VxWorks5.5 平台下矢量字体显示的实现①

凌云锋

(江苏自动化研究所，连云港 222006)

摘 要：VxWorks5.5 采用点阵字库实现字体显示，这种字库设计简洁，应用广泛，但一个字库只能对应一种字体的一种大小，在不确定使用何种字体的情况下，这种传统的字体显示方式便不能够满足需求。通过使用 TrueType 字库和 FreeType 字体引擎相结合的方式，能实现多种字体、任意大小的显示功能。主要介绍了 TrueType、FreeType 技术的基本原理，以及在 VxWorks5.5 下如何将 WindML、FreeType 和 TrueType 三者相结合实现矢量字体显示的方法。

关键词：矢量字体；编码转换；字体库加载；粗体；斜体

Implementation of Outline Font in VxWorks5.5

LING Yun-Feng

(Jiangsu Automation Research Institute, Lianyungang 222006, China)

Abstract: Bitmap fonts are used to display in VxWorks5.5, which is simple and widely used. But it will not meet the demand when we don't make sure which font size is ok as each bitmap font file has only one font and one size. We can achieve a variety of fonts and sizes by using the TrueType fonts and FreeType font engine. This paper describes the basic principle of TrueType and FreeType technology, and how to realize outline font display by using WindML, FreeType and TrueType fonts .

Key words: outline font; encoding translating; font file loading; bold; italic

1 VxWorks5.5点阵字库的局限性

VxWorks5.5 是美国风河公司开发的嵌入式操作系统，图形系统采用 WindML3.0，支持点阵字显示，不支持矢量字体显示。点阵字采用内存模式加载，使用前需要将字体库先加载到内存，再通过 WindML 图形接口实现点阵字显示。点阵字库采用.c 文件方式储存信息，每个字信息都包含在一个数据结构中，其中包含了字体的所有点信息，字体显示时只要根据字的宽高将点信息直接送入显存显示。这种方式的字体实现简单、显示速度快，但是一个字体文件只包含一种字体、一个大小的信息，使用具有一定的局限性。

随着嵌入式软件的不断发展，在设计象嵌入式浏览器这样的人机界面的软件时，发现点阵字库已经远远不能满足设计要求，嵌入式浏览器对字体的需求是根据网页内容来决定的，在网页上任何类型，任何大小的字体都可能出现，点阵字库要将所有字体类型，每种字体的所有大小都包括是不可能的，这种局限性大大降低了浏览器的显示效果。TrueType 字库引入到 VxWorks5.5 系统下，有效的解决了字体的问题，所有 Windows 下的 TrueType 字库都可以在 VxWorks5.5 系统直接使用，资源非常丰富，能满足嵌入式系统对字库的新需求。

2 TrueType字库原理及FreeType字体引擎

TrueType 是 Apple 公司和 Microsoft 公司合作开发的页面描述语言（简称 TTF），采用了直线和二次贝赛尔曲线来描述字符的轮廓，结合了光栅技术和矢量技术的优点，克服了以往所有点阵字体、矢量字体和向量轮廓字体的缺点，字体可以任意放大、缩小、旋转和变形而不会影响输出质量，提供了真正的设备无关

① 收稿时间:2010-10-30;收到修改稿时间:2010-11-25

性，二次贝赛尔曲线既能保证轮廓曲线的光滑性，又有利于提高字形还原的速度[1]。如下图 1 所示。

图 1　TrueType 字体轮廓图

FreeType 是一个完全免费的、高品质的可移植的字体引擎，它提供同一的接口访问多种字体格式，包括 TrueType，openType，CID，CFF 等。支持单色位图，反走样位图的渲染，FreeType 库是高度模块化的程序库，它使用 ANSI C 开发，但采用面向对象的思想，FreeType 用户可以灵活地对它进行裁剪。

3　VxWorks5.5下矢量字库的实现

VxWorks5.5 下矢量字库采用开放源代码的 Freetype 库和 Windows 下的 TrueType 字库结合实现，通过 WindML 图形系统将矢量字应用到 VxWorks5.5 系统中。矢量字使用前先初始化 WindML 图形系统，再初始化矢量字库，并将矢量字库的接口函数挂接到图形系统下，在应用矢量字库时只需调用 WindML 接口函数，调用方式和点阵字库一致，实现了与 WindML 的无缝挂接。TrueType 字库根据加载方式不同分为动态加载和静态加载两种方式，动态加载方式是将 TrueType 字库拷贝到目标机硬盘，根据应用程序的设计要求在程序运行时动态加载字库；静态加载方式是将 TrueType 字库在系统启动时便加载到目标机内存，应用程序可以直接调用字库信息。动态加载方式优点在于节省内存和加载灵活，缺点在于不同字体切换时消耗的时间长，不适合需字体的频繁切换的应用程序；静态加载方式优点在于不同字体切换时消耗的时间短，适合需字体的频繁切换的应用程序，缺点在于内存消耗大，加载不灵活。

3.1 矢量字体的初始化

矢量字库的初始化主要有矢量字体设备创建和矢量字体设备注册两部分组成。先创建矢量字体设备，如果创建成功则将矢量字体设备注册到系统中，如果创建不成功则退出程序。

矢量字体设备创建函数 UGL_FONT_DRIVER *uglFT2FontDriverCreate(UGL_UGI_DRIVER*pDriver, UGL_FT2_FONT_DRV_CFG *pFT2FontConfig)，参数 pDriver 为图形系统设备号，取值 graphicsDevID 为 WindML 初始化时创建的图形系统设备号；参数 pFT2FontConfig 为字体配置结构，根据字体加载的方式不同参数也不同，具体见 3.4 章节；返回值 ft_fontDevID 为矢量字体设备号。

设备注册函数 UGL_STATE uglRegistryAdd (UGL_UINT32 type, UGL_UINT32 data, UGL_UINT32 id, char *name)，参数 type 为矢量字体设备类型，需定义一个新设备类型 UGL_FONT_ENGINE_FTTYPE，取值为 13；参数 data 为图形系统设备 ID，取值(UGL_UINT32)graphicsDevID；参数 id 为矢量字体设备号，取值(UGL_UINT32) ft_fontDevID；参数 name 取值 0。

3.2 字体单双字节编码转换

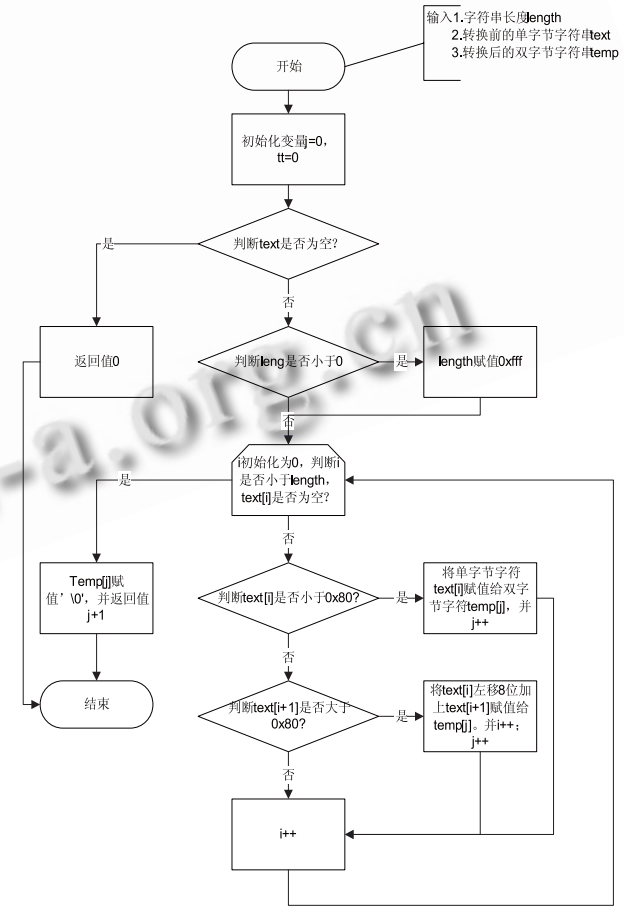

图 2　单字节字符转换成双字节字符

字符编码根据长度分为单字节和双字节两种编码

方式，单字节编码包括英文字母、数字和特殊字符等，双字节编码包括汉字和自定义字符等。

WindML 字体显示分双字节显示和单字节显示两种方式，双字节显示是两个字节作为字体编码对字库进行查询，找到字符位图并显示；单字节显示是单个字节作为字体编码对字库进行查询，找到字符位图并显示。当英文字符显示时，可以使用单字节显示或双字节显示，当中文字符显示或中英文混合字符显示时必须使用双字节显示[2,3]。

VxWorks 下字体采用 GB2312 编码，中文字符编码的每个字节都大于 0x80，英文字符编码都小于 0x80，在进行双字节显示时，需要将单字节字符转换成双字节字符。在字符转换时，先获取整个字符串长度，再判断每个字节是否大于 0x80，如果小于 0x80，则将单字节扩展成双字节；如果大于 0x80，则将这个字节与后个字节组合成一个双字节；计算双字节数并返回，如上图 2 所示。

3.3 字体编码转换

VxWorks5.5 下汉字采用 GB2312 编码，而 FreeType 在处理汉字时只能识别 Unicode 编码[4]，在处理汉字前需要将 GB2312 编码先转换成 Unicode 编码，GB2312 与 Unicode 的编码转换表采用二维数组保存数据，共有 7000 多组对应项，如果采用遍历数组的方式来进行编码转换，那么平均每个汉字编码转换需要做 3000 多次的编码比较，这非常影响汉字的处理速度。

为了提高编码转换的处理速度，编码转换时采用折半查找方式来实现，使用折半查找需要先将 GB2312 编码从小到大排列，每个 GB2312 编码对应一个 Unicode 编码。在使用折半查找时，先取 first=0 end=数组长度，然后(first+end)/2 得到一个中间编号，再通过中间编号获取相应的 GB2312 编码和显示汉字编码比较大小，如果中间值大，则将 first=0 end=中间编码组合再进行折半查找；如果中间值小，则将 first=中间编码 end=数组长度 组合再进行折半查找；如果相等，则将 GB2312 编码对应的 Unicode 编码提交程序处理。使用折半查找一个汉字最多只需查找 13 次，大大提高了汉字 Unicode 编码的查找速度，加速了汉字显示。

3.4 字库加载

字体库加载方式分动态和静态两种，两者之间互有优缺点，可根据用户的不同需求自主选择加载方式。

3.4.1 字库动态加载

字库动态加载方式是将 windows 下的 TrueType 字体库文件（*.ttf，*.ttc）拷贝到目标机目录下，根据用户需求在程序执行过程中动态加载字库。动态加载的实现方法：先声明两个结构变量，

1) UGL_FT2_FONT_DRV_CFG ft_font_cfg;

2) UGL_FT2_FONT_PATH_DESC FontPathDesc;

接着设置 FontPathDesc 信息，FontPathDesc.PFontSearchPath="/ata0a/ttf/"; FontPathDesc.filter="*.ttf"; pFontSearchPath 为字体文件搜索路径，filter 为文件过滤器。再设置 ft_font_cfg 信息，ft_font_cfg.numFontPathDesc=1;ft_font_cfg.pFontPathDesc=&FontPathDesc; ft_font_cfg.defaultCharset=FT_ENCODING_UNICODE; numFontPathDesc 为字体搜索路径的个数，pFontPathDesc 为搜索路径，defaultCharset 为设置矢量字体的编码模式。最后按照 3.1 章节对矢量字库进行初始化。

3.4.2 字库静态加载

字体库静态加载方式是将 windows 下的 TrueType 字体库文件（*.ttf，*.ttc）编译生成一个.o 文件，并在应用程序执行前先加载到内存。静态加载的实现方法：在编译生成.o 字库文件前，先确定需要加载的 TrueType 字体库文件，例如需要将 f:/font/目录下的 simsun.ttc，simkai.ttf 文件编译生成一个.o 文件，先创建一个 udft2cfg.s 文件，将需要编译的字库信息填入文件，再使用编译命令 ccpentium -mtune=pentium –march =pentium -O2 -nostdlib -fno-builtin -fno-defer -pop -DCPU=PENTIUM -DTOOL_FAMILY=gnu -D_WRS_KERNEL-DVXWORKS -xassembler-with-cpp -g -c udft 2cfg.o udft2cfg.s 编译字体文件，生成 udft2cfg.o 文件。

静态加载方式在矢量字体初始化时所用的字体结构信息与动态加载有一些区别，静态加载所用字体信息已经明确，在初始化时需要将字体信息在结构中描述清楚，结构 UGL_FT2_FONT_MEMBUF_DESC 描述字体名称、起始地址，终止地址，并挂接到结构 UGL_FT2_FONT_DRV_CFG 下，最后按照 3.1 章节对矢量字体进行初始化。

3.5 矢量字体的粗、斜体实现

矢量字体显示方式包括正体、粗体、斜体、和粗斜体四种方式，FreeType 字体引擎已经实现了对各种显示方式的支持，但要在 VxWorks5.5 上支持粗、斜体，

需要修改 udft2fnt.c 和 uglfont2.c 的部分代码。udft2fnt.c 修改代码如下：

1) 在 UGL_FT2_FONT 结构中增加两个结构变量，UGL_SIZE weightsize; UGL_SIZE italicsize;

2) 在 uglFT2FontCreate 函数中，去除三个条件 pFontDef->weight >=pFT2FontDesc->header.weight.min && pFontDef->weight <= pFT2FontDesc-> header. weight.max && pFontDef->italic == pFT2FontDesc-> header. Italic；增加字体结构变量 weightsize，italicsize 的赋值，pFT2Font->weightsize = pFontDef->weight; pFT2Font->italicsize=pFontDef->italic；

3) 在 uglFT2FontInfo 函数中，增加粗体、斜体信息的设置和获取代码，

case UGL_FONT_WEIGHT_SET: pFT2Font-> weightsize = *((UGL_SIZE *)pInfo);break;

case UGL_FONT_WEIGHT_GET: (*(UGL_SIZE *)pInfo) = pFT2Font->weightsize;

status = UGL_STATUS_OK;break;

case UGL_FONT_SLANT_ANGLE_SET: pFT2 Font->italicsize = *((UGL_SIZE *)pInfo);break;

case UGL_FONT_ SLANT_ANGLE _GET: (*(UGL_SIZE *)pInfo) = pFT2Font->italicsize;

status = UGL_STATUS_OK;break;

4) 在 ft2DrawStringImageCache 函数中，增加矢量字体在斜体时的矩阵值；增加矢量字体在粗、斜体时字体位图索引的获取。因为矢量字体在粗、斜体时矩阵值和位图索引号的获取和正体有些差异，所以在处理时需和正体分开处理。

5) 在 ft2DrawStringSmallBitmaps 函数中，增加矢量字体在斜体时的矩阵值；因为矢量字体在粗、斜体时使用 ft2GetGlyphIndex 函数不能正确获取位图索引，修改为 FT_Get_Char_Index 来获取位图索引；增加在粗、斜体时的矢量字体位图的处理。

uglfont2.c 修改代码如下：

修改 uglConstructFontDef 函数，增加斜体信息赋值，pFontDefinition->italic = pFontDescriptor-> italic。去除语句 pFontDefinition->weight = (pListArray [matchIndex].fontDesc.weight.min + pListArray [matchIndex].fontDesc.weight.max)/2;，使用语句 pFont Definition->weight = (pFontDescriptor-> weight.min + pFontDescriptor->weight.max)/2;替换。

4 结论

矢量字库已应用于嵌入式浏览器、嵌入式阅读器等多个软件开发项目，实际工程应用表明，矢量字体的切换速度、显示速度都能满足应用要求，并且字体大小的无级缩放、粗斜体显示、以及旋转显示等效果能使人机界面更加友好，使用更加便捷。

参考文献

1 孙枫,陈业夫,郭勇鹏.在 VxWorks 系统中使用 TrueType 字库.应用科技, 2003,30(11):53−55.

2 WindML DDK 3.0 PROGRAMMER’S GUIDE. Wind River Systems, Inc.2002.

3 WindML SDK 3.0 PROGRAMMER’S GUIDE. Wind River Systems, Inc.2002.

4 孔祥营,柏桂枝.嵌入式实时操作系统 VxWork 及其开发环境.北京:中国电力出版社, 2002.