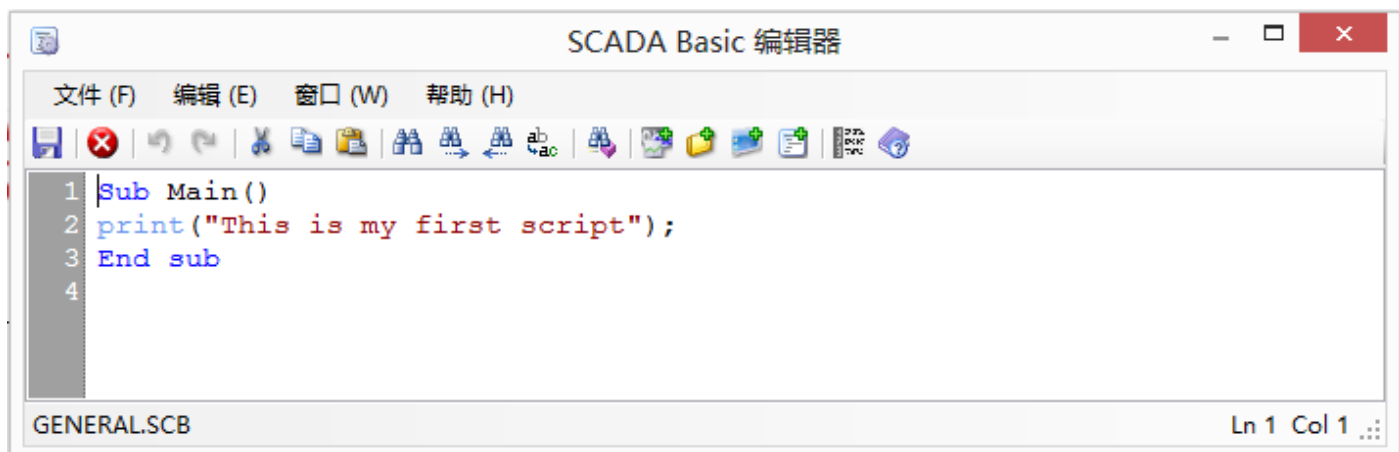


图 8

```
Sub Main()
  Print("This is my first script");
End Sub
```

步骤 2. 装载程序。

步骤 3. 在已装载的程序列表中选择该程序。

步骤 4. 点击 执行 按钮。



图 9

步骤 5. 点击 结果显示 按钮来检查是否出错。

4.2.4 执行一个函数

函数可以通过多种触发器来执行。比如在 HMI 上点击一个特定动画来改变数据库中的值（被称为事件）。我们将在高级模块中阐述可用的触发器。

使用 SCADA Basic 工作区来执行函数的唯一方式是执行主函数，主函数可以调用用户所要执行的其他函数。

实际上，用户可以从一个函数调用另外一个函数。调用函数在被调用函数执行之后继续执行。

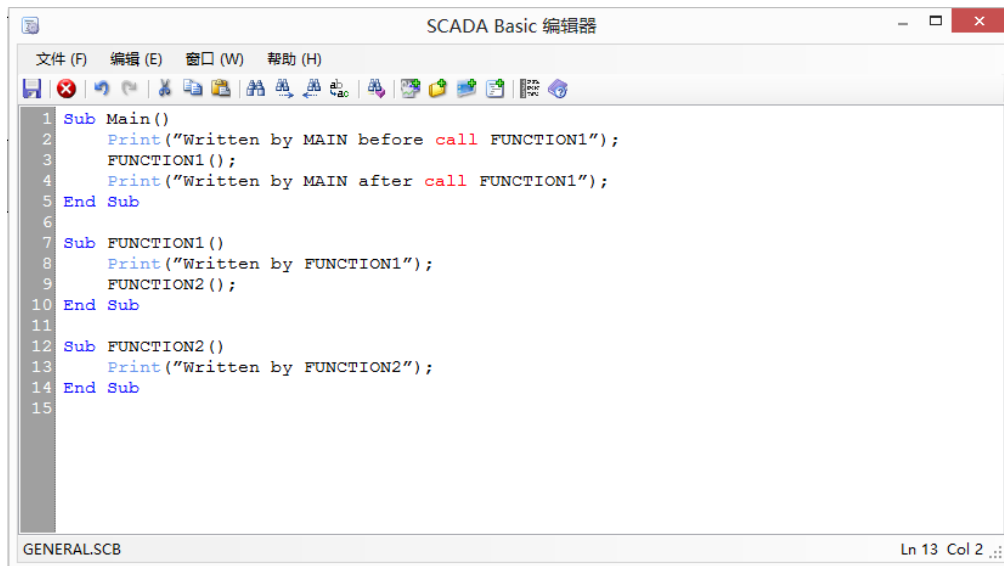


图 10

```
Sub Main()  
    Print("Written by MAIN before call FUNCTION1");  
    FUNCTION1();  
    Print("Written by MAIN after call FUNCTION1");  
End Sub  
  
Sub FUNCTION1()  
    Print("Written by FUNCTION1");  
    FUNCTION2();  
End Sub  
  
Sub FUNCTION2()  
    Print("Written by FUNCTION2");  
End Sub
```

上述脚本将会生成以下结果：

```
Written by Main before call Function1  
Written by Function1  
Written by Function2  
Written by Main after call Function1
```

图 11



不需要按调用函数的顺序来声明各函数。比如，函数 2 可以在函数 1 之前声明。

4.2.5 传递参量

如 4.2.2 章节所示，函数可以获得一系列参量作为输入。每个参量都被当作变量。（一会我们将会说到变量。）

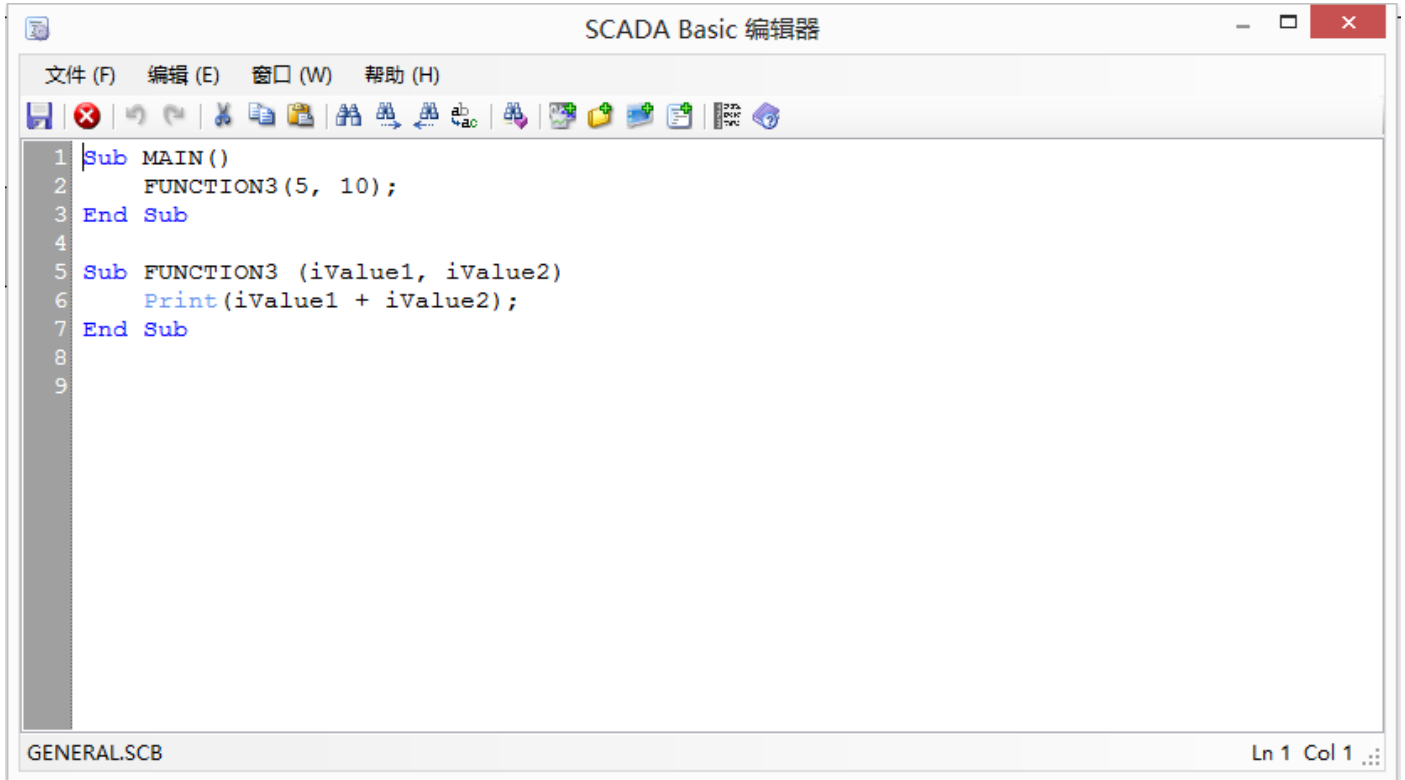


图 12

```
Sub MAIN()  
    FUNCTION3(5, 10);  
End Sub  
  
Sub FUNCTION3 (iValue1, iValue2)  
    Print(iValue1 + iValue2);  
End Sub
```

上述脚本将得到以下结果： 15

如你所见，虽然参量的类型没有指定 – 但是 SCADA Basic 通过参量使用的类型来决定其类型。

4.2.6 返回结果

前面我们提到一个函数可以返回结果。调用的函数可以将结果存储在一个变量中或者作为一个其他函数的一个参数。要返回函数的结果，必须使用内置的指令：**Return (Value)**。



图 13

```
Sub MAIN()  
    Print(FUNCTION4());  
End Sub  
  
Sub FUNCTION4()  
    Return(10);  
End Sub
```

函数 4 返回值为 10，该值通过参量传递给内置的 **Print** 指令。

4.2.7 通过动画触发一个函数

可以通过点击图形对象来触发程序。可用运行程序 动画来配置。

步骤 1. 选择图形对象。

步骤 2. 选择 动画/ 运行/程序 ... 命令。

步骤 3. 输入要运行的 程序 和 函数。点击确定按钮。

4.3 变量

与其他编程语言一样，SCADA Basic 也需要处理变量。可以将变量看成是某个时间段内所分配的内存空间。

有两种类型的变量：

- ★ 内部变量
- ★ 数据库变量

4.3.1 内部变量

4.3.1.1 内部变量类型

内部变量是在程序中使用的，它们不是变量数据库的一部分。

变量再创建时需要定义变量的类型。以下是临时变量的类型列表：

类型	描述	范围
Integer	2 字节整型	-32768 到 +32768
Long	4 字节整型	-2147483648 到+2147483648
Single	4 字节单精度实型	-3.37E+38 到+3.37E+38
Double	8 字节双精度实型	-1.67D+308 到+1.67D+308
Str	字符串	总计 255
Const	数字常量	-1.67D+308 +1.67D+308

4.3.1.2 声明内部变量

必须在书写代码前声明好变量（通常在 **Sub** 声明之后）。

要声明内部变量必须使用以下语法：

Dim <变量名称> **as** <变量类型>;

对于常量声明，必须使用以下语法：

Const <常量名称> = <数值>;

内部变量再创建之后，它的数值为 0，如果是字符型，那么则为空字符串。

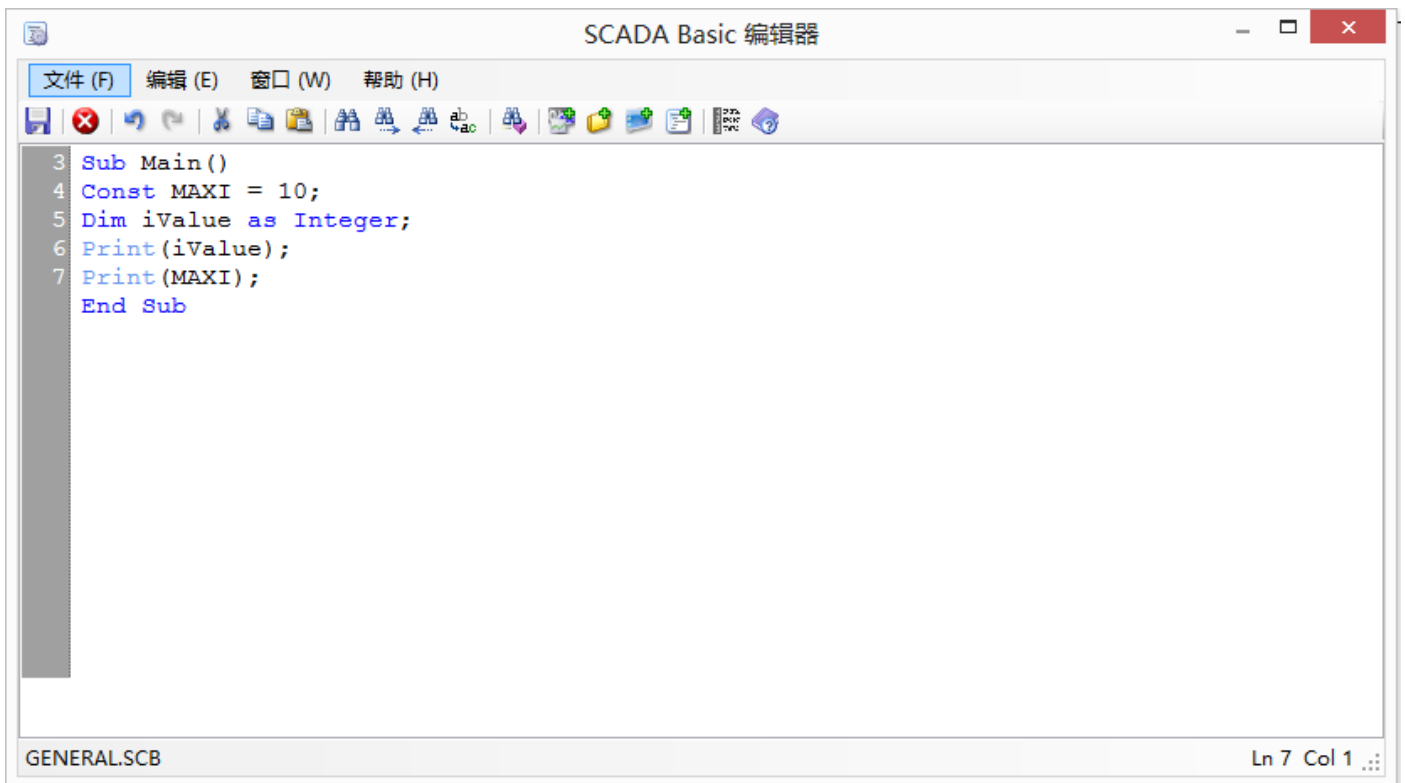


图 14

```
Sub Main()  
    Const MAXI = 10;  
    Dim iValue as Integer;  
    Print(iValue);  
    Print(MAXI);  
End Sub
```

上述脚本将会获得以下结果：

0

10

4.3.1.3 使用内部变量

有多种方式来应用内部变量：

- ★ 赋值
- ★ 设置其他变量的值
- ★ 传递参量
- ★ 返回结果

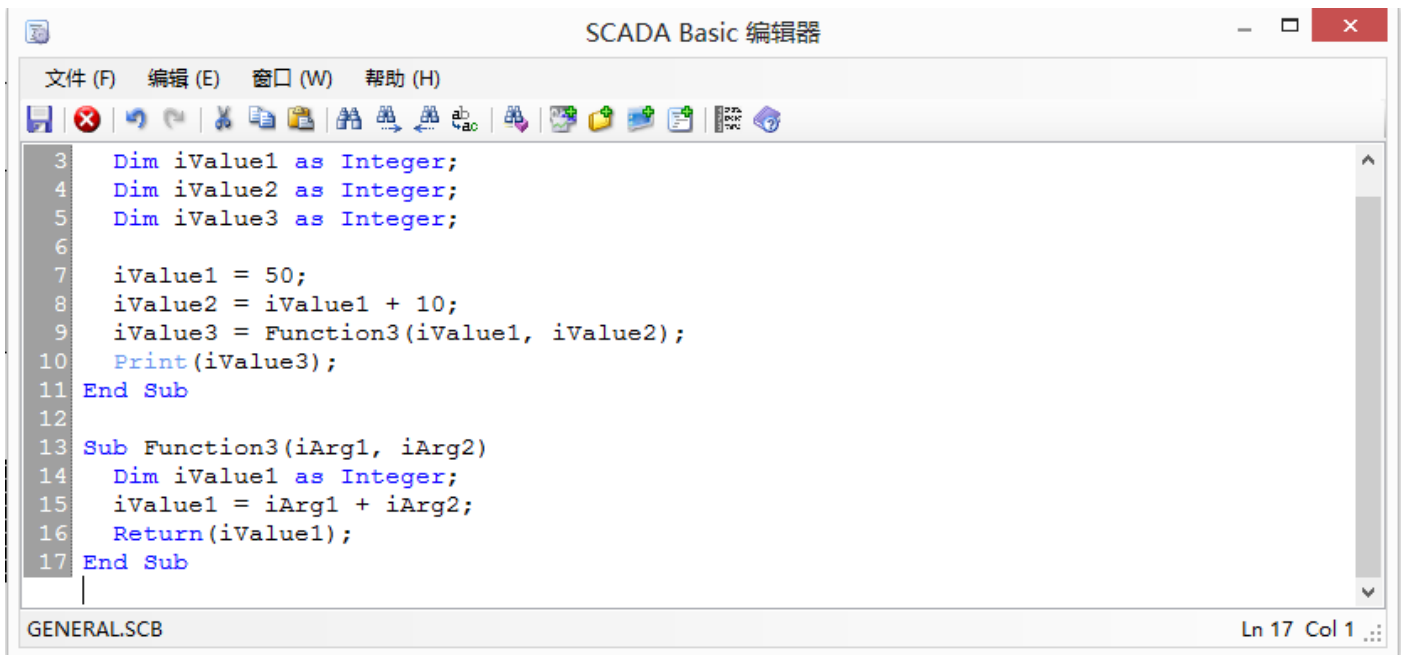


图 15

```
Sub Main()
    Dim iValue1 as Integer;
    Dim iValue2 as Integer;
    Dim iValue3 as Integer;

    iValue1 = 50;
    iValue2 = iValue1 + 10;
    iValue3 = Function3(iValue1, iValue2);
    Print(iValue3);
End Sub

Sub FUNCTION3(iArg1, iArg2)
    Dim iValue1 as Integer;
    iValue1 = iArg1 + iArg2;
    Return(iValue1);
End Sub
```

上述脚本将获得以下结果：

110

4.3.2 变量树变量

用户可以与内部变量一样在程序中使用变量树变量，但是需要遵循一些规则。

- ★ 变量树变量能够直接使用它们的全称来调用。
- ★ 如果变量需要在程序中被修改，那么该变量必须勾选“命令”选项。
- ★ 所有变量树变量都可以在程序中使用，因为它们是公共的变量。
- ★ 在程序中引用变量树变量时，必须保证程序中的变量类型是以下类型：
 - 开关量：整型
 - 模拟量：单精度实型（默认情况下为单精度，但是可以在高级选项中改成双精度实型）
 - 文变量：字符串型 (Str)

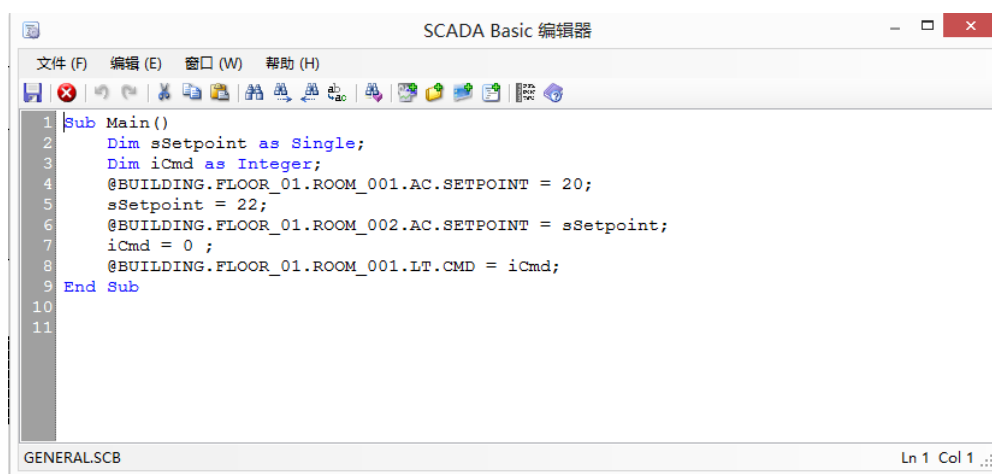


图 16

```
Sub Main()  
    Dim sSetpoint as Single;  
    Dim iCmd as Integer;  
    @BUILDING.FLOOR_01.ROOM_001.AC.SETPOINT = 20;  
    sSetpoint = 22;  
    @BUILDING.FLOOR_01.ROOM_002.AC.SETPOINT = sSetpoint;  
    iCmd = 0 ;  
    @BUILDING.FLOOR_01.ROOM_001.LT.CMD = iCmd;  
End Sub
```



可以使用命令 **编辑 / 高级 / 变量...** 或 **CTRL + I, CTRL + V** 组合键来插入数据库变量。然后双击所要变量来插入。

4.4 内置指令

PcVue 提供超过 150 个的内置指令。因为他们由全局程序装载，并可用于所有程序。

使用内置指令的语法与函数的一致。

本课程的目的不是学习所有指令而是教大家如何找到所需要的函数及其应用。

事实上我们将要学习如何使用 SCADA Basic 帮助文档。

4.4.1 查找正确指令

编程中常见的问题是如何找到相应的指令来实现所需要的功能。在 SCADA Basic 帮助中有一节 可以帮到用户：指令分类。

按以下方式可以找到： [编程语言 / Scada Basic / 语言参考 / 指令分类](#)。

如果点击[指令分类 / 关于指令](#)主题将会看到分类列表。然后点击该分类将可以看到相应的指引列表。

比如，如果在脚本中需要使用当前日期和时间，可以使用 [时间和日期](#) 分类中的函数。看了之后，你就会发现 DATETIME 指令将符合你的需求。

时间和日期

参见 指令

DATETIME

DATETIMESTRING

DATETIMEVALUE

TOHMS

作用

返回一个日期和时间字符串的组成。

将时间和日期（以DOUBLE保存）转换成一串字符。

返回从1980年起的毫秒数。

将秒值转换成格式：时：分：秒。

图 17

4.4.2 指令的使用

当找到了正确的指令，那么就需要知道如何使用这些指令。在 SCADA Basic 帮助中有一节：指令从 A 到 Z 的参考。

按以下方式找到： [编程语言 / Scada Basic / 语言参考 / 指令从 A 到 Z 的参考](#)。

当选择一个指令是，将在右手面板中显示该指令的帮助。页的格式是一样的：

- ★ 最开始是，一个[参见](#) 超链接显示与该指令相关的指令和一个[示例](#) 超链接来显示实际例子。
- ★ 起始部分将是该函数的简介。
- ★ 紧接着是该函数的各种实用方式。**方式**一直都是函数的第一个参数，它可以用数字或助记符表示。



我们建议用户使用助记符，因为助记符简洁明了，一看就知道使用哪个方式。否则的话，将需要查找帮助查看相应数字的含义。

- ★ 每个方式，都有相应的语法格式。
- ★ 各语法的完整解释。
 - 语法里显示着由“[]”代表的可选参量。
 - 返回值
 - 使用该语法时各方式的含义
- ★ 使用该函数的例子。

以下截图显示着 DATETIME 函数的帮助首页，带有各方式及第一个语法。

DATETIME		
参见 示例 返回一个日期和时间字符串的组成。		
方式	助记符	语法
1	DAY	<u>1</u>
2	MONTH	<u>1</u>
3	YEAR	<u>1</u>
4	HOUR	<u>1</u>
5	MINUTE	<u>1</u>
6	SECOND	<u>1</u>
7	MILLISECOND	<u>1</u>
8	WEEKDAY	<u>1</u>
9	YEARDAY	<u>1</u>
10	ISLEAPYEAR	<u>1</u>

语法1	
IntVal = DATETIME([方式 [, DateTime]]); 返回类型为 INTEGER.	
自变量	含义
DateTime	要运算的日期和时间。如果不提供，使用当前日期和时间。类型STR。

执行		
方式	助记符	作用
1	DAY	返回年月日的日（1至31）。
2	MONTH	返回月份（1至12）。
3	YEAR	返回年份（1980至2106）。
4	HOUR	返回小时（0至23）。
5	MINUTE	返回分钟数（0至59）。
6	SECOND	返回毫秒数（0至999）。
7	MILLISECOND	返回星期（0至6：星期天=0，星期一=1等）。
8	WEEKDAY	返回星期（0至6：星期天=0，星期一=1等）。

图 18

4.5 自定义窗口

报警，日志和趋势窗口提供带有许多工具的工具栏，以供用户操作。但或许工具栏看起来不符合您的品味，或者不服务与您客户端的特殊要求？例如，您不想使用报警窗口的工具栏，取而代之使用两个按钮：一个用于确认选择的报警，一个用于打印。



图 19

要解决这个问题，PcVue 提供三条内部指令来自定义窗口。

窗体	内部指令
报警窗口	ALARMDISPLAY
日志窗口	LOGDISPLAY
趋势窗口	TREND

在 PcVue 当前版本中，ALARMDISPLAY 有 46 种模式！！

提取自帮助文档

26	FILTER	2
27	ACK_SELECTED	1
28	ACK_DISPLAY	1
29	MASK_SELECTED	1
30	UNMASK_SELECTED	1
31	PRINT_SELECTED	1, 11
32	PRINT_DISPLAY	1, 11
33	PROGMASK	4

图 20

快速查看后，我们发现模式 ACK_SELECTED 和 PRINT_SELECTE 符合我们的要求。

★ ACK_SELECTED 使用语法 1:

ALARMDISPLAY(*Mode*, *Window*, *Branch*, *Identity*);

- *Mode*: ACK_SELECTED 或 PRINT_SELECTED;
- *Window*: 带有报警窗口的动画控件;
- *Branch*: 动画控件的分支;
- *Identity*: 报警窗口的 ID 号。每个窗体都有一个 ID 号，因为用户可以在同一个窗口中添加多个窗体。要找个该窗体的 ID 号，必须使用绘图浏览器。可以通过点击 ID 号来修改。



图 21

★ PRINT_SELECTED 同样使用语法 1:

ALARMDISPLAY (*Mode, Window, Branch, Identity*); 也可以选择语法 11:

ALARMDISPLAY (*Mode, Window, Branch, Identity [, PrintFormat]*);

除了带有可选参数 PrintFormat 用于改变报警打印的方式外, 该语法与语法 1 类似。在本例中, 我们使用语法 1。

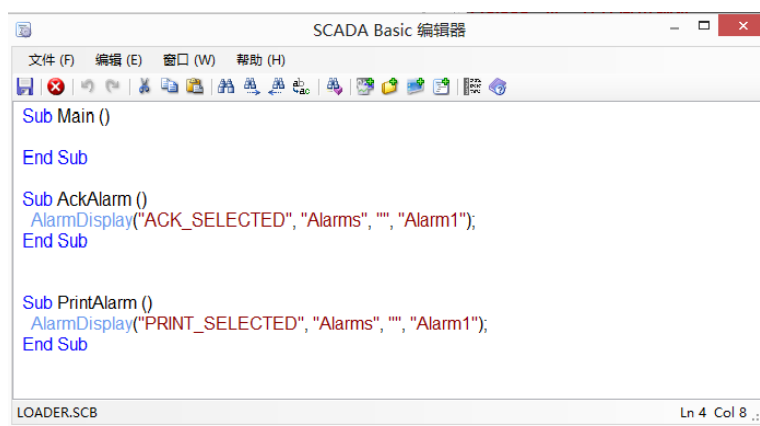


图 22

4.6 触发函数的不同方式

有多种方式来触发函数：

- ★ 通过事件来触发
- ★ 通过按键组合来触发
- ★ 周期性（循环）
- ★ 使用时间表来触发
- ★ 从另外一个程序中触发

4.6.1 事件

配置一个事件来运行 SCADA Basic 程序：

步骤 1. 选择命令 组态 / 操作 / 事件...

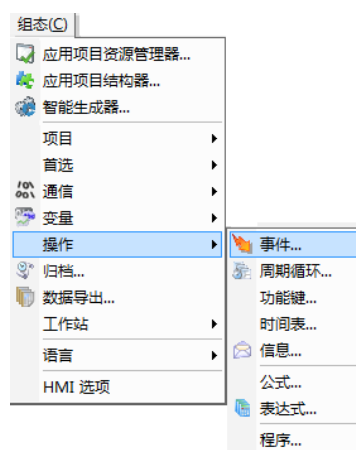


图 23

提示：用户也可以在应用项目资源管理器中进行配置。

步骤 2. 点击命令 添加事件来触发程序...

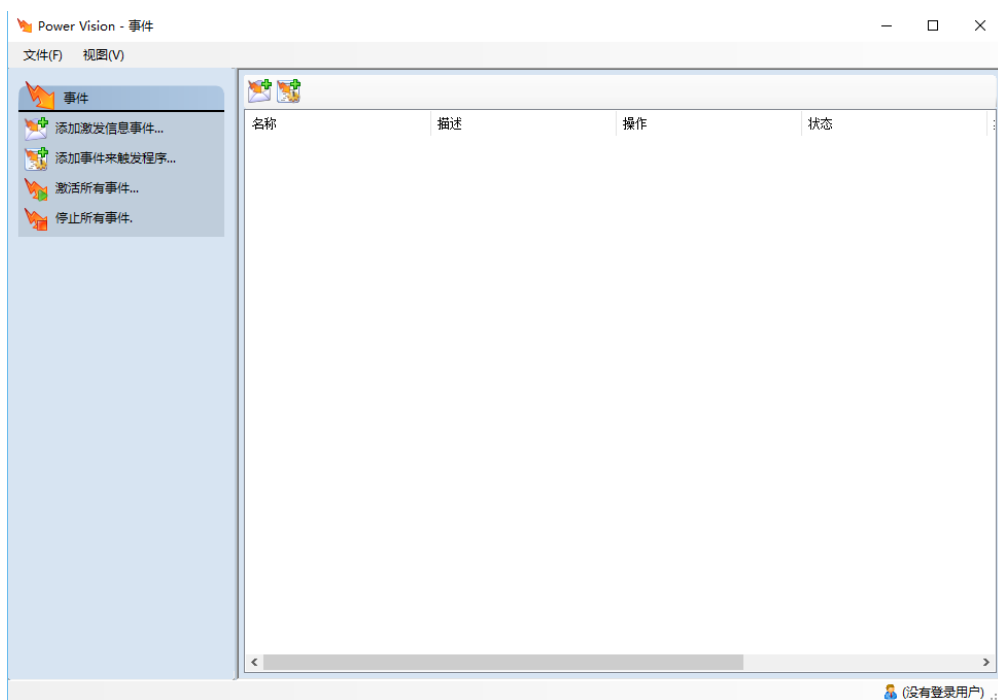


图 24

- 步骤 3.** 选择触发选项卡。
- 步骤 4.** 选择触发函数的变量。
- 步骤 5.** 在 触发 选项卡中配置传输条件。



图 25



可用的传输条件的描述可以在帮助文档中找到：组态 / 周期操作/ 事件触发表达式。

步骤 6. 在程序 选项卡中选择程序，分支和函数。

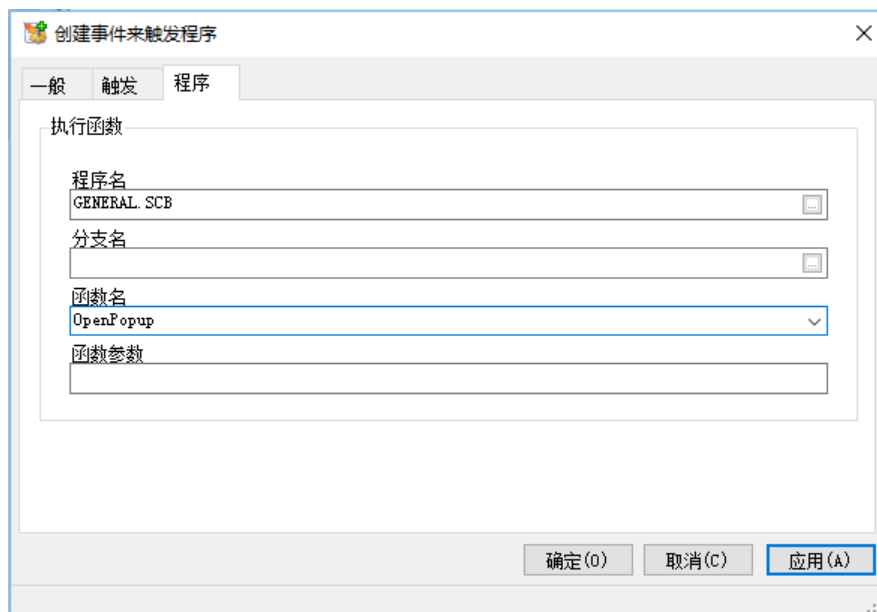


图 26

步骤 7. 点击 确定 按钮。



通过事件调用程序，该程序严格禁止使用内置指令 **DELAY**。

4.6.2 按键组合

配置一个组合键来触发函数：

步骤 1. 选择命令 组态/操作/功能键...

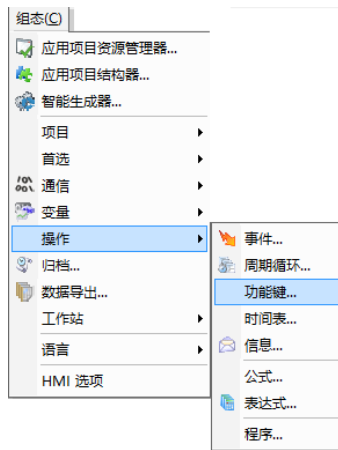


图 27

步骤 2. 点击 创建... 按钮。

步骤 3. 选择触发器。

步骤 4. 选择 执行功能 选项。

步骤 5. 选择好程序，分支和函数。

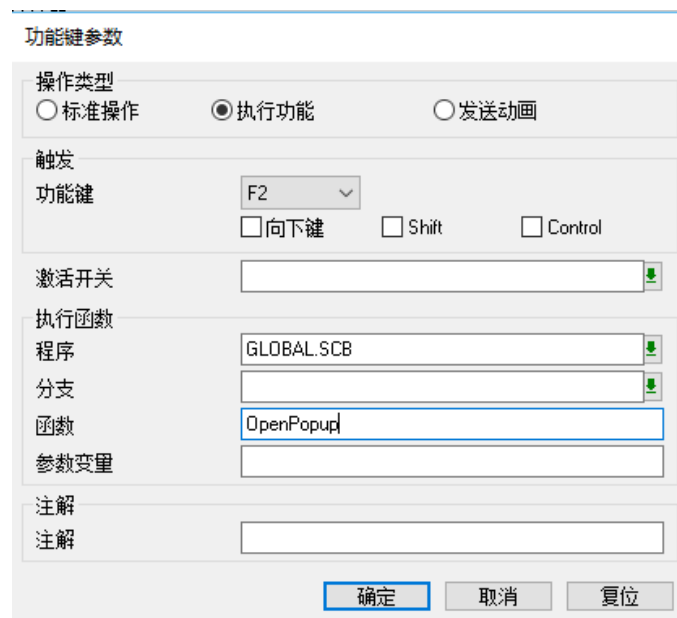


图 28

步骤 6. 点击确定按钮并和保存按钮。

4.6.3 周期循环

配置一个周期操作来触发函数：

步骤 1. 选择命令 配置 / 操作 / 周期循环...

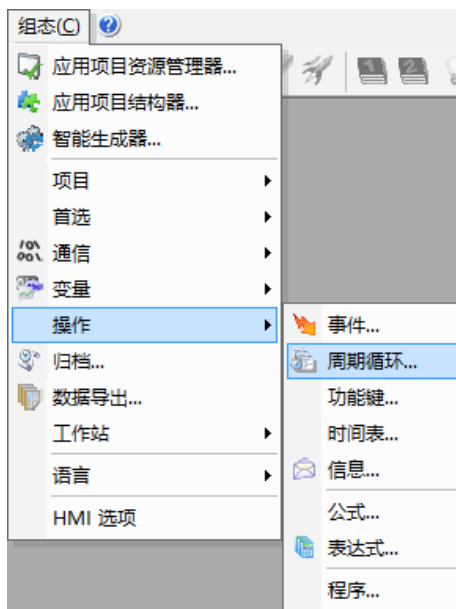


图 29

提示：用户也可以在应用项目资源管理器中配置。

步骤 2. 在工具栏，点击添加周期... 按钮。

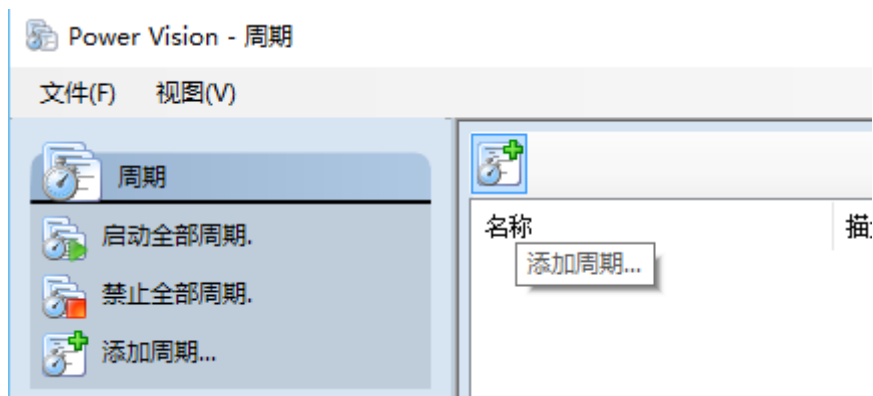


图 30

步骤 3. 在触发选项卡，输入周期来触发函数。

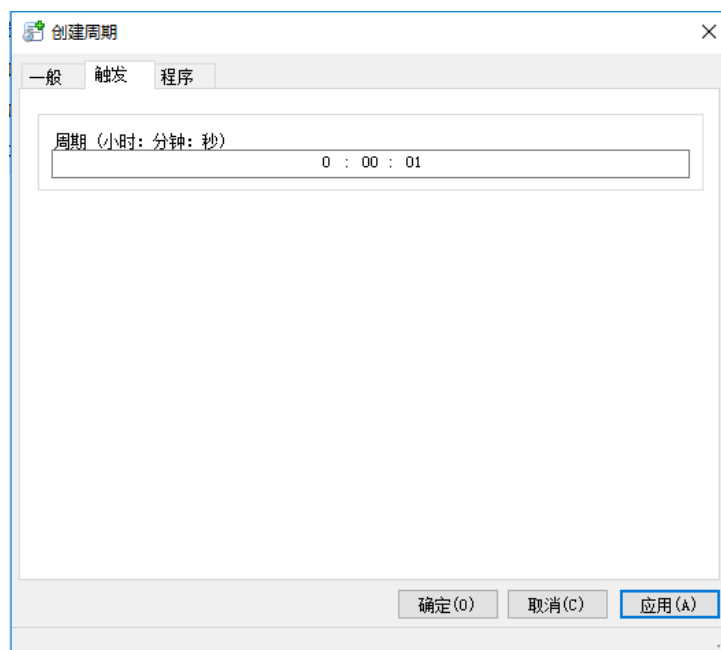


图 31

步骤 4. 在程序选项卡，输入程序，分支和函数。

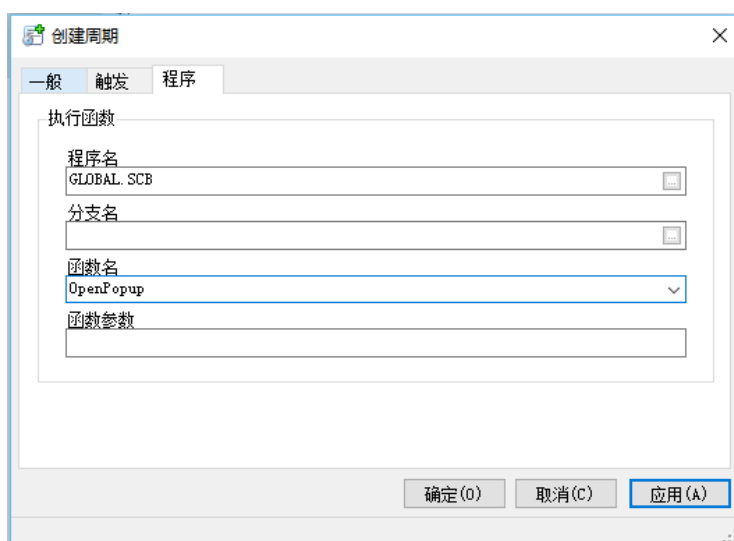


图 32

步骤 5. 点击确定按钮和保存按钮。



通过周期循环操作调用的程序严格禁止使用内置指令 DELAY 。

4.6.4 时间表

配置一个时间表操作来触发函数：

步骤 1. 打开命令 组态/操作/时间表...

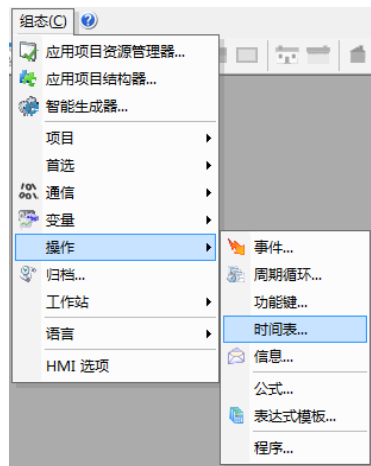


图 33

步骤 2. 点击 创建... 按钮。

步骤 3. 选择函数 操作类型。输入程序，分支和函数。

步骤 4. 在议程 中选择所要触发程序的频率和时间。

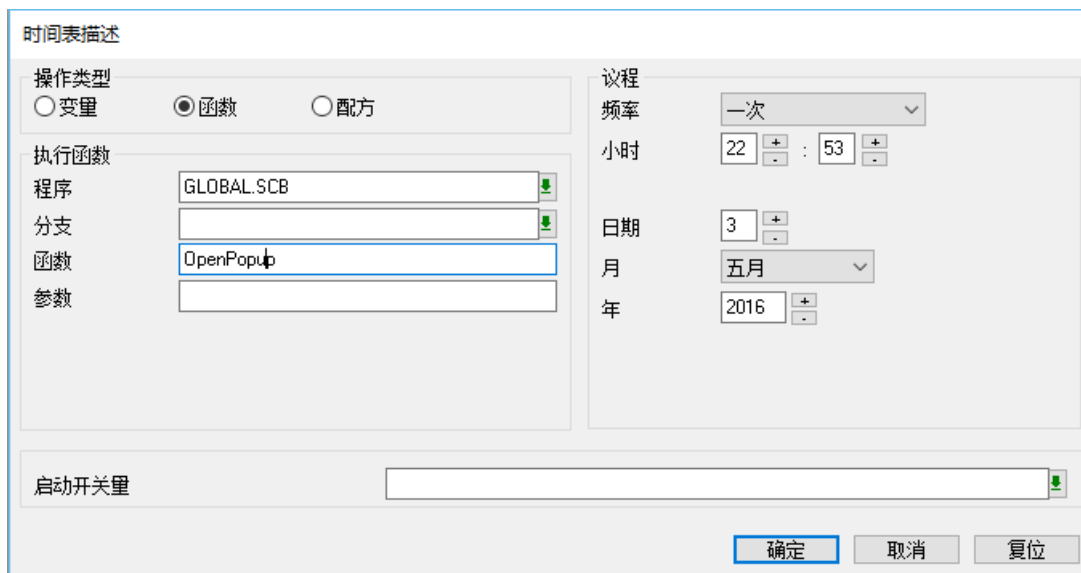


图 34

步骤 5. 点击确定按钮和保存按钮。

4.6.5 从另一个程序调用函数

用户可以使用内置指令中的 **PROGRAM** 指令来调用别的程序中的函数。但是要注意的是，只在调用函数结束时才调用该函数，该调用被称为**异步**调用。

在帮助中可以找到：

PROGRAM		
参数	示例	
执行、加载、卸装或停止一个程序或函数。		
方式	助记符	语法
0	STOP	<u>1</u>
1	START	<u>1</u> , <u>3</u>
2	IS_LOADED	<u>1</u>
3	PRELOAD	<u>1</u> , <u>3</u>
4	UNLOAD	<u>1</u>
5	EXECUTE	<u>1</u>
9	FUNCTION	<u>2</u>
10	IS_FUNCTION	<u>2</u>

图 35

我们所要使用的模式时 **FUNCTION** 的语法 2。

语法 2: PROGRAM(*Mode*, *ProgName*, *Branch*, *Function*[,*Farg*])

- *Mode*: FUNCTION。
- *ProgName*: 该函数所在的程序名称。
- *Branch*: 程序装载的分支。
- *Function*: 所要执行的函数。
- *Farg*: 可选参数。参数列表。

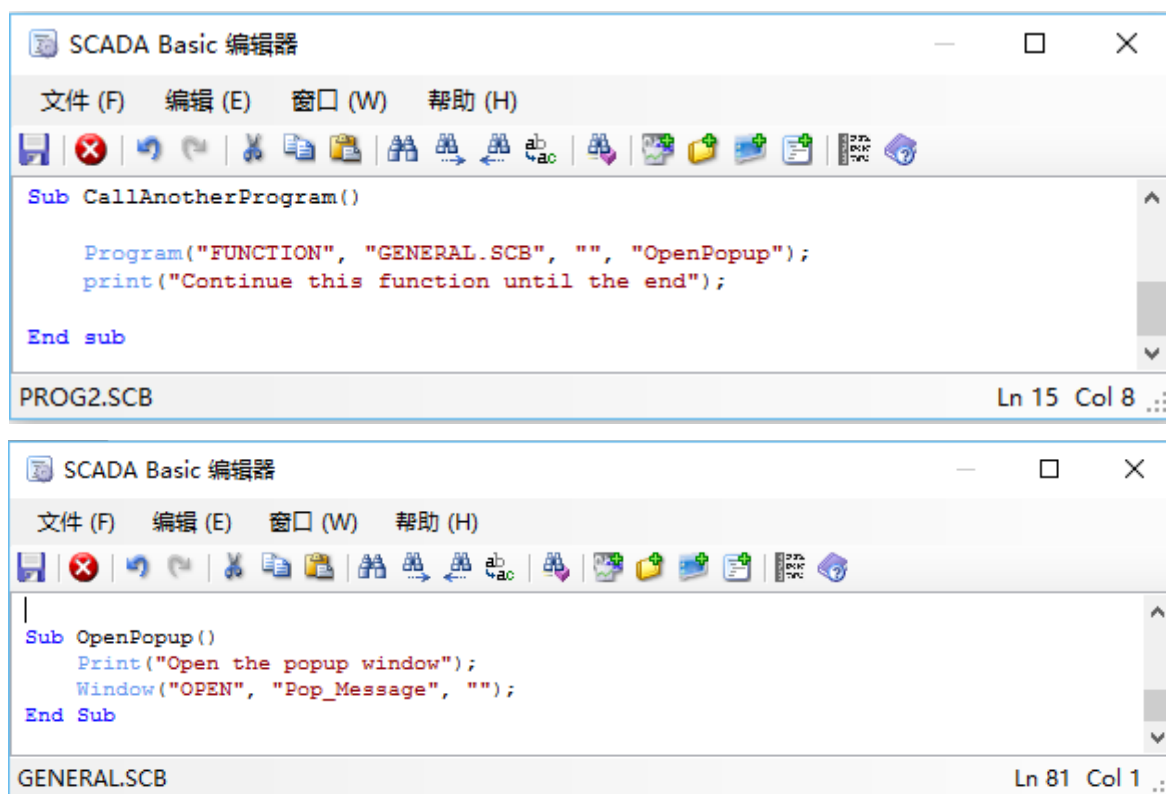


图 36

```
Sub CallAnotherProgram()  
Program("FUNCTION", "GENERAL.SCB", "", "OpenPopup");  
Print("Continue this program to the end");  
End Sub  
  
Sub OpenPopup()  
Print("Open the popup window");  
Window("OPEN", "Pop_Message", "");  
End Sub
```

以上脚本将会得到以下结果：

Continue this function until the end
打开 popup 窗口。

5 总结

- ★ SCADA Basic 是一个完整、功能强大的编程语言，它提供了几乎所有 PcVue 对象的编程接口。
- ★ SCADA Basic 是程序的集合。
- ★ 程序是函数的集合。
- ★ 函数可以接收多个参量作为输入，也可以返回一个数值作为输出。
- ★ 如果需要使用内置指令，全局程序是必须的。
- ★ 所有在全局程序中声明的函数和变量都是公共的。
- ★ 帮助文档可以帮你找到所有内置指引的使用说明。