硕人时代

STEC2000 控制器

用户手册

二次开发环境说明书

STEC2000 二次开发环境说明书

概述

STEC 控制器的二次开发通过高效、灵活的脚本语言——SRScript 来实现。 SRScript 是硕人科技开放给控制工程师和二次开发用户的专用脚本语言，具有严谨的语法及规则解释，提供了从界面显示到控制状态等多种功能强大的函数及命令接口，并支持同时在实时或非实时状态下运行。使用 SRScript，用户可以简单、快速的实现按自己的需求对控制器的功能进行定义。

SRScript 同时也提供了强大的调试环境，开发人员可以方便的对自己的程序进行调试。

阅读对象

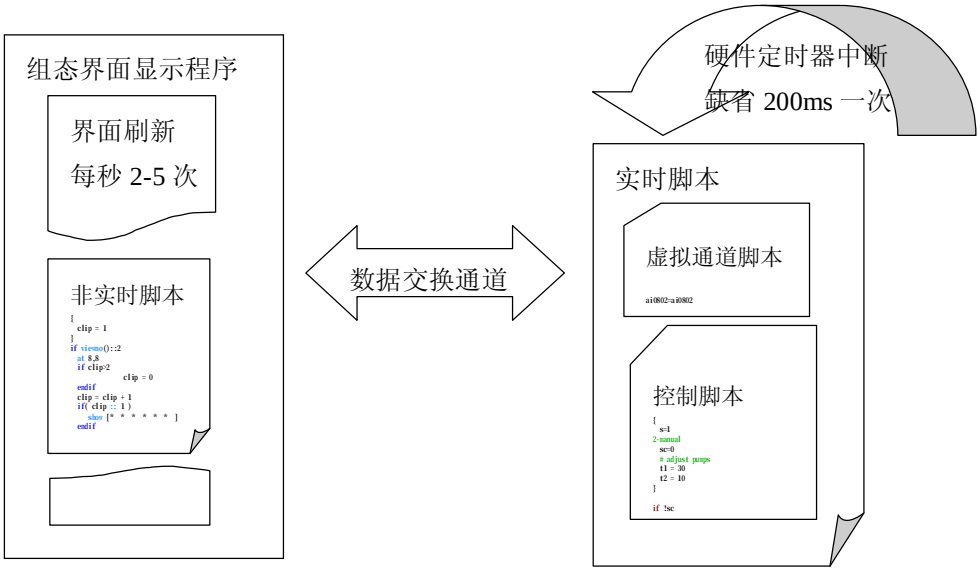
该说明书为控制工程师二次开发时的函数速查手册；

同时也是工程师系统了解 STEC 控制器二次开发支持的手册。

运行机制

SRScript 程序从运行状态上可分两类：实时脚本和非实时脚本。在控制器软件系统启动后，实时控制脚本先运行，之后非实时脚本开始运行。

下图为 SRScript 运行机制示意图：



- 实时脚本：分为实时控制脚本程序和虚拟通道脚本程序。
  - 实时控制脚本程序在每个控制周期（STEC2000 标准控制周期为 200ms）中都会被执行一次，且执行时不会被中断。因此在此脚本中实现控制器的实时性要求较高的控制策略，例如随 AI/DI 输入量的瞬时变化来设置 AO/DO 的输出值。
  - 虚拟通道脚本中的输入量，如虚拟 AI、DI，在控制器软件系统取虚拟通道值时被调用，脚本程序的最后一条语句值为本虚拟通道值（返回值），执行过程也不会被中断。
  - 虚拟通道脚本中的输出量，如虚拟 AO，DO，在此通道输出最置值，包括系统置值和脚本函数置值，时最调用。调用时，内置脚本变量 param 会被置为所设参数值。

  注意：每个控制器中只能存在一个实时控制脚本程序，但可以随系统中配置的虚拟通道数而拥有多个虚拟通道脚本程序。

- 非实时脚本：在 STEC 监控画面相关程序运行时被不断重复执行。可以在此脚本程序中实现监控画面表现、人机界面交互功能和对实时性要求不高的控制策略。

脚本说明

以下内容是 SRScript 的详细介绍，由于非实时脚本与实时脚本在功能上及运行状态都有所不同，所以在语法上也有区别，我们对不同之处将会分别说明。

非实时、实时相同部分

常量：

脚本中支持数字常量，支持科学记数法。数字常量在内部全部作为浮点数来处理，1 对应布尔型的“真”，0 对应“假”。

例：以下均为合法的常数：

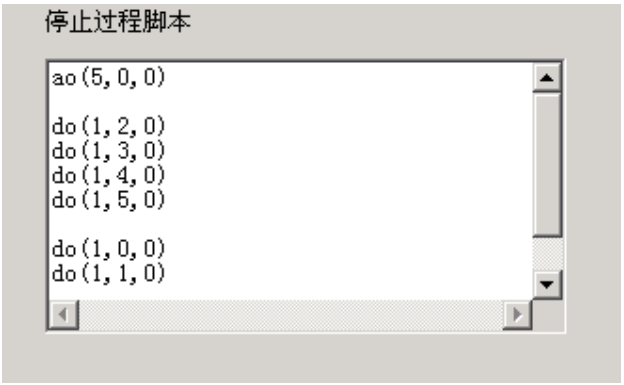
1.25　1　0.25　1e5

保留字：

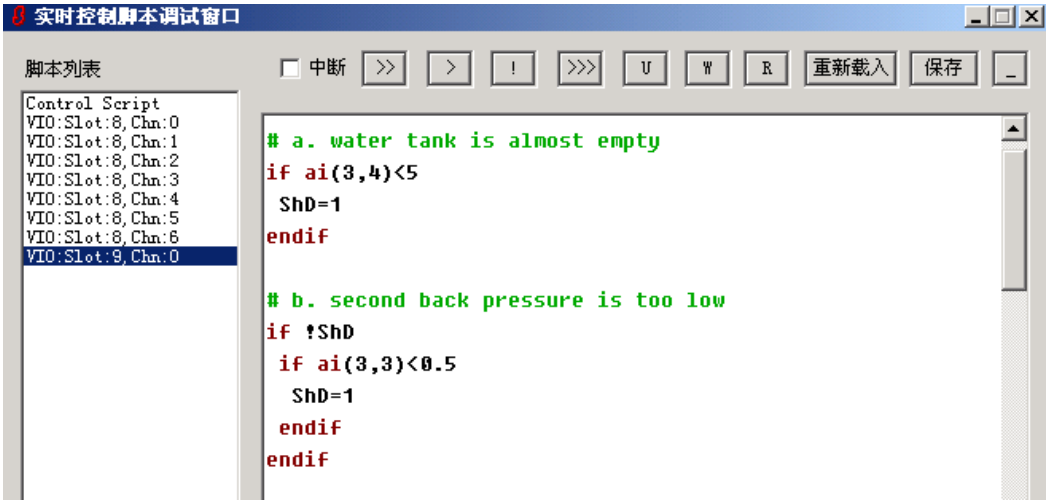
SRScript 中的保留字是指在任何脚本程序的任何地方都可以直接引用的一些特殊变量名，它对应一些常用的固定值或者是当前系统相关的变化值，如时间信息等。在命名其它变量时不能再使用保留字。SRScript 中支持的保留字如下：

e 或 E：自然指数值
pi 或 PI：圆周率

ShD ：这是一个内置的紧急状态的脚本变量。会在最初被设定为 0。当此变量被设定为大于 0.1 时，系统会启动在组态环境中的停止过程脚本并停止运行。
例：某工程要求出现紧急故障时将接在 AO(5,0)的蒸汽调节阀关闭，并将接在 DO 0 至 6 通道上的的泵与阀关闭。
停止过程脚本定义如下：



在实时脚本中当凝水箱水位（接在 ai(3,4)）过低或二次回水压力过低时，将 ShD 置为 1，则系统会进入紧急停止状态。



变量：

脚本中变量的命名与其它高级语言基本相同，不支持下划线，必须以字母开头，可包含数字，但长度不能超过 10 个字符。

例：下面均为合法的变量名：

OutdoorT

Supply

FlagWarn

X1

Y1

注意：

- 变量为大小写敏感；

  如 supplyT 和 SupplyT 会被认为是不同的变量。
- 变量无需指定类型；
- 变量名不能与保留字冲突；
- 实时脚本程序中的变量可在非实时脚本程序中使用，可以认为是同一变量，而实时脚本不能引用非实时脚本的变量。 当非实时脚本执行时遇到不可识别的变量时，其与实时脚本通讯，查询实时脚本中是否有此变量，如果有，则非实时脚本将这一变量作为实时脚本的引用变量。对该变量的操作与一般没有什么不同。这一机制主要用于非实时脚本与实时脚本的通讯。例如：实现根据用户输入的值，修改供水温度设定值:

  TSet 是实时脚本中供水温度设定值，TInput 是非实时脚本中用户的输入值。要改变供水温度的设定值，只须下面这条语句即可：

  TSet = TInput

表达式：

支持标准表达式，运行符函数可嵌套。

算术运算支持 符： +、-、*、/

关系运算符：>, <, >=, <=

逻辑运算符：！——非，& ——与， | ——或

其它运算符：（,）

注意：

为了增加实时脚本的高度可靠性，实时脚本不支持等号关系运行符。非实时脚本支持等号运算符，参见“非实时脚本专有语法”。

在编写实时脚本时应尽量减少嵌套。

例：下面均为合法的表达式：

```
hour*60+minute
0.5 * ai(2,0)/(tin + tout)
```

```
(di(1,0)|di(1,3))&(ai(0,0)>5.0)
```

程序块

条件执行块 —根据条件判断执行的语句序列

if （条件）

语句 1

……

else //可以没有

或 else if //注意 else 和 if 之间有空格

语句 2

……

endif //注意 end 和 if 之间没有空格

说明：

当条件为真（非 0）时执行语句 1 及相应序列

当条件为假（0）时执行语句 2 及相应序列

例：下面的程序段实现了供水温度的分段控制

```
if( OutdoorT > 0)
    if( OutdoorT > 5 )
        # 5 度以上时
        Tset = 60
    else
        #0 至 5 度时
        Tset = 65
    endif
else
    if(OutdoorT <-5)
        #-5 度以下
        Tset = 78
    else
        #-5 到 0 度
        Tset = 73
    endif
endif
```

循环块 —循环执行循环体中语句序列

loop（n）

循环体语句

……

endl

说明：

直接执行循环体语句序列 n 次，n 可以是表达式。

例:

下面的非实是脚本程序在屏幕上显示插在第 1 槽上 AI 卡上各通道的值:

```
iChn = 0
loop(8)
    at iChn+1,1
    output(ai(1,iChn))
    iChn = iChn + 1
endl
end
```

初始化段

以 { } 包围的语句段为初始化段，在脚本程序中只能出现一次，并只会在程序初始运行时执行一次（也就是说在控制器上运行时，每重启一次控制器，这部分代码才会被执行）。这实时脚本中，这部分代码会最先执行。虚拟通道脚本中不能有初始化段，所需要的初始化工作应在实时控制脚本中的初始化段中完成。

例: 下面的非实时脚本的初始化段将保存的几个参数载入：

```
{
    load Tset2, ValPosSet
}
```

关于 load 词句的使用请参看相关说明。

定时执行块

```
$var
语句 1
…
}
```

包括起来的脚本段将每 var 秒执行一次，var 可以是常量或变量。

例：下面的程序每 30 分钟改最一次二次供水温度的设定值：

```
$1800
    if(OutdoorT>0)
        Tset2 = 65
    else
```

```
        Tset2 = 70
    Endif
}
```

函数：

- ai

格式：ai(slot,chn)

功能：取得 ai 卡 slot 槽， chn 通道的模拟输入量

参数：slot—槽口号 chn—通道号

返回值：第 slot 槽，第 chn 通道对应的模拟输入量

说明：

返回值为转换后的量；

slot,chn 均从 0 开始，可以是虚拟的槽口号，通道号

例：

下面的词句将第 0 槽 第 2 通道的转换后量赋给变量 roomT

```
roomT = ai(0,2)
```

- di

格式：di(slot,chn)

功能：取得 di 卡 slot 槽， chn 通道的开关输入量

参数：slot—槽口号 chn—通道号

返回值：1—开 0—关

说明：

slot,chn 均从 0 开始，可以是虚拟的槽口号，通道号

例：

设有一块 DI 卡插在第 2 槽上，其第 3 通道联接着低水位开关，下面的程序段实现当低水位时开启联接在第 6 槽第 2 通道的补水泵：

```
if di(2,3)
   do(6,2,1)
endif
```

do 函数的用法参见 do 函数。

- ci

格式：ci(slot,chn)

功能：取得 slot 槽， chn 通道的计数器输入量

参数：slot—槽口号 chn—通道号

返回值：第 slot 槽，第 chn 通道对应的计数器输入量

说明：

返回值为转换后的量；

slot,chn 均从 0 开始，可以是虚拟的槽口号，通道号

ci 通道是可选的 DI 卡的前三个通道，可在组态环境中配置。

例：下面的语句将第 2 槽第 0 通道计数器测量转换后的值赋给变量 flow

```
flow = ci(2,0)
```

- ao

格式：ao(slot,chn,v)

功能：设置 ao 卡 slot 槽， chn 通道的模块输出为 v

参数：slot—槽口号 chn—通道号

返回值：无

说明：

变量 v 乘以配置文件中相应的修正系数为最终输出值。

当该通道配置为电压输出时，v 的范围是 0 至 10，单位是伏。当该通道配置为电流输出时，v 的范围是 0 至 20，单位是毫安。

例：

设一块 AO 卡插在第 7 槽第 0 通道，联接在一台变频器的输入端，配置为电压输出。该变频器接受 0 至 10 伏电压输入，对应频率输出为 0 到 50Hz，下面的词句将此变频器的频率设定为 30Hz:

```
ao(7,0,6.0)
```

- do

格式：do(slot,chn,v)

功能：设置 do 卡 slot 槽， chn 通道的模块输出

参数：slot—槽口号 chn—通道号 v—输出值

返回值：无

说明：

当 v 为 0 时通道输出置为关，当 v 为 1 时通道输出置为开

例：

设有一块 DI 卡插在第 2 槽上，其第 3 通道联接着低水位开关，下面的程序段实现当低水位时开启联接在第 6 槽第 2 通道的补水泵：

```
if di(2,3)
    do(6,2,1)
endif
```

do 函数的用法参见 do 函数。

- abs

格式：abs(v)
功能：得到变量 v 的绝对值
参数：需要返回绝对值的变量名
返回值：变量 v 的绝对值
说明：

- ln

格式：ln(v)
功能：返回 v 的自然对数
参数：v
返回值：v 自然对数
说明：

- log

格式：log(v)
功能：返回 v 的对数
参数：v
返回值：v 对数
说明：

- sin

格式：sin(v)
功能：正弦函数
参数：v:角度（角度制）
返回值：v 的正弦
说明：

- cos

格式：cos(v)
功能：余弦函数
参数：v:角度（角度制）
返回值：v 的余弦
说明：

- tan

格式：tan(v)
功能：正切函数
参数：v:角度（角度制）

返回值：v 的正切
说明：

- min

格式：min(x1,x2)
功能：返回两数的最小值
参数：x1,x2
返回值：x1,x2 中的最小值
说明：

- max

格式：max(x1,x2)
功能：返回两数的最大值
参数：x1,x2
返回值：x1,x2 中的最大值
说明：

- sqrt

格式：sqrt(v)
功能：计算数的开方
参数：v
返回值：v 的开方
说明：

- pow

格式：pow(x,p)
功能：指数运算
参数：x：根，p:幂
返回值：x 的 p 次幂
说明：x 和 p 均可为浮点数

- setidlim (ID,uppper,down)

功能：设定 ID 所对应的数据源（Data_tag.ID）的量程
参数：ID－表示要改变的 Data_tag.ID
upper－表示上限
down－表示下限
返回值：无
说明：在组态环境中也可设置量程，此函数用于量程在控制器运行之后还需要变化的应用。

- setwarn (ID,uppper,down)

  功能：设定 ID 所对应的数据源（Data_tag.ID）的报警上下限。

  参数：ID－表示要改变的 Data_tag.ID

  upper－表示上限

  down－表示下限

  返回值：无

  说明：在组态环境中也可以设置报警上下限，此函数用于报警上下限在控制器运行之后还需要改变的场合。

- xv(index)

  脚本提供 1000 个可供使用的通用变量，编号为 0 至 999。通过 xv 函数可以得到其值。

  例：

  xv(100)

  返回第 100 个通用变量的值。

- sxv(index,value)

  脚本提供 1000 个可供使用的通用变量，编号为 0 至 999。通过 sxv 函数设定其值。

  如：

  sxv(100，12.6)将第 100 个通用变量的值设为 12.6。

其它：

- **注释：**

  脚本程序的注释行以＃开头，注释行中内容将不会被解析执行。本脚本只支持行注释。

- **语句：**

  脚本程序的每行为一条完整的执行语句，不支持换行符。每行字符个数最多为 128 个。

  脚本程序以小写 end 结束，即在所有语句之后，一定有单独的 end 作为终结符。

注意事项：

由于脚本在控制器上是解析运行，特别是实时控制脚本，是在实时态解析运行，因此应注意以下几点来保证脚本执行效率：

1） 命名的变量名应尽可能的短

2） 常数尽可能短

3） 嵌套层次尽可能浅

实时脚本专有部分

有部分语法和函数等只在实时部分有效，说明如下：

保留字：

err：表示调用 I/O 函数后，函数执行状况。

err 为 0 表示执行正常，不为 1 表示出错。

注意： err 为只读保留字。不可用作它用。

该保留字不能在非实时脚本中使用。

函数：

- pidon

格式：pidon(no)

功能：设置第 no 路 pid 控制为开

参数：no—对应于组态环境中配置的 pid 控制路数序号，序号从 1 开始

返回值：无

说明：

系统运行时 pid 控制缺省值为打开，本函数用于打开已经确定被关闭的 pid 控制

例：

下面的程序段实现第一路 PID 控制在外温（接在 ai(0,0)）在-5 至+5 内时有效。

```
if( ai(0,0)>5|ai(0,0)<-5)
    pidoff(1)
else
    pidon(1)
endif
```

- pidoff

格式：pidoff(no)

功能：设置第 no 路 pid 控制为关

参数：no—对应于组态环境中配置的 pid 控制路数序号，序号从 1 开始

返回值：无

说明：

系统运行时 pid 控制缺省值为打开，本函数用于关闭打开的 pid 控制

例：

见 pidoff 例

- pidv

格式：pidv(no,v)

功能：设置第 no 路 pid 控制值为 v，序号从 1 开始

参数：no—对应于组态环境中配置的 pid 控制序号　v—pid 控制值

返回值：无

说明：

系统运行时 pid 控制缺省设定值为在组态环境中的设定值。可此函数可在动行过程中改变设定值。

例：

下面的语句按公式将第一个 PID 控制回路的设定值设定为室外温度（接在 ai(0,0)）的函数。

```
$300
    pidv(1,18+0.1*ai(0,0))
}
```

- pidcoef

格式：pidcoef(no,coefP, coefI, coefD), 序号从 1 开始

功能：设置第 no 路 pid 控制的比例、积分和微分系数

参数：no—对应于组态环境中配置的 pid 控制序号

coefP:比例系数

coefI:积分系数

coefD:微分系数

返回值：无

说明： 此函数用于运行过程中改变比例、积分和微分系数

- bvws

发布时间：2003-7-30

格式：bvws(sw)

功能：AI 量和计数器量超量程时是否产生报警的开关

参数：sw—为 1 时产生报警，为 0 时不产生报警

返回值：无

说明： 此函数用于关闭或开启原始测量值超量程报警功能。

脚本命令：

l 延时赋值

功能：指定某变量值在一定时间后变为指定值。

语法格式：

变量名=值 ~ 时间延迟（单位为秒）

例如：

a=17 ~ 2

变量 a 将在 2 秒钟后变为 17

说明：

当同一个变量被不同的词句指定延迟赋值时，只有离现在最近时间的赋值起作用；

直接赋值语句会立即生效，并取消此变量的延迟赋值。

时间延迟小于半控制周期时，如为 0 时，该语句会取消之前的延时赋值，但不会改变变量的目前值。即 a=0 与 a=0~0 是不等价的。

即用如 a=???~0 的语句可以取消对该变量的延时赋值，但不改变量的当前值。

实时脚本注意事项：

- *实时脚本每控制周期执行一次（STEC2000 的标准设置为 200ms），执行中不会被中断；*
- *实时脚本程序中可命名的变量个数限制为不超过 128 个。*

非实时脚本专有部分

保留字：

year：当前日期年份

month：当前日期月份

day：当前日期天数

hour：当前时间小时数，24 小时制

minute：当前时间分钟数

second：当前时间秒数

例，当前时间为 2002 年 2 月 7 日下午 2 点 35 分 17 秒时，上述各变量值分别为：

year: 2002

month: 2

day: 7
hour:14
minute:35
second:17

常量：

非实时脚本程序中支持字符常量，字符常量由两个单引号括起，中间是为字符，值为 ASCII 码值。例如’c’就是一个合法的字符常量。

表达式：

支持等于关系运算符：::
例：下面的程序等待用户按键，用户按“a”键时，显示“1 键按下”，按其它键则不显示。

```
InputKey = inkey()
if (InputKey::'1')
   show 1 键按下
endif
```

程序块：

循环语句—循环执行循环体中语句序列

```
while(v)
语句 1
…
endw
```

说明：
　　当常量或变量 v 为真（非 0）时持续执行语句序列。
例：　下面的程序不断重复直到“0”键按下

```
key = ‘?’
while(!(key::’0’))
      key = inkey()
      #……

      #….
endw
```

函数：

- color

格式：color(r,g,b)

功能：指定当前前景色

参数：r—红色色值 g—绿色色值 b—蓝色色值

返回值：无

说明：

r, g, b 参数的取值范围为 0—255；

当前前景色设置影响之后所有的绘图操作。

缺省的前景色为兰色，对应 r,g,b 分别为 0,0,255。

改变前景色可改变包括字符在内的颜色。

- colorf

格式：colorf(r,g,b)

功能：指定当前填充色

参数：r—红色色值 g—绿色色值 b—蓝色色值

返回值：无

说明：

r,g,b 参数的取值范围为 0—255；

当前填充设置影响之后所有的绘图的填充操作，包括文字的背景色。

缺省的填充色为黑色。

- format

格式：format(len,decimal)

功能：设定当前数据输出格式

参数：len—输出格式长度　decimal—输出小数点位数

返回值：无

说明：

函数执行后数据输出格式将不会再改变，直到下一次设定时为止

例：　下面的语句将输出格式设定为总长为名，小数点后位数为 2：

format(7,2)

- output

格式：output(v)

功能：在当前绘图位置显示表达式 v

参数：v—需要显示的变量

返回值：无

说明：

可用 at 和 atp 命令行改变当前绘图位置。输出格式用 format 函数设定。

- moveto

格式：moveto(x,y)

功能：移动当前绘图坐标起始点到（x,y）

参数：x，y—坐标点，从 0 开始，范围为 x:[0,319],y[0,239]

返回值：无

说明：

x,y 值可以为 0，单位为象素

- lineto

格式：lineto(x,y)

功能：从当前绘图坐标起始点绘制线段到（x,y）并设定绘图坐标起始点为（x,y）

参数：x，y—坐标点，从 0 开始，范围为 x:[0,319],y[0,239]

返回值：无

说明：

x,y 值可以为 0，单位为象素；

函数执行后当前绘图坐标起始点变为（x,y）；

线段颜色为当前前景色。

- rect

格式：rect(x,y,width,height,bFill,lineWidth)

功能：以坐标（x,y）为左上顶点绘制矩形

参数：x，y—坐标点，从 0 开始，范围为 x:[0,319],y[0,239]

width—矩形宽　　height—矩形高

bFilll—是否填充　　lineWidth—边框宽度

返回值：无

说明：

x,y 值可以为 0，单位为象素；

x,y 为矩形的左上角坐标点；

矩形边框颜色为当前前景色，如果 bFill 不为 0 则以当前填充色填充所绘制矩形。

- circle

格式：circle(x,y,radius,bFill,lineWidth)

功能：以坐标（x,y）为圆心绘制圆

参数：x，y—坐标点　　radius—圆半径

bFilll—是否填充　　lineWidth—边宽度

返回值：无

说明：

x,y 值可以为 0，单位为象素；

圆边颜色为当前前景色，如果 bFill 不为 0 则以当前填充色填充所绘制圆。

- input

格式：input(len)

功能：在当前绘图坐标点等待输入

参数：len—等待输入长度

返回值：输入的字符串或浮点数

说明：

本函数为阻塞方式运行；

输入内容将会在当前绘图坐标点显示；

回车键表示输入完成，最长输入字符个数为 len。

- inkey

格式：inkey()

功能：等待按键输入

参数：无

返回值：输入的按键值

说明：

本函数为阻塞方式运行；

输入的按键不会显示在当前界面中。

注意这是一个阻塞方式运行的函数。

- viewno

格式：viewno()

功能：返回当前显示监控界面在组态环境中组态的序号

参数：无

返回值：组态环境中对应当前监控界面的组态序号 1 至 6，当前显示的画面是主界面（组态配置的顶级菜单）时，返回 0。

说明：

序号取值从 1 开始。

在组态环境中配置的监控画面与非实时脚本是共同运行的。系统在刷新组态生成的监控画面的同时运行非实时脚本。非实时脚本通过 viewno()函数可以知道当前所显示的组态生产的监控画面是第几个，并由此决定执行哪部分程序。

例：

下面的程序在显示组态的第二个监控画面上显示流动的动画效果：

```
{
  clip = 1
```

```
}
if viewno()::2
  at 8,8
  if clip>2
    clip = 0
  endif
  clip = clip + 1
  if( clip :: 1 )
     show [*  *  *  *  *  *  ]
  endif
  if( clip :: 2 )
     show [ *  *  *  *  *  * ]
  endif
  if( clip :: 3 )
     show [  *  *  *  *  *  *]
  endif
endif

end
```

稍加改动即可做出流水等效果。

- chshtime(t)

功能：改变阻塞函数超时的时间。

参数：t—超时的时间，单位为秒。

返回值：无

说明：一般的现场监控希望可自动从阻塞的用户交互中返回。如 inkey()函数执行时，如果用户不输入而离开，且没有超时自动退出机制，则程序将一直等待，而使得脚本中其它功能不能执行。因此，缺省情况下，用户在 10S 之内不操作，非实时脚本将退出此次执行。这个函数可以用来改变超时值。

- setid(Tw,T1h,T2g,T2h,P2h)

功能：设置需要显示的各个量的 Data_tag.ID，并初始化历史数据显示。

参数：Tw—表示室外温度的 Data_tag.ID

T1h—表示一次回水温度的 Data_tag.ID

T2g—表示二次供水温度所对应的 Data_tag.ID

T2h—表示二次回水温度所对应的 Data_tag.ID

P2h—表示二次回水压力所对应的 Data_tag.ID

返回值：无

参看：showhis 函数

说明：这个函数是专为热力控制所提供的。

- showhis()

功能：显示历史记录

参数：无

返回值：无

说明：此函数必须在 setid(Tw,T1h,T2g,T2h,P2h)调用之后才能使用。

这个函数是专为热力控制所提供的。

注意：这是一个阻塞函数

- showfail()

功能：显示故障记录表格。

参数：无

返回值：无

注意：这是一个阻塞函数。

- showwarn()

功能：显示报警记录表格。

参数：无

返回值：无

注意：这是一个阻塞函数。

- setyear(year)

功能：设定控制器时间年份

参数：year－本地时间的年份

返回值：无

- setmonth(month)

功能：设定控制器时间月份

参数：month－本地时间的月份

返回值：无

- setday(day)

功能：设定控制器时间天

参数：day－本地时间的天

返回值：无

- sethour(hour)

功能：设定控制器时间小时

参数：hour－本地时间的小时

返回值：无

- setminute(minute)

  功能：设定控制器时间分钟

  参数：minute－本地时间的分钟

  返回值：无

- setsecond(second)

  功能：设定控制器时间秒

  参数：second－本地时间的年秒

  返回值：无

- warnbar(show)

  打开或关闭报警提示。show 为 1 时打开，show 为 0 时关闭。缺省状态为关闭。

  该命令不影响报警检测与故障上传功能。

  例：

  warnbar(1)

  打开报警提示。

- failurebar(show)

  打开或关闭故障提示。show 为 1 时打开，show 为 0 时关闭。缺省状态为关闭。

  该命令不影响报警检测与故障上传功能。

  例：

  failurebar(1)

  打开故障提示。

控制语句：

键盘检测—根据检测键值执行语句序列

onkey v

语句 1

……

endk

说明：

当键值为 v 的键被按下时，onkey v 和 endonkey 之间的语句序列会被执行。

注意：

onkey 不支持嵌套，即只能是如下结构：

onkey ‘1’

#…

#…

```
    endk
    onkey ‘2’
        #…
        #…
    endk
    onkey ‘3’
        #…
        #…
    endk
```

不能是如下结构：

```
    onkey ‘1’
        #…
        onkey ‘2’
            #…
            #…
        endk
        #…
    endk
```

脚本命令：

u cls

功能：清屏

说明：用目前的背景名清屏

例：下面将屏幕清成红色

```
colorf(255,0,0)
cls
```

u at line, column

功能：将当前绘图位置移到第 line 行，第 column 列

说明：

行和列取值都从 1 开始，每行为 16 像素，每列为 8 像素

例如：

at 5,2

执行后当前绘图坐标就移至第 line 行，第 column 列。

u atp y, x

功能：将当前绘图位置移到（x,y）坐标处

说明：

y,x 的取值都从 1 开始，此命令可以对绘图坐标精确定位

例如：

atp 100,50

执行后当前绘图坐标就移至坐标（50,100）。

- show msg

功能：在当前绘图位置显示 msg

说明：

msg 将会原样显示，此命令不支持显示变量

例如：

show 硕人时代科技

执行后当前绘图位置处就显示“硕人时代科技”字符串。

- invalidview

功能：要求重画组态环境配置的画面

说明：有时候，希望组态环境配置的画面重画一次，可调用此命令。

例：

```
动画效果：
invalidview
x = x+5
if x>100
   x = 1
endif
atp 100,x
show hello!
end
```

说明：这个例子只是一个示意。画面效果并不好。重画背景很耗资源，不是非用不可尽量不用。

- showbmp bmpfilename

功能：在当前绘图位置显示位图文件

说明：

bmpfilename 为显示的位图文件名，此文件之前必须在组态环境中按  按钮，从图库中选出。

例如：

首先在组态环境工具条上按  按钮，从图库中选出为 bmpBack 的位图，然后在非实时脚本程序中加入命令

show bmpBack

当前界面的绘图位置处就会显示位图 bmpBack。

- save v1,v2,v3……

功能：保存变量 v1,v2,v3……等

说明：

被保存后的变量值断电后不会丢失；

注意：不能频繁调用 save，间隔应至少在 5 分钟以上，否则可能会造成数据丢失。

例如：

save openvalue closevalue

将会保存 openvalue 、closevalue 变量值。

- load v1,v2,v3……

功能：重新载入已保存的变量 v1,v2,v3……等

说明：

载入时必须于保存时的顺序保持一致，不能任意载入；

本命令一般用于系统启动时读出上次保存的变量值。

注意：不能频繁调用 load，间隔应至少在 5 分钟以上，否则可能会造成数据丢失。

例如：

load openvalue closevalue

将重新载入 openvalue 及 closevalue 变量值，但不能仅读取 closevalue 值。

非实时脚本注意事项：

- *非实时脚本可命名变量个数不能超过 256 个；*
- *非实时脚本随监控制画面程序运行不断被执行，执行频率不定，平均约每秒 2 至 5 次。*
- *非实时脚本的函数和命令中有几个是阻塞方式执行的，使用时应特别注意。当用户不进行操作超过一定的时间后（缺省为 10 秒），脚本自动退了此次执行。*

调试环境

任何程序在编写完成后都离不开调试过程，为了使二次开发人员能够方便的对SRSCript 脚本程序进行调试，我们把 SRScript 调试环境与虚拟机集成在一起：每当运行虚拟机时，SRScript 的调试环境（包括非实时脚本和实时脚本）也同时启动。开发人员可以直接在调试环境的脚本编辑窗口中编写自己的脚本程序，也可以在其它编辑环境中编写，之后在调试环境中载入。程序完成之后就可以在调试环境中直接运行并能够以多种方法来进行调试了。

界面

调试环境的界面如图 1 所示。图左侧为**脚本列表框**，框内显示了系统中目前所有的实时脚本程序，其中第一个为控制脚本，其它项目对应为虚拟通道脚本程序。右侧的**脚本编辑窗口**中显示了当前打开的脚本程序。而界面右侧的下半部分是各种辅助信息显示窗口，可以显示变量值或是出错信息等。界面上方的**工具栏**中按钮支持了多种调试功能，如单步调试，设置断点等。

功能说明

实时脚本和非实时脚本调试环境的使用方法基本相同，我们以实时脚本为例来介绍调试环境的使用方法。下图中标明了界面各部分的基本功能：

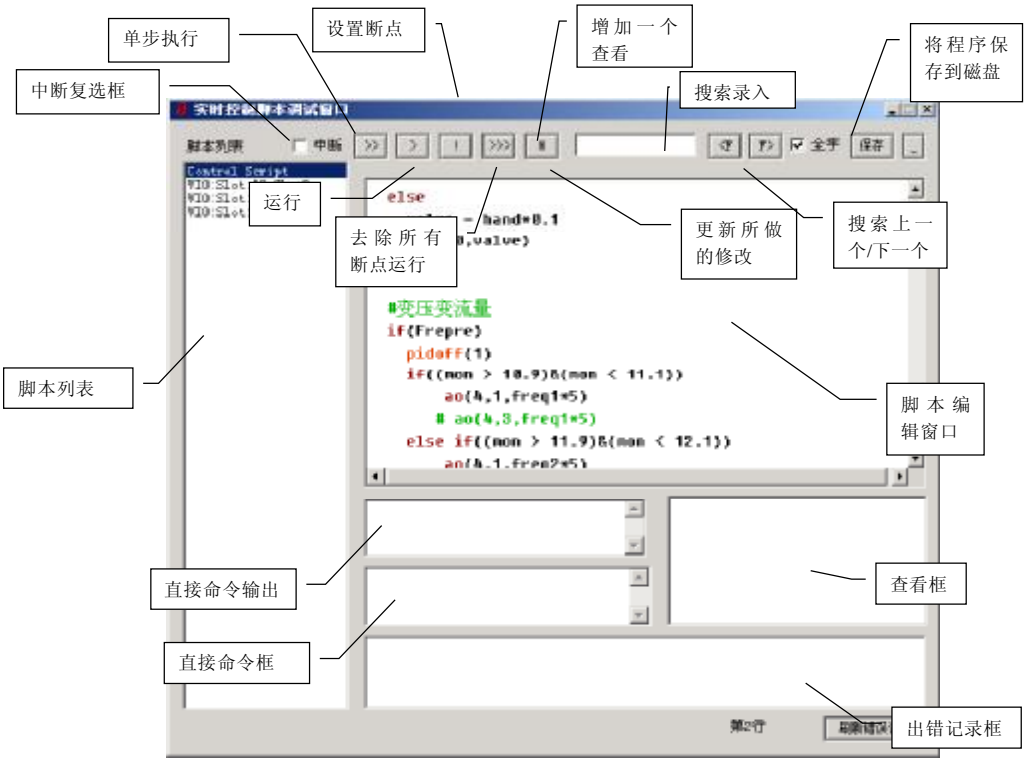


图 2　调试环境功能说明

脚本列表框

功能：以列表形式显示当前系统中的脚本程序，双击列表框中选中的脚本程序项可以在编辑窗口中打开对应的脚本程序进行编辑。

说明：

在实时脚本调试环境下，实时控制脚本显示在第一项，其余为虚拟通道脚本项；

在非实时脚本调试环境下，只显示用户脚本一项。

脚本编辑窗口

功能：编辑当前选中的脚本程序，用法与其它编辑环境相似。

说明：

正常编辑状态下，脚本关键字为蓝色显示，注释行为绿色显示；

调试状态下，设置断点的程序行显示为红色，行首被插入”@”；

单步执行时，下一条将被执行的语句会加上下划线来提示。

查看框

功能：可以查看指定的当前脚本程序中变量值

说明：

在编辑框中选中变量后，可通过增加查看按钮或单击鼠标右键添加到查看框中；

查看框中左侧显示查看的变量名，右侧显示变量对应值；

被查看的变量所显示值只在中断时才会更新；

鼠标双击变量名可删除查看。

直接命令框

功能：用于直接执行单句脚本，为调试的辅助工具。例如，可以在此对变量直接赋值。

说明：

在命令框中单句脚本的执行结果将会在直接命令输出框输出；

在命令框中直接输入变量或保留字，将会在直接命令输出框输出对应值。

直接命令输出框

功能：显示直接命令框语句执行结果或显示命令框中输入的变量、保留字对应值。

出错纪录框

功能：纪录脚本程序中错误情况。

说明：

同一错误可能会不断显示，直至错误排除；

鼠标双击错误记录项，可在脚本编辑窗口定位出错程序位置。

刷新错误纪录按钮

功能：单击此按钮清空当前错误纪录框中错误信息项。

注意：由于高度环境支持动动态更新，即不中止脚本而支持更新程序，因此，在动态更新时，可能会出现错误误报的情况。按此按钮可纠正这种情况，真正的错误在按**刷新错误纪录按钮**后还会出现。

中断复选框

功能：被选中时脚本运行中断，此时脚本编辑窗口中当前执行程序语句会以下划线提示。

单步执行按钮

功能：在中断复选框选中状态下单击此按钮脚本程序将单步执行。

运行按钮

功能：单击此按钮以使脚本程序从中断状态恢复运行。

设置断点按钮

功能：单击此按钮将会把当前光标所在行设为中断点，脚本程序每次运行到此就会中断。

 **去除所有断点运行按钮**

功能：单击此按钮将删除所有已经设置的断点，程序正常运行。

 **增加查看按钮**

功能：用于将当前脚本编辑框所选择的文本增加为一个查看，显示在查看框中。

 **保存按钮**

功能：用于将脚本程序存入磁盘。

注意：按保存按钮，脚本才会被真正保存。并且，还需要在组态环境中同时进行保存，程序才会保存到项目文件中。保存后重启虚拟机，脚本才能生效。

最后需要注意的是，由于脚本的特性，在一些条件块中的脚本如果没有被执行，其正确性就不会得到检查，因此，脚本调试过程中必须保证所有的逻辑部分都试过。

本公司保留对产品型号及技术说明进行更改的权利,恕不另行通知!

地址：北京市海淀区上地中黎科技园 2 号楼 A 座 602
邮编：100085
电话：010－62979299
传真：010－62695262
http://www.shuoren.com
E-mail: develop@shuoren.com