

Barcode nicht erkannt - Was tun?

**Eine Hilfestellung für QS-Barcode SDK,
QS-DocumentAssembler, QS-Beleg
und die Barcode Testapplikation bcTester**

Zusammengestellt von unserem Support-Team

**QS QualitySoft GmbH
Zum Fürstenmoor 11
21079 Hamburg**

**Tel: +49 (0) 40 790 100 40
Fax: +49 (0) 40 790 100 44
info@qualitysoft.de**

**Neueste Infos im Internet:
www.qualitysoft.de**

© 2008 QS QualitySoft GmbH

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Was ist ein Barcode?

Barcodes dienen zur Codierung von Informationen und bestehen aus einer Folge von Strichen und Lücken. Daher werden Barcodes im Deutschen auch Strichcodes oder Balkencodes genannt. Durch die Variation der Breite der Striche und Lücken können Ziffern und Buchstaben verschlüsselt werden.

Die Erfassung mit Barcodes ermöglicht eine erhebliche Verringerung des Arbeitsaufwandes im Vergleich zum Eintippen der Daten. Zum Beispiel können Artikelnummern von Produkten beim Einkauf an der Kasse als Barcode per Handscanner eingelesen werden, statt die zwölf Ziffern zu tippen.

Im Vergleich zu einer Schrifterkennung (OCR) aus Bilddateien sind die Ortung und das Lesen von Barcodes schneller und fehlersicherer.

Alle herkömmlichen „linearen“ Barcodes beginnen und enden mit einem Strich. Auf einen Strich folgt immer eine Lücke und auf eine Lücke immer ein Strich. Es existieren etliche verschiedene Arten von Barcodetypen, jede von ihnen zeichnet sich durch spezifische Anordnungen und Breiten ihrer Elemente aus.

Lineare Barcodes (Strichcodes)

Typ **Code 2/5**, Inhalt: 6 Zeichen



Typ **EAN 13**, Inhalt: 13 Zeichen



Es gibt Typen, bei denen nur 2 Breiten der Striche und Lücken unterschieden werden und andere, bei denen bis zu 4 unterschiedliche Dicken verwendet werden.

Die Barcodetypen haben spezifische Start- und Endemuster, die zur Typbestimmung und zur Überprüfung der äußeren Vollständigkeit der Striche dienen.

Zur Überprüfung der Dateninhalte wird ein zusätzliches Kontrollzeichen „Prüfziffer“ empfohlen, bei einigen Typen ist sie sogar Pflicht.

Wesentliche Unterschiede gibt es im möglichen Zeichensatz, in einfachen Codes können nur Ziffern verschlüsselt werden, in anderen auch Buchstaben.

Neben dem oben besprochenen Strichcode gibt es immer mehr Einsatzgebiete, in denen die neuen 2D (zweidimensionale) Barcodes eingesetzt werden, bei denen die Elemente flächig angeordnet sind.

2D Barcodes (Flächencodes)

PDF417, Inhalt: 165 Zeichen



Data Matrix,
57 Zeichen



QR Code,
25 Zeichen



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Durch das System der 2D Barcodes ist eine noch stärkere Komprimierung von Inhalten auf noch kleinerem Raum möglich geworden.

Die bekanntesten 2D Barcodes sind der PDF 417, der QR Code, der Aztec Code und der Data Matrix Code. Diese Barcodes können in sehr unterschiedlichen Größen erzeugt werden und über 2.000 Zeichen beinhalten, die auch verschlüsselt sein können. Prominente Einsatzgebiete sind die „STAMPIT“ Marken der Deutschen Post und der Elster Barcode der Steuerverwaltung. Die Fahrkarten der Deutschen Bahn enthalten alle relevanten Daten in einem Aztec Code.

Produkte zur Barcode-Erkennung aus Bilddateien

QS QualitySoft bietet eine Reihe von Produkten zur Barcode-Erkennung, die hier kurz vorgestellt werden. Mehr Informationen zum Einsatz von Barcodes und zur Lesung der Barcodes aus Bilddateien finden Sie auf unserer Website <http://www.qualitysoft.de>

Für die Analyse der Barcodelesung sollten Sie unsere **kostenlose Barcode-Testapplikation** verwenden, die Sie bei http://www.bcTester.de/download/bctester_de.zip finden.

bcTester ist ein anwenderfreundliches Windows-Dialogprogramm. bcTester enthält Beispieldateien mit unterschiedlichen Barcodes und eine ausführliche Hilfefunktion mit zahlreichen Informationen über Barcodes. Mit bcTester können Sie eine Bilddatei öffnen und die Barcode-Erkennung testen. Sie können bei Bedarf die Einstellungen für die Erkennung anpassen oder die automatische Analyse verwenden.

Barcodes, die mit bcTester gelesen werden, sind auch mit den anderen Produkten von QualitySoft lesbar.

Wir bieten mit dem **QS-DocumentAssembler** ein Programm, mit dem automatisiert Barcodes massenhaft aus Bilddateien gelesen werden (Stapelmodus). Dabei können die Dateien indiziert oder aufgrund der Barcodeinformationen strukturiert werden.

Mit dem **QS-Barcode SDK** (Software Development Kit) können Programmierer die Funktionen zur Barcodelesung leicht in ihre Programme integrieren.

Auf der <http://www.qualitysoft.de> Website finden Sie **kostenlose Evaluationsversionen** der Produkte „QS-DocumentAssembler“ und „QS-Barcode SDK“ zum intensiven Test.

Hinweis:

Wenn Ihr Barcode mit bcTester richtig erkannt wird, aber in einer QS-Barcode SDK Anwendung bzw. dem QS-DocumentAssembler **aus der „3“ eine „1“ liest**, so ist der **Demomodus** aktiviert. Dies geschieht automatisch, wenn keine Lizenzdatei „qsbcl.lic“ gefunden wird.

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Barcodes nicht erkannt - was tun?

Immer wieder treten Fälle auf, in denen Barcodes nicht auf Anhieb mit einem unserer Produkte erkannt werden. Dies bedeutet keineswegs, dass das Programm unfähig ist, den betreffenden Barcode zu erkennen. Oft sind vielmehr leicht zu behebbende Fehler der Grund für mangelnden Erfolg beim Erfassen.

Die zu lesenden Barcodes müssen in einer Bilddatei oder einem Adobe PDF-Dokument vorliegen. Wenn Sie Probleme haben, die Bilddatei in der Testapplikation zu laden, müssen Sie die Bilder evtl. in andere Formate konvertieren.

Generell lassen sich die Ursachen für eine fehlgeschlagene Barcodeerfassung in drei große Gruppen einteilen:

- **1. Fehler beim Druck des Barcodes**
- **2. „Unglückliche“ Scannereinstellungen**
- **3. Unpassende Parameter für die Erkennung**

In den folgenden Kapiteln geben wir Hinweise zu den häufigsten Fehlern und Tipps zur Vermeidung. Außerdem werden die Möglichkeiten der **Bildverbesserung** vor der Barcodeerkennung diskutiert.

Führt dies nicht zum Erfolg, bleibt Ihnen immer noch die Möglichkeit, mit QualitySoft in Kontakt zu treten.

Analysen von Barcodes führen wir kostenlos durch!

Senden Sie einfach die Bilder mit einer kurzen Fehlerbeschreibung an

support@qualitysoft.de

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

1. Fehlerursache: Fehler beim Druck des Barcodes

In der Praxis kommen immer wieder Fehler beim Erzeugen und Druck der Barcodes vor, die eine spätere Lesung verhindern. Offensichtliche Fehler sind:

- unterschiedliche Dicken von Strichen und Lücken, die gleich sein sollten (durch zu geringe Auflösung, schlechte Barcode-Fonts oder Fehler im Druckertreiber)
- Weglassen der Ruhezone rechts und links vom Barcode
- Abschneiden von einzelnen Strichen an den Rändern der Barcodes

Die letzten beiden Fehler treten vor allem beim Etikettendruck auf. Beim Etikettendruck wird häufig an den Rand gedruckt, so dass die Ruhezonen rechts und links vom Barcode fehlen oder sogar einzelne Striche ganz fehlen. Dies kann auch bei einigen Druckprogrammen passieren, wenn das Feld für den Barcode zu klein ist (Abschneiden).

Wenn der Barcode auf dem gescannten Bild gut aussieht, aber kein Barcode erkannt wird, sind häufig die **Start- und Endezeichen** beim Druck vergessen worden. Bei vielen Barcode-typen kann man durch Zählen der Striche leicht prüfen, ob alle notwendigen Striche vorhanden sind.

Der prinzipielle Aufbau sei anhand des Typs „Code39“ erläutert. Bei diesem Code wird jedes Zeichen durch 9 Elemente (5 Striche und 4 Lücken) dargestellt, von denen 3 dick sind.

Zeichentabelle Code 39 (es werden nur die Ziffern dargestellt, der Code 39 besitzt außerdem Codierungen für A-Z und die Sonderzeichen .\$/+%).

0		4		8	
1		5		9	
2		6		0	
3		7		*	

Das * (Stern-) Zeichen dient als Start- und Endezeichen. Die Zahl 1234 wird also richtig als *1234* codiert und hat dann 6x5=30 Striche. Zur Verdeutlichung sind im folgenden Bild die Zeichen einzeln dargestellt und die Start- und Endezeichen in Rot.



Ohne Lücken sieht der Code dann so aus:



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Die Start-Ende-Zeichen einiger wichtigen Barcodetypen:

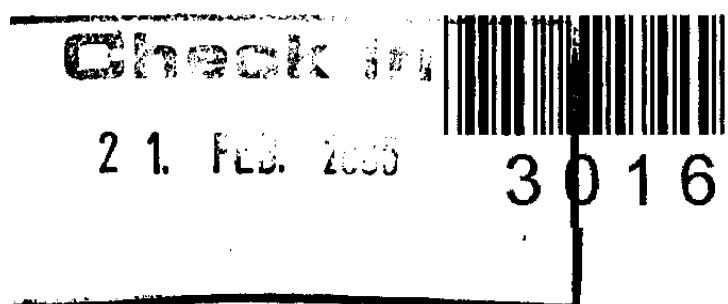


Die Start-Ende-Zeichen sind Pflicht! Sie sind typspezifisch und helfen bei der automatischen Bestimmung des Typs bei der Barcodelesung. Die Berechnung der Strichanzahl eines Barcodes ist abhängig vom Typ und den Zeichen im Barcode. Die Formel ist nicht immer so einfach wie beim Code 39, weil es einige Typen mit Komprimierungen, Prüfziffern und Steuer- bzw. Umschaltzeichen gibt.

Wenn Striche der Barcodes fehlen, kann der gesamte Barcode nicht erkannt werden. Dieser Fehler geschieht auch, wenn Barcodes nach der Erzeugung nicht in die vorgesehenen Ausgabefelder passen und beim Drucken abgeschnitten werden.

Übrigens sollte auf den Barcode nicht geschrieben oder gestempelt werden, auch das kann die Lesung verhindern.

In diesem Muster zerstört der Eingangsstempel den Barcode.



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

2. Fehlerursache: „Unglückliche“ Scannereinstellungen

Häufig beruhen Leseschwierigkeiten von Barcodes auf Fehlern beim Scannen der Barcodes. Diese Fehler sind häufig auf dem eingescannten Bild erkennbar.

Mögliche Fehlerursachen sind:

- eine zu geringe Auflösung beim Scannen
- sowie eine zu dunkle oder zu helle Einstellung beim Scannen.

Im Folgenden diskutieren wir anhand von zwei Barcode-Beispielen die typischen Fehler beim Einscannen. Zuerst der **Idealfall**:



Dieser Code hat nur 2 Dicken, es gilt:

Breite der Striche:

- alle schmalen Striche sind gleich breit
- alle breiten Striche sind gleich breit
- die breiten Striche sind dreimal so breit wie die schmalen

Breite der Lücken:

- alle schmalen Lücken sind gleich breit
- alle breiten Lücken sind gleich breit
- die breiten Lücken sind dreimal so breit wie die schmalen

Verhältnis Striche/Lücken:

- die schmalen Striche haben die gleiche Breite wie die schmalen Lücken
- die breiten Striche haben die gleiche Breite wie die breiten Lücken



Dieser Code hat 4 Dicken, es gilt:

Breite der Striche:

- Es sind vier verschiedene Breiten der Striche möglich. Sie können zweimal, dreimal und viermal so breit sein wie der schmalste mögliche Strich.

Breite der Lücken:

- Es sind vier verschiedene Breiten der Lücken möglich. Sie können zweimal, dreimal und viermal so breit sein wie die schmalste mögliche Lücke.

Verhältnis Striche/Lücken:

Die Breiten der Striche entsprechen den Breiten der Lücken.

Gescannte Barcodes haben im Vergleich zum gedruckten Original immer Fehler, z.B. Lücken und Punkte am Rand, Treppenkurven, etc. Bei einem guten Bild treten diese Fehler nur selten auf, die Striche und Lücken sind fast überall in der Sollbreite vorhanden.

Recht guter Scan:



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Fehlerursache: zu geringe Auflösung beim Scan

Bei zu geringer Auflösung können feine Linien nicht getrennt werden. Benachbarte Linien laufen zusammen, die Lücken verschwinden. Dadurch kann das Programm die Lücke nicht als solche erkennen und die Ziffer kann nicht entschlüsselt werden.

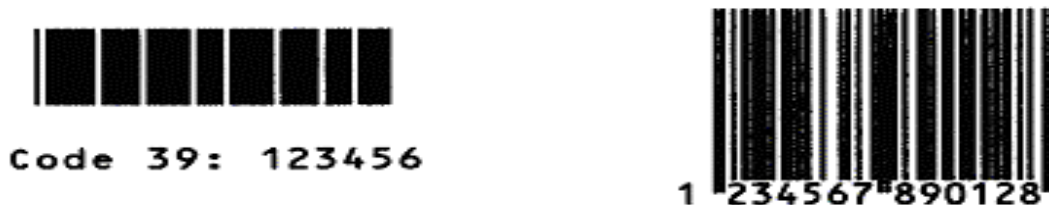
Beispiele mit einer zu geringen Scanauflösung von 50 dpi:



Fehlerursache: Scaneinstellung zu dunkel

Wird der Barcode zu dunkel gescannt, verbreitern sich die Linien auf Kosten der Lücken, dünne Lücken verschwinden, der Barcode ist nicht lesbar. Eine mögliche Fehlerabhilfe ist es, die Scaneinstellung heller zu wählen. Wenn nicht heller gescannt werden kann, weil auf dem Bild andere helle Elemente verschwinden würden, muss der Barcode deutlich breiter gedruckt werden.

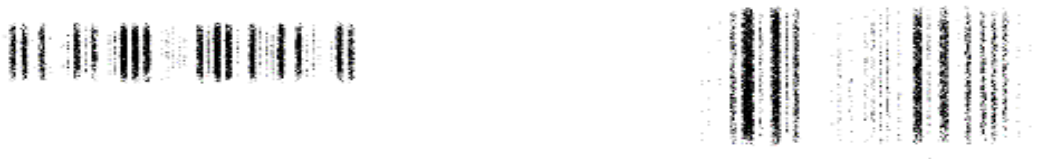
Beispiele wurden mit einer viel zu dunklen Scaneinstellung gescannt:



Fehlerursache: Scaneinstellung zu hell

Bei zu hellem Scan fehlen für die Erkennung Striche.

Beispiele wurden mit einer viel zu hellen Scaneinstellung gescannt:



Barcodes nicht erkannt – Was tun?

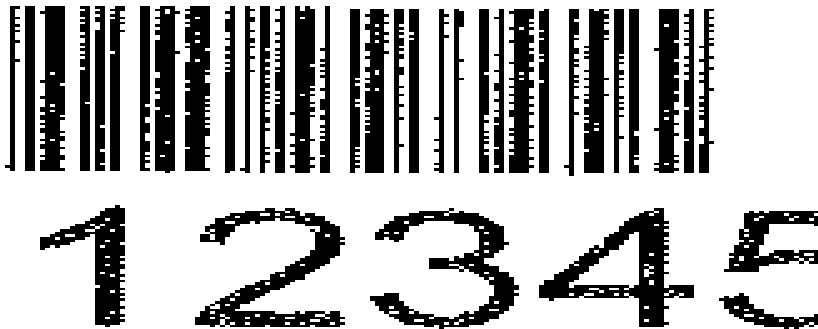
Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Hier noch einige Bilder aus der Praxis, bei denen aufgrund von Scannereinstellungen unlesbare Bilder erzeugt wurden, zum Teil sind diese Fehler nur mit der "Lupe" erkennbar (bei Vergrößerung des Bildes).



Konturen oder Konturenverstärkung

Diese Scannereinstellung kann dazu führen, dass bei breiten Strichen nur noch der Rand dargestellt wird. In der Mitte sind diese dicken Striche dann weiß, der Barcode ist nicht lesbar.



Dithering (dt.=Rasterung)

Bei der Rasterung werden Grautöne durch eine entsprechende Verteilung von schwarzen und weißen Punkte erzeugt.

Wenn der Barcode nicht richtig schwarz gedruckt ist, werden beim Scannen weiße Punkte erzeugt, die die Erkennung erschweren. Die Punkte sind auch im Text sichtbar, wenn er die gleiche Helligkeit hat.



starke jpg Komprimierung

Die jpg Bildkomprimierung erzeugt sehr kleine Bilder. Allerdings arbeitet das Verfahren bei hoher Komprimierung mit "Verlusten" in der Bildqualität. Die Barcodestriche werden verschmiert und die Breiten können bei der Erkennung nicht ermittelt werden.

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

3. Fehlerursache: Unpassende Erkennungs-Parameter

Zur Lesung des Barcodes kann es notwendig sein, die als Standard voreingestellten Parameter zu verändern, um alle codierten Informationen lesen zu können. Dieses Kapitel gibt Hinweise darauf, welche Veränderungen an den Parametern hilfreich sein können.

Vorgehen

Testen Sie zunächst mit den Standard-Einstellungen von **bcTester**, beginnen Sie danach, unter „Einstellungen“ einzelne Parameter zu verändern. Im Folgenden werden einige dieser Parameter dazu detailliert erläutert.

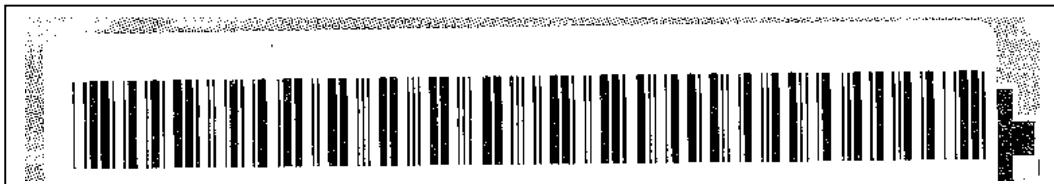
Ruhezone (engl. LightMargin)

Vor und hinter einem Barcode wird ein weißer Bereich als Ruhezone erwartet. Dieser weiße Bereich muss breiter sein als jede weiße Lücke innerhalb des Barcodes und breiter als jeder schwarze Strich in dem Barcode. Die Abbildung rechts verdeutlicht dies noch einmal.



Die Einstellungen zur Ruhezone finden Sie im bcTester im Dialog „Einstellungen“ unter <Erweiterte Parameter>. Die Angabe erfolgt in Pixel (Bildpunkte). Der Standard-Wert von 18 Pixeln bezieht sich auf gängige 200dpi Bilder. Sie sollten mit einem höheren Wert testen, wenn Ihre Scans eine hohe Auflösung haben.

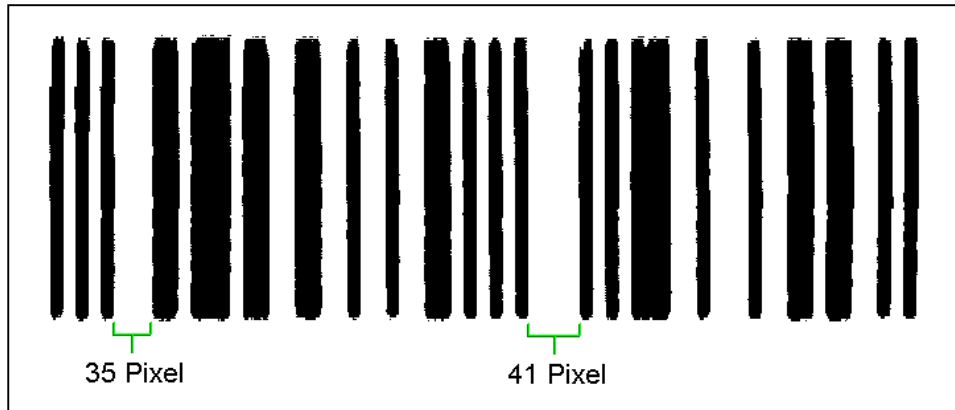
Das folgende Beispiel zeigt ein Barcode-Etikett, auf dem der Barcode zu nah an den Rand gedruckt wurde. Die weiße Ruhezone hinter dem Barcode ist zu kurz, so dass sie nicht als Ende des Barcodes erkannt wird und der Strich auf dem Papier neben dem Etikett mit zum Barcode gerechnet wird. Dadurch wird eine korrekte Erkennung verhindert. In diesem Fall ist auch die Reduzierung der Ruhezone um einige Pixel nicht hilfreich, weil dann weiße Lücken innerhalb des Barcodes als Ruhezone erkannt werden würden.



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Der folgende Ausschnitt wurde mit 600dpi gescannt. Die hohe Auflösung führt dazu, dass innerhalb des Barcodes weiße Lücken auftreten, die deutlich über dem Standardwert 18 Pixel liegen. Erhöht man den Ruhezone-Wert auf 45 Pixel, wird der Barcode problemlos erkannt.

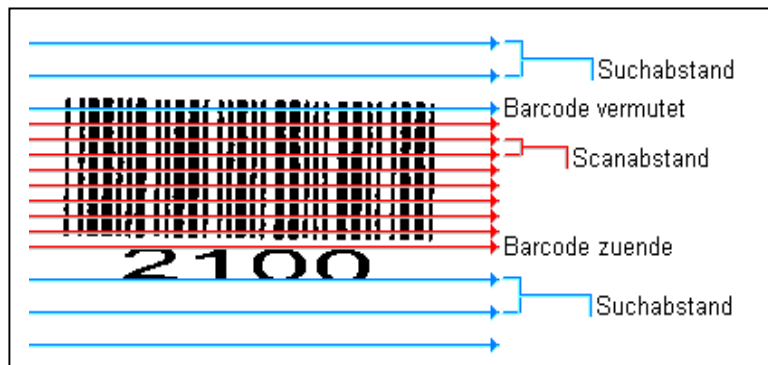


Suchabstand und Scanabstand

(engl. ScanDistance = Suchabstand, ScanDistanceBarcode = Scanabstand)

Der Algorithmus zur Barcodesuche tastet sich zeilenweise durch das Bild. Häufig ist es nicht nötig, jede Zeile zu analysieren. Das spart Zeit.

Die Standardeinstellung sieht einen „Suchabstand“ von 15 Zeilen vor, wird in einer Zeile ein Barcode vermutet, schaltet der Algorithmus den Tastabstand auf den „Scanabstand“ (Standardwert: 3 Zeilen) herunter, um innerhalb von Barcodes mehr Informationen zu sammeln.



Der „Scanabstand“ sollte immer kleiner oder gleich dem „Suchabstand“ gewählt werden. Beide Parameter sind unter „Einstellungen“ unter <Erweiterte Parameter> zu finden.

Optimale Resultate im Hinblick auf die Erkennung erzielt man, wenn man beide Parameter auf 1 setzt, dann wird *jede* Zeile im Bild berücksichtigt. Dies bremst natürlich die Erkennungsgeschwindigkeit. Zum Test ist ein Vorgehen sinnvoll, bei dem Sie zunächst beide Werte auf 1 setzen, wenn die Erkennung dann klappt, können Sie die Werte dann wieder schrittweise erhöhen.

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Barcodeanzahl: BC_ONE_BREAK

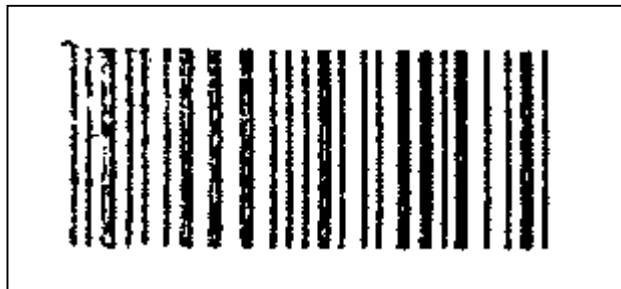
(engl. ReadMulti)

In Sonderfällen, vor allem wenn Barcodes sehr schlecht gedruckt oder gescannt sind, können Sie eine Erkennung mit dem speziellen Parameter „BC_ONE_BREAK“ versuchen.

Wie immer wird bei dieser Einstellung das Bild zeilenweise abgetastet. Sobald in einer Zeile ein Barcode erkannt wird, wird anders als sonst die Erkennung sofort abgebrochen und der Barcode als Ergebnis gemeldet.

Die üblichen Prüfungen zur Qualität und Sicherung, ob das gefundene Strichmuster wirklich ein Barcode ist, z.B. ob die Mindesthöhe erfüllt ist, finden in diesem Modus nicht statt.

Weitere Vor- und Nachteile dieser Einstellung können Sie der folgenden Tabelle entnehmen.



Vorteile

Die Einstellung „BC_ONE_BREAK“ ermöglicht die Erkennung von stark beschädigten Barcodes noch bei nur einer lesbaren Abtastzeile

Die Einstellung „BC_ONE_BREAK“ ermöglicht eine sehr schnelle Erkennung, weil früh abgebrochen wird

Nachteile

Es wird immer nur höchstens ein Barcode gefunden

Es besteht die Gefahr, dass Schrift oder Grauschattierungen fälschlicherweise als Barcode interpretiert werden und ein falsches Ergebnis geliefert wird!
Bei anderen Einstellungen erfolgt zur Absicherung gegen diesen Fehler eine Analyse mehrerer Abtastzeilen.

In vielen Anwendungsfällen wird es vorgezogen, dass kein Barcode erkannt wird, statt dass ein fehlerhafter oder gar nicht existenter Barcode gemeldet wird. In solchen Fällen sollte man auf jeden Fall auf die Einstellung „BC_ONE_BREAK“ verzichten.

Die Angabe einer exakten Länge, Orientierung und Typangabe der Barcodes reduziert die Fehlergefahr.

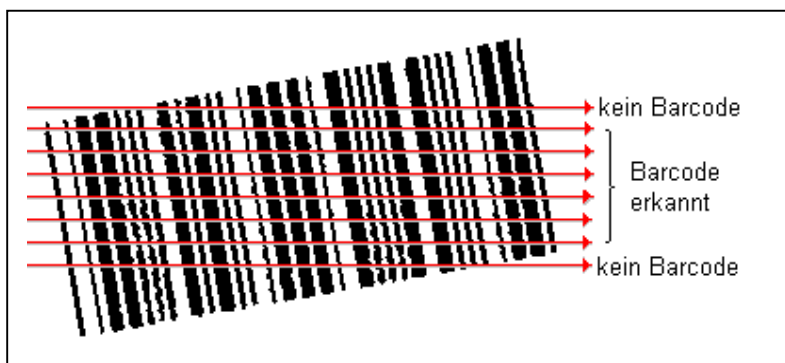
Der Beispiel-Barcode in der Darstellung ist von schlechter Qualität, gerade die ersten Striche sind größtenteils beschädigt. Bei dieser Qualität sollte ein Barcode verwendet werden, der eine Prüfziffer enthält, damit Falscherkennungen bemerkt werden!

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

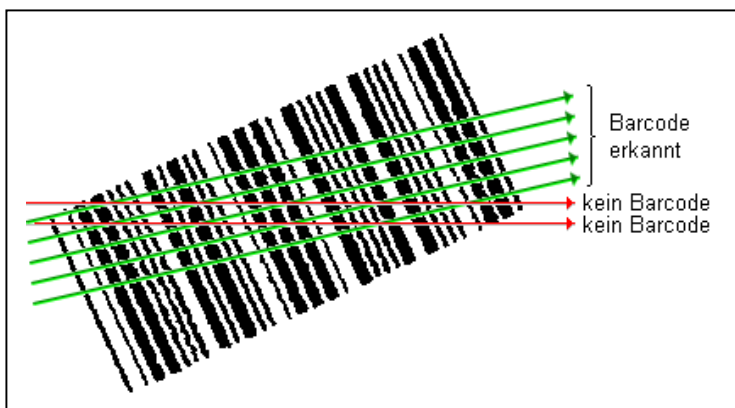
Verdrehungen

Standardmäßig werden Barcodes gesucht, die mit 0° / 180° und 90° / 270° auf dem Bild aufgebracht sind. Leichte Verdrehungen von bis zu 8° , wie sie z.B. beim Einscannen entstehen, werden hierbei toleriert. Für den Fall, dass Barcodes sehr viel stärker verdreht auftreten, sollte der Parameter „maximale Verdrehung“ auf 21° , 34° oder 46° gestellt werden. Bei der Einstellung 46° werden alle Richtungen durchsucht.



Der links dargestellte Barcode wird bei eingestellter Orientierung 0° erkannt, da trotz der Verdrehungen genug Linien vollständig erkannt werden.

Der rechts dargestellte Barcode wird bei eingestellter Orientierung 0° nicht erkannt (**rote Pfeile**), der Schalter „maximale Verdrehung“ von 21° erlaubt aber eine Erkennung, weil die Bilddatei **zusätzlich** noch schräg abgetastet wird (**grüne Pfeile**).



Bildverbesserung vor der Lesung

Übersicht

Grundsätzlich ist es immer besser, durch Veränderung der Bilderzeugung eine Bildqualität zu erreichen, bei der eine Barcodeerkennung ohne weitere Maßnahmen gewährleistet ist. Manchmal ist dies jedoch nicht möglich, z.B. weil die Bildgewinnung nicht beeinflusst werden kann. Im Folgenden werden einige Möglichkeiten aufgezeigt, wie die Bildqualität nachträglich verbessert werden kann, so dass in vielen Fällen eine Erkennung doch noch möglich ist. Meistens senken diese Maßnahmen die Erkennungsgeschwindigkeit deutlich.

Es werden Funktionen für folgende Fehlerfälle zur Verfügung gestellt:

- durch Punkte verschmutzte Bilder
- Löcher in den Strichen der Barcodes
- weitere Bildfehler, besonders Fehler in den Strichen und Lücken
- extreme Helligkeitswerte für Farb- oder Graustufenbilder
- ungleichmäßige Beleuchtung bei Farb- oder Graustufenbildern

Die Bildverbesserungsmethoden werden über Befehle gesteuert, deren Format weiter unten erläutert wird. Beim **QS-DocumentAssembler** und beim **bcTester** werden diese Befehle aus einer Datei namens `QSBC.ini` im Programmverzeichnis ausgelesen, falls diese vorhanden ist. Auch Anwendungen, die die **QS-Barcode** SDK Schnittstellen `F_DLL`, `H_DLL` oder `F_OCX` nutzen, verarbeiten die `QSBC.ini` Einträge zur Bildverbesserung.

Aufbau der Steuerdatei `qsbc.ini`

Die Barcode-Erkennung prüft also jeweils im Anwendungsverzeichnis, ob eine Datei mit dem Namen `QSBC.ini` vorhanden ist. Die Datei hat das übliche Format der Windows-Parameterdateien und kann mit einem Texteditor erstellt werden.

Beispielsweise könnte eine `QSBC.ini` für einen Test wie folgt aussehen:

```
[Imaging]
Steps=5
Step1=GraphLib:SubImage("c:\_test1.tif");
Step2=FreeImage:Threshold,160;
Step3=GraphLib:SubImage("c:\_test3.tif");
Step4=GraphLib:DespeckEx(10,2);
Step5=GraphLib:SubImage("c:\_test5.tif");
```

Mit diesen Befehlen sollen Grau- oder Farbbilder binarisiert werden (Befehl `Threshold`) und dann störende schwarze und weiße Schmutzpunkte entfernt werden (Befehl `DespeckEx`). Die Befehle werden in den folgenden Abschnitten erläutert.

Der Eintrag `Steps=5` gibt an, dass 5 Schritte durchgeführt werden. Für jeden Schritt muss dann ein Eintrag der Form `StepX=Befehl` vorhanden sein, wobei X für die Schrittnummer steht, und Befehl einer der möglichen und weiter unten beschriebenen Befehle ist. Wichtig ist das abschließende Semikolon in den Befehlszeilen.

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Die Bildverbesserung wird nur auf einer Kopie des zu verarbeitenden Bildes durchgeführt, die Eingabedatei wird dabei nicht verändert. Alle angegebenen Schritte werden unabhängig von der Barcodeerkennung durchgeführt.

Zum Testen der Bildverbesserung dienen die Befehle:

```
Step1=GraphLib:SubImage("c:\_test1.tif");  
Step3=GraphLib:SubImage("c:\_test3.tif");  
Step5=GraphLib:SubImage("c:\_test5.tif");
```

Hierdurch wird der aktuelle Zustand des bearbeiteten Bildes jeweils unter dem angegebenen Dateinamen gespeichert. So kann man anhand der 3 „_test“-Bilder beurteilen, wie das verbesserte Bild aussieht. Das Dateiformat wird durch die Dateinamenendung bestimmt, möglich sind hier „tif“ oder „bmp“.

Schmutzpunkte entfernen

Bei vielen Vorlagen gibt es nach der Bilderzeugung und der Binarisierung (dem Wandeln der Bilder in schwarzweiß) schwarze und weiße Schmutzpunkte.

Das Eingabebild



weist viele schwarze Punkte zwischen den Barcodelinien auf. Dadurch ergeben sich bei der zeilenweisen Abtastung zusätzliche Wechsel, die eine Erkennung des Barcodes verhindern.

Durch den Befehl Despeck (englisch „speck“ = Fleck, Schmutzpunkt)

```
GraphLib:DespeckEx(10,10);
```

können sowohl schwarze Punkte zwischen als auch weiße Löcher in den Barcodelinien entfernt werden. Die beiden Zahlen geben den maximalen Durchmesser der zu entfernenden Punkte in Pixel (Bildpunkte) an, die erste für schwarze Punkte, die zweite für weiße. Es werden nur „alleinstehende“ Punkte in der angegebenen Größe entfernt, nicht welche, die Kontakt mit anderen haben.

Das Ergebnis dieser Bildverbesserung sieht dann so aus und kann korrekt erkannt werden:



Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Anmerkung: Punkte entfernen funktioniert nur auf monochromen (schwarzweiß) Bildern. Farbige Bilder und Graustufenbilder müssen daher erst umgewandelt werden, eine mögliche QSBC.ini sieht z.B. so aus:

```
[Imaging]
Steps=3
Step1=FreeImage:GrayScale;
Step2=GraphLib:Dynamic;
Step3=GraphLib:DespeckEx(10,10);
```

Näheres zu den Befehlen zum Umwandeln von farbigen Bildern und Graustufenbildern in monochrome Bilder wie `Dynamic` findet sich weiter unten.

Bildfehler-Ausgleich

Eine Alternative zum „Punkte entfernen“ ist die Option „Bildfehler-Ausgleich“, die etwas anders arbeitet und daher auch andere Vor- und Nachteile bietet:

Vorteile:

- korrigiert Bildfehler und Schmutz im Barcode, sowohl in den weißen Lücken wie auch in schwarzen Balken.
- arbeitet viel schneller als „Punkte entfernen“, der Zeitaufwand dieser Korrekturfunktion geht gegen null!
- Kann auch ganze senkrechte Linien, wie sie durch fehlerhafte Scanner- oder Fax-Optik oder Schmutz auf der Optik entstehen, ausblenden.
- Funktioniert auch bei Verwendung der QS-Barcode SDK Schnittstelle P_LIB

Nachteile:

- Muss abhängig von den eingesetzten Barcodes und der Auflösung der Bilddateien konfiguriert werden
- Fehlerhafte Einstellungen können Barcode-Erkennung komplett verhindern

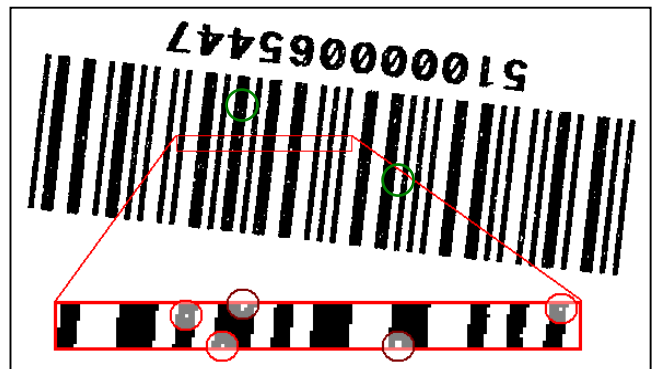
Wie funktioniert es?

Beim Abtasten der Bildzeilen werden ganz **kurze schwarze oder weiße Stellen** überlesen. In Laufrichtung der Abtastung werden alle kurzen Stellen überlesen, die eine konfigurierbare Breite an Pixeln unterschreitet.

In der Abbildung rechts bewirken die Einstellungen

```
[CleanBresenham]
WhitePoints=2
BlackPoints=0
```

dass alle umkreisten Punkte entfernt werden. Bei `WhitePoints=1` würden die zwei Punkte in den grünen Kreisen nicht entfernt, weil sie zu breit sind.



Die Parameter für die Option „Bildfehler-Ausgleich“ befinden sich im Abschnitt `[CleanBresenham]` der `qsbc.ini`-Datei, es können entweder weiße Stellen mit `WhitePoints=N` oder schwarze Stellen mit `BlackPoints=M` oder auch beides in

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Kombination überlesen werden. N bzw. M gibt dabei die maximal zu überlesene Anzahl der Pixel an.

Achtung! N und M müssen immer so gewählt werden, dass reguläre Lücken oder Striche im Barcode nicht auch überlesen werden, diese müssen also immer deutlich breiter als die gewählten Werte sein.

Ab und zu treten in den Barcodebildern sehr feine Linien auf („Haarlinienfehler“), die die Lesung verhindern. Häufig sind diese Fehler erst bei einer Vergrößerung des Bildes sichtbar. In der Regel sind die Ursachen Defekte beim Druck oder im Scanner. In diesen Fällen hilft der oben beschriebenen „Bildfehler-Ausgleich“.

Die Fehler werden damit korrigiert, wie die folgenden Abbildungen verdeutlichen:



Bildfehler: Senkrechter weißer Strich im Bild



Senkrechter Strich rot hervorgehoben



Beim Abtasten der Bildzeilen mit der Einstellung „Bildfehler Ausgleich“ wird der dünne weiße Strich „überlesen“ (rote Markierung)

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Zu helle oder zu dunkle Barcodes - unterschiedliche Methoden zur Binarisierung

Sind die Bilder zu hell oder zu dunkel gescannt, liegen aber als Farb- oder Graustufenbilder vor, lässt sich dies möglicherweise noch bei der Umwandlung (Binarisierung) in ein Schwarzweißbild (monochrom) korrigieren.

Diese Umwandlung geschieht normalerweise automatisch vor der Barcode-Erkennung. Diese Standardumwandlung kann im bcTester im Menüpunkt **<Effekte> <Wandeln in Monochrom>** durchgeführt werden. Das binarisierte Bild wird dann angezeigt.

Bei der Umwandlung (Binarisierung) wird ein Schwellwert (engl. threshold) verwendet. Punkte, die heller sind als dieser Schwellwert, werden weiß, die anderen schwarz. Mögliche Werte für die Schwelle liegen zwischen 0 (schwarz) und 255 (weiß).

Bei der internen Umwandlung wird der Schwellwert 160 verwendet. Dieser Wert hat sich für sehr viele Fälle als gut erwiesen. Ist beim **<Effekte> <Wandeln in Monochrom>** das Bild zu hell oder zu dunkel, kann der Standardwert überschrieben werden. Das geschieht durch den Eintrag eines neuen Werts in der qsbc.ini. Soll das Bild nach der Wandlung heller werden, so hilft vermutlich in der qsbc.ini im Abschnitt [Imaging] die Zeile Threshold=128 .

Weitere Binarisierungsmöglichkeiten

Wenn der Barcode im Ergebnisbild zu hell oder dunkel ist, kann durch die folgenden Funktionen die Umwandlung beeinflusst werden. Die nachfolgende Tabelle enthält mögliche Befehle für verschiedene Binarisierungsverfahren, die sich darin unterscheiden, wie der Schwellwert ermittelt wird.

Hinweis

Die folgenden GraphLib Befehle arbeiten nur auf Graustufenbildern (8 Bit) . Farbbilder müssen einem vorhergehenden Schritt in Graustufenbilder gewandelt werden:

```
Step1=FreeImage:GrayScale;
```

Befehl	Beschreibung
GraphLib:Dynamic;	Hierbei wird der Schwellwert aus dem im Bild vorkommenden Helligkeitswerten ermittelt. Dies entspricht der „automatischen Helligkeitseinstellung“, wie sie in vielen Kopiergeräten zu finden ist. Es wird für die Ermittlung des „dynamischen Schwellwerts“ das ganze Bild verwendet, nicht nur der Barcodebereich.
GraphLib:DynamicEx("20");	Der Schwellwert wird wie bei Dynamic bestimmt, allerdings noch um einen Versatz verschoben. Dabei bewirkt ein positiver Versatz eine Verdunklung, ein negativer eine Aufhellung des Bildes.
FreeImage:Threshold,128;	Bei dieser Methode wird der zu verwendende Schwellwert direkt angegeben. Mögliche Werte für die Schwelle liegen zwischen 0 (schwarz) und 255 (weiß).

Barcodes nicht erkannt – Was tun?

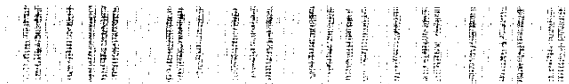
Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Normalerweise liefert `Dynamic` gute Ergebnisse. Manchmal ist der automatisch berechnete Schwellwert jedoch nicht geeignet, beispielsweise, weil aufgeklebte Barcodeetiketten heller gedruckt sind als der Rest des Bildes. Dann lässt sich durch eine Verschiebung des Schwellwertes mit `DynamicEx` oder durch direkte Angabe des zu verwendeten Schwellwerts mittels `Threshold` oft doch noch eine Erkennung erreichen.

Beispiel



Zu hell gescannter Barcode



`Threshold (128) ;`
der mittlere Wert ist zu niedrig



`Threshold (200) ;`
der höhere Wert passt für dieses Ausgangsbild



`Dynamic ;`
dynamisch wird ein guter Wert ermittelt

Ungleichmäßige Beleuchtung

Das Eingabebild



ist in der rechten oberen Ecke heller als in der linken unteren. Für die Barcodeerkennung wird dieses Bild in ein Schwarzweißbild (monochrom) umgewandelt, wie im vorherigen Abschnitt beschrieben. Da ein Schwellwert für das gesamte Bild verwendet wird, ergibt sich bei diesem Bild folgendes Ergebnis:



Dieser Barcode ist nicht mehr erkennbar.

Dass der Barcode (in der Originalaufnahme) von einem menschlichen Betrachter gut zu erkennen ist, liegt daran, dass das Auge lokal beurteilt, was schwarz und was weiß ist.

Mit dem Befehl

```
Step... =GraphLib:LocalThreshold(0,20,"0.9");
```

wird bei der Binarisierung ebenfalls ein „lokaler Schwellwert“ (local threshold) verwendet.

Drei Zahlen steuern diese Bildverbesserung. Die erste 0 ist ein Reserveparameter für spätere Verbesserungen dieser Funktion und kann nicht verändert werden.

Barcodes nicht erkannt – Was tun?

Eine Hilfestellung für QS-Barcode, QS-DocumentAssembler, QS-Beleg und die Barcode Testapplikation bcTester

Die zweite Stelle steuert die Größe der Region, in der der lokale Schwellwert ermittelt wird. 20 hat sich hier als guter Wert erwiesen. Ein kleinerer Wert erhöht zwar die Geschwindigkeit, aber ebenso die Gefahr, dass in einheitlich weißen oder schwarzen Regionen durch kleinere Helligkeitsvariationen Löcher entstehen.

Die „0.9“ in der letzten Stelle wiederum besagt, dass alle Bildpunkte, deren Helligkeit über 90% der mittleren Helligkeit der Umgebung liegt, als weiß betrachtet werden.

Das Ergebnis



weist keine Schattierungseffekte mehr auf und der Barcode lässt sich gut erkennen. Allerdings ist diese Funktion sehr rechen- und daher zeitaufwendig.

Hinweis: QualitySoft bemüht sich, diese Dokumentation immer aktuell zu halten. Hinweise auf Fehler oder unverständliche Stellen nehmen wir gern entgegen. Die QS-Barcode Produkte werden stetig gepflegt und verbessert.