

Implementation of UCDOS Font Interface Based on QT

Jian Huang, Qian Zhao, GuoMing Li

School of Communication and Information Engineering

Xi'an University of Science and Technology, Xi'an, China,710054

huangjian_work@163.com

Keywords: Embedded system; QT; Font library; UCDOS

Abstract. After analyzing the interface work principle of the lattice font supported by QT and UCDOS font feature, It is found that the open source QT is lack of the lattice font more than 16*16 size. For this shortcoming, the UCDOS font is combined with the QT seamlessly. Thus, the QT can use any font in the UCDOS and support the big font and many font styles, greatly economizes storage cost. This interface can be widely applied in embedded GUI interface design and development based on QT.

Introduction

Qt is cross-platform application and UI framework. It includes cross-platform class library, integrated development tool and cross-platform. Developing applications only needs one time by using Qt, and it can arrange these applications without programming source codes once more by crossing different desktops and embedded operating systems to. Presently the font library that the open source embedded Qt version supports just provides that of `unifont_160_50.qpf` which supports Chinese only. The size of the font is no more than 16×16 pixels and the style is single. Therefore, this article based on the Qt3 the last open source version Qt 3.3.8b realizes the interface of UCDOS dot matrix font library (the size is about 500 K), empowers that it's able to use the ample dot matrix font library source of UCDOS on Qt.

The dot-matrix library supported by embedded QT

QT supports TTF vector font library (it needs the assistance of the third-part component----FreeType2 interface library), but the storage cost of TTF vector font library is higher than embedded system and it's vector processing algorithm adds the processor's afford. In addition, QT internally installs two kinds of dot matrix font libraries which are QPF and BDF, of which BDF (The Glyph Bitmap Distribution Format) is the bitmap dot matrix font library established by Adobe while QDF is really the one that supports font cache which designed by QT. The storage principles of QPF and BDF are the same that they both belong to bitmap dot matrix. So the main topic of this article is discussing the interface of QPF font library.

1) QPF Dot Matrix Font Library

QT installs internally QPF dot matrix font library interface and supports rapid internal memory loading and buffer memory. Moreover, it has higher property while retrieving a character font from any other font libraries (TTF vector font library or BDF font library), first of all, QT transforms it into a dot matrix bitmap (QPF font of dot matrix format) whose size is the same as the character font and then adds it to the font cache list of memory pool (the font cache list can be save as document and the suffix is qpf), finally, submits it to the display driver to draw on the graphical interfaces. QPF font library loads all the font data to the storage (the font cache list) at one time, thus, the retrieval speed and display is very fast. Accordingly, the QPF font library has advantages over the storage cache dot matrix library, but the amount of its font is much larger than that of pure dot matrix and every pixel of it reaches as much as 256 class gray levels (every font character is a Qglyph instance object). Therefore, the show effect is better than that of 2 class gray levels's resisting dot matrix saw smooth. However, the size of QPF font library file is largening along with the font does, adding the

storage cost of the embeded system.

2)The Method of Abstracting QPF Dot Matrix Font Library Under QT

As like as the map of the memory font library, the QPF interface is the middle link from searching to showing of all the font libraries.Thus,this passage will connect UCDOS font library with it,making that QT can use Chinese font library in several styles under UCDOS and the size of UCDOS font library is no more than 0.8M,enormously saving the storage space.

According to the codes above,the customization of QPF dot matrix font library comes from memory manager in fact(every QPF font is a Qglyph inatance object).Yet,it doesn't have its own font style.In the case of the same size of font,QPF dot matrix(every pixel occupies 8 bits---256 class gray levels) font liabrury spares approximately as much as 8 times storage space than UCDOS(every pixel occupies 1 bits---2 class gray levels).

The feature analyzing of UCDOS Chinese font library

In terms of references,there are 7445 character symbles in GB2312- 80.Please refer to document [3] on the relationship between the specific location code and Chinese character code.UCDOS font library is stored according to the dot matrix font realized by standard region-position code.To abstract the inputting of the dot matrix font library,you must first get the internal code of the Chinese character and locate the offset address where the Chinese character in the font library by calculating the corresponding area and position of the Chinese character.

The UCDOS non-graphical characters and Chinese characters are stored by separated,so the first Chinese character ‘啊’ of UCDOS Chinese font library is stored from the zero offset address of the file. The bytes that the Chinese character ‘啊’ occupies is different with the different size of the font library.Taking the 24*24 dot matrix font library for example,every Chinese characte is displayed in the area where there are 24 point pixels in length and breadth and it needs 72 bytes to store only one Chinese character.The internal code of Chinese character ‘啊’ is 0xB0A1,which becomes the starting point that the offset distance is zero in the font library file.Thus,for abstracting the data of a Chinese character in the UCDOS Chinese font library file,you should first calculate the internal code offset distance(offset = Chinese character internal code-B0A1) and then you can locate the data offset position of the Chinese character in the font library file.

For the font whose size is larger than 24*24,such as UCDOS HZK24H,you will see that all the font are accumbent according to the displaying way of HZK16.This is because there is difference between big font library and general Chinese font library.This kind of big typehead Chinese font library is specially for printing and the typeheads are put reversely for the convenience of printing.It just needs to transpose the bit information of the typehead.

The designing of UCDOS font library driver based on QT

The QfontManager under QT can read the configuration file of the fontdir library.Every font library corresponds a font library factory instance object and they are all loaded and managed by QfontManager class.

1) QT Interface Analysis of UC DOS Lattice Font

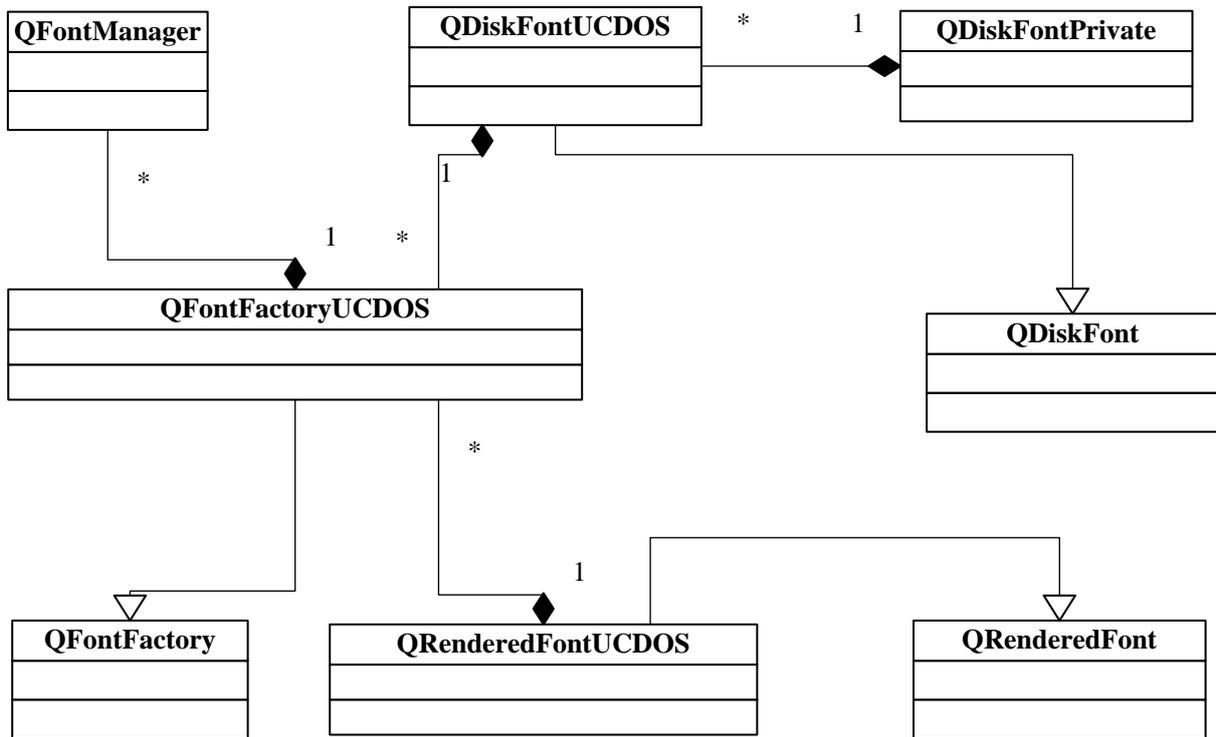


Fig2. Relation diagram of derivative and custom font driving factory class

QFontManager class based on QT is for reading configuration files of fontdir font, each font corresponds to an instant object of font which is loaded and managed by Q Font Manager class. Therefore, the interface instantiation of multi-language lattice font designed in this paper must realize function extension from the factories of this class[5].

As shown in Figure2, the custom font factory QFontFactoryUCDOS derives from QFontFactory base class, and the most important member is QRenderedFontUCDOS which derives from QRenderedFont base class. QFontFactoryUCDOS needs to make cover definition on 3 virtual functions:

1) get function : it is responsible for constructing instant object of QRenderedFontCST and returning the base-class indicator of QRenderedFont.

2) load function: it is in charge of loading font data. The load pattern designed in this paper only needs to load file font to index into buffer pool which can adopt QT template to define QMap less than unsigned short, unsigned int greater than or equal to create an instant object corresponding to a key value (realizing the corresponding of Unicode to offset of data). When some Unicode font is demanded to be searched, a key value needs to be transferred the key value on find member function of an instant object, which can retrieve rapidly the corresponding position of the font data in font file.

3) unload function: it is in charge of unloading font data. The unload patten designed in the paper only needs to empty the key values of QMap and close font file.

Derived class of QRendered Font UC DOS mainly is mainly responsible for implementing the retrieval of font with Unicode and creating QPF lattice bitmap. And the following 3 virtual functions of QRendered Font base class must be covered and inherited:

Unicode function: it is responsible for judging if Unicode code of the font to be retrieved has the corresponding font data in font.

InFont function: it is in charge of judging if Unicode code of the retrieved font is in the current loaded font index range.

Render function: it is responsible for retrieving font, inputting the parameter of Unicode code

and returning QGlyph instant object (carrying lattice bitmap data).

There is a member variable in QGlyph, that is, QGlyphMetrics. And other important member variables including advance, bearingx,bearingy,linestep, width and height correspond to the font data structure designed in the paper. Data member variable of QGlyph stores the font data which has been searched out. Because the font size designed in this paper is less than 256-level gray (referring to chapter 3.3), the parameter bitPerPixel of the font system is needed to quantify and return 8 bit value on each pixel (256-level gray).

Render function is for inquiring font bitmap data according to Unicode code of font when QT font needs to display font.Render Font Data function is the function recovering bitmap data based on gray level.

2) The Design of UC DOS Font Drive Subclass

As shown in Fig1, UC DOS font (drive) factory QFontFactoryUCDOS is derived form the QFontFcatory base class, and its most important members is QRenderedFontUCDOS (derived form the QRenderedFont base class).The QFontFactoryUCDOS Need to define four virtual function.

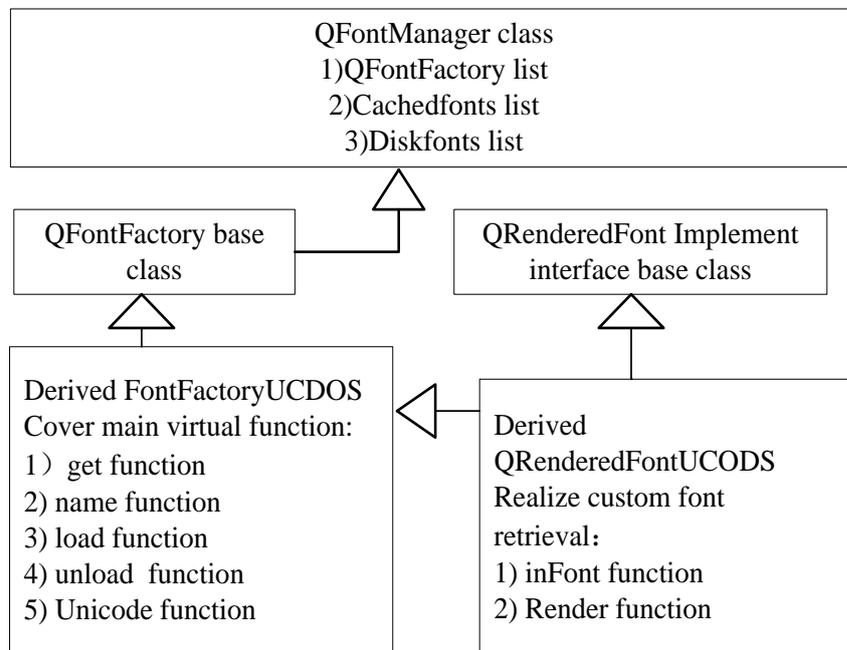


Fig1. UC DOS Font Drive Subclass

- get function: responsible for create QRenderedFontUCDOS instance object, back to QRenderedFont base class pointer.
- name function: responsible for the return of the Font name (Default returns the string "UCDOS").
- load function: responsible for loading the font data, open UC DOS font file, use QMap key to load GB2312 and Unicode code table, for the target class of the class template definition,its key is QMap<short,short > GBToUCMap, achieve UC DOS and QT Unicode compatible.
- unload function: responsible for unloading the font data, close the opened UC DOS font file, empty Unicode code table key value pairs.
- unicode function: Whether there is a corresponding font data to determine the font that you want to retrieve Unicode code point in the font.

The QRenderedFontUCDOS derived class is primarily responsible for the implementation of the font (for Unicode code) retrieval, and generate QPF dot matrix bitmap.We must cover the two the inherited from QRenderedFont base class virtual function is.

- inFont function: responsible for judging whether the retrieve font Unicode code is within the font index scope of the currently loaded.

- render function: responsible for retrieving fonts, the input parameters is Unicode code, return to the QGlyph instances of objects (include lattice bitmap data). Because UCDOS are monospaced font, the the font width FontWidth and the font height FontHeight belongs changeless value. ASCII code font width equal to half the width of the Chinese character fonts.

3) UCDOS Font Retrieval

UCDOS Chinese character belongs to the GB2312 encoding, This is not compatible with QT default font support for Unicode encoding mode. Therefore, before the retrieval, we should firstly do the UCDOS font to Unicode to GB2312 encoding conversion. This article refer to [6], developed a corresponding code table gbMapUnicode which is corresponding with GB2312 and Unicode. We can use the code table to determine whether a Unicode code exists in the font, its corresponding internal code.

4) UCDOS Font and QPF Font Docking

By getUcdosFontData function to retrieve and read UCDOS dot font data, It also need to be docked with the QPF(Converted to a dot matrix bitmap). In the the QT image display, visualization unit (picture and text) will be converted to bitmap mode to facilitate cache and display. When ASCII character g (Chinese character display empathy) is converted to bitmap display, it need five important parameters: advance, bearingx, bearingy, width and height. By those font dot matrix can be realized convert bitmap display.

QRenderedFontUCDOS derived class member function render is to complete the conversion work and back to QGlyph instance of an object, then cached to memory manager memorymanager; QGlyph as a primitive class have an important member: QGlyphMetrics parameter object which include 6 parameter: advance, bearingx, bearingy, linestep (occupied by each line equal to the byte value with the width), width and height. QGlyph data member variables store the font data which have been retrieved. Because UCDOS is each pixel a bit (two gray levels) data, 1-bit must be converted to 8-bit data (256 gray levels).

The configuration and application of UCDOS font

After the design of QFontFactoryUCDOS class factory, we need to add the following code in the Font Manager QFontManager's constructor: factories.append(new QFontFactoryUCDOS());

The new design of the font' use need to be configured QT font configuration files: \$QTDIR/lib/fonts/fontdir. In those files, the \$ DTDIR belongs to environment variable (store the QT application root directory), we only need to add a line of configuration parameters in fontdir file. For example, UcdosFont HZ24S.bin UCDOS n 50 240 u, Specific syntax, see [6], in QT program, we call the QApplication instance object gAPP which has been established, then call gAPP->setFont(QFont("UcdosFont",24,QFont::Normal)); now it can load UCDOS lattice font and support Chinese character display.

Conclusion

After the actual test and UCDOS font antialiasing, the font only 1-bit(2 grayscale) data each pixel, but when it was docked to the QT QPF dot bitmap (QGlyph), the font is clear with out distortion, and UCDOS font significant savings in storage costs. Its retrieval and display performance is better than the TTF vector font, saving the processor resources; the design can be extended to support other dot matrix font and extended multi-language multi-style dot matrix font for embedded QT provides a wealth of font resources, and has broad application prospects.

Acknowledgment

This work was supported the Science and Technology Plan Industrial Application Technology Research and Development Project of Xi'an(Program No.CXY1440④), the Xi'an City Beilin

District Application Technology Research and Development Project (Program No. GX1209), and the Scientific Research Program funded by the Shaanxi Provincial Education Department (Program No.12JK0508; 12JK0535).

References

- [1] Trolltech.Qt 3.3 Whitepaper. <http://www.trolltech.com>. 2005
- [2] Adobe Developer Technologies.Glyph Bitmap Distribution Format (BDF) Specification.
http://partners.adobe.com/public/developer/en/font/5005.BDF_Spec.pdf.1993
- [3] National Institute of Standards Administration. Basic set for the exchange of information with the coded character set of Chinese characters.1981.
- [4] Xu Yu-ming,Jiang Sheng-yi. The analysis of UCDOS curve outline font. Journal of Hengyang Normal University.2000.vol21.No.3.P45~48
- [5] Huang Jian, Zhao Qian;Wu,YanHai. Design of embedded multi-language lattice font based on QT. Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on. P2252-2255
- [6] Trolltech.Qt Reference Documentation (Open Source Edition)[Z].
<http://doc.qt.nokia.com/3.3/index.html>.2005