

Semesterarbeit im Fach Bildverarbeitung

QR-Code

Eingereicht am: 10. Juli 2020

Josef Prothmann
geboren am 16. Dezember 1998
Email: j.prothmann@stud.hs-wismar.de

Betreuer: Prof. Dr. H. Litschke

Abstrakt

Diese Arbeit beschäftigt sich mit der Generierung von QR Codes, mit einer Android Applikation Implementierung.

Abstract

This thesis is facing the generation of an qr codes, with an android application implementation.

Inhaltsverzeichnis

1	Einleitung	2
2	Grundlagen	3
2.1	QR-Code	3
2.1.1	Generierung	4
2.2	DataMatrix-Code	6
3	Konzept	7
4	Implementierung	8
4.1	Generator	8
4.2	Auslesen	10
5	Schlussfolgerung	i
	Literaturverzeichnis	ii
	Abbildungsverzeichnis	iii
	Selbstständigkeitserklärung	iv

1 Einleitung

QR-Codes spielen aktuell eine große Rolle in der Industrie, Werbewelt und im Datenaustausch. Zwischen dem Text und dem fertigen Code liegen eine Menge an Prozessierungsschritten die einen nicht bewusst sind. In diesem Projekt wurde die Theorie Grundlage zu QR-Codes erläutert und eine einfache Implementierung für ein Android Endgerät entwickelt.

2 Grundlagen

2.1 QR-Code

QR-Code steht für Quick Response - Code, was auf deutsch “schnelle Antwort” bedeutet, ist ein zweidimensionaler Code, der auf Grund seiner automatischen Fehlerkorrektur überaus robust und hierdurch weit verbreitet ist. Entwickelt wurde dieser im Jahr 1994 im Japan von der Firma n Firma Denso Wave Incorporated, welches eine Tochterfirma von Toyota ist. Der Ursprüngliche Zweck war das Markieren von Teilstücken in der industriellen Fertigung, womit Arbeitsschritte automatisiert werden konnten. Dieser Code kann als eine Weiterentwicklung des EAN-Barcodes, welcher 1977 in Europa eingeführt wurde, gesehen werden. Unterschied dieser beiden Methoden ist die Dimensionale Darstellung ein Barcode gilt als eine 1D Darstellung da es gradlinige Informationsdarstellung getrachtet werden kann. Der QR-Code dagegen besitzt Informationen sowohl in der Horizontalen als auch in der Vertikalen Richtung.

Ein QR-Code besteht aus einer quadratischen Matrix aus schwarzen und weißen Quadraten, welche die kodierte Informationen binär darstellen. Spezielle Markierung in drei der vier Ecken des Quadrats geben die Orientierung an. Die Informationen im QR-Code sind mittels einen fehlerkorrigierenden Code ergänzt. Hierdurch wird ein Verlust von bis zu 30 % des Codes toleriert. Für die Fehlerkorrektur sind Verschiedene Level möglich können folgende 4 Level verwendet werden.

- Level L (Low) 7%
- Level M (Medium) 15%
- Level Q (Quartile) 25%
- Level H (High)30%

Im Code enthalten sind die Versionsinformation, das benutzte Datenformat und der Datenteil. Dieser beinhaltet die kodierte Informationen in redundanter Beschaffenheit. Zur Feldbegrenzung beinhaltet der QR-Code in ausschließlich drei seiner Ecken ein bestimmtes Muster. Über das fehlende Muster in der vierten Ecke erkennt das

Lesegerät die Orientierung. Mit zunehmender Größe des Codes werden alternative Muster hinzugegeben, um die Ausrichtung des Codes besser wahrnehmbar zu schaffen. Zwischen den drei Hauptpositionsmarkierungen steckt eine Linie aus einer Folge streng abwechselnder Bits, worüber sich die Matrix definiert. Wichtig ist gleichermaßen der weiße Rand (Quiet Zone) um den Code in dieser Art von sonstigen Fakten abzukoppeln.

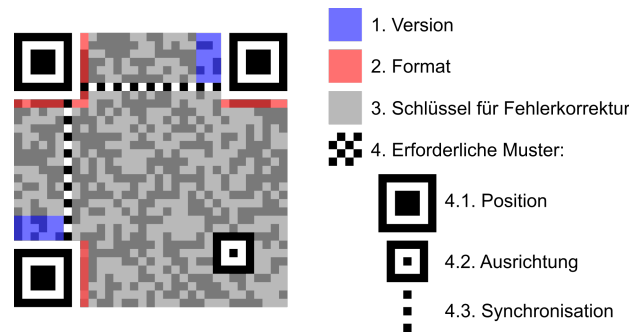


Abbildung 2.1: QR-Code Struktur

Ein QR-Code besitzt die Fähigkeit bis zu 4296 Alphanumerische Zeichen zu speichern. QR-Codes werden mehr im öffentlichen Rahmen benutzt, so werden diese für Kommerzielle Zwecke bei Werbungen, Webseiten, Plakaten und ähnlichem benutzt.

2.1.1 Generierung

1. Anhand der Länge des Textes und dem Grad der Fehlerkorrektur wird bestimmt, wie groß der QR-Code sein muss.
2. Eine weißen Fläche wird erzeugt, auf der nach und nach alle Elemente des QR-Codes dargestellt werden.
3. Die Erkennungsmuster, die nicht von dem Text abhängen, werden zuerst auf die Fläche gebracht. Das sind die Positionsmuster, die Ausrichtungsmuster und die Synchronisationslinien.
4. Aus dem Text wird eine Bitfolge generiert.
5. Zu der Text-Bitfolge wird eine weitere Bitfolge für die Fehlerkorrektur generiert.

6. Die Text-Bitfolge wird zusammen mit der Fehlerkorrektur-Bitfolge dort in das Feld gezeichnet, wo noch Platz ist. Dies geschieht von rechts nach links in Schlangenlinien.
7. Um ungefähr gleich viele schwarze und weiße Pixel auf dem QR-Code zu generieren und um Muster zu vermeiden, werden nacheinander acht verschiedene Masken über das Symbol gelegt. Die Maske, welche das beste Ergebnis liefert, wird beibehalten.
8. Die Kennnummer der verwendeten Maske in das Bild gezeichnet.

2.2 DataMatrix-Code

Der DataMatrix-Code bekannter 2D-Code, welches älter als der QR-Code ist. Der äußere Ring zeigt an wie die Matrix einzulesen ist, zur Orientierung sind zwei schwarze Balken auf der linken und unteren Seite festgelegt. Im Gegensatz zum QR-Code ist eine Fehlerkorrektur von bis zu 30% möglich, es ist nicht möglich weniger als 30% Fehlerkorrektur einzustellen. Für die Fehlerkorrektur existiert in Kombination mit einem Verfahren zur Fehlererkennung und zwei verschiedenen Verfahren zur Fehlerkorrektur. Ein DMC besitzt die Fähigkeit bis zu 2335 Alphanumerische Zeichen zu speichern. Im Gegensatz zum QR-Code wird der DMC in der Industrie verwendet um Komponenten zu markieren.

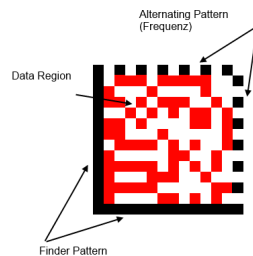


Abbildung 2.2: DMC Struktur

3 Konzept

Das Hauptaugenmerk soll bei der Generierung und dem Auslesen von QR-Codes liegen. Die Applikation soll aus einer Hauptseite bestehen, welche zu den jeweiligen Activity Pages führt. Die Generierung eines QR-Codes lokal in der Activity geschehen. Dazu soll ein Text eingetragbar sein. Das Auslesen und Entschlüsseln der Informationen des QR-Codes, soll mithilfe der Smartphone Kamera ebenfalls direkt auf dem Endgerät stattfinden.

4 Implementierung

Im Rahmen dieses Projektes wurden verschiedene Ansätze betrachtet um eine Android Applikation zu entwickeln. Eine eigene Implementierung des QR-Code Generierungsalgorithmus hätte zu viel Ressourcen gekostet. Deswegen wurde eine vorgefertigte Bibliothek ¹ verwendet.

4.1 Generator

Wie bereits erwähnt wurde für das Generieren in der Applikation eine externe Bibliothek verwendet, diese macht es dem User sehr einfach eigene Codes zu erzeugen. Die Bibliothek ermöglicht das Erstellen von DMC, Barcodes und QR-Codes, sowie das Auslesen dieser Codes. In Abbildung 3.1 ist die Startseite der Applikation zu sehen, von dort aus kann man direkt auf die beiden Funktionen (Scannen und Generieren) zugreifen. In Abbildung 3.2 ist ein bereits generierter Code zu sehen. Um einen neuen Code zu generieren muss in das obige Textfeld ein Text eingegeben werden, nachdem auf den “Generieren”-Button gedrückt wurde wird nach der Generierungsmethode wie in Absatz 2.1.1 beschrieben die Bitmap erstellt und angezeigt.

¹<https://github.com/zxing/zxing>



Abbildung 4.1: Main Activity

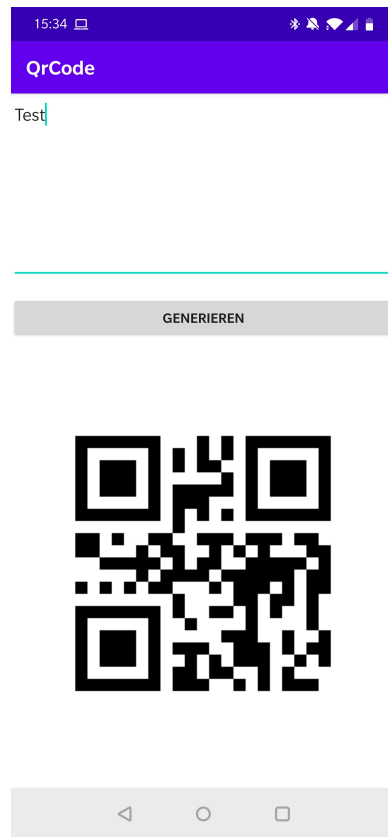


Abbildung 4.2: Generate Activity

4.2 Auslesen

Für das Auslesen wurde die gleiche Bibliothek wie für das generieren verwendet. Wie in Abbildung 3.3 ist zu sehen wie die Scan-Activity aussieht. Um ein QR-Code abzuscannen ist es nötig dieses vor das helleren Kästchen zu halten. Sobald dieses abgescannt wurde, erklingt ein Scanton und ein Popup mit dem Inhalt des Codes wird angezeigt. Handelt es sich um eine Webseite steht kann diese durch einen einfachen Knopfdruck besucht werden. Sind die beinhalteten Daten reine Informationen so werden diese nur dargestellt. Dies soll dem Nutzer die Möglichkeit bieten vor dem weiterleiten, eine Selbsteinschätzung abzugeben ob diese Webseite vertraulich ist.

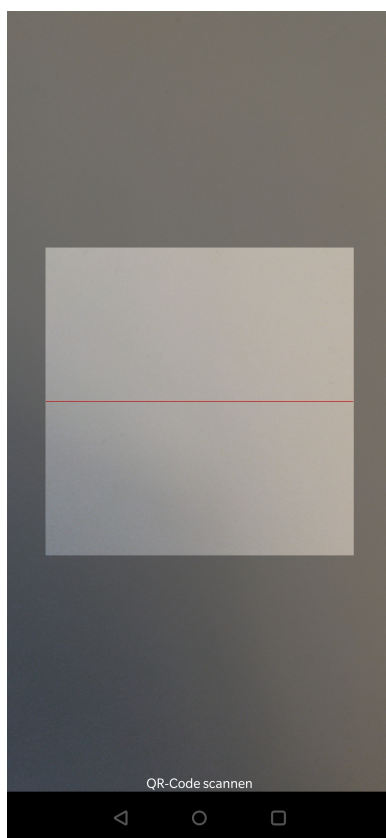


Abbildung 4.3: Scan Activity

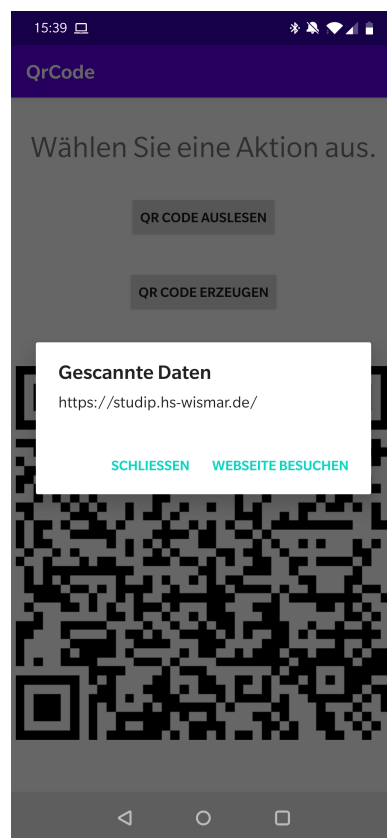


Abbildung 4.4: Scan PopUp

5 Schlussfolgerung

Gegenüber dem DMC ist der QR-Code weitaus mehr bekannt und besitzt eine Etablierte Rolle, dadurch sind QR-Codes im Rahmen der Werbung sehr Wertvoll und bieten dem Ersteller und Verteiler die Möglichkeit mit wenig Aufwand viel Traffic zu erreichen. Durch vorgefertigte Implementierungen ist es sehr einfach für eigene Projekte solche Codes zu generieren.

Während der Ausarbeitung dieses Projektes gab es noch keine Signifikanten Hinweise auf einen Nachfolger des QR-Codes.

Literaturverzeichnis

- [1] TENZER, F.: *Marktanteile von Android und iOS am Smartphone-Absatz in Deutschland bis März 2020.* <https://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-absatz-in-deutschland/>.
Version: Mai 2020

Abbildungsverzeichnis

2.1	QR-Code Struktur	4
2.2	DMC Struktur	6
4.1	Main Activity	9
4.2	Generate Activity	9
4.3	Scan Activity	10
4.4	Scan PopUp	10

Selbstständigkeitserklärung

Hiermit erklären wir, dass wir die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der aufgeführten Hilfsmittel angefertigt haben.

Wismar, den 10. Juli 2020

Ort, Datum

Unterschrift