Qr Decoder in Qt

This article demonstrates how to create a Barcode Decoder program, using the C++ part of ZXing project to process the image and decode it.

Warning: This project is now been merged in the QZXing library's project page. You are advised to search there for the newest version. The decoding operations are gathered in a library file (QZXing) and has support to QML. (17/10/2011)

QQrDecoder project is developed entirely in Qt SDK + using the CPP part of the ZXing library. It supports the latest version of Qt in Symbian (4.7.x) and the camera is controlled using the Camera APIs from Qt Mobility project. Thus we reasure the compatibility with future releases of Qt and the compatibility with newer device with Symbian^3, Belle and Anna (and S60 5th).

The project's folder contains the code as well as the an installation file in case you want try it out without compiling the project.

Note: This project, since it uses Camera APIs from Qt Mobility breaks the backward compatibility with non touch-screen devices. So this project can't be used at device with S60 3rd Edition.

It currently supports decoding:

- UPC-A and UPC-E
- EAN-8 and EAN-13
- Code 39
- Code 93
- Code 128
- ITF
- Codabar
- RSS-14 (all variants)
- RSS Expanded (most variants)
- QR Code
- Data Matrix
- Aztec ('beta' quality)
- PDF 417 ('alpha' quality)

The three parts of the project

Qr Code containing the text "Qr Decoder in Qt"

- `CameraImageWrapper`:

  The porting layer between the ZXing and Qt. Inherits from `LuminanceSource` (zxing/LuminanceSource.h) which is an abstract class representing a grey-scale image. This class holds the image information in a QImage and 3 methods were needed to be implemented: `getWidth()`, `getHeight()` and `getPixel(int,int)`. The first 2 are trivial, the 3rd needed to return an unsigned char representing the grey-scale value of the pixel given as argument so since QImage will contain RGB image in this project, the return value was given by qGray(QImage::pixel(x,y)). New functions added to comply with the latest version of ZXing libary `getRow(int, unsigned char*)` and `getMatrix()`.

- `QCameraControllerWidget`:

  This widget is responsible for the camera. All the camera operations are taken from QCamera Example. This is the widget where the image from the camera is shown. When the user presses the button to take a picture, that picture is decoded. If and barcode is found, it is reported with a QMessageBox else nothing happens.

- `QQrDecoder`:

  the main window class which contains a *QCameraControllerWidget* instance. **QCameraControllerWidget::imageCaptured(QImage)** signal is connected to **QQrDecoder::decodeImage(QImage)** slot, so each captured image is processed to find the "hidden" message inside the Qr Code.

Prerequisites to compile

- Have Qt SDK installed. This will give you all the tools needed such as the Qt SDK, Qt Mobility APIs and the Qt Creator IDE.

```
http://qt.nokia.com/downloads/
```

Prerequisites to install and run to device

- Have Qt 4.7.3 (and newer) binaries installed to the device.
- Have Qt Mobility 1.1.0 (and newer) binaries installed to the device.

NOTE: those binaries, if Qt SDK is installed, can be found under the path \QtSDK\Symbian\sis\Symbian^X\

Importing the project

For this step, Qt Creator IDE will be used (contained in Qt SDK). The project can be imported/opened with 2 ways.

NOTE: The project folder must be placed under the same drive letter as the installed Qt SDK. For example "C:\QtSDK" and "C:\QtProject\QQrDecoder"

1. Go the the project's folder and double-click the QQrDecoder.pro file. This will result the Qt Creator to open. (This requires that Qt Creator is selected as the predefined program to open such files which is automatically set when installing Qt SDK.)
2. Alternatively, open Qt Creator manually, and go under: "File" -> "Open File or Project...", go to the project's folder and select once again the QQrDecoder.pro.

Whichever step from the above you have followed, now you are led to screen named "Project Setup". Here you select the Target Platform. You can select "Symbian Device" to target directly for a physical device, or "Qt Simulator" to first test it to the build-in simulator.

Building the project

At this point you have imported the project and you are ready to compile it. At the left toolbar in Qt Creator, above the green arrow you can see the current target platform selected. Press to select which you want.

Now by pressing Ctrl+B or the "hammer icon" on the left at the bottom. Normally this should be completed without errors (might be lots of wrning though which we selectively reject them from our mind :P)

This step compiles the project BUT doesn't produces the .sis file (the installation file). To create the installation file you have to press the "Run button". It that "Green arrow" on your left, or by simply pressing Ctrl+R. Regardless if you have a device or not, the .sis file will be created.

Now you can install it to your device.

Version changes

- 1.3 Support of Aztec barcodes. Speed improvement in 1D decoding.
- 1.2.1 Removed automatic detection (unsupported operations from Qt Mobility). Detection happens by pressing a button.
- 1.2 Added zoom functionality. Added automatic detection (no need to press any button).
- 1.1 Camera Plug-in replaced by Camera API from Qt Mobility. New UI. Now the project is completely written on Qt.
- 1.0 Initial release. Was using Camera Plug-in for S60 3rd.

Download

- QZXing library's project page
- File:ZXingBarcodeReader.zip
  This is the same implementation but without Qt. The GUI is implemented with Symbian C++ and again Open C++ is required (for the ZXing library to work).

Licence

- For the QCamera Example see **Nokia_Licence.txt**
- For the ZXing code see **ZXing_Licence.txt**

Useful Links

- ZXing- Multi-format 1D/2D barcode image processing library
- QCamera Example

Qr Decoder for Windows phone 7

Following open source libraries are available for Windows Phone 7 :

- Windows Phone 7 Silverlight ZXing Barcode Scanning Library
- messagingtoolkit

NOTE : please look for any copyright/commercial use policy (if any) in the above links.

favoritas37 18:28, 14 May 2012 (EEST)