

PLC梯形图中内嵌C语言编程的实现

黄小强, 严 义, 邬惠峰, 吴盼盼
(杭州电子科技大学 计算机学院, 浙江 杭州 310018)

摘要: 针对现有梯形图语言适合实现逻辑控制而不适合实现复杂算法这一问题,从基于构件技术的梯形图实现原理出发,对构件的属性定义以及编译方法进行了分析,提出了一种PLC梯形图内嵌C语言实现混合编程的方法,并在CASS(computer assist special system)梯形图算法生成平台上进行了验证。实验结果表明,提出的方法适于复杂算法的实现,并且能够减少程序的代码量。

关键词: 梯形图;C语言;混合编程;CASS

中图分类号: TH39;TP311

文献标志码:A

文章编号:1001-4551(2012)04-0421-04

Implementation of embedding C language programming in PLC ladder diagram

HUANG Xiao-qiang, YAN Yi, WU Hui-feng, WU Pan-pan
(College of Computer, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Aiming that the ladder diagram (LD) language is incompetent in realizing complex algorithms, a mix programming method of ladder diagram and C language was proposed. The component technology for ladder diagram implementation was analyzed and the way to embed C language inside ladder diagram was introduced. Experiments were carried out based on the computer assist special system (CASS) platform. Results show that the proposed method is fit for complex algorithm implementation, and the code size could also be reduced.

Key words: ladder diagram(LD); C language; mixed programming; computer assist special system(CASS)

0 引 言

可编程逻辑控制器(PLC)^[1]是一种以微处理器技术、电子信息技术、网络通信技术和自动控制技术为依托的新型通用自动控制装置,它在工业控制领域备受青睐,特别是在各种工业过程控制、生产线自动控制及各类机电一体化设备控制中得到了极为广泛的应用,与数控技术、CAD/CAM技术和机器人技术一起被称为现代工业控制四大支柱。

根据IEC61131-3标准^[2],梯形图^[3](LD)作为PLC的图形化编程语言,适用于逻辑控制程序设计,具有使用简单、直观明了的优点,是目前最流行的PLC控制编程语言。然而,随着PLC应用领域的扩展和被控设备功能复杂程度的增加,如针对模拟量的闭环控制、数字量的智能控制等,基于梯形图语言的程序设计已经显得力不从心。尽管国内外已有学者在这

方面做了研究,例如文献[4]给出了在西门子S7-200 PLC的梯形图上实现模糊神经网络算法,但是该算法实现困难,并且会使代码量大大增加,延长开发周期。因此,研究一种能将复杂的算法嵌入到梯形图中实现混合编程的方法,具有重要的现实意义。

目前,复杂算法嵌入研究集中在可编程自动控制器PAC上,主要采用流程图和脚本语言相结合^[5-6]的方法,在梯形图上采用混合编程的方法实现复杂算法嵌入的研究较少。考虑到C语言作为计算机高级编程语言,在编写算法时具有高效简洁、实现方便的特点,若能将在C语言编写的算法嵌入梯形图中,将极大地提高梯形图算法设计的方便性以及系统的开发效率。

本研究基于CASS梯形图算法生成平台^[7],提出一种梯形图内嵌C语言的方法,并通过实验对该方法的有效性进行验证。

1 CASS 梯形图算法生成平台

CASS 梯形图算法生成平台是一种可扩展开放式的软件系统,是嵌入式系统与软 PLC 的结合,为各种类型的嵌入式芯片提供友好、高效的通用编程环境,平台界面如图 1 所示。

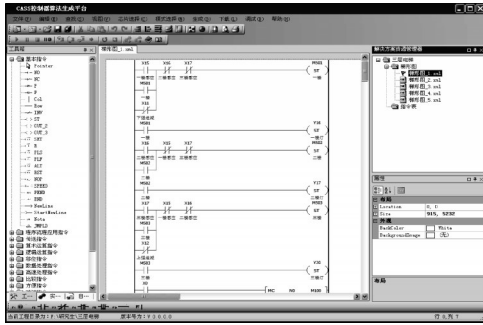


图 1 CASS 梯形图算法生成平台

它具有如下的显著优点:

(1) 开放式的硬件体系结构,相比传统 PLC,具有更加丰富的指令集;

(2) 基于 IEC61131-3 语言标准,具有开发便捷、易于复用、缩短产品开发周期的优势;

(3) 开放的架构和标准使得产品具有高性价比,而且可以同时加入到 PLC 网络和标准计算机网络中。

其主要功能可概括如下:

(1) 采用拖拉的方式实现快捷、高效的梯形图编程;

(2) 为各种类型芯片提供定制、编译、下载等操作;

(3) 提供实时监控与调试功能,以实现单步控制程序运行等。

目前,CASS 平台已广泛应用于工业自动化控制领域,如电梯系统、伺服系统、医疗器械、纺织控制等。

1.1 软件构件技术

CASS 梯形图算法生成平台是基于软件构件技术进行设计的。软件构件技术^[8]是支持软件复用的核心技术,用于避免软件开发中的重复劳动,其主要内容包括构件获取、构件模型、构件描述语言、构件分类与检索、构件组装等。全过程描述如下:

(1) 利用领域工程等技术,进行有目的的构件生产和从已有系统中挖掘提取构件^[9];

(2) 研究构件的本质特征及构件间的关系,得到相应的构件模型;

(3) 以构件模型为基础,研究构件分类、组织模式及检索策略,建立构件库系统;

(4) 通过构件的接口使软件构件相互连接,从而构造应用系统;

(5) 对已经形成的构件模型和构件库系统进行标准化。

因此,本研究将构件技术应用到梯形图中,采用统一的构件模型整合梯形图元素,利用构件组装技术生成梯形图程序,可以极大地提高软件开发的效率和产品的质量,方便软件维护升级。

1.2 梯形图的构件模型

CASS 梯形图算法生成平台采用形式化方式描述构件模型的语义、结构和非功能特性,它是分析和评价构件行为和性能的依据。对梯形图构件模型的描述为:

$LDC < Name, ID, PI, RI, Property, Specification >$

其中:

(1) Name 为构件的名称,用于描述构件实现的功能;

(2) ID 为构件的标识符,在梯形图程序中作为唯一的标识;

(3) PI 为构件能够提供服务的接口集合,包括梯形图构件的输出数据接口、右侧串联接口、向下并联接口和部分辅助数据接口;

(4) RI 为构件所需求服务的接口集合,包括梯形图构件的输出数据接口、左侧串联接口、向上并联接口和部分辅助数据接口;

(5) Property 为构件内部所包含属性的集合,作为构件的基本构成元素,包括位置、大小、指令参数、注释、状态值等信息;

(6) Specification 为功能说明,以具体的文字、公式或框架模板说明构件实现的方法及功能。

梯形图基本构件分为触点构件、功能或功能块构件、线圈构件、横线竖线构件、换行符、注释等,功能块构件包括形式化描述的所有信息,如图 2 所示。触点构件不含输入数据接口和输出数据接口,线圈构件不含输入数据接口、输出数据接口和并联接口,横线构件和竖线构件不含数据接口。

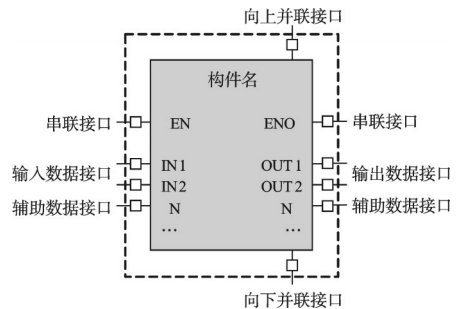


图 2 功能块构件的图形化表示

2 梯形图内嵌 C 语言的实现方法

2.1 基于构件模型的 C 语言嵌入

在上述构件模型基础上,本研究通过增加 C 语言编程构件,实现梯形图内嵌 C 语言。C 语言编程构件模型有两种,如图 3 所示。其中,不含输入/输出数据接

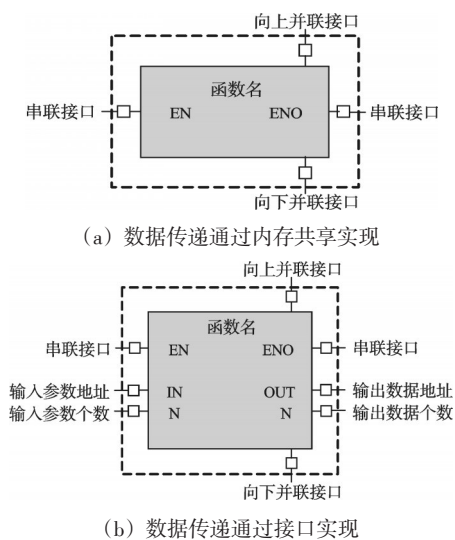


图3 C语言嵌入的控件模型

口,参数传递及数值返回通过内存共享实现图3(a)所示;输入数据接口作为C语言模块的参数传递变量,输出数据接口作为数值返回通道图3(b)所示。

两种模型各有优缺点:模型(a)通过内存共享方式进行数据处理,具有很高的数据传输效率,并且能够进行大批量数据处理,但访问数据区(M区、D区等)的方式较为复杂,对编程人员要求较高。模型(b)中数据的传递通过构件模型的输入输出接口实现,具有直观、易于操作的优点,但在批量数据处理和参数传递效率等方面相对欠缺。本研究采用模型(a)来说明梯形图内嵌C语言的方法。

2.2 内嵌C语言的梯形图编译方法

梯形图的编译过程可分为4个步骤:

- (1) 对梯形图构件组合、逻辑关系、语法、语义的正确性进行分析;
- (2) 采用XML文件对梯形图的逻辑关系进行描述,并经AOV图转换成IL指令^[10];
- (3) 对转换过程中的错误进行检查;
- (4) 根据不同类型的芯片,调用相应的编译器将IL指令编译成可执行文件。

在梯形图中嵌入C语言构件之后,需要对上述的编译过程进行实现。C语言构件在梯形图中的作用有两点:作为梯形图构件,保证梯形图的完整性;C语言算法的调用点。

因此,编译过程可分两个步骤实现:

- (1) C语言构件模型编译。C语言构件模型可按原来的梯形图编译方法进行编译:首先根据梯形图语法库,通过拓扑抽象转化为有向图,分析梯形图拓扑结构错误;根据串联/并联规约,生成二叉分解树;最后根据IL语法库,执行语义翻译算法,形成IL指令,经宏处理后加载到应用程序工程中,作为调用C语言编程模块的入口点。

- (2) C语言编程模块编译。C语言编程模块编译可分为预编译和编译器编译。预编译阶段可描述为:读取梯形图内部地址映射表,根据内部地址映射表,将C语言算法中涉及的相关地址转化为嵌入式硬件系统中对应的逻辑存储地址,并将经过转化的C语言算法内容保存到应用程序工程中,供编译器编译调用。编译器编译是将经过预编译后的代码通过GCC编译器,以命令行的形式完成。

3 实验结果与分析

本研究以工业过程控制中被广泛应用的PID算法为例,对内嵌C语言的梯形图实现方法作简要介绍,并与传统的梯形图实现方法做比较。对于单输出控制系统,PID算法可描述如下:

- (1) 读取当前的测量值PV,并根据设定值SP,计算偏差值 $E = SP - PV$;

- (2) 根据设定的比例系数 K_p 、积分算子 K_i 、微分算子 K_d ,将偏差值通过式(1)转换为有效值CV输出,用于调节当前值,使之接近SP值:

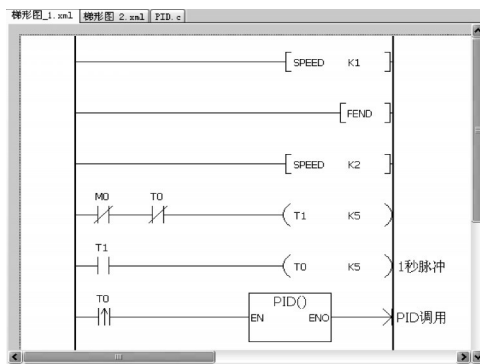
$$Cv = K_p \cdot e(t) + K_i \cdot \sum e(t) + K_d \cdot \{e(t) - e(t-1)\} \quad (1)$$

式中: $e(t)$ — t 时刻的偏差值。

- (3) 重复步骤(1)~(2)。

3.1 PID算法的混合编程实现

内嵌C语言的PID算法如图4所示,其中,PID的



(a) PID的梯形图调用

```

1 #ifndef PIDFlag
2 #define INADDRESS 100 //输入参数偏移地址
3 #define OUTADDRESS 150 //输出参数偏移地址
4 void PID()
5 {
6     float *InAddr, *OutAddr;
7     //通过内存共享传递参数
8     InAddr = (float*) &CassMem[DStart+INADDRESS];
9     //InAddr[0]:采样值; InAddr[1]:设定值;
10    //InAddr[2]:Kp比例系数; InAddr[3]:Ki积分系数;
11    //InAddr[4]:Kd微分系数; InAddr[5]:上一次偏差;
12    //InAddr[6]:本次偏差; InAddr[7]:累积偏差;
13    InAddr[5] = InAddr[6]; //上一次偏差
14    InAddr[6] = InAddr[1] - InAddr[0]; //本次偏差
15    InAddr[7] += InAddr[6]; //累积偏差
16    OutAddr = (float*) &CassMem[DStart+OUTADDRESS];
17    //Output = Output + Kp*Error + Ki*ErrorSum
18    // + Kd*(Error - LastError);
19    *OutAddr = *OutAddr + InAddr[2]*InAddr[6] //输出
20    + InAddr[3]*InAddr[7] +
21    InAddr[4]*(InAddr[6]-InAddr[5]);
22 }
23 #endif
    
```

(b) PID的C语言算法实现

图4 内嵌C语言的PID算法

梯形图调用如图4(a)所示,梯形图逻辑过程控制中,当常闭触点M0为逻辑0时,定时器T0和T1互锁产生1s脉冲,T0每秒获得一个上升沿,执行一次PID算法。针对如图4(b)所示的C语言编程模块,参数传递是采用共享内存实现的,输入、输出参数的地址是通过宏定义来指定相应的内存偏移地址,PID参数存于VD区,以便于修正参数,整个PID算法核心代码共有6条语句。

3.2 PID算法的梯形图实现

PID算法的梯形图代码如图5所示,右边的注释语句为对应的C语言代码。当T0获得一个上升沿时,执行PID算法。梯形图的运算指令要求输出与输入的每一个参数共用同一个地址,因此需要4个字节的额外内存空间(VD50~VD53),核心语句共有16条,整个代码主要使用数据传送指令(MOVD)、浮点数的加法指令(EADD)、减法指令(ESUB)、乘法指令(EMUL)。

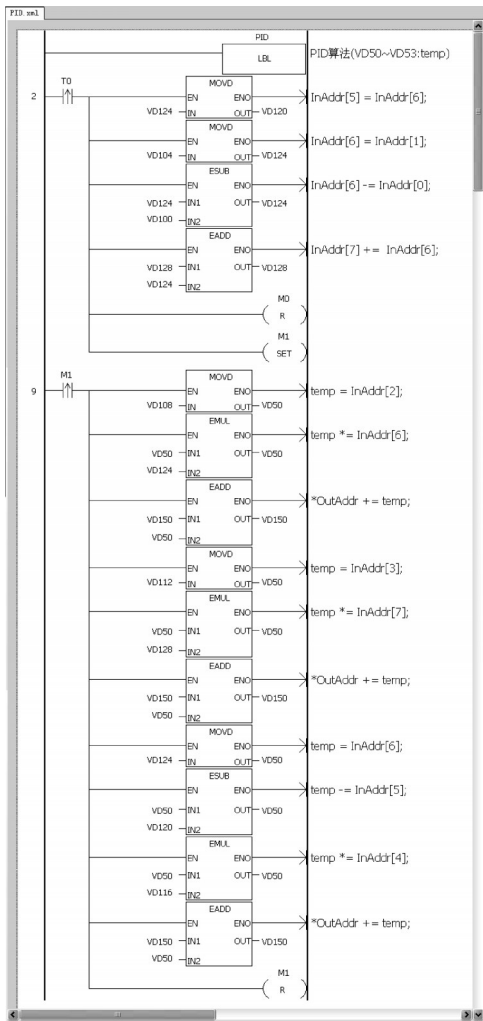


图5 梯形图实现的PID算法

3.3 混合编程与梯形图方法对比分析

从代码量、数据内存空间、可读性角度,本研究对分别采用梯形图和内嵌C语言实现的PID算法作简要分析对比,对比情况如表1所示,可以看出,基于内嵌C

语言实现的PID算法,其代码量比基于梯形图减少了11.8%;基于内嵌C语言实现的PID算法比基于梯形图实现的PID算法占用更少的数据内存空间。此外,C语言代码简洁,具有更好的可读性。

表1 PID算法的梯形图实现方法和内嵌C语言实现方法对比

实现方式	代码量 /KB	数据内存空间 /Bytes	可读性
梯形图实现	4.50	40	冗长,可读性一般
内嵌C语言实现	3.97	36	精简,可读性较好

4 结束语

本研究以基于CASS的梯形图算法生成平台为基础,提出了一种PLC梯形图内嵌C语言的方法,并对分别采用C语言和梯形图实现的PID算法进行了对比分析。实验结果表明:梯形图内嵌C语言的方法具有实现简单、可读性好、生成代码量小等优点。

在接下来进一步的研究中,还可以整合各种算法语言嵌入梯形图中并建立不同语言的算法库,以便工业控制使用。

参考文献(References):

- [1] 程立英,野莹莹,彭泽波. PLC教程 原理·应用·设计[M]. 北京:人民邮电出版社,2009.
- [2] JOHN K H, TIEGELKAMP M. IEC 61131-3 International Standard, Programmable Controllers, Part 3: Programming Languages[S]. IEC, 2003.
- [3] 彭 瑜. 工控编程语言IEC61131-3的现状与发展[J]. 国内外机电一体化技术, 2004, 7(1): 42-49.
- [4] GUO Hui, FU Cheng-hua, GAN Shu-chuan. A Fuzzy Neural Network Algorithm applied to S7-200 PLC [C]//Proceedings-2011 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Shanghai: [s.n.], 2011: 480-484.
- [5] 王善永,陈思宁,施 冲,等. MB系列智能可编程逻辑控制器[J]. 电力系统自动化, 2005, 29(10): 82-84, 91.
- [6] 郭华芳,李智文. 智能控制算法在PAC上应用[J]. 微计算机信息, 2007, 23(8): 52-54.
- [7] 罗 川. 基于CASS数控绕线机非线性算法的设计与实现[D]. 杭州:杭州电子科技大学计算机学院, 2010.
- [8] 杨芙清,梅宏,李克勤. 软件复用和软件构件技术[J]. 电子学报, 1999, 27(2): 68-75.
- [9] 周 欣,陈向葵,孙家骥,等. 面向对象系统中基于度量的可复用构件获取机制[J]. 电子学报, 2003, 31(5): 649-653.
- [10] YAN Yi, ZHANG Hang-ping. Compiling ladder diagram into instruction list to comply with IEC 61131-3[J]. Computers in Industry, 2010, 61(5): 448-462.

[编辑:李 辉]