



---

# PROJEKT CODIERUNG UMSETZEN

---

Barcode



9. MAI 2024

JANNE CHARTRON  
Modul 114

## Inhalt

Beschreibung .....	2
Einleitung.....	2
Technologie.....	2
Funktionsweise.....	2
Detaillierte Code Beschreibung .....	3
Generierter Code.....	4

# Beschreibung

## Einleitung

Diese Dokumentation beschreibt die Entwicklung und Funktionsweise eines Barcode-Decoders, der mithilfe einer Kamera Barcodes lesen und decodieren kann. Die Implementierung verwendet Bildverarbeitungstechniken, um Barcodes in Echtzeit zu erkennen und die enthaltenen Informationen zu extrahieren.

## Technologie

Der Barcode-Decoder basiert auf einer Kombination von Technologien:

**Kamera:** Es wird eine Kamera benutzt welche im Laptop vorhanden ist, diese wird mit dem code aktiviert.

**Barcode-Bibliotheken:** Zur Decodierung der Barcodes verwenden wir spezialisierte Bibliotheken und Software.

## Funktionsweise

**Bildaufnahme:** Die Kamera erfasst ein Bild, das einen Barcode enthält.

**Vorverarbeitung:** Das Bild wird vorverarbeitet, um Kontrast und Schärfe zu optimieren sowie störende Elemente zu entfernen.

**Barcode-Erkennung:** Anhand spezieller Algorithmen wird der Barcode im Bild lokalisiert. Dazu können verschiedene Techniken wie Kanten- oder Mustererkennung eingesetzt werden.

**Decodierung:** Sobald der Barcode erkannt ist, wird er anhand des darin enthaltenen Musters und der Struktur decodiert. Dieser Schritt kann auf spezialisierte Bibliotheken zurückgreifen, die die Decodierung für verschiedene Barcode-Typen durchführen können.

**Ausgabe der Informationen:** Die decodierten Informationen (z. B. Produktnummer, URL usw.) werden extrahiert und für die weitere Verarbeitung oder Anzeige verwendet.

# Detaillierte Code Beschreibung

```
/**
 * Decodes the given barcode string.
 *
 * @param barcode The barcode string to decode
 * @return The decoded information from the barcode
 */
static public String decode(String barcode) {
    String decodedData = "Unknown"; // Default decoded data

    // Check if the barcode starts with a known format identifier
    if (barcode.startsWith("UPC")) {
        // Example: Decode UPC barcode
        decodedData = decodeUPC(barcode);
    } else if (barcode.startsWith("QR")) {
        // Example: Decode QR code
        decodedData = decodeQR(barcode);
    } else {
        // Add more specific decoding logic for other barcode formats
        // Here, we assume the barcode format is unknown
        decodedData = "Unknown barcode format";
    }

    return decodedData;
}
```

In dem gezeigten Codeausschnitt wird eine Methode `decode` implementiert, die eine Eingabezeichenkette (`barcode`) entgegennimmt und versucht, diese anhand des Formats zu decodieren, um Informationen aus dem Barcode zu extrahieren.

```
private static String decodeUPC(String barcode) {

    // Beispiel: Extrahiere und interpretiere UPC-Daten

    String upcData = barcode.substring(3); // Annahme: UPC-Format beginnt mit "UPC"

    return "UPC: " + upcData;

}
```

Funktionsweise:

- Die Methode `decodeUPC` nimmt einen Barcode-String (`barcode`) als Eingabe entgegen.
- Mit der Annahme, dass der UPC-Barcode mit dem Präfix "UPC" beginnt, extrahiert die Methode `decodeUPC` die Daten aus dem Barcode-String, beginnend ab dem 4. Zeichen (`barcode.substring(3)`). Dies ist eine vereinfachte Annahme und muss je nach tatsächlichem Format des UPC-Barcodes angepasst werden.
- Die extrahierten Daten (`upcData`) werden dann als Teil einer interpretierten Zeichenkette zurückgegeben, die "UPC: " gefolgt von den extrahierten Daten enthält.

```
private static String decodeQR(String barcode) {
    // Beispiel: Extrahiere und interpretiere QR-Code-Daten
    String qrData = barcode.substring(3); // Annahme: QR-Format beginnt mit "QR"
    return "QR Code: " + qrData;
}
```

Funktionsweise:

- Die Methode decodeQR nimmt einen Barcode-String (barcode) als Eingabe entgegen.
- Mit der Annahme, dass der QR-Code mit dem Präfix "QR" beginnt, extrahiert die Methode decodeQR die Daten aus dem Barcode-String, beginnend ab dem 4. Zeichen (barcode.substring(3)). Dies ist eine vereinfachte Annahme und muss je nach tatsächlichem Format des QR-Codes angepasst werden.
- Die extrahierten Daten (qrData) werden dann als Teil einer interpretierten Zeichenkette zurückgegeben, die "QR Code: " gefolgt von den extrahierten Daten enthält.

## Generierter Code

Das ist mein Generierter Code den rest habe ich selber gemacht.

```
private static String decodeQR(String barcode) {
    // Beispiel: Extrahiere und interpretiere QR-Code-Daten
    String qrData = barcode.substring(3); // Annahme: QR-Format beginnt mit "QR"
    return "QR Code: " + qrData;
}
```