

# 用 Visual Basic 制作控制回路的算法模块

卫 明

(南京师范大学动力工程学院, 南京, 210042)

[摘要] 简述了用 Visual Basic 语言制作控制回路算法模块的方法. 将控制回路功能模块分为算法模块、输入/输出端子块和信号流线, 并给出具体的算法模块的 VB 程序设计.

[关键词] 模块; 控制回路; 语言; 过程

[中图分类号] TPB19; [文献标识码] B; [文章编号] 1008-1925(2001)01-0077-04

## 0 引言

在仿真和 DCS 系统中对控制回路的编程通常在功能块语言系统中采用预先构造的功能模块进行. 系统中的每一个功能模块都仅具有一个特定的功能, 把一系列的功能模块按要求串接起来, 再与仿真对象的模型或 DCS 系统外部的一次设备及执行机构连接, 形成一个完整的控制回路. 由此可见, 功能模块是仿真和 DCS 系统中控制回路组态的基本元素, 制作合适有效的功能模块是控制回路编程的基础. 目前, 在各种软件包中这些模块大多是由 C 语言和汇编语言创建的, 因语言本身的限制, 需用大段程序来描述对象的属性, 因而创建起来较为困难. 现在由于 VB 语言所具有的面向对象编程的特点, 采用 VB 语言制作功能模块, 只要“画”出模块的外观, 从而使工作重点转移到对过程的设计和调试, 节省了大量的时间和精力.

## 1 功能模块的分类

根据控制回路的需要, 可将控制回路分为 3 大类: 算法模块、输入/输出端子块和信号流线. 具体分类如下:

### 1.1 算法模块

算法模块是用图形表示各种标准算法的模块, 是控制回路中的主要组成单元. 按功能可分为如下几组:

- (1) 四则运算: 加、减、乘、除、开平方、绝对值、指数、对数.
- (2) 逻辑运算: 与、或、非、异或、触发器.
- (3) 比较选择: 高选、低选、最大值、最小值、信号选择.
- (4) 折线积算: 定时器、流量积算、计算器、斜波函数、折线函数、时间运算、时间判定、事件驱动.
- (5) 报警限制: 幅值报警、开关报警、偏差报警、速率报警、幅值限制、速率限制.
- (6) 控制算法: 手操器、PID 调节、超前滞后、一阶滞后、二阶滞后、微分、一阶惯性、二阶惯

收稿日期: 2000-11-01

作者简介: 卫明, 1963-, 南京师范大学动力工程学院工程师, 主要从事工业过程自动控制方面的研究.

© 1994-2013 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

性、无扰切换.

1.2 输入/输出端子块

输入/输出端子块用来指明算法模块的处理对象,传递算法模块之间的信息.

输入端子块连接到算法模块的输入项上,在功能块语言中相当于函数的实参,用来向算法模块传递参与运算的参数.

输出端子块被连接到算法模块的输出项上,用来把算法模块的运算结果赋给该端子块所代表的变量.

1.3 信号流线

信号流线是用来在算法模块与算法模块之间,算法模块与变量之间进行数据传输的,信号总是从信号流线的起点到终点,其传输是单向的.

将上述 3 种模块有机集合在一起,即可组成一个控制系统.而这 3 种模块,都可由 VB 语言设计.

2 算法模块的 VB 程序设计

2.1 设计原则

一个算法模块在功能块语言中可以看成为一个函数,将此模块的输入项、输出项和算法名分别看成函数的输入变量、输出变量和函数名称.一个算法模块在用 VB 进行设计时由窗体和过程两部分组成.窗体的格式如图 1.



图 1 算法模块窗体外观

该窗体模块在功能块语言中是一个小算法块,其外观尺寸的大小必须规范,以便于组态时的布置.根据经验将具有一个输入项和一个输出项的算法块的面积设计为 102×56 个像素点(不包括标题栏),将此面积定义为一个图形单元,一个功能块所占的单元数等于该算法块的输入项端子的最大个数,且将这些单元纵向连续排列.



图 2 参数设置对话框

为对模块中的参数进行设置,可通过参数设置对话框实现(如图 2),在算法窗体上右击则可弹出参数设置对话框.参数设置对话框中参数的数量与算法的类型有关.对话框的大小与参数的数量有关.

过程中输入/输出项数据的保存和传递由实时库完成,而各模块中的参数通过参数库解决保存和传递的问题.

2.2 算法模块设计举例

下面以加法模块为例说明模块的设计方法.

加法模块的图形单元形式见图 3,该窗体模块的部分属性值见表 1.

表 1 加法模块属性值表

属性	值
Name	Add1
Capation	加法
ControlBox	false
ScaleMode	3- pixel
ScaleHeigh	112
ScaleWidth	102
Visible	true



图 3 加法模块窗体外观

该窗体中共包含四个文本框, 分别用来表示两个输入项  $I_1$ 、 $I_2$ , 一个输出项  $AV$  和计算名. 该模块的功能为:

$$AV(K) = I1(K) \times P1 + I2(K) \times P2$$

若只有一个输入端  $X$  则:  $AV(K) = I_X(K) \times P_X$

式中:  $P1$ 、 $P2$  为加数系数, 其值由参数设置对话框设置.

设置加权系数的对话框(设计态)如图 4 所示, 由两个标签、3 个文本框、一个按钮和一个数据控件组成. 其中  $TEXT3$  和数据控件  $DATA1$  的  $Visible$  属性值设置为  $False$ ,  $TEXT1$  和  $TEXT2$  的  $Enabled$  属性值设置为  $False$ , 只有在点击“设置”按钮后才为  $True$ . 在两个文本框中输入参数值后点击“确定”按钮, 则将参数值保存到参数库中.



图 4 加法模块参数设置对话框(设计态)

在下列过程中,  $js1.mdb$  为参数设置中间数据库,  $in\_out.mdb$  为输入/输出量实时中间数据库, 其中  $js1.mdb$  中包含有“NAME”、“ $P1$ ”和“ $P2$ ”三个字段, 而实时库  $in\_out.mdb$  中包含“NAME”、“ $IN1$ ”、“ $IN2$ ”和“OUT”四个字段, 两者都以“NAME”字段建立索引. 实时库中“ $IN1$ ”、“ $IN2$ ”两个字段的值由信号流线或输入端子块更新, “OUT”字段的值由信号流线或输出端子块取出. 组态时  $js1.mdb$  中的“NAME”字段的值由全局变量  $Arithmetic$  从算法窗体中传递到参数设置窗体, 并与  $Text1$  和  $Text2$  的  $Text$  属性值一起保存. 在过程执行时, 依“NAME”字段从参数库  $js1.mdb$  中取得两个加权系数, 再依“NAME”字段从实时库中取得两个输入项, 运算结果写入到库中的“OUT”字段中. 该模块的过程如下:

Public Arithmetic As String

主过程:

Private Sub main()

Load Form1

Form1.Show

End Sub

Form1 中的过程:

Private Sub Form\_Activate()

Dim db1, db2 As Database

Dim rs1, rs2 As Recordset

‘打开参数库中间文件

Set db1 = OpenDatabase("D:\js1.mdb")

‘设置表格型记录集变量

Set rs1 = db1.OpenRecordset("js1")

‘定义当前索引

rs1.Index = "name"

‘查找记录

rs1.Seek "=", Text4.Text

If Not rs1.NoMatch Then

p1 = rs1.Fields("P1")

p2 = rs1.Fields("P2")

End If

‘打开实时库中间文件

Set db2 = OpenDatabase("D:\in\_out.mdb")

‘设置表格型记录集变量

Set rs2 = db2.OpenRecordset("in\_out")

‘定义当前索引

rs2.Index = "name"

‘查找记录

rs2.Seek "=", Text4.Text

If Not rs2.NoMatch Then

i1 = rs2.Fields("in1")

i2 = rs2.Fields("in2")

‘使编辑有效

rs2.Edit

‘改变输出值

rs2("out") = i1 \* p1 + i2 \* p2

‘保存变化

rs2.Update

End If

End Sub

‘右击窗体事件设置参数

Private Sub Form\_MouseDown(Button As Integer,

```
_Shift As Integer, X As Single, Y As Single)
If Button = 2 Then
    Arithmetic = Text4.Text
    Load Form2
    Form2.Show
End If
End Sub
```

Form 2 中的过程:

```
Private Sub Command1_Click()
If Command1.Caption = "设置" Then
    If Data1.Recordset.RecordCount > 0 _
Then Data1.Recordset.MoveLast
    Text1.Enabled= True
    Text2.Enabled= True
    Data1.Recordset.AddNew
    Text1.SetFocus
    Command1.Caption = "确定"
Else
    Text3.Text = Arithmetic
```

```
If Text3.Text <> "" And IsNumeric(_
Text1.Text) And IsNumeric(_
Text2.Text) Then
    Data1.Recordset.Update
Else
    MsgBox " 输入数据非法,重新设置 ",48+0
End If
Unload Me
Form1.Show
End If
End Sub

Private Sub Form_Activate()
If Data1.Recordset.RecordCount = 0 Then
    Data1.Recordset.AddNew
Else
    Data1.Recordset.MoveLast
End If
End Sub
```

3 结束语

上述模块运行过程中用到的数据库由功能块语言事先建立,模块运行时可直接访问数据库中记录的字段. 上述模块运行正确. 过程语言中的其它算法模块、输入/输出端子块和信号流线也可依此法设计,但需注意不同类型模块中参数项的数目不一定相同,虽可采用同一个参数库或实时库,但应在功能块语言中设计采用不同的数据表.

[ 参考文献]

[ 1] 东箭工作室. Visual Basic 5.0 中文版程序设计[ M]. 北京:清华大学出版社, 1997  
[ 2] 王常力, 罗安. 集散型控制系统选型与应用[ M]. 北京: 清华大学出版社, 1996  
[ 3] 陈华生, 牛又奇, 孙建国. Visual Basic 程序设计教程[ M]. 苏州: 苏州大学出版社, 1999

Arithmetic Module of Control Loop in Visual Basic

Wei Ming

( College of Power ,Nanjing Normal University, Nanjing, 210042,PRC)

**Abstract:** This paper discusses the method to design the arithmetic module of the control loop in Visual Basic. The control loop is divided into arithmetic module, input/ output module and signal line, with the particular VB program for the arithmetic modules given.

**Key words:** module; control loop; language; process

[ 责任编辑: 严海琳]