**开源QR码C函数库qrencode介绍**

来源: 石博文博客 | 浏览: 7868 | 评论: 8发表时间: 2014-02-09 18:37:01

[« QT 中使用 QUdpSocket 进行通信](http://blog.sbw.so/Article/index/title/QT%20%E4%B8%AD%E4%BD%BF%E7%94%A8%20QUdpSocket%20%E8%BF%9B%E8%A1%8C%E9%80%9A%E4%BF%A1.html)

[C++11新特性: lambda表达式的使用介绍 »](http://blog.sbw.so/Article/index/title/C++11%E6%96%B0%E7%89%B9%E6%80%A7:%20lambda%E8%A1%A8%E8%BE%BE%E5%BC%8F%E7%9A%84%E4%BD%BF%E7%94%A8%E4%BB%8B%E7%BB%8D.html)

QR码是当前最流行的二维码之一,它具有可靠性高,识别速度快等特点.而qrencode则是一款由C语言(完全兼容C++)写成的一个QR码生成与解码的函数库.它以GNU LGPL协议发布,至今己有将近八年历史,是当前最常用的QR码识别函数库.

**下载源代码**

Libqrencode是完全开源的函数库,可以到它的[项目主页](http://u.sbw.so/o)下载源程序.下载完成后,使用下面的命令解压:

[?](http://blog.sbw.so/Article/index/title/%E5%BC%80%E6%BA%90QR%E7%A0%81C%E5%87%BD%E6%95%B0%E5%BA%93qrencode%E4%BB%8B%E7%BB%8D.html)

|  |  |
| --- | --- |
| 1  2 | tar -zxvf qrencode-xxx.tar.gz  tar -jxvf qrencode-xxx.tar.bz2 # 如果下载的是 .tar.bz2 后缀的文件 |

**编译安装(Linux)**

解压完成后,得到一个目录,里面就是qrencode的代码了,在Shell中进入这个目录,编译并安装qrencode.

[?](http://blog.sbw.so/Article/index/title/%E5%BC%80%E6%BA%90QR%E7%A0%81C%E5%87%BD%E6%95%B0%E5%BA%93qrencode%E4%BB%8B%E7%BB%8D.html)

|  |  |
| --- | --- |
| 1  2  3 | ./configure  make  sudo make install |

默认情况下,库文件会安装在/usr/local/lib目录下,而头文件会安装在/usr/local/include目录下,除此之外,还会在/usr/local/bin下面安装一个qrencode的执行文件,它是使用libqrencode做成的一个可以生成QR码图片的实例程序,如果不想安装此程序,可以在配置时添加"--without-tools"选项.

在这个目录下,有一个test目录,在这个目录下有许多使用qrencode库的示例代码,在使用这个库函数之前,先看看这些例子会很有帮助的:) 如果想查阅qrencode的相关文档,可以转到[这里](http://u.sbw.so/p).

**注意**

1. qrencode库依赖PNG库或者SDL库.

2. 编译使用了qrencode库的项目时要加上 -lqrencode 选项.

最后,上一张笔者使用Qt + libqrencode做的一个图形界面的QR码生成小程序:



### 【原创】移植二维码QR库qrencode到ARM

2015-03-19 10:40:01|  分类： [QT](http://heyunhuan513.blog.163.com/blog/#m=0&t=1&c=fks_084069084085086070087082087095085080081065084082084) |  标签：[qt](http://heyunhuan513.blog.163.com/blog/#m=0&t=3&c=qt)  [qr](http://heyunhuan513.blog.163.com/blog/#m=0&t=3&c=qr)  [二维码](http://heyunhuan513.blog.163.com/blog/#m=0&t=3&c=二维码)  |举报|字号 订阅





[下载LOFTER](http://www.lofter.com/app?act=qbbkrzydb_20150408_01)[我的照片书  |](http://yxp.163.com/)

网上大部分使用都将其做成库文件，提供库文件下载的地方倒是很多，但是不太方便嵌入式下使用。

昨天闲着无事，准备移植到ARM里。

硬件环境：ARM9+Linux

软件环境：QT

qrencode是开源的二维码QR码编码库，主要C语言编写的，这样方便移植到各种平台下。

生成二维码图片很多人使用开源的图形库libpng来生成png图片，既然QT里面有QPainter，索性不需要移植libpng这个开源库，直接使用QPainter来完成绘图。

1) 将qrencode源码中的(\*.h  \*.c)加入到工程中

2) 将源码中的config.h.in文件修改成config.h并加入工程

3) 在QT的pro文件中添加DEFINES += HAVE\_CONFIG\_H 定义全局宏定义

4) 重新定义 MAJOR\_VERSION、MICRO\_VERSION、MINOR\_VERSION、VERSION

这样修改后，即可编译。

使用方法：

QRcode \*qrcode;

//QR\_ECLEVEL\_Q 容错等级

qrcode = QRcode\_encodeString("http://heyunhuan513.blog.163.com/", 2, QR\_ECLEVEL\_Q, QR\_MODE\_8, 0);

运行结果：



经过测试：可以在Windows、Linux x86、Linux armv4 下运行。

# [我写的一个Qt 显示二维码（ QR Code）的控件](http://blog.csdn.net/liyuanbhu/article/details/44599031)

2015-03-24 21:34 7472人阅读 [评论](http://blog.csdn.net/liyuanbhu/article/details/44599031#comments)(3) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/liyuanbhu/article/details/44599031#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

编程杂项（49） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg 图像处理（27） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg Qt 编程（24） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

最近一个项目需要显示二维码，所以花了点时间（只用了一个晚上，写的很不完善），写了个显示二维码的控件。当然这个控件用到了些开源的代码，比如qrencode，所以我也打算把我的代码开源。

我的代码参考了

<http://stackoverflow.com/questions/21400254/how-to-draw-a-qr-code-with-qt-in-native-c-c>

基本就是按照这里面的思路来写的。

首先要下载 libqrencode，这是一个c 语言的QR code 生成库。QR Code 可以容纳 7000 个数字或者4000个字符，可以承载的信息量很大，用起来也很方便。关于QR Code更详细的信息可以自行 google.

Libqrencode 暂时只支持 QR Code model 2，如果需要ECI 或者FNC1模式的话，还要想别的办法。

编译Libqrencode 我用的是 MSYS2，直接 configure 的话还遇到了点小问题，报的错误如下：

**[plain]** [view plain](http://blog.csdn.net/liyuanbhu/article/details/44599031) [copy](http://blog.csdn.net/liyuanbhu/article/details/44599031)

[在CODE上查看代码片](https://code.csdn.net/snippets/627042)

1. ...
2. checking for pkg-config... no
3. checking for strdup... yes
4. checking for pthread\_mutex\_init in -lpthread... yes
5. checking for png... no
6. configure: error: in `/home/Ivan/qrencode-3.4.4':
7. configure: error: The pkg-config script could not be found or is too old.  Make sure it
8. is in your PATH or set the PKG\_CONFIG environment variable to the full
9. path to pkg-config.
11. Alternatively, you may set the environment variables png\_CFLAGS
12. and png\_LIBS to avoid the need to call pkg-config.
13. See the pkg-config man page for more details.
15. To get pkg-config, see <http://pkg-config.freedesktop.org/>.
16. See `config.log' for more details

大体的意思就是我的编译环境中没有 pkg-config，不过没关系，按照作者的说法，Libqrencode 不依赖于任何第三方的库。在configure 时加一个参数--without-tools ，就可以顺利通过了。

编译之后在 .lib 目录中生成一个 libqrencode.a ，再加上 qrencode.h 这两个文件就够了。我用的Qt 开发环境是 VS2010+Qt4.5.1 。Libqrencode.a 在 VS2010 中也是可以用的，另外还需要libwinpthread.dll.a 这个文件，因为Libqrencode中用到了libpthread 的一些函数。

 补充一下，经过测试，这里生成的  libqrencode.a 在 VS2010 中使用还是有些问题的，表现为 Debug 模式下运行正常，可是一旦将程序编译为 Release 模式就无法运行。看来还需要用 vs2010 编译libqrencode。估计不那么简单，等有时间了折腾一下。

将 config.h 文件中

/\* Define to 1 if using pthread is enabled. \*/  
#define HAVE\_LIBPTHREAD 1

改为：

//#define HAVE\_LIBPTHREAD 1

就可以去掉对 libpthread 的依赖，而且编译出的库文件可以在 vc2010 的release 模式下使用。

我写的控件很简单，具体的看代码吧

**[cpp]** [view plain](http://blog.csdn.net/liyuanbhu/article/details/44599031) [copy](http://blog.csdn.net/liyuanbhu/article/details/44599031)

[在CODE上查看代码片](https://code.csdn.net/snippets/627042)

1. #ifndef QRWIDGET\_H
2. #define QRWIDGET\_H
4. #include <QWidget>
5. #include "qrencode.h"
7. **class** QRWidget : **public** QWidget
8. {
9. Q\_OBJECT
10. **public**:
11. **explicit** QRWidget(QWidget \*parent = 0);
12. ~QRWidget();
13. **void** setString(QString str);
14. **int** getQRWidth() **const**;
15. **bool** saveImage(QString name, **int** size);
16. **private**:
17. **void** draw(QPainter &painter, **int** width, **int** height);
18. QString string;
19. QRcode \*qr;
20. signals:
22. **protected**:
23. **void** paintEvent(QPaintEvent \*);
24. QSize sizeHint() **const**;
25. QSize minimumSizeHint() **const**;
26. **public** slots:
27. };
29. #endif // QRWIDGET\_H

**[cpp]** [view plain](http://blog.csdn.net/liyuanbhu/article/details/44599031) [copy](http://blog.csdn.net/liyuanbhu/article/details/44599031)

[在CODE上查看代码片](https://code.csdn.net/snippets/627042)

1. #include "qrwidget.h"
2. #include <QPainter>
3. #include <QImage>
4. QRWidget::QRWidget(QWidget \*parent) : QWidget(parent)
5. {
6. qr = NULL;
7. setString("Hello QR Code");
8. }
10. QRWidget::~QRWidget()
11. {
12. **if**(qr != NULL)
13. {
14. QRcode\_free(qr);
15. }
16. }
18. **int** QRWidget::getQRWidth() **const**
19. {
20. **if**(qr != NULL)
21. {
22. **return** qr->width;
23. }
24. **else**
25. {
26. **return** 0;
27. }
28. }
30. **void** QRWidget::setString(QString str)
31. {
32. string = str;
33. **if**(qr != NULL)
34. {
35. QRcode\_free(qr);
36. }
37. qr = QRcode\_encodeString(string.toStdString().c\_str(),
38. 1,
39. QR\_ECLEVEL\_L,
40. QR\_MODE\_8,
41. 1);
42. update();
43. }
44. QSize QRWidget::sizeHint()  **const**
45. {
46. QSize s;
47. **if**(qr != NULL)
48. {
49. **int** qr\_width = qr->width > 0 ? qr->width : 1;
50. s = QSize(qr\_width \* 4, qr\_width \* 4);
51. }
52. **else**
53. {
54. s = QSize(50, 50);
55. }
56. **return** s;
57. }
59. QSize QRWidget::minimumSizeHint()  **const**
60. {
61. QSize s;
62. **if**(qr != NULL)
63. {
64. **int** qr\_width = qr->width > 0 ? qr->width : 1;
65. s = QSize(qr\_width, qr\_width);
66. }
67. **else**
68. {
69. s = QSize(50, 50);
70. }
71. **return** s;
72. }
73. **bool** QRWidget::saveImage(QString fileName, **int** size)
74. {
75. **if**(size != 0 && !fileName.isEmpty())
76. {
77. QImage image(size, size, QImage::Format\_Mono);
78. QPainter painter(&image);
79. QColor background(Qt::white);
80. painter.setBrush(background);
81. painter.setPen(Qt::NoPen);
82. painter.drawRect(0, 0, size, size);
83. **if**(qr != NULL)
84. {
85. draw(painter, size, size);
86. }
87. **return** image.save(fileName);
88. }
89. **else**
90. {
91. **return** **false**;
92. }
93. }
95. **void** QRWidget::draw(QPainter &painter, **int** width, **int** height)
96. {
97. QColor foreground(Qt::black);
98. painter.setBrush(foreground);
99. **const** **int** qr\_width = qr->width > 0 ? qr->width : 1;
100. **double** scale\_x = width / qr\_width;
101. **double** scale\_y = height / qr\_width;
102. **for**( **int** y = 0; y < qr\_width; y ++)
103. {
104. **for**(**int** x = 0; x < qr\_width; x++)
105. {
106. unsigned **char** b = qr->data[y \* qr\_width + x];
107. **if**(b & 0x01)
108. {
109. QRectF r(x \* scale\_x, y \* scale\_y, scale\_x, scale\_y);
110. painter.drawRects(&r, 1);
111. }
112. }
113. }
114. }
116. **void** QRWidget::paintEvent(QPaintEvent \*)
117. {
118. QPainter painter(**this**);
119. QColor background(Qt::white);
120. painter.setBrush(background);
121. painter.setPen(Qt::NoPen);
122. painter.drawRect(0, 0, width(), height());
123. **if**(qr != NULL)
124. {
125. draw(painter, width(), height());
126. }
127. }

下面是软件界面：



# [最近用到二维码，vs2010 qrencodelib库编译。](http://blog.csdn.net/u012307430/article/details/49004133)

标签： [二维码](http://www.csdn.net/tag/%e4%ba%8c%e7%bb%b4%e7%a0%81)[qrcode](http://www.csdn.net/tag/qrcode)

2015-10-09 19:23 340人阅读 [评论](http://blog.csdn.net/u012307430/article/details/49004133#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/u012307430/article/details/49004133#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Qt（1） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

创建VC2010工程，静态库类型。

添加qrencode-3.0.3目录下所有的.c、.h文件到工程中，移除qrenc.c文件。

注释掉文件rscode.c中的modnn函数的inline前缀。

编译，生成QrEncode.lib。

创建一个QWidget工程；

关联lib库和连接，头文件；

**[cpp]** [view plain](http://blog.csdn.net/u012307430/article/details/49004133) [copy](http://blog.csdn.net/u012307430/article/details/49004133)

[print?](http://blog.csdn.net/u012307430/article/details/49004133)

1. #include <QtGui/QWidget>
2. #include "ui\_app\_qrencode.h"
3. #include "../../qrencode.h"
5. **class** app\_qrencode : **public** QWidget
6. {
7. Q\_OBJECT
8. **public**:
9. **explicit** app\_qrencode(QWidget \*parent = 0);
10. ~app\_qrencode();
11. **void** setString(QString str);
12. **int** getQRWidth() **const**;
13. **bool** saveImage(QString name, **int** size);
14. **private**:
15. **void** draw(QPainter &painter, **int** width, **int** height);
16. QString string;
17. QRcode \*qr;
18. **public** slots:
19. **void** setImage();
21. **protected**:
22. **void** paintEvent(QPaintEvent \*);
23. QSize sizeHint() **const**;
24. QSize minimumSizeHint() **const**;
25. **private**:
26. Ui::app\_qrencodeClass ui;
27. };

具体实现：

**[cpp]** [view plain](http://blog.csdn.net/u012307430/article/details/49004133) [copy](http://blog.csdn.net/u012307430/article/details/49004133)

[print?](http://blog.csdn.net/u012307430/article/details/49004133)

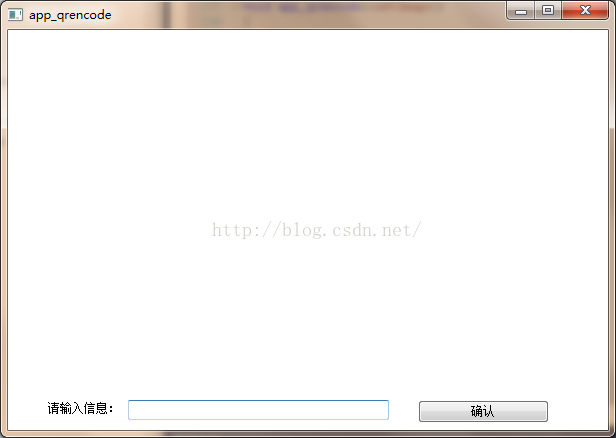
1. #include "app\_qrencode.h"
2. #include <QPainter>
3. #include <QImage>

6. app\_qrencode::app\_qrencode(QWidget \*parent) : QWidget(parent)
7. {
8. ui.setupUi(**this**);
9. qr = NULL;
10. /\*  setString("Hello QR Code");  \*/
11. connect(ui.pushButton,SIGNAL(clicked()),**this**,SLOT(setImage()));
12. }
14. app\_qrencode::~app\_qrencode()
15. {
16. **if**(saveImage("123.png",50))
17. {
18. **return**;
19. }
20. **if**(qr != NULL)
21. {
22. QRcode\_free(qr);
23. }
24. }
26. **int** app\_qrencode::getQRWidth() **const**
27. {
28. **if**(qr != NULL)
29. {
30. **return** qr->width;
31. }
32. **else**
33. {
34. **return** 0;
35. }
36. }
38. **void** app\_qrencode::setString(QString str)
39. {
40. string = str;
41. **if**(qr != NULL)
42. {
43. QRcode\_free(qr);
44. }
45. qr = QRcode\_encodeString(string.toStdString().c\_str(),
46. 1,
47. QR\_ECLEVEL\_L,
48. QR\_MODE\_8,
49. 1);
50. update();
51. }
52. QSize app\_qrencode::sizeHint()  **const**
53. {
54. QSize s;
55. **if**(qr != NULL)
56. {
57. **int** qr\_width = qr->width > 0 ? qr->width : 1;
58. s = QSize(qr\_width \* 4, qr\_width \* 4);
59. }
60. **else**
61. {
62. s = QSize(50, 50);
63. }
64. **return** s;
65. }
67. QSize app\_qrencode::minimumSizeHint()  **const**
68. {
69. QSize s;
70. **if**(qr != NULL)
71. {
72. **int** qr\_width = qr->width > 0 ? qr->width : 1;
73. s = QSize(qr\_width, qr\_width);
74. }
75. **else**
76. {
77. s = QSize(50, 50);
78. }
79. **return** s;
80. }
81. **bool** app\_qrencode::saveImage(QString fileName, **int** size)
82. {
83. **if**(size != 0 && !fileName.isEmpty())
84. {
85. QImage image(size, size, QImage::Format\_Mono);
86. QPainter painter(&image);
87. QColor background(Qt::white);
88. painter.setBrush(background);
89. painter.setPen(Qt::NoPen);
90. painter.drawRect(0, 0, size, size);
91. **if**(qr != NULL)
92. {
93. draw(painter, size, size);
94. }
95. **return** image.save(fileName);
96. }
97. **else**
98. {
99. **return** **false**;
100. }
101. }
103. **void** app\_qrencode::draw(QPainter &painter, **int** width, **int** height)
104. {
105. QColor foreground(Qt::black);
106. painter.setBrush(foreground);
107. **const** **int** qr\_width = qr->width > 0 ? qr->width : 1;
108. **double** scale\_x = width / qr\_width;
109. **double** scale\_y = height / qr\_width;
110. **for**( **int** y = 0; y < qr\_width; y ++)
111. {
112. **for**(**int** x = 0; x < qr\_width; x++)
113. {
114. unsigned **char** b = qr->data[y \* qr\_width + x];
115. **if**(b & 0x01)
116. {
117. QRectF r(x \* scale\_x, y \* scale\_y, scale\_x, scale\_y);
118. painter.drawRects(&r, 1);
119. }
120. }
121. }
122. }
124. **void** app\_qrencode::paintEvent(QPaintEvent \*)
125. {
126. QPainter painter(**this**);
127. QColor background(Qt::white);
128. painter.setBrush(background);
129. painter.setPen(Qt::NoPen);
130. painter.drawRect(0, 0, width(), height());
131. **if**(qr != NULL)
132. {
133. draw(painter, width(), height());
134. }
135. }
137. **void** app\_qrencode::setImage()
138. {
139. QString str = ui.lineEdit->text();
140. ui.label->hide();
141. ui.pushButton->hide();
142. ui.lineEdit->hide();
143. **if** (str.isEmpty())
144. {
145. **return**;
146. }
147. setString(str);
148. }

**[cpp]** [view plain](http://blog.csdn.net/u012307430/article/details/49004133) [copy](http://blog.csdn.net/u012307430/article/details/49004133)

[print?](http://blog.csdn.net/u012307430/article/details/49004133)





输入百度。扫描后就能打开百度网页！www.baidu.com。关闭程序会在目录下生成123.png图片。