

文章编号:1006-544X(2003)04-0378-04

单片机可视化通用开发系统模型研究与实现

麦范金, 蒋存波, 崔建明, 陆刚

(桂林工学院 电子与计算机系, 广西 桂林 541004)

摘要: 目前单片机开发系统的开发界面不友好, 而且各种开发系统通常都是针对某一特定类型 CPU 的, 从而制约了单片机的广泛应用与发展. 在介绍单片机应用程序汇编语言源代码结构特点的基础上, 通过对可视化、通用性的分析研究, 给出一个汇编语言的单片机可视化通用开发系统的系统模型与软件的实现, 并提供了 8051 系列 CPU 和 8096 系列 CPU 单片机应用程序的基本功能模块.

关键词: 通用性; 单片机; 开发系统; 系统模型

中图分类号: TP311.52; TP368.2

文献标识码: A^①

单片机系统^[1]包括硬件和软件两部分, 其中软件部分的特点是直接面向具体应用, 且多数情况下无操作系统支持, 因此, 软件的设计尽管可以用汇编语言或 C 语言等实现, 其开发还是存在很大的难度. 目前汇编语言的单片机应用程序开发系统, 大都是单片机厂家针对自己的产品而专门推出, 不支持其他类型的 CPU, 因而在一定程度上制约了单片机的应用与发展. 而可视化的内涵早已应用到 PC 软件设计中, 若能移植到单片机应用程序开发中, 必将大大提高其开发的效率和质量.

1 单片机应用程序结构特点

基于过程的单片机应用程序一般结构组成可分为主程序及功能模块(过程或实时中断服务)两大部分. 对于一般的任务, 可以以模块的形式被主程序调用, 而实时要求高的任务则以中断服务的形式进行处理. 主要特点有^[1~4]: ①由分支、顺序、循环等程序结构组成, 即程序具有结构化的特点; ②为了完成相应的功能, 一般都会使用到一些特定的功能模块, 即具模块化特点; ③单片机程序一般都被固化在 ROM 中, 由于 ROM 的容量有限, 所以要求代码的长度尽可能短; ④单片机应用程序一般是在系统加电后即开始运行,

直到掉电时才停止.

2 系统模型

2.1 可视化

所谓可视化编程是指在程序设计的过程中运用可视化形成的表达式, 例如图形、窗口、图标、菜单等使得程序设计的过程直观化^[5]. 目前, 有多种图形方式可以表示程序的整体结构, 如程序流程图、盒图、PAD 图等, 鉴于程序流程图的描绘直观、便于掌握, 且已被人们所熟悉并接受, 故选择程序流程图方式作为单片机应用程序设计实现可视化.

单片机的程序设计过程一般为: 在问题定义和建立数学模型的基础上, 划分出各个功能模块, 在软件结构设计阶段, 将这些模块以一定的程序结构连接起来, 即先设计出各个子程序模块, 然后在主程序中使用各种程序结构来控制这些子程序的运行. 据此, 在开发系统中应有各种图标以及它们与各种模块建立对应关系(图1)的定义, 然后通过拖动图标的方式来建立应用程序流程图, 而流程图的每一图标都与相应的子程序关联, 最终生成整个应用程序源代码.

系统应能正确、自动识别出所用 CPU 类型的

① 收稿日期: 2003-06-30; 修订日期: 2003-09-14

基金项目: 广西区教育厅资助项目(桂教科研[2002]121号)

作者简介: 麦范金(1963-), 男, 广西玉林人, 讲师, 计算机科学与技术专业.

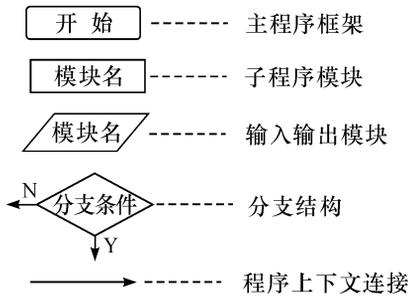


图 1 图标与程序结构对应关系

Fig. 1 Corresponding relations between icon and procedure structure

各个模块，故必须制定出各个图标所对应模块的汇编代码的书写规则。

①图标“主程序框架”所对应的程序

```
sjmp main
; 中断程序入口
main:
; 各功能调用
; end of main
```

②图标“子程序模块”所对应的程序

```
模块名:
实现该模块功能的汇编源代码 ret
```

③图标“输入输出模块”所对应的程序

```
模块名:
实现输入输出功能的汇编源代码 ret
```

④图标“分支结构”所对应的程序

```
分支模块名:
; here is the entry of the JUDGE
JZ 模块名
LCALL 模块名
; end of judge
```

2.2 通用性

目前，大多数单片机开发系统都是各个厂家针对自己的产品专门推出的，在一种开发系统中通常无法开发其他类型 CPU 的单片机应用程序，这就意味着当开发人员需要开发多种不同类型 CPU 的应用程序时，必须掌握多种开发系统的操作方法，这样造成的重复劳动是显而易见的。

从单片机程序结构特点分析可知：不论何种单片机，其应用程序都是由分支、顺序、循环等程序结构及运算模块、中断模块、输入输出模块等组成，且各种单片机的应用程序设计过程类同。据此，在开发系统中应增加对 CPU 类型选择的功能。通过选择 CPU 类型，即确定了相应的汇编指令系统及其汇编器，以及图标所对应的程序功能模块，使

开发系统能够支持多类型 CPU 而达到通用性。

由于系统具通用性，即系统中必须有不同类型 CPU 的汇编器及对应的基本功能模块，故应加以分类管理，拟采用树型目录结构进行管理（图 2）。即在系统所在目录下建立一个名为“cpu”的子目录，该子目录用来存放各种不同类型的单片机 CPU 的模块及汇编器，如子目录“8051”用来存放 8051 系列单片机的各种模块及其汇编器，子目录“8096”用来存放 8096 系列单片机的各种模块及其汇编器等。当用户需要添加自己所需的类型 CPU 时，只需按上面所述建立相应目录，并将汇编器及模块的代码复制到相应的目录下即可，使用户可根据需要对系统进行扩充。其中：

子目录 assembler 中存放汇编器。

子目录 model 中存放用户自定义的模块。

子目录 stdmodel 中存放系统提供的标准模块。

2.3 系统模型

通过对单片机开发系统可视化、通用性的分析研究，可以给出单片机可视化通用开发系统的系统模型（图 3）。模型由 9 个部分组成。

①CPU 类型选择：根据所选的 CPU 决定使用的汇编器及指令系统。

②图标模块库：主要负责绘制表示各个模块及程序结构的图标，供用户构建应用程序流程图用。

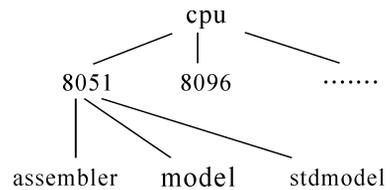


图 2 目录结构

Fig. 2 Directory structure

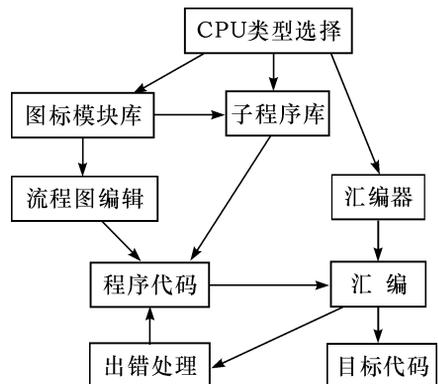


图 3 单片机可视化通用开发系统模型

Fig. 3 Microcontroller unit visual system development model

③子程序库：对基础平台的扩展部分提供所需的功能模块库，如运算模块、输入输出模块等，以文件形式存放在规定目录中。系统实现中仅提供 8051 及 8096 系列单片机汇编语言的运算模块的多字节加、减、乘、除运算和键盘输入、显示等模块的汇编源程序，其它功能模块用户可以随时添加。

限于篇幅，以下仅给出 8051 系列单片机多字节加法的汇编源程序，其余单片机及其它模块源代码略。

8051 (文件名: add.asm):

```

ADVADD:
MOV A, @R0
ADDC A, @R1
MOV @R0, A
INC R0
INC R1
DJNZ R2, ADVADD
RET

```

④流程图编辑器：实现应用程序流程图的工具。

⑤编译器：由 CPU 类型得到相应的汇编器。

⑥程序代码：根据应用程序流程图及子程序库导出应用程序源代码。

⑦汇编：调用相应的汇编器汇编源代码。

⑧出错处理：对汇编器的输出信息进行分析，根据出错信息，将其定位到文本编辑器中源代码的出错处。

⑨目标代码：由汇编器汇编得到的目标代码。

系统运行流程为：首先选择目标代码所要运行的 CPU 类型，在图标模块库中选取所需的模块，并添加到流程图编辑器中(以图标表示)，利用子程序库生成相应的汇编源代码。然后通过相应的汇编器汇编源代码，利用出错处理捕获出错信息且反馈给用户，并定位出错信息，最后得到目标代码。

3 软件实现

系统以目前最为流行的 Windows 操作系统作为平台，使用 VC++ 6.0 作为开发工具，整个系统采用面向对象的方法实现（限于篇幅，系统对象模型图、功能模型图及动态模型图略）。系统主界面见图 4。以下仅就系统可视化、通用性的实现进行讨论。

3.1 可视化实现

可视化包括图标操作（拖放、移动、命名和删除等）、应用程序流程图与汇编源代码的对应关

系、流程图的存取等 3 部分。

首先设计一个类，对系统中定义的 5 种图标的绘制、移动、删除及命名等操作进行封装。当用户拖动图标时，在 View 类的 OnMouseMove () 函数中记录鼠标移动的位置，并将该位置信息传递给正在移动的模块，然后使整个客户区无效，在 OnDraw () 函数中调用模块的绘制函数来对整个客户区重绘。当用户选择删除该图标（即模块）时，从模块链表中删除该模块，然后调用模块的绘制函数重绘整个客户区。

当用户需要产生代码时，根据该模块（图标）在流程图中的位置信息，在模块链表中查找该模块的前驱节点，然后根据前驱节点在源文件中的位置，就可以得到该模块在源文件中的位置，然后再将该模块子程序的源代码插入源文件的子程序区。流程图中的各个模块在内存中使用链表存储。

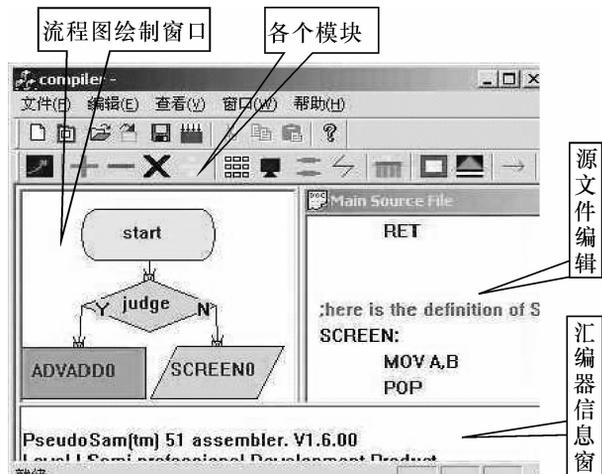


图 4 主界面

Fig. 4 Main page

对流程图存取的实现，采用一个链表结构存储流程图信息。在存储其信息时，指针的值无法存取，因此在存取时将该链表的各个模块节点按其前后顺序存入文件中，读入时根据文件中的节点信息重建一个链表，由于模块（图标）在流程图中的后继节点的信息也需要指针来存储，而在读入时无法得到该指针，故另外用一个整型数值表示各个模块在链表中的相对位置，这样，后继节点的信息就可以用整数来表示，因此访问链表比较方便、直观。

3.2 通用性的实现

包括 CPU 及汇编器选择、汇编器输出信息的处理、错误的定位、关键字的着色等。CPU 及汇编器选择的实现较为简单,在此不赘述。

对于汇编器输出信息处理,由于汇编器都是在 DOS 模式下,其输出信息采用重定向技术定向到程序中的显示窗口。即利用操作系统的 API 函数 CreateProcess 函数调用汇编器,然后用重定向符“>>”将输出信息定向到文件中,待汇编器汇编结束后,将该文件内容读入到程序中显示信息的窗口,然后将该文件中的信息删除,准备下一次汇编使用。

汇编器的输出信息中,包含一些关于出错位置的信息,如以下字符串:“unrecognized character --blank substituted in line 2, column =1”表示在第2行有错误,对这些表示出错信息的字符串进行分析,发现其有一定的输出格式,如在单词“line”后紧跟的数字指明了错误发生的行数,由此得出下列方法:在字符串中分离出紧跟“line”的数字,然后在文本编辑器中定位到该数字所指的行,这样就可以完成错误定位的功能,很显然,该方法不是对所有的汇编器都适用,这是在以后需要改进的地方。

关键字的着色是指在源代码中的指令关键字的现实颜色与其他(如操作数、注释等)的不同(图4)。实现过程中将关键字以文件的形式存放在磁盘上,在用户选择 CPU 类型及编程语言时将相应的关键字调入内存,当用户输入字符后,取得用户输入光标所在行的所有字符,并在其中查

找是否有注释符,如有则将该注释符后的所有该行字符改变颜色为注释行颜色(即紫色),然后在剩下的字符串中分离出各个单词,且与关键词逐一比较,如是关键词,则改变其颜色为蓝色;如果不存在注释符则将该行中的各个单词分离出来,分别判断其是否为关键词,是则改变其颜色为蓝色,从而实现了关键词的着色。

4 结束语

通过可视化、通用性的分析研究,建立了单片机可视化通用开发系统模型,并用 VC++6.0 进行了系统实现,使单片机应用程序开发人员能以直观流程图方式进行单片机应用程序的开发,即实现了单片机应用程序的可视化开发。同时,系统支持多种类型的 CPU,实现了可以在同一开发系统环境中开发不同类型 CPU 的单片机应用程序,系统具通用性。

参考文献

- [1] 袁涛. M65HC08 系列单片机原理与应用 [M]. 北京:北京航空航天大学出版社,1998.
- [2] 何立民. MCS-51 系列单片机应用系统设计——系统配置与接口技术 [M]. 北京:北京航空航天大学出版社,2000.
- [3] 孙涵芳,徐爱卿. MCS-96 系列 16 位单片微型计算机 [M]. 北京:北京航空航天大学出版社,2000.
- [4] 金磐石. Intel 96 系列单片机应用详解 [M]. 北京:电子工业出版社,2000.
- [5] 李愚. 可视化设计及其应用 [J]. 微型计算机,1995(5):251-257.

Microcontroller unit all-purpose visual development and implementation

MAI Fan-jin, JIANG Cun-bo, CUI Jian-ming, LU Gang

(Department of Electronics and Computer Science, Guilin Institute of Technology, Guilin 541004, China)

Abstract: At present, the interface of microcontroller unit's application development is restricted. Generally the development is working for a specific kind of CPU, not for all. Therefore the application of microcontroller unit in many fields is not wide. The structure of microcontroller unit's source code is introduced. Based on the research of the visual and all-purpose programming of the microcontroller unit, the system model of the microcontroller unit all-purpose visual development is established, and the software is achieved by VC, providing some basic function models of 8051 and 8096 CPU.

Key words: all-purpose; microcontroller unit; development system; system model