第35卷 第23期 计 算 机 工 程 2009年12月
Vol.35 No.23 Computer Engineering December 2009
·工程应用技术与实现· 文章编号：1000—3428(2009)23—0246—03 文献标识码：A 中图分类号：N945

基于IEC61131-3标准的PLC编程环境

秦 华[1,2]，夏清国[1]，付 钰[3]
(1. 西北工业大学计算机学院，西安 710129；2. 桂林空军学院3系，桂林 541003；3. EMC中国(上海)研发中心，上海 200433)

摘 要：针对目前PLC编程环境存在的问题，设计一种基于IEC61131-3标准的新型PLC编程开发环境，给出开发环境的模块构成，介绍各个模块实现时用到的数据结构和算法，分析其中较难实现的梯形图语言向指令表语言翻译的算法思想和实现步骤。实验结果表明，该系统用户界面友好、操作简单、功能全面，可支持多种PLC语言开发，方便用户对PLC的开发和使用，具有广阔的应用前景。
关键词：可编程逻辑控制器；梯形图；指令表；IEC61131-3标准

PLC Programming Environment Based on IEC61131-3 Standard

QIN Hua[1,2], XIA Qing-guo[1], FU Yu[3]
(1. School of Computer, Northwestern Polytechnical University, Xi'an 710129; 2. No. 3 Dept., Guilin Air Force Academy, Guilin 541003; 3. EMC China(Shanghai) R & D Center, Shanghai 200433)

【Abstract】Aiming at the limitation and lack of the current editing software for PLC, this paper designs a new programming environment for PLC based on the IEC61131-3 standard. After given the development environment modules, it introduces data structure and algorithm of every module, focuses on the main ideas and realization steps of the translation algorithm between the Ladder Diagram(LD) and the Instruction List(IL). Experimental results show the programming system has a friendly user interface and complete functions, it is easy to operate and can support more than one kind of PLC language development, and what's more, this system provides great convenience for user developing and using PLC, so it has a broad application prospect.
【Key words】Programmable Logic Controller(PLC); Ladder Diagram(LD); Instruction List(IL); IEC61131-3 standard

1 概述

可编程逻辑控制器(Programmable Logic Control, PLC)是以微型计算机为核心的工控装置。目前，PLC功能要求不断增长，传统PLC编程方式的开发时间和错误率都相对增加，对大型程序的纠错和维护也非常困难。与此同时，多数公司的编程软件只针对自己的产品，产生了不同PLC程序可移植性、可复用性差，缺乏封装能力等一系列问题。

1993年国际电工委员会颁布IEC61131标准，其中，第3部分对PLC编程进行规范，这一标准为工控软件的发展起到了举足轻重的推动作用[1]。但是，多种软件开发平台并存且没有针对组态软件设计的规范，使组态软件在设计上大相径庭，即使同一种组态功能实现时也各有侧重点。现阶段PLC编程软件普遍存在提供的编程语言较少、人机交互界面设计简单粗糙、功能少、缺乏整体的视觉效果容易使用户对组态工作产生混乱等问题。

本文通过对IEC61131-3标准的研究，设计一个基于PC的PLC统一编程环境。为编程人员提供一个将自己的要求转化成程序的友好平台，编程环境要求具有良好的可视化人机界面，程序开发简单、维护方便、系统运行稳定，具有智能的图形及文本编辑系统，提供使用符合标准的5种语言，并能实现图形语言至文本语言转换。

2 IEC61131-3的软件模型

IEC61131-3软件模型在最上层把软件概括为“配置”，专指一个特定类型的控制系统，包括硬件装置、处理资源、I/O通道的存储地址和系统能力。在一个“配置”中可定义一个或多个“资源”，反映控制器的物理结构，在程序和控制器的物理I/O通道之间提供一个接口，只有在装入“资源”后才能执行程序。在一个“资源”内可以定义一个或多个“任务”，配置后控制一组程序组织单元(Program Organization Units, POU)周期执行或事件驱动执行。标准定义了5种编程语言，其中，图形语言有梯形图(Ladder Diagram, LD)、顺序功能图(Sequential Function Chart, SFC)和功能块图(Function Block Diagram, FBD)，文本语言有指令表(Instruction List, IL)和结构化文本(Structured Text, ST)。

3 编程环境体系结构设计

为满足IEC61131-3各项要求，系统可划分为编辑模块、检查模块、翻译模块、监视模块、通信模块5个部分。

(1)编辑模块

包括图形及文本编辑环境，图形语言支持使用鼠标拖拽编辑，文本语言需要对用户的输入进行提示。

(2)检查模块

对文本语言程序进行基本的词法和语法错误检查，对图形语言程序进行图元组合错误检查，同时提示错误。

(3)翻译模块

图形语言不能被PLC识别，要转换成文本语言，以便调用编译器进行编译。

作者简介：秦 华(1980－)，女，硕士研究生，主研方向：网络与信息安全；夏清国，副教授；付 钰，硕士
收稿日期：2009-07-10 **E-mail：**babycat218@163.com

(4)监视模块

当程序在 PLC 中运行时反映其执行和参数变化情况，以便及时调整和维护。

(5)通信模块

负责 PC 与 PLC 硬件之间的数据通信，包括用户程序下载、操作命令和硬件配置参数、内存等的记取。

除了这些主要功能模块外的其他基本功能可以内嵌到以上模块中。系统工作流程如图 1 所示。

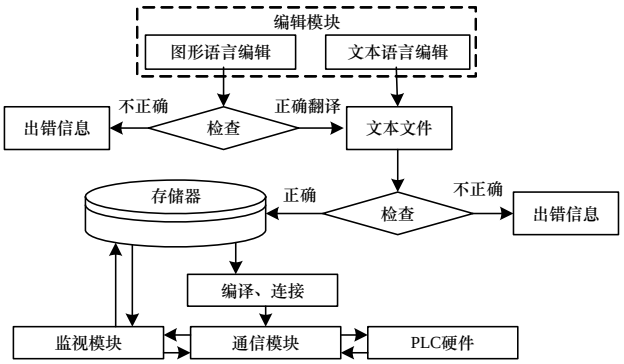

图 1　软件系统工作流程

4　软件模块具体实现

4.1　编辑模块

软件设计的第 1 阶段选用 LD 和 IL 作为编辑语言。

4.1.1　LD 编辑模块

(1)LD 对象类定义

LD 采用逻辑元件和逻辑关系图表示程序。标准规定可采用的元件图符有电源轨线、连接元素、触点、线圈、功能和功能块等，可设计如图 2 所示的类图对 LD 对象进行封装。

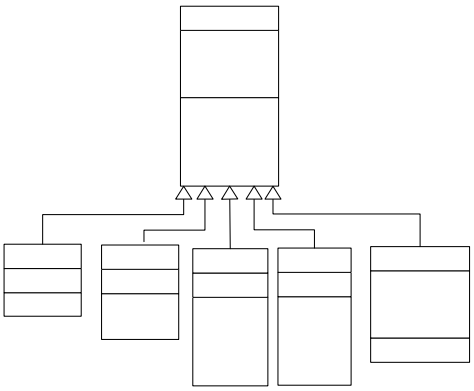

图 2　LD 类图

(2)LD 程序的存储

LD 程序常用的存储结构有双向链表、邻接链表和十字链表等。为了对 LD 程序进行正确存储、显示并方便后续处理，除了需要存储图符的全部信息，还要正确表示图符间的拓扑关系。一个 LD 程序可看成是一个有向图，其中，图符可看作图的顶点，连接符可看成图的弧。在后续处理中，还经常需要使用各个顶点的入度和出度。双向链表无法清晰的表示出图的结构，邻接表在求出度、入度时较为复杂，因此，采用十字链表存储 LD 程序。

4.1.2　IL 编辑模块

IL 类似于汇编语言，由一系列指令组成，指令包括操作符和操作数。为了支持 IL 编辑，可使用 CEditView 派生出 CTextView 类。CEditView 已经具有文字编辑功能，支持多窗口操作并有文件预览功能，由此派生的 ETextView 能支持 IL 程序在编辑视窗中进行编辑。

4.2　检查模块

4.2.1　LD 检查模块

本模块主要是规范使用者的编程，给出出错信息引导用户做出修改[2]。从软件的易用性考虑，不只在程序设计结束后进行检查，而是将检查模块分为实时和非实时 2 部分设计。

(1)实时检查

对编程过程中出现的明显错误弹出对话框提示，引导编程人员规范的编写程序，提高编程效率。

(2)非实时检查

程序编辑完成后全面检查，保证通过检查的 LD 程序是正确的，主要是检查元件参数错误、短路和断路错误、混合连接错误，下面给出检查的思路。

1)元件参数错误：在遍历元件链表时获取每个元件的参数，判断是否合法。

2)短路错误：如果网络(回路)中某个部分并联了几条支路，其中一条支路全由直线或者直线和空元件组成的，那么该部分其他支路被短路。

3)开路错误：按一定原则把行划分为若干小行，然后检查小行的开始、中间和结束元件是否合法。小行划分标准是：如果发现空元件则其前面所有元件为一小行；如果某元件的列号比上一个元件的列号大 2 及以上，则该元素前所有元素为一小行；元件链表的结束也是划分小行的标志。

4)检查混合连接错误：若网络中存在任何一部分(行数和列数都至少为 3，并且行与行之间相互联系)，这部分的行不能在任何一列处交汇到一个点，否则就发生了混合连接错误。

4.2.2　IL 检查模块

指令表错误处理主要是对指令表文本进行词法和语法分析，对操作码和操作数有效性进行校验。

4.3　翻译模块

目前国内多数 PLC 开发系统中采用的 LD 至 IL 翻译算法是将 LD 映射成 AOV(Activity On Vertex)图，再采用拓扑排序对图进行遍历，从而实现转换[3]。该方法通过相应的图符含义来获得 LD 中的逻辑关系，并不通用，且不适用于串、并联关系较复杂的程序。所以，本文采用将 LD 映射成 AOV 图，再由 AOV 图建立二叉树，并对其进行后序遍历的方法[4]，步骤如下：

(1)LD 程序映射成 AOV 图

AOV 图用顶点表示活动，用弧<*i*,*j*>表示活动 *i* 必须在活动 *j* 开始之前完成。可以将 LD 中的图符抽象成顶点，图符之间的连接关系抽象成弧。

映射时首先对 LD 程序进行一次从左到右，从上到下的扫描，统计其中图符和弧的数目，得到各图符的前驱节点、后继节点、入度和出度等信息，并把 LD 并连线抽象成虚节点。然后进行第 2 次扫描，建立 AOV 图的结构。

(2)AOV 图转换成相应的二叉树

基本思路是：入度为零的顶点设置为根节点，每个图符对应二叉树的一个叶子节点。如果图符的出度为 1，则建立一个“与”节点；若图符的出度大于等于 2，则建立一个“或”节点。

AOV 图转换成二叉树算法流程如图 3 所示，其中，*P*1 表示图符顶点指针；*F*1 表示 *P*1 当前指向图符的访问标志；*N*1 表示 *P*1 指向图符的入度；*N*2 表示 *P*1 指向图符的出度。

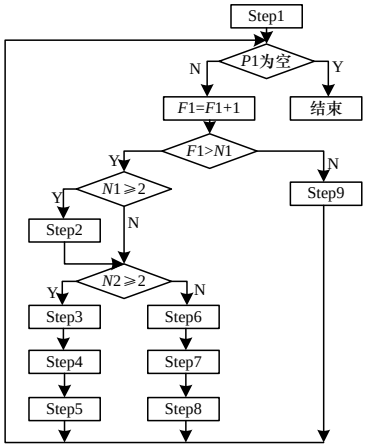

图 3　由 AOV 图建立二叉树算法流程

对图 3 的说明如下：

Step1 创建与堆栈 *S*1 或堆栈 *S*2，初始化 *P*1、二叉树节点指针 *P*2，将 *F*1 清 0。*P*1 指向入度为 0 的节点。

Step2 从 *S*1 中弹出“与”节点指针，并赋给 *P*2。

Step3 创建“与”节点，并赋给 *P*3。

Step4 如果 *P*2 左子树为空，则 *P*3 作为 *P*2 的左子树，否则 *P*3 作为 *P*2 的右子树。

Step5 *P*1 作为 *P*3 的左子树，*P*2 指向 *P*3 对应的节点，*P*1 指向 *P*1 的第 1 个直接后继。

Step6 创建 2 个“与”节点 *P*3, *P*4，创建 *N*2-1 个“或”节点 *P*3, *P*4,…, *PN*。

Step7 如果 *P*2 左子树为空，则 *P*3 作为 *P*2 的左子树，否则 *P*3 作为 *P*2 的右子树。

Step8 P4 作为 *P*3 左子树，*P*1 作为 *P*4 左子树，*P*6 作为 *P*5 左子树，*P*7 作为 *P*6 左子树，…，*PN* 作为 *PN*-1 左子树。*P*5~*PN* 压入 *S*2，*P*1 的直接后继压入 *S*2。*P*1 指向其第 1 个直接后继，*P*2 指向 *PN* 对应的节点。

Step9 从 *S*2 中弹出 2 个对象指针，分别赋给 *P*1, *P*2。

(3)对生成的二叉树进行后序遍历，访问每个节点时进行相应的处理，生成对应的指令表。

4.4　在线监视模块

监视画面可以根据 PLC 程序的控制流程和中间变量，用直观的编程方式完成，并提供出错报警，以达到随时了解程序执行和参数变化情况的目的。模块由图形界面和实时数据库组成。

图形界面可以依照操作系统的图形标准，采用面向对象技术通过图形来反应参数变化情况，生成图形的数据驱动来源于数据采集过程中自动生成的数据表。在连接过程中，从数据表选择驱动数据源，最终生成图形目标应用系统供图形运行环境运行时使用。实时数据库中应定义数据库的结构、数据连接、数据类型及相关的各种参数，其数据的主要来源为数据采集时生成的数据表。

4.5　通信模块

PLC 和 PC 的通信通常通过 PC 的串口以“命令-响应”的交互方式进行。设计时建一个能够实现串口通信的类，类中设计 2 个函数：ReadData, SendData，按照 PLC 通信格式建立起上位机和 PLC 的通信以后，即可完成所需功能。

4.6　工程管理器

工程管理器对配置层进行管理。配置层是 IEC61131-3 软件模型中的最上层，对应于整个控制系统。工程管理器负责配置中各个部分的组织和联络，并要求能对系统中所有资源进行统一的管理。

工程管理器按照树形结构进行组织，并应具备如下功能：

(1)登记新创建的文件；

(2)从其他项目导入文件；

(3)显示所有已经存在的 POU；

(4)更名或删除 POU；

(5)显示整个项目的信息结构。

5　软件测试

VS2005 拥有完善的集成开发环境，包括可视化设计器、代码编辑器以及程序设计语言，为提高开发速度和程序执行效率，选用 VS2005 作为开发工具。系统主界面及翻译模块运行结果如图 4 所示。

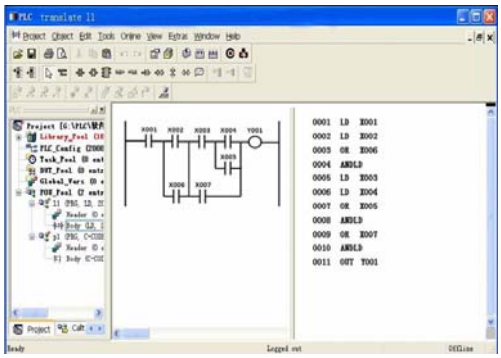

图 4　系统主界面及翻译模块运行展示

通过实验表明：

(1)系统软件开发遵循 IEC61131-3 标准，用户界面友好、操作简单、功能全面。

(2)工程管理器反映了 IEC61131-3 软件模型的结构，使用户清晰地了解所进行组态工作的全貌。管理器能够以不同的树型视图来显示不同的工程部分，操作基于 Windows 标准。通过界面良好的工程管理器，用户能够对工程中的各种资源进行管理。

(3)用户可通过对编辑工具栏中 LD 组件的拖拽轻松完成 LD 程序编辑工作。可使用不同语言进行交叉编程，编程方式灵活。对程序编辑过程中出现的一般错误，软件正确进行了提示，避免不必要的错误，极大提高了程序开发的效率。

(4)系统正确高效地完成了 LD 向 IL 的转换，编译生成的可执行程序在 PLC 中正确运行。运行过程中各个参数的变化在监视界面中以图形方式正确显示，方便进行程序的修改和维护。

(5)软件正确进行了异常处理，运行稳定。

6　结束语

本文对 IEC61131-3 标准进行介绍，给出基于 IEC61131-3 的 PLC 编程环境框架结构，对其中各个模块进行说明，给出主要的算法和数据结构，并针对较难实现的 LD 向 IL 转换的算法思想和实现步骤进行说明。实验证明该软件界面友好、编程方式方便灵活、易于维护，在一定程度上可以解决目前各种品牌 PLC 编程方法无法统一的问题，提高编程效率，减少培训、维护费用，具有广阔的应用前景。

该软件框架的设计思想为以后的开发打下较为坚实的基础，但是第 1 阶段只进行了初步的开发，要进一步提供全部 5 种语言，全面兼容 IEC61131 标准，还需要进行大量的工作。

（下转第 251 页）