

用于嵌入式系统的汉字点阵提取方法

王诗彬,林聪仁

(厦门大学 信息科学与技术学院 福建 厦门 361005)

摘要:嵌入式系统人机交互性能的重要指标之一是汉字显示打印功能,利用汉字点阵实现此功能是一种简单实用的方法,因此从现有汉字系统中提取汉字点阵是嵌入式系统开发的第一步工作。提取汉字点阵有两种方法,一是从点阵字库中提取,二是从矢量字库中提取,这两种方法在实际开发工作中都得到了应用。详细分析了这两种方法的原理及编程实现,提出了自动扫描嵌入式应用程序生成汉字点阵的方法,这种方法特别适用于建立小型汉字点阵库。

关键词:嵌入式系统;汉字点阵;点阵字库;矢量字库;自动扫描

中图分类号: TP368.1

文献标识码: B

文章编号: 1004-373X(2007)07-162-04

The Methods of Abstracting Chinese Dot Matrix for Embedded Systems

WANG Shibin, LIN Congren

(School of Information Science and Technology, Xiamen University, Xiamen, 361005, China)

Abstract: It is a simple and practical method to use the Chinese dot matrix to display and print Chinese characters in embedded systems. Therefore, abstracting dot matrixes from the existing Chinese character OS is the first step of developing embedded system. We can abstract dot matrixes from the dot matrix library or from the vector library. Authors have applied both ways in practical applications. This paper analyses the principles and programming of both ways in detail. In addition, this paper proposes a method of auto-scanning source files, this method is best for creating mini dot matrix library.

Keywords: embedded system; Chinese dot matrix; dot matrix library; vector library; auto-scan

嵌入式系统要支持汉字显示打印功能需要两步基础工作,一是从现有汉字系统提取所需汉字的汉字点阵,二是根据嵌入式系统的不同要求,逐行或逐列读取每个汉字的点阵信息。本文就第一步工作做详细讨论,第二步工作不同的嵌入式系统开发环境有关,以后另文讨论。汉字点阵的提取有两种方法,一是从点阵字库提取,二是从矢量字库提取,这两种方法的原理和适用范围各不相同。现有汉字点阵提取工具种类不少,但功能并不一定完全满足各种嵌入式系统开发的需要,为了编制适合自己的嵌入式系统需要的汉字点阵生成工具,必须了解点阵字库和矢量字库的组织结构,掌握汉字点阵提取的编程原理,根据各种嵌入式系统对汉字显示打印功能的不同要求选择不同编程方法。显示打印单一字体字号汉字的嵌入式系统可采用从点阵字库提取汉字点阵的方法;显示打印多种字体字号汉字的嵌入式系统可采用从矢量字库提取汉字点阵的方法。笔者在实际开发工作中研究了这两种方法的原理并用 Visual C++ 6.0 编制了相应的汉字点阵提取工具。本文在简要介绍点阵字库和矢量字库结构的基础上,详细分析这两种方法的编程原理和流程。这两种方法都必须从含有汉字参数的嵌入式应用源程序中提取无重复汉字字符串,实现这一功能

最快速有效的方法是源程序自动扫描。

1 从点阵字库提取汉字点阵

1.1 方法分析

诸如 UC DOS, SP DOS 等早期使用的汉字操作系统都是使用点阵字库来实现汉字功能,按不同点阵规格和不同字体,将 GB2312-80 通用汉字字符集的 6763 个汉字的点阵数据存于不同的点阵字库文件中。尽管那些汉字操作系统已很少使用,但这些点阵字库文件还有很大用处。了解点阵字库文件结构就可以从中提取所需的汉字点阵。

点阵字库文件按机内码顺序存储全角字符或汉字,分 94 区,每区 94 个全角字符或汉字,为了与 ASCII 码区分,全角字符或汉字的机内码的高低字节都是从 0A1H 开始编号。嵌入式系统源程序内的汉字都是以 2 个字节的机内码形式存储的,用此机内码查询点阵字库文件可得到该汉字的汉字点阵。以 16×16 点阵字库为例,每个汉字的点阵数据占 32 个字节,已知某汉字机内码,求此汉字点阵数据首地址的公式为:

$$32 \times [(\text{机内码高字节} - 0A1H) \times 94 + (\text{机内码低字节} - 0A1H)] \quad (1)$$

从此地址开始连续读取 32 个字节,就是此汉字的汉字点阵。

收稿日期:2006-11-16

基金项目:厦门大学 985 二期信息创新平台项目资助

1.2 原理流程

从点阵字库提取汉字点阵方法的流程如图 1 所示,图中列出了每项操作所用的函数。首先用源程序自动扫描方法生成汉字字符串文本文件(hzzfc.txt)(详见第 3 部分),此文件存储要提取汉字点阵的所有汉字的机内码。打开点阵字库文件(HZK16)后,用 fseek 将汉字字符串文本文件指针移至文件头,顺序取出每个汉字的机内码,根据式(1)计算点阵数据的首地址,用 fseek 定位至此地址,用 fgetc 连续读取 32 个字节存入汉字点阵文件(Mychar.c),所有汉字的汉字点阵提取完毕后,用 fprintf 添加汉字索引数组。最后用 fprintf 在头文件(Mychar.h)添加数据类型定义和外部数组声明。程序运行结果如图 2 所示。

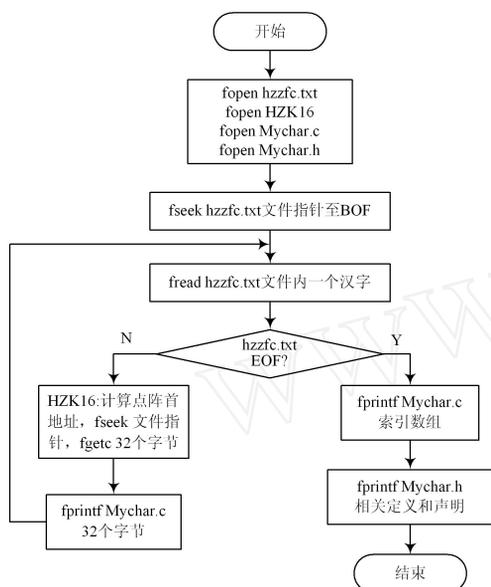


图 1 从点阵字库提取汉字点阵流程图

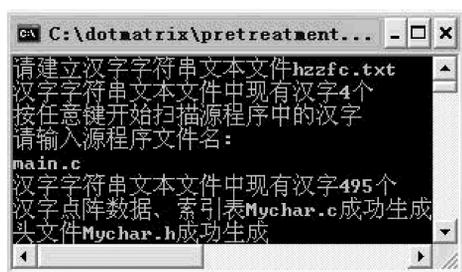


图 2 从点阵字库提取汉字点阵程序运行结果

1.3 生成文件及使用

如图 1 所示,汉字点阵提取程序生成 2 个文件供嵌入式应用源程序使用,不同的嵌入式开发系统对这 2 个文件的格式要求可能略有不同。头文件 Mychar.h 如表 1 所示,定义了数据类型,声明了外部数组。此头文件要添加至嵌入式应用的工程项目内并 include 到嵌入式应用源程序中。表 2 是汉字点阵及索引文件 Mychar.c 的一个简化例子,此文件要添加至嵌入式应用的工程项目内。索引数组是一维数组,汉字点阵数组是二维数组,行下标为序号,

列数为 32。嵌入式应用源程序先查询索引数组,得到汉字序号,再根据序号读取汉字点阵。

表 1 Mychar.h 文件内容

```
#ifndef _MYCHAR_H
#define _MYCHAR_H
typedef unsigned int UINT
typedef unsigned char UCHAR
extern const UCHAR WordIndex[]
extern const UINT WordLib[][32]
#endif
```

表 2 Mychar.c 文件内容

```
const UCHAR WordIndex [] = "汉字显示 ..."; // 索引表
const UINT WordLib[][32] = { // 点阵库
{0x00,0x00,0x40,0x08,0x37,0xfc,0x10,0x08,
0x82,0x08,0x62,0x08,0x22,0x10,0x09,0x10,
0x11,0x20,0x20,0xa0,0xe0,0x40,0x20,0xa0,
0x21,0x10,0x22,0x08,0x24,0x0e,0x08,0x04,}, // 汉
{0x02,0x00,0x01,0x00,0x3f,0xfc,0x20,0x04,
0x40,0x08,0x1f,0xe0,0x00,0x40,0x00,0x80,
0x01,0x04,0xff,0xfe,0x01,0x00,0x01,0x00,
0x01,0x00,0x01,0x00,0x05,0x00,0x02,0x00,}, // 字
...};
```

2 从矢量字库提取汉字点阵

2.1 方法分析

Windows 等操作系统提供了丰富的矢量字库来实现汉字功能。矢量字库有字型美观、可不失真缩放等优点,但占用存储空间比点阵字库大得多,如“仿宋体(TrueType)”SIMFANG.TTF 大小为 3 904 kB。大部分中低档嵌入式系统对汉字质量要求不高,将矢量字库转换为汉字点阵可大大降低存储器成本。

矢量字库用坐标、半径及弧度等参数描述汉字的笔划信息,有多种不同的数学模型,每种矢量字库又由多个表格组成,结构复杂。若直接使用矢量字库算法非常复杂,对中低档嵌入式系统不适用。显示器上每个点对应显示缓冲区的一个象素点,在白色背景下显示黑色汉字,然后读取一个汉字对应的所有象素点,就可得到此汉字的汉字点阵。矢量汉字显示和象素点读取功能有现成的 API 函数可以调用,添加代码可将象素点信息转换成以字节为单位的汉字点阵。

2.2 编程实现

笔者用 MFC 制作了从 TrueType 字库提取汉字点阵的软件,界面如图 3 所示,为明了起见并节省篇幅,“编辑”和“格式”两菜单并列显示在同一图中。软件基本框架程序用“MFC App Wizard”生成,在基本菜单“编辑”下增加“打开文本”、“扫描文本”和“提取点阵”子功能,增加“格

式“菜单”及“字体”和“自动换行”子功能。在“打开文本”通用对话框可以选择 txt、c、asm 扩展名的文件显示于编辑窗口;“扫描文本”提取无重复汉字覆盖编辑窗口;“提取点阵”先用“另存为”通用对话框指定文件名和扩展名,然后将提取的汉字点阵数据存于该文件中并生成数据类型定义和外部数组声明的头文件。“格式”中的“字体”功能使用“字体通用对话框”,其中,字号选择 12 可以得到 16 × 16 点阵;“自动换行”功能使编辑窗口中的文本适应窗口宽度。下面介绍几个主要功能的编程实现。



图 3 从矢量字库提取汉字点阵软件界面

2.2.1 提取点阵

提取点阵函数 OnMatrixPickup() 流程如图 4 所示,先调用 CFileDialog::DoModal() 函数实现“另存为”通用对话框功能,然后按顺序进行如下操作:

(1) 调用 CDC::SelectObject() 函数将创建的字体选入关联设备(device context)中。

关联字体设备的成员函数是 CFont * SelectObject(CFont * pFont);

(2) 调用 CDC::GetTextExtent() 函数获取要显示的每个汉字的大小。

函数原型为 CSize GetTextExtent(LPCTSTR lpszString,int nCount);

(3) 调用 CDC::SelectObject() 函数,根据汉字大小创建位图并选入关联设备中。

关联位图设备的成员函数是 CBitmap * SelectObject(CBitmap * pBitmap);

(4) 调用 CDC::TextOut() 函数绘制汉字于内存的位图空间。

函数原型为 BOOL TextOut(int x,int y,const CString & str);

(5) 循环调用函数 CDC::GetPixel() 函数扫描内存位图上相应区域的像素点以获取汉字点阵信息。

函数原型为 COLORREF GetPixel(POINT point), 返回值为由参数 point 所指定的像素点的 RGB 值。默认白色背景 RGB(255,255,255),黑色字体 RGB(0,0,0),逐行扫描每一像素点,转换为字符型数组,某点为 RGB(0,0,0),对应位置 1,否则置 0。一行扫描结束后,调用

CFile::Write() 函数将数组写入.c 文件。扫描一行像素点的代码如下:

```
pChar = new BYTE[iBytes];
memset(pChar,0,iBytes);
for(j=0;j<sWordSize.cx;j++)
{if(pDC->GetPixel(CPoint(j,i))=RGB(0,0,0))
pChar[j/8]|=0x80>>(j%8);}
```

(6) 调用 CGdiObject::DeleteObject() 函数释放关联设备,避免内存泄漏。

函数原型为 BOOL DeleteObject()。

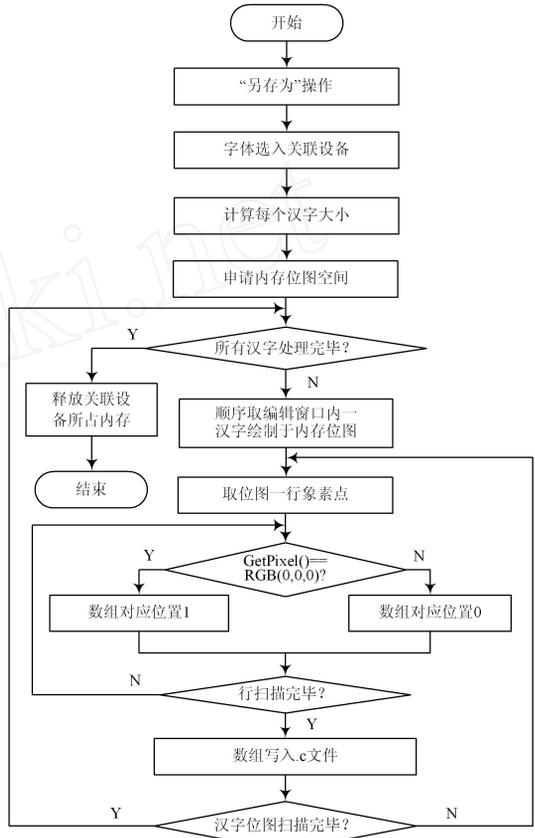


图 4 OnMatrixPickup() 函数流程图

2.2.2 选择字体

与选择字体有关的程序代码有 3 段:一是在视图类头文件中声明公有型(public)成员变量 LOGFONT m_LogFont 和 CFont m_Font;二是在消息 WM_CREATE 处理函数 OnCreate() 中初始化字体属性结构体 LOGFONT,运行 CFont 成员函数 m_Font.CreatePointFont(120, "Fixedsys");三是在“字体”菜单消息处理函数 OnFont() 中创建新字体,主要代码如下:

```
LOGFONT lf;
CFileDialog cf(&lf);
if(cf.DoModal()==IDOK) // 创建字体选择对话框
{cf.GetCurrentFont(&lf); // 获取字体信息
this->m_Font.DeleteObject(); // 删除旧字体
this->m_Font.CreateFontIndirect(&lf); // 创建新字体
Invalidate();
UpdateWindow(); // 重绘视图
}
```

2.2.3 其他功能

“编辑”菜单下“打开文本”消息处理函数 OnTextOpen() 用 CString 类申请字符型暂存区,用 CStdioFile::ReadString() 读文件内字符至暂存区,用 CWnd::SetWindowText() 显示文本。“自动换行”消息处理函数 OnAutoreturn() 用 CWnd::ShowScrollBar() 控制水平滚动条显示/关闭。这两项不是主要功能,在此仅作简要介绍。“扫描文本”消息处理函数 OnEditScanText() 主要功能是源程序自动扫描,基本流程与图 5 相同,不同的是所用函数是 MFC 成员函数,而第一部分所用函数是标准 C 文件操作函数。

3 源程序自动扫描

3.1 方法分析

不同嵌入式系统对汉字点阵库的大小要求不同。诸如 PDA、手机等嵌入式系统必须存储和使用完整的 GB2312-80 汉字字符集,建立完整汉字点阵库只需按汉字机内码顺序提取所有汉字的汉字点阵,编程较为容易,这里不做讨论。完整的 16×16 汉字点阵库占内存 200 多 kB,大部分中低档嵌入式系统只需显示或打印部分汉字,为了节约硬件成本,使用小型汉字点阵库为宜。

建立小型汉字点阵库有不少现成的“字模提取工具”可供选用,但这些工具都存在某些缺点,不一定完全适应用户的需要。最大的缺点是需手工输入所需汉字字符串到编辑框或文本文件中,比较费时且容易遗漏或重复,嵌入式系统中使用的汉字字数增加或减少时需手工修改已存的汉字字符串,重新生成汉字点阵库,增加很多重复工作量。笔者在使用过多种字模提取工具后,为克服上述缺点,编制了源程序自动扫描程序。在嵌入式应用程序编程过程中若汉字参数数量有增减,则在调试前运行自动扫描程序,就能无重复地将所需汉字存入汉字字符串文本文件中,大大提高了工作效率。从点阵字库或矢量字库提取汉字点阵时,源程序自动扫描的原理流程相同,所用函数不同,下面以第一种情况为例说明其原理。

3.2 原理流程

源程序自动扫描生成汉字字符串文本文件的流程如图 5 所示。源程序 main.c 是纯文本文件,存储的都是西文字符和控制符的 ASCII 码及中文字符的机内码。ASCII 码是单字节,二进制最高位为 0,汉字及全角符号的机内码为 2 个字节,都大于 0A0H,据此可把汉字区分出来。从文件开始到结束扫描源程序 main.c 的每个字节,当遇到注释符时文件的位置指针移至注释内容后继续扫描,当遇到 ASCII 码时位置指针加 1,当遇到大于 0A0H 的字节时,即是汉字机内码的高字节,下一字节即为汉字机内码的低字节。

作者简介 王诗彬 男,1982 年出生,硕士研究生。

得到一个汉字机内码后,顺序与汉字字符串文本文件中已存的每个汉字进行比较,若已有此汉字,则不存入此汉字,若无此汉字,则添加至文件末尾,这样可以保证汉字字符串文本文件中没有重复汉字。上述的顺序查找方法流程较简单,但效率较低,若要提高程序运行速度,可以采用排序及对分搜索的方法,限于篇幅,不再赘述。

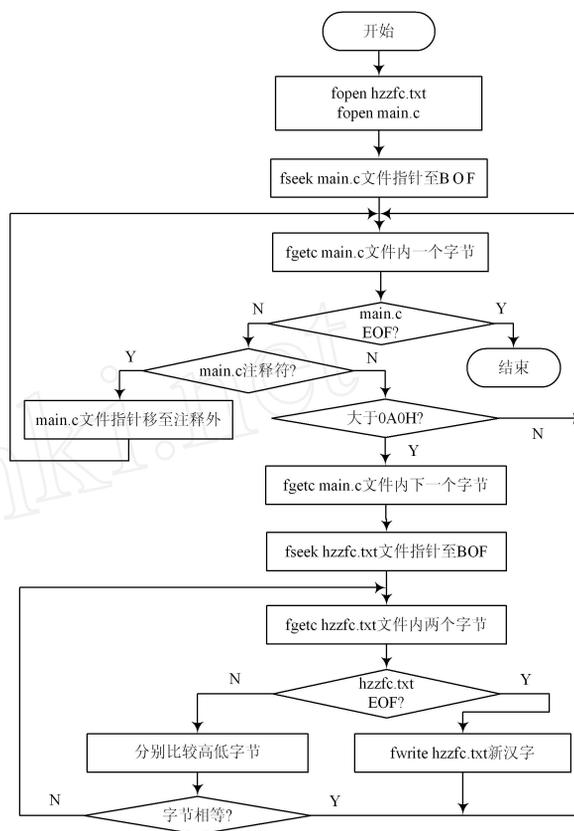


图 5 源程序自动扫描方法流程图

4 结语

汉字点阵是嵌入式系统实现汉字显示和打印功能的基础,本文剖析了从点阵和矢量两类字库提取汉字点阵的原理及关键技术,源程序自动扫描方法克服了手工方法及字模提取工具的缺点,大大提高了工作效率。嵌入式系统开发人员必须掌握这些实用的技术和方法。

参考文献

- [1] 安学军,张永军,梁祥,等.汉字的动态编码与显示方案[J].单片机与嵌入式系统应用,2003(6):183-185.
- [2] 万伟,陈明宏,夏洪星.矢量字库在嵌入式机顶盒中的应用[J].单片机与嵌入式系统应用,2004(10):53-56.
- [3] 赵艳红,费洪晓.基于.NET的任意点阵字库生成方法[J].现代计算机,2004(6):72-75.