

# Softwarehandbuch

## Wertbeutelfahnenendrucker

**Express** 



*Stand: Mai 2003*

3092.01.094.90.06



# Inhaltsverzeichnis

1	ESC-Sequenzen .....	4
1.1	Erstellen von Datensätzen .....	4
1.1.1	Symbole und Konventionen .....	4
1.1.2	Benutzung von Steuersequenzen .....	5
1.1.3	Benutzung von Objektblöcken .....	5
1.1.4	Datensatzaufbau .....	5
1.1.5	Beispiel für einen Datensatz.....	6
1.2	Steuersequenzen .....	6
1.2.1	Kartenzahl .....	7
1.2.2	Bildhöhe.....	7
1.2.3	Bildbreite.....	7
1.2.4	Variables Objekt: Logo .....	7
1.2.5	Ländercode.....	8
1.2.6	Transponderdaten .....	8
1.2.7	Variables Objekt: Text / Barcode .....	9
1.3	Objektsequenzen.....	9
1.3.1	Objekt-Attribute .....	9
1.3.2	Y-Vergrößerung.....	10
1.3.3	X-Vergrößerung.....	10
1.3.4	Zeichenabstand .....	11
1.3.5	Positionierung.....	11
1.3.5.1	X-Koordinate.....	12
1.3.5.2	Y-Koordinate.....	12
1.3.6	Internes Logo .....	12
1.3.7	Fortschaltung.....	12
1.3.8	Drehung.....	13
1.3.9	Transponder beschreiben .....	13
1.3.10	Variable Objekte.....	14
1.3.11	Linie und Rahmen .....	14
1.3.12	Bildhintergrund .....	15
1.4	Objekte .....	15
1.4.1	Text - Objekt.....	16
1.4.2	Logo - Objekt.....	16
1.4.3	Barcode - Objekt .....	18
1.4.3.1	Einführung .....	18
1.4.3.2	Barcode: Code 2 of 5 Interleaved.....	20
1.4.3.3	Barcode: Code 39 .....	21
1.4.3.4	Barcode: Code 128 .....	22
1.4.3.5	Barcode: EAN-8 .....	23
1.4.3.6	Barcode: EAN-13 .....	24
1.4.3.7	Barcode: EAN-128 .....	25
1.4.3.8	Barcode: PDF-417.....	27
1.5	Bevorzugte Sequenzen .....	30
1.5.1	Statusmeldung .....	30
1.5.2	RFID-Status.....	30
1.5.3	Software-Reset.....	30
2	Anhang .....	31
2.1	Anhang A: Fehlermeldungen.....	31
2.1.1	Fehlerebene 1 - Warnung .....	31
2.1.2	Fehlerebene 2 - Fehler.....	34
2.1.3	Fehlerebene 3 - Hardware .....	34
2.2	Anhang B: Befehlsübersicht .....	35
2.2.1	Steuersequenzen .....	35
2.2.2	Objektsequenzen .....	36
2.2.3	Bevorzugte Sequenzen .....	36



## 1 ESC-Sequenzen

Dieser Teil enthält die Druckerkommandos zum direkten Ansteuern des Druckers und ist in 4 Kapitel (Steuersequenzen, Objektsequenzen, Objekte und bevorzugte Sequenzen) unterteilt.

### 1.1 Erstellen von Datensätzen

Hier wird der Aufbau eines Datensatzes und die Verwendung verschiedener Sequenzen erläutert.

#### 1.1.1 Symbole und Konventionen

Im weiteren Verlauf der Beschreibung werden einige besondere Symbole und Konventionen angewandt:

**Fettgedruckte** Zeichen sind Schlüsselwörter und müssen genauso eingegeben werden.

*Kursivschrift* innerhalb einer Sequenz stellt Platzhalter dar, die der Anwender durch bestimmte Werte ersetzt.

Unterstrichene Werte sind Standard-Werte.

Nichtdruckbare Steuerzeichen werden in spitzen Klammern eingeschlossen, z.B. <C<sub>R</sub>>.

Die Eingabe von Steuerzeichen kann je nach Programm unterschiedlich sein. Hier einige Beispiele:

<S <sub>X</sub> >, <STX>	⊔	STX-Zeichen, 02 Dezimal, 02 Hexadezimal, CTRL/B
<E <sub>T</sub> >, <EOT>	⊔	EOT-Zeichen, 04 Dezimal, 04 Hexadezimal, CTRL/D
<E <sub>Q</sub> >, <ENQ>	⊔	ENQ-Zeichen, 05 Dezimal, 05 Hexadezimal, CTRL/E
<A <sub>K</sub> >, <ACK>	⊔	ACK-Zeichen, 06 Dezimal, 06 Hexadezimal, CTRL/F
<C <sub>R</sub> >, <CR>	⊔	CR-Zeichen, 13 Dezimal, 0D Hexadezimal, CTRL/M
<E <sub>C</sub> >, <ESC>	⊔	ESC-Zeichen, 27 Dezimal, 1B Hexadezimal, CTRL/[

Wahlweise Eingabemöglichkeiten werden in Rechteckklammern [ ] dargestellt.

Für Zahlenwerte im ASCII-Eingabeformat gelten folgende Konventionen: Jede Ziffernstelle bekommt einen Platzhalter in Form eines Buchstabens. Der Buchstabe kennzeichnet das Zahlenformat. Jede Zahl wird durch ihren ASCII-Wert dargestellt.

dddd bedeutet: 4-stellige Dezimalzahl (0000 - 9999) im ASCII-Format

hhh bedeutet: 3-stellige Hexzahl (000 - fff) im ASCII-Format

Ist die Eingabe nicht an ein festes Format gebunden, so wird dies mit 2 Punkten (..) angezeigt:

d.. steht z.B. für: 1, 12, 123, 012, ...

Für Zahlenwerte im binären Eingabeformat gelten folgende Konventionen: Jede 8-bit-Binärzahl wird durch ihren hexadezimalen Wert dargestellt. Jede Hexziffer bekommt einen Platzhalter (1 Hexziffer = 4 Bit).

HH bedeutet: 2-stellige Hexzahl (00 - ff) im Hexformat

af bedeutet: 175 dezimal, af hexadezimal, 10101111 binär

Für Charakterwerte gelten folgende Konventionen: Jede Zeichenstelle bekommt einen Platzhalter in Form eines Buchstabens. Der Buchstabe kennzeichnet das Zeichenformat. Jedes Zeichen wird durch seinen ASCII-Wert dargestellt.

c bedeutet: 1 Charakter (A - Z, a - z)

a bedeutet: 1 Charakter (A - Z, a - z, 0 - 9)

0|1 stellt 0 oder 1 als Eingabemöglichkeit zur Verfügung.

Tiefgestellte Texte dienen lediglich zur Information.

d<sub>typ</sub> bedeutet: 1-stellige Typnummer, dezimal.



### 1.1.2 Benutzung von Steuersequenzen

Die Steuerung des Druckers wird beeinflusst durch die Gerätekonfiguration. Die Gerätekonfiguration kann entweder über das Menü oder über Steuersequenzen verändert werden.

Eine Steuersequenz hat folgenden Aufbau:

Beginn mit <ESC> und einem darauffolgenden Kleinbuchstaben als Schlüsselwort.

Alle Steuersequenzen werden mit einem <CR> beendet.

Die Summe aller aufeinanderfolgender Steuersequenzen bilden den Steuerblock.

Fehlerhafte Steuersequenzen lösen Fehlermeldungen mit dem Hinweis 'Steuersequenz' aus.

### 1.1.3 Benutzung von Objektblöcken

Objektblöcke bestimmen das eigentliche Druckbild, nachdem die Gerätekonfiguration abgeschlossen ist.

Der Express verfügt über eine eigene Steuersprache, die nicht kompatibel zu anderen zeilenorientierten Matrixdruckern ist. Dafür bietet das Gerät eine große Auswahl an Gestaltungsmöglichkeiten zum Erstellen eines Layouts. Textzeilen, Barcodes und Logos können beliebig auf eine Karte in verschiedener Form und Richtung gedruckt werden.

Um eine Karte zu bedrucken, muss zunächst das gesamte Daten-Layout in Form von Objektblöcken definiert werden. Erst nach Erhalt des kompletten Layoutblocks kann ein Datensatz gedruckt werden.

Objektblöcke haben folgenden Aufbau:

Ein Objektblock besteht aus Objektsequenzen, die mit <ESC> und einem darauffolgenden Großbuchstaben als Schlüsselwort beginnen.

Objektsequenzen innerhalb eines Objektblocks können wahlweise mit oder ohne <CR> beendet werden.

Alle Objektblöcke zusammen bilden einen Layoutblock. Ein Layoutblock muss zur Unterscheidung von Steuerblöcken in <STX>.... <EOT> eingeschlossen werden.

Die meisten Fehlermeldungen, die von fehlerhaften Objektsequenzen ausgelöst werden, haben im Fehlerfenster den Hinweis 'Objektsequenz'.

Variable Objektsequenzen werden formal wie Steuersequenzen behandelt.

Alle Objektparameter haben bereits eine Grundeinstellung, z.B. Zeichenabstand = 1 Dot.

### 1.1.4 Datensatzaufbau

Der Datensatz hat folgenden Aufbau:

Datensatz	= {Steuerblock, Layoutblock, Steuerblock}
Steuerblock	= { Steuersequenz 1 [Steuersequenz n] } n=[2...]
Steuersequenz	= { <ESC>x... <CR> } x=[a b ... z]
Layoutblock	= { <STX> Objektblock 1 [Objektblock n] <EOT> } n = [2...]
Objektblock	= { [Objektsequenz n] Objekt } n = [1...]
Objektsequenz	= { <ESC> X ... [<CR>] } X = [A C ... Z]
Objekt	= { <ESC> B T L ... [<CR>] }

#### *Datensatz*

Beschreibt das komplette Karten-Layout einschließlich Druckerkonfiguration und Druckvorgang.

#### *Steuerblock*

Beschreibt die komplette Druckerkonfiguration und Druckvorgang.

#### *Steuersequenz*

Setzt einen Steuerparameter für Druckerkonfiguration bzw. Druckvorgang.

#### *Objektblock*

Beschreibt das komplette Karten-Layout.



## ESC-Sequenzen

## Objektsequenz, Objekt

Das Karten-Layout enthält Datenfelder unterschiedlichster Art. Eine Objektsequenz liefert einen Parameter zur Spezifikation dieses Feldes. Ein Objekt ist formal eine Objektsequenz und spezifiziert den Feldtyp (Text, Barcode, Logo/Linie/Rahmen). Diese muss immer an letzter Stelle in einem Objektblock stehen. Die vorhergehenden Objektsequenzen beziehen sich auf dieses Objekt.

Datensätze werden zunächst im Eingabespeicher zwischengespeichert und nacheinander aufbereitet. Ein bereits aufbereiteter Datensatz belegt keinen Eingabespeicher mehr. Folgende Angaben im Standard-Fenster geben Aufschluss über den momentanen Speicherzustand:

- >    ⊔ Daten können eingelesen werden
- |       ⊔ Eingabespeicher voll
- ⊔ kein Datensatz aufbereitet
- n      ⊔ Datensatz aufbereitet
- #      ⊔ Anzahl Karten, die vom aktuellen Datensatz noch zu drucken sind

## 1.1.5 Beispiel für einen Datensatz

ESC-Sequenz	Erklärung
<ESC>j100<CR>	<b>Beginn des Datensatzes</b> Steuersätze: (1. Steuerblock) Dieser Block kann frei bleiben
<STX>	<b>Beginn Layoutblock</b>
<ESC>I35<CR> <ESC>G150<CR> <ESC>R0<CR> <ESC>BEAN13;H60;B3;P1>4012 34567890<CR>	Objektblock 1 (=Barcodesatz)
<ESC>I35<CR> <ESC>G20<CR> <ESC>R270<CR> <ESC>D1<CR> <ESC>C2<CR> <ESC>F3<CR> <ESC>TARIAL18F;Drehung 270°<CR>	Objektblock 2 (=Textdaten)  Text = ' Drehung 270° '
<EOT>	<b>Ende Layoutblock</b>
<ESC>#1<CR>	Steuersätze: (2. Steuerblock) Druckauftrag <b>Ende des Datensatzes</b>

Nachdem diese Daten zum Drucker übertragen worden sind, sollte der Ausdruck eines Datensatzes erfolgen. Steht in der Anzeige nach dem Senden des Datensatzes eine Fehlermeldung, so kann die Fehlerursache über den Fehlercode im Anhang nachgesehen werden. Eine mögliche Fehlerursache könnte sein, dass die Verbindung Rechner-Drucker nicht korrekt hergestellt ist oder die Übertragungsparameter nicht korrekt eingestellt sind.

## 1.2 Steuersequenzen

Aufbau:

Steuerblock	= { Steuersequenz 1 [Steuersequenz n] }	n=[2...]
Steuersequenz	= { <ESC>x.... <CR> }	x=[a b ... z]



## ESC-Sequenzen

**Auflistung der möglichen Steuersequenzen:**

Kartenanzahl  
 Bildhöhe  
 Bildbreite  
 Variables Objekt: Logo  
 Ländercode  
 Variables Objekt: Text / Barcode

Auf den nachfolgenden Seiten sind die einzelnen Steuersequenzen detailliert beschrieben.

**1.2.1 Kartenanzahl**

<i>Menüanwahl:</i>	KARTENDATEN / Kartenanzahl / <u>0000</u>
<i>Steuersatz:</i>	<ESC>#d.<CR>
<i>Beispiel:</i>	<ESC>#10<CR>

Die Steuersequenz veranlasst den Ausdruck eines Druckauftrages mit der angegebenen Anzahl von Karten.

Parameter: *d* = Kartenanzahl

**1.2.2 Bildhöhe**

<i>Menüanwahl:</i>	KARTENDATEN / Bildhöhe / <u>1440</u>
<i>Steuersatz:</i>	<ESC>bd.<CR>
<i>Beispiel:</i>	<ESC>b240<CR>

Die *Bildhöhe* legt die Höhe des Druckbereiches fest.

Parameter: *d* = Bildhöhe \* 12/mm (Anzahl Motorschritte),

Mindesthöhe = 10 mm (d = 120), Maximalhöhe = 500 mm (d = 6000)

**1.2.3 Bildbreite**

<i>Menüanwahl:</i>	KARTENDATEN / Bildbreite / <u>960</u>
<i>Steuersatz:</i>	<ESC>cd.<CR>
<i>Beispiel:</i>	<ESC>c650<CR>

Die *Bildbreite* legt die Breite des Druckbereiches fest.

Parameter: *d* = Bildbreite \* 12/mm (Anzahl Dots),

Mindestbreite = 5,3 mm (d = 64), Maximalbreite = 80 mm (d = 960)

Der Wert für die *Bildbreite* kann jederzeit kleiner als das tatsächliche Kartenformat sein. Der resultierende Druckbereich wird dadurch mittig verschoben.

**1.2.4 Variables Objekt: Logo**

<i>Steuersatz:</i>	<ESC>   a <sub>obj.</sub> ; d. <sub>Breite</sub> ; d. <sub>Höhe</sub> ; HH <sub>Logodaten</sub> <CR >
	a <sub>obj.</sub> : Objekt-Name
	d. <sub>Breite</sub> : Logo-Breite : Anzahl Dots in X-Richtung
	d. <sub>Höhe</sub> : Logo-Höhe : Anzahl Dots in Y-Richtung
	HH <sub>Logodaten</sub> : Logodaten in Binärform
	Jedes Bit eines Bytes repräsentiert 1 Punkt
	Bit=0: Punkt nicht drucken, =1: Punkt drucken
	Datenbit: 7 6 5 4 3 2 1 0
	Punkt: 1 2 3 4 5 6 7 8
<i>Beispiel:</i>	<ESC>   a ; 8 ; 5 ; 08 08 C8 28 10 <CR>
	<ESC> # 1 <CR>

Das *Logo* muss zuvor als variables Objekt definiert worden sein. (siehe 1.3.10 Variable Objekte)



## ESC-Sequenzen

## 1.2.5 Ländercode

<b>Menüwahl:</b>	ZEICHENSATZ / Ländercode / <u>USA</u>
<b>Steuersatz:</b>	<ESC>n<CR>
<b>Beispiel:</b>	<ESC>n2<CR>

Die Steuersequenz ermöglicht die Einstellung verschiedener Ländercodes.

Parameter:

$d = 0$	USA
$d = 1$	England
$d = 2$	Deutschland
$d = 3$	Dänemark
$d = 4$	Frankreich
$d = 5$	Schweden
$d = 6$	Italien
$d = 7$	Spanien
$d = 8$	Norwegen
$d = 9$	Holland

## 1.2.6 Transponderdaten

<b>Steuersatz:</b>	<ESC> u $d_{\text{Offset}}$ ; $d_{\text{Länge}}$ ; $a_{\text{Aktion}}$ ; [ $HH_{\text{Daten}}$ ] <CR>	
	$d_{\text{Offset}}$ = Offset in Blöcke:	0 ... Anzahl Blöcke
	$d_{\text{Länge}}$ = Länge in Byte:	0 ... Transpondergröße *
	$a_{\text{Aktion}}$ = r / w	r -> Lesen / w -> Schreiben
	$HH_{\text{Daten}}$ = Transponderdaten:	Daten in Binärform
<b>Beispiel:</b>	<ESC>u1;5;r <CR>	
	Der erste Block wird übersprungen und anschließend 5 Bytes ausgelesen.	
	* In einer Sequenz können maximal 216 Bytes gelesen und maximal 32 Bytes geschrieben werden.	

Der Drucker liest den Inhalt oder beschreibt den Transponder und gibt die entsprechenden Daten an den Host weiter. (siehe auch 1.3.9)

Die Ausgabe der Daten erfolgt nach der ISO 15693 und wird über die Serielle- und Parallele Schnittstelle übertragen.

Transponder Tag-it HF / I-CODE:

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Beisp.					H	a	l	l	o								
Offset	0				1				2				3				4

Transponder MY-D:

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Beisp.									H	a	l	l	o				
Offset	0								1								2

**Beispiel:**

ESC-Sequenz	Erklärung
<ESC>u0;0;r;<CR>	Anforderung der Ser.-Nr. des Transponders Der Drucker liefert 9 Bytes zurück 1. Byte = Transpondertyp 2.-9. Byte = Seriennummer des Transponders
<ESC>u0;0;w;<CR>	Transponder komplett löschen Rückgabe: Anzahl der gelöschten Bytes
<ESC>u2;12;r;<CR>	Es werden ab dem 2. Block 12 Bytes ausgelesen. Rückgabe: 12 Bytes
<ESC>u2;12;w;1234567Hallo<CR>	Es werden ab dem 2. Block 12 Bytes geschrieben. Rückgabe: Anzahl der tatsächlich geschriebenen Bytes.



Transpondertypen	Blockgröße in Bytes	Anzahl Blöcke	Nutzbarer Bereich in Blöcke	Kapazität in Bytes
Philips I-CODE 1	4 Bytes	11	von 0 bis 10	44 Bytes
Philips I-CODE SLI	4 Bytes	28	von 0 bis 27	112 Bytes
Texas Instruments Tag-it HF	4 Bytes	8	von 0 bis 7	32 Bytes
Infineon MY-D	8 Bytes	125	von 3 bis 127	1000 Bytes

### 1.2.7 Variables Objekt: Text / Barcode

<b>Steuersatz:</b>	<b>&lt;ESC&gt; v a<sub>obj.</sub> ; Daten &lt;CR&gt;</b>	
	a <sub>obj.</sub> :	Objekt-Name
	Daten :	Text-Daten oder Barcode-Daten
<b>Beispiel:</b>	<ESC> v a ; Text <CR>	Text-Objekt
	<ESC> # 1 <CR>	
	<ESC> v a ; <i>Barcodedaten</i> <CR>	Barcode-Objekt
	<ESC> # 1 <CR>	

Der *Text* bzw. *Barcode* muss zuvor als variables Objekt definiert worden sein. (siehe 1.3.10 Variable Objekte)

## 1.3 Objektsequenzen

Aufbau:

Datensatz	= { Steuerblock, Layoutblock, Steuerblock }
Layoutblock	= { <STX> Objektblock 1 [Objektblock n] <EOT> } n = [2...]
Objektblock	= { [Objektsequenz n] Objekt } n = [1...]
Objektsequenz	= { <ESC> X ... [<CR>] } X = [A C ... Z]
Objekt	= { <ESC> B T L ... [<CR>] }

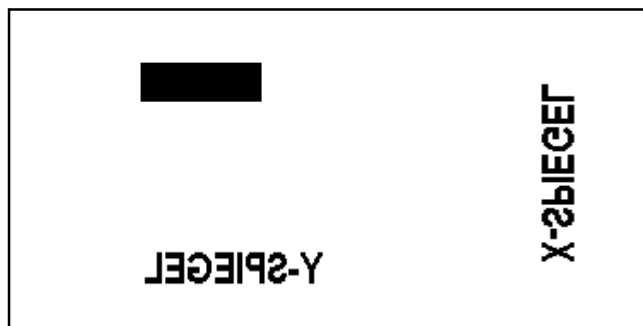
Mittels Objektsequenzen wird ein Daten-Layout (Layoutblock) definiert.

Über mehrere Objektsequenzen kann ein Objekt in Form und Position bestimmt werden. Alle Objektsequenzen, die ein Objekt beschreiben, stellen einen Objektblock dar. Die letzte Objektsequenz eines Objektblockes definiert den Objekttyp (Text, Barcode, Logo/Rahmen/Linie). Die Anzahl der Objekte ist nicht begrenzt.

Alle Objektblöcke zusammen ergeben einen Layoutblock für 1 Karte. Auf den nachfolgenden Seiten sind die einzelnen Objektsequenzen detailliert beschrieben.

### 1.3.1 Objekt-Attribute

<b>Objektsequenz:</b>	<b>&lt;ESC&gt; A dddd [&lt;CR&gt;]</b>
ddd	= 0001 : Objekt invertieren
	= 0002 : Objekt an X-Achse spiegeln
	= 0004 : Objekt an Y-Achse spiegeln
	= 0010 : Transparenz ausschalten (keine Veroderung)
<b>Beispiel:</b>	<ESC>A0001 Default: A0000



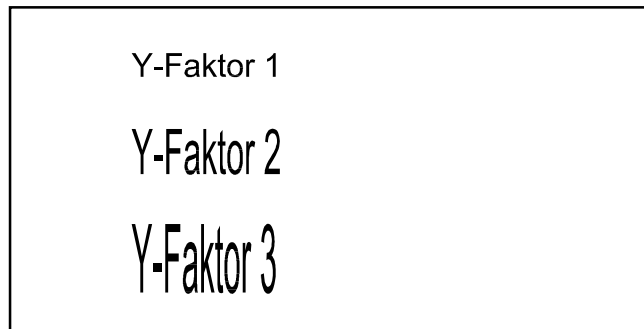
Es können auch mehrere Attribute gleichzeitig definiert werden. Hierzu muss lediglich die Summe der Einzelattribute gebildet werden.

**Beispiel:**

ESC-Sequenz	Erklärung
<ESC> A 0003	invertiert und an X-Achse gespiegelt

## 1.3.2 Y-Vergrößerung

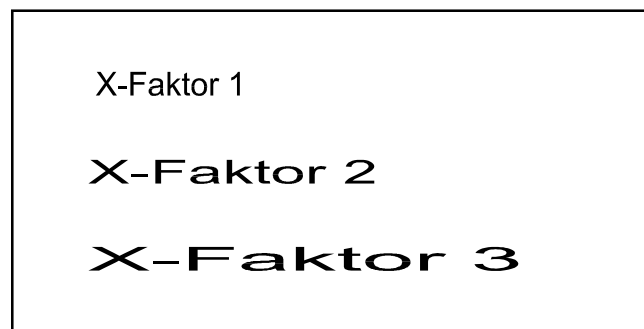
**Objektsequenz:** <ESC> **C** *d*.. [<CR>]  
*d* = 1 ... 255 : Y-Vergrößerungsfaktor  
**Beispiel:** <ESC>C2 Default: C1

**Beispiel:**

```
<STX>
<ESC>G50<ESC>I35<ESC>C1<ESC>TARIAL14f;Y-Faktor 1
<ESC>G50<ESC>I70<ESC>C2<ESC>TARIAL14f;Y-Faktor 2
<ESC>G50<ESC>I135<ESC>C3<ESC>TARIAL14f;Y-Faktor 3
<EOT>
```

## 1.3.3 X-Vergrößerung

**Objektsequenz:** <ESC> **D** *d*.. [<CR>]  
*d* = 1 ... 255 : X-Vergrößerungsfaktor  
**Beispiel:** <ESC>D2 Default: D1

**Beispiel:**

```
<STX>
<ESC>G50<ESC>I50<ESC>D1<ESC>TARIAL14f;X-Faktor 1
<ESC>G50<ESC>I90<ESC>D2<ESC>TARIAL14f;X-Faktor 2
<ESC>G50<ESC>I140<ESC>D3<ESC>TARIAL14f;X-Faktor 3
<EOT>
```

### 1.3.4 Zeichenabstand

<b>Objektsequenz:</b>	<ESC> F d.. [<CR>]
	d = 1 ... 255 : Anzahl Leerdots
<b>Beispiel:</b>	<ESC>F3 Default: F1

**Zeichenabstand 1**  
**Zeichenabstand 2**  
**Zeichenabstand 3**  
**Zeichenabstand 4**  
**Zeichenabstand 5**

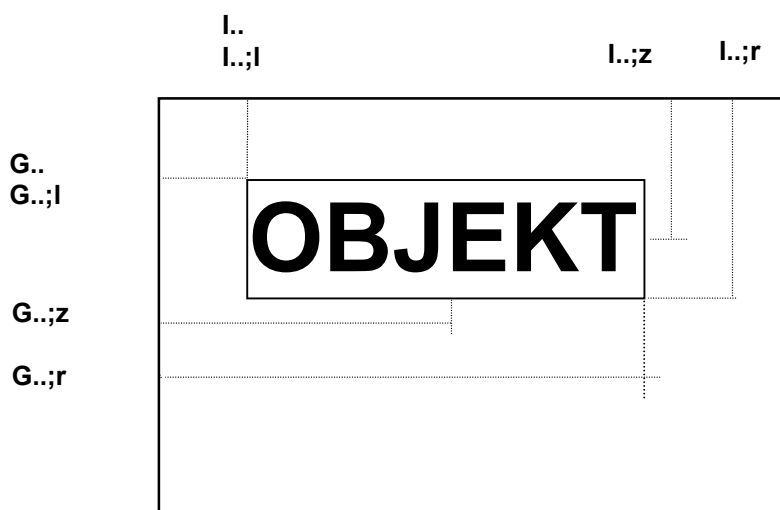
Es wird der Abstand zwischen 2 Zeichen festgelegt. Die hier definierten Leerdots werden zusätzlich zu dem im Zeichenfont vorgegebenen Zeichenabstand eingefügt.

**Beispiel:**

```
<STX>
<ESC>G50<ESC>I50<ESC>F1<ESC>TARIAL14f;Zeichenabstand 1
<ESC>G50<ESC>I100<ESC>F2<ESC>TARIAL14f;Zeichenabstand 2
<ESC>G50<ESC>I150<ESC>F3<ESC>TARIAL14f;Zeichenabstand 3
<ESC>G50<ESC>I200<ESC>F4<ESC>TARIAL14f;Zeichenabstand 4
<ESC>G50<ESC>I250<ESC>F5<ESC>TARIAL14f;Zeichenabstand 5
<EOT>
```

### 1.3.5 Positionierung

Das Positionieren der Objekte erfolgt durch Angabe von X- und Y-Koordinate. Ohne weitere Angabe bezieht sich der so definierte Koordinatenpunkt auf die linke obere Ecke des betreffenden Text-, Barcode- oder Logo-Objektes. Durch zusätzliche Parameter kann als Bezugspunkt aber auch die Mitte oder der rechte bzw. untere Rand festgelegt werden. Hierdurch ist es möglich, sämtliche Objekte eines Kartenlayouts sowohl in X- als auch in Y-Richtung zu zentrieren als auch an beiden Rändern auszurichten. Bei der Positionierung ist auf einen Mindestabstand von 3mm zum unteren Bildrand zu achten.



## 1.3.5.1 X-Koordinate

Objektsequenz:	<b>&lt;ESC&gt; G</b> <i>d.</i> [ ; x-Ausrichtung ] [ <b>&lt;CR&gt;</b> ]	(X - Koordinate)
	<i>d</i> = 1 ... 960	Anzahl Dots vom linken Bildrand
	x-Ausrichtung = l	linksbündig
	r	rechtsbündig
	z	zentriert
Beispiel:	<b>&lt;ESC&gt;G10;z</b>	

## 1.3.5.2 Y-Koordinate

Objektsequenz:	<b>&lt;ESC&gt; I</b> <i>d.</i> [ ; y-Ausrichtung ] [ <b>&lt;CR&gt;</b> ]	(Y - Koordinate)
	<i>d</i> = 1 ... 6000	Anzahl Dots vom oberen Bildrand
	y-Ausrichtung = l	linksbündig
	r	rechtsbündig
	z	zentriert
Beispiel:	<b>&lt;ESC&gt;I10</b>	

## 1.3.6 Internes Logo

Objektsequenz:	<b>&lt;ESC&gt; M</b> <i>Bildname</i> ; [ <b>&lt;CR&gt;</b> ]
	<i>Bildname</i> = Name des intern abgelegten Logos
Beispiel:	<b>&lt;ESC&gt;M FDLogo;</b>

Mit Hilfe dieser Sequenz können Logos, die im Drucker implementiert sind, in einen Datensatz mit eingebunden werden. Die im Drucker abgespeicherten Logos kann man sich im Menü *Zeige Infos / Logos* anzeigen und ausdrucken lassen.

## 1.3.7 Fortschaltung

Objektsequenz:	<b>&lt;ESC&gt; Q</b> <i>d..w;d..z [;d..f [;d..B [;d..A]]]</i> [ <b>&lt;CR&gt;</b> ]
<i>d..w</i> = -9 ... +9	: Fortschaltungswert
<i>d..z</i> = 1 ... 255	: Fortschaltungszyklus 1-254: Anzahl Karten ohne Fortschaltung 255 : Fortschaltung nach jedem Auftrag
<i>d..f</i> = <u>0</u>   1	: Filter für führende Nullen ( Textobjekte ) 0 : führende Nullen drucken 1 : führende Nullen unterdrücken
<i>d..B</i> = <u>1</u> ...	: Beginn des Fortschaltungsfeldes Position der 1.Ziffer des Fort.feldes
<i>d..A</i> = <u>0</u> ...	: Feldgröße: Anzahl Ziffern = 0 : bis Ende Text-, Barcodestring : > 0 : Anzahl der relevanten Ziffern
Beispiel:	<b>&lt;ESC&gt;Q1;1</b>

Die Fortschaltungsfunktion kann angewendet werden für **Text**- und **Barcode**-Objekte.

Wird die Felddefinition so gewählt, dass das definierte Feld nicht komplett innerhalb des betreffenden Text- oder Barcodestrings liegt, so wird keine Fortschaltung getätigt.

**Beispiele:**

ESC-Sequenz	Erklärung
<b>&lt;ESC&gt;Q1;1</b>	nach jeder Karte wird der komplette Textstring um +1 fortgeschaltet. Führende Nullen werden gedruckt.
<b>&lt;ESC&gt;Q-1;2;1</b>	nach jeweils 2 Karten wird der komplette Textstring um -1 fortgeschaltet. Führende Nullen werden durch Leerzeichen ersetzt. Ist der Wert des betreffenden Feldes = 0, so wird am Ende eine '0' gedruckt.



## ESC-Sequenzen

ESC-Sequenz	Erklärung
<ESC>Q1;1;0;4	nach jeder Karte wird das relevante Ziffernfeld um +1 fortgeschaltet. Führende Nullen werden ausgedruckt. Das relevante Ziffernfeld beginnt mit der 4. Stelle des betreffenden Text-, Barcodestrings und endet am Stringende.
<ESC>Q1;1;0;4;3	nach jeder Karte wird das relevante Ziffernfeld um +1 fortgeschaltet. Führende Nullen werden ausgedruckt. Das relevante Ziffernfeld beginnt mit der 4. Stelle des betreffenden Text-, Barcodestrings und erstreckt sich über insgesamt 3 Ziffern.
<ESC>Q1;255	nach jedem abgeschlossenem Druckauftrag wird der komplette Textstring um +1 fortgeschaltet. Führende Nullen werden gedruckt. Ein Druckauftrag wird mit der Steuersequenz: ESC># d.. eingeleitet.

## 1.3.8 Drehung

**Objektsequenz:** <ESC> R 0 | 90 | 180 | 270 [<CR>]  
**Beispiel:** <ESC> R0

Jedes Objekt kann um 90°, 180° oder 270° gedreht werden.

**Beispiel:**

```
<STX>
<ESC>G396;z<ESC>I20<ESC>R0<ESC>TARIAL20f;Drehung 0°
<ESC>G700<ESC>I240;z<ESC>R90<ESC>TARIAL20f;Drehung 90°
<ESC>G396;z<ESC>I400<ESC>R180<ESC>TARIAL20f;Drehung 180°
<ESC>G20<ESC>I240;z<ESC>R270<ESC>TARIAL20f;Drehung 270°
<EOT>
```

## 1.3.9 Transponder beschreiben

**Objektsequenz:** <ESC> U d..Offset ; d..Länge ; HH<sub>Daten</sub> <CR>  
 $d_{\text{Offset}}$  = Offset in Blöcke: 0 ... Anzahl Blöcke  
 $d_{\text{Länge}}$  = Länge in Byte: 0 ... 32 Bytes  
 $HH_{\text{Daten}}$  = Transponderdaten: Daten in Binärform

**Beispiel:** <ESC>U1;5;Hallo<CR>  
Der erste Block wird übersprungen und anschließend 5 Bytes (Hallo) geschrieben.

Der Transponder wird vor dem Drucken beschrieben, gleichzeitig wird eine Schreibkontrolle durchgeführt. Bei erfolgreichem Beschreiben ertönt ein kurzer Signalton und bei einem Schreibfehler ertönt ein langer Signalton mit Anzeige des Fehlers. Im Fehlerfall wird das Etikett nicht bedruckt.



## ESC-Sequenzen

Der Transponder ist in Blöcke von jeweils 4 Byte / 8 Byte aufgeteilt. Beschreibt man 1 Block nicht vollständig, wird der Rest des Blocks mit FF gefüllt. (siehe 1.2.6)

## 1.3.10 Variable Objekte

<b>Objektsequenz:</b>	<b>&lt;ESC&gt; V</b> $a_{obj}$ [CR>]
	$a_{obj}$ = Objekt-Name : 0-9 , A-Z , a-z (1 Zeichen)
<b>Beispiel:</b>	<ESC>V1

Jedes Objekt (Barcode-, Text- und Logo-Objekt) kann als variabel definiert werden. Dies eröffnet die Möglichkeit, außerhalb des Layoutblocks diesem Objekt neuen Inhalt zuzuweisen, ohne dass der komplette Datensatz neu übertragen werden muss.

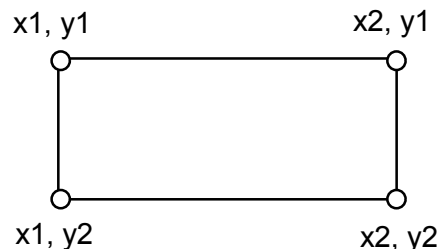
Das im Layoutblock definierte Originalobjekt gibt die maximale Feldgröße vor. Später kann dieses Objekt also maximal mit gleich vielen Daten gefüllt werden. Die Übergabe neuer Daten erfolgt mit der Steuersequenz:

<ESC>va <sub>obj</sub> .	für Text- / Barcode-Objekte	(siehe 1.2.6)
<ESC>la <sub>obj</sub> .	für Logo-Objekte	(siehe 1.2.4)

## 1.3.11 Linie und Rahmen

<b>Objektsequenz:</b>	<b>&lt;ESC&gt; X</b> $d_{..x1}$ ; $d_{..y1}$ ; $d_{..x2}$ ; $d_{..y2}$ ; $d_{..Breite}$ [ ; $d_{..Füllen}$ ] [<CR>]
	$d_{..x1}$ , $d_{..y1}$ : Koordinate 1 : 1 ... Kartenbreite
	$d_{..x2}$ , $d_{..y2}$ : Koordinate 2 : 1 ... Kartenhöhe
	$d_{..Breite}$ : Linienbreite: Anzahl Dots : 1 ... Kartenbreite
	$d_{..Füllen}$ : Rahmen füllen

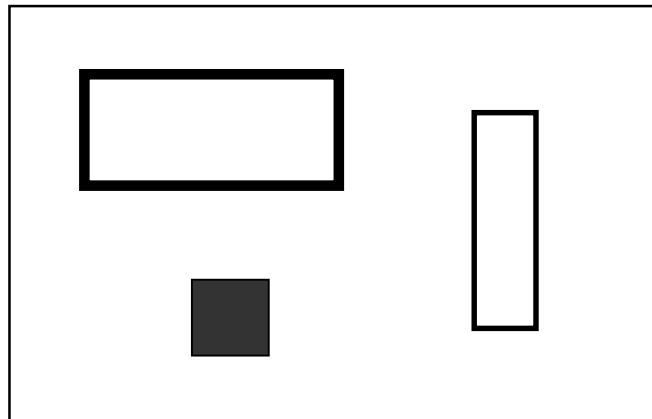
Über diese Objektsequenz können sowohl Linien als auch Rahmen erzeugt werden. Falls [x1,y1] und [x2,y2] eine diagonale Linie beschreiben, wird ein Rahmen gedruckt, ansonsten eine waagerechte oder senkrechte Linie.



Es ist zu beachten, dass das 2. Koordinatenpaar einen höheren Wert haben muss als das erste. Linien werden nach innen verbreitert.

**Beispiel:**

```
<STX>
<ESC>X20;20;250;150;6
<ESC>X300;40;350;330;3
<ESC>X120;220;200;300;1;1
<EOT>
```



### 1.3.12 Bildhintergrund

Objektsequenz: `<ESC> Y HHBilddaten <CR >`  
 $HH_{\text{Bilddaten}}$  Bilddaten in Binärform  
 Jedes Bit eines Bytes repräsentiert 1 Punkt  
 Bit = 0: Punkt nicht drucken, = 1: Punkt drucken  
 Datenbit: 7 6 5 4 3 2 1 0  
 Punkt: 1 2 3 4 5 6 7 8

Normalerweise ist der Bildhintergrund bzw. Bildspeicher gelöscht (weiß). Mit Hilfe dieser Sequenz kann man den Bildhintergrund beschreiben. Es handelt sich hierbei um keine echte Objektsequenz, da die Bildsequenzen direkt im Bildspeicher landen und somit Objektattribute nicht anwendbar sind. Dafür kann aber der gesamte Bildbereich ohne 64K-Begrenzung beschrieben werden, ohne dafür Arbeitsspeicher zu benötigen. Bei sofortiger Bearbeitung des Datensatzes muss im Eingabespeicher nur 1 Bildzeile zwischengespeichert werden.

Während Objekte normalerweise in den Bildbereich geodert werden, überschreiben Bild-Sequenzen den Bildspeicher. Aus diesem Grund sind Bild-Sequenzen vor Objekten an den Drucker zu übertragen.

Je ESC-Sequenz wird nur 1 Bildzeile übertragen. Mit jeder Sequenz wird automatisch die nächste Bildzeile angewählt und beschrieben. Die max. Anzahl an Bildsequenzen ist mit der Bildhöhe identisch. Zuviel Bild-Sequenzen werden mit einer WARNUNG-Meldung ignoriert.

Die Anzahl Bilddaten je Sequenz errechnet sich aus der Bildbreite / 8. Kommastellen werden aufgerundet. Die Bildbreite ist identisch mit der Kartenbreite, sofern explizit kein anderer Bildbereich definiert wurde. Falsche Anzahl Bilddaten führt zu unvorhersehbaren Fehlern und sollten nach Möglichkeit vermieden werden.

Objektsequenz: `<ESC> Zd.. <CR >`  
 $d$  = Anzahl leerer Bildzeilen

Zur Reduzierung der Datenmenge können Bildzeilen, in denen nichts gedruckt wird, durch `<ESC>Zd` übersprungen werden. Der Parameter  $d$  gibt die Anzahl leerer Bildzeilen an.

#### Beispiel:

Kartenbreite = 800 Dots: Je Bildzeile sind 100 Bilddaten zu senden.

Kartenhöhe = 480 Dots: Es dürfen max. 480 Bild-Sequenzen zum Drucker geschickt werden.

## 1.4 Objekte

Ein Objekt wird über eine spezielle Objektsequenz definiert. Diese Angabe muss immer als letzte Objektsequenz in einem Objektblock stehen. Die Angaben in den voranliegenden Objektsequenzen beziehen sich auf dieses Objekt.

Der Express unterscheidet folgende Objekte:

`<ESC> T ...` TEXT-Objekt  
`<ESC> B ...` BARCODE-Objekt  
`<ESC> L ...` LOGO-Objekt

Nachfolgend sind die einzelnen Objekte detailliert beschrieben.



## 1.4.1 Text - Objekt

<b>Objektsequenz:</b>	<code>&lt;ESC&gt; Tfonttyp ; Textdaten [CR]</code>
<i>fonttyp</i>	Textbeispiele:
<b>COURI06F</b>	Courier 06 fett
<b>COURI08F</b>	Courier 08 fett
<b>COURI10F</b>	Courier 10 fett
<b>COURI12F</b>	Courier 12 fett
<b>COURI14F</b>	Courier 14 fett
<b>ARIAL08F</b>	Arial 08 fett
<b>ARIAL09F</b>	Arial 09 fett
<b>ARIAL10F</b>	Arial 10 fett
<b>ARIAL12F</b>	Arial 12 fett
<b>ARIAL14F</b>	Arial 14 fett
<b>ARIAL16F</b>	Arial 16 fett
<b>ARIAL18F</b>	Arial 18 fett

Text-Objekte werden definiert durch Angabe des Zeichenfonts und des betreffenden Zeichenstrings. Bei falscher oder fehlender Font-Angabe wird als Ersatztyp Courier 08 fett selektiert.

Über das Menü ZEICHENSATZ/TABELLE kann zwischen ANSI-, ASCII- oder multilingualer Zeichensatztable gewählt werden.

**Beispiel:**

```
<ESC>TARIAL18F;Textstring
```

## 1.4.2 Logo - Objekt

<b>Objektsequenz:</b>	<code>&lt;ESC&gt; L d..Breite ; d..Höhe ; I ; HH<sub>Logodaten</sub> &lt;CR &gt;</code>
<i>d..Breite</i> :	Logo-Breite : Anzahl Dots in X-Richtung
<i>d..Höhe</i> :	Logo-Höhe : Anzahl Dots in Y-Richtung
<b>I</b>	Logotyp (kleines L)
<i>HH<sub>Logodaten</sub></i>	Logodaten in Binärform
	Jedes Bit eines Bytes repräsentiert 1 Punkt
	Bit = 0: Punkt nicht drucken, = 1: Punkt drucken
	Datenbit: 7 6 5 4 3 2 1 0
	Punkt: 1 2 3 4 5 6 7 8
<code>&lt;CR&gt;</code>	muss direkt hinter den Logodaten stehen !!

Als Logo wird ein frei programmierbares Bild verstanden, dessen Punktmuster (Bitmap) dem Drucker als Daten übergeben werden. Ein Logo-Objekt kann wie andere Objekte auch durch die entsprechenden Objektsequenzen in Form und Position näher festgelegt werden.

**Beispiel:**

Logo	Bit: 7654 3210	Hexwert	dezimal
... n ...	0000 1000	08	8
... n ...	0000 1000	08	8
nn . n ...	1100 1000	C8	200
. . n . n . .	0010 1000	28	40
. . . n . . . .	0001 0000	10	16



---

ESC-Sequenzen

```
<STX>  
<ESC>G50<ESC>I35<ESC>R0<ESC>C4<ESC>D4  
<ESC>L8;5;l;08h 08h C8h 28h 10h<CR>  
<EOT>
```

Logo-Objektsequenz in BASIC-Notation:

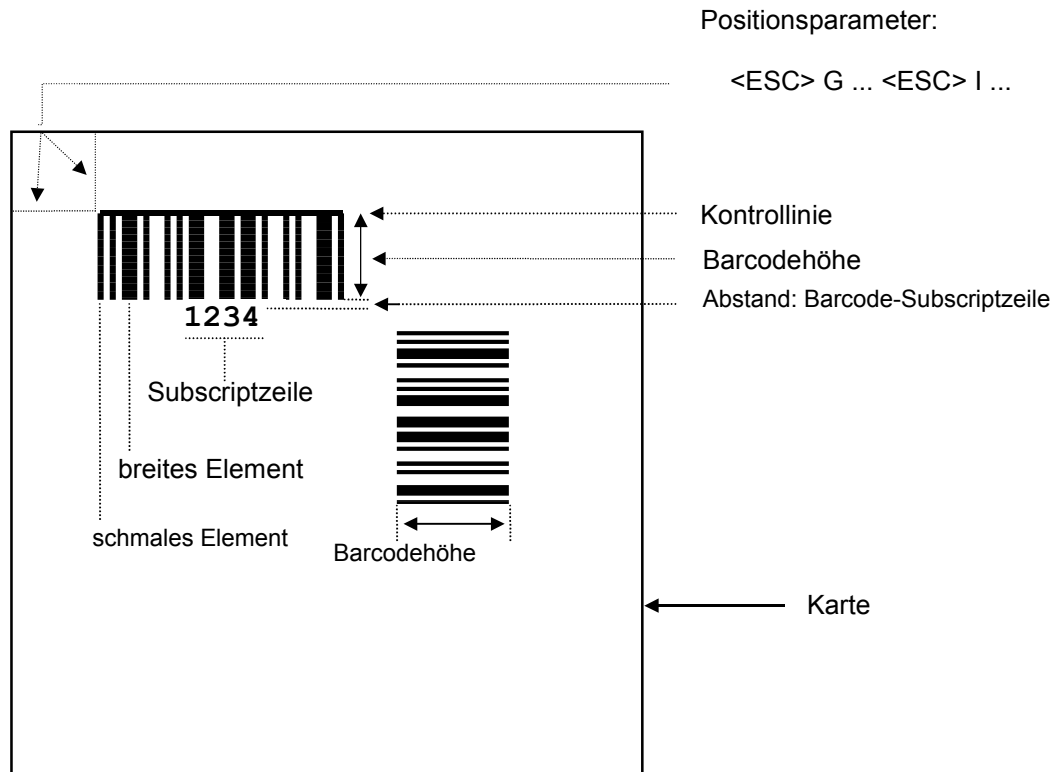
```
CHR$(27)+"L8;5;l;" + CHR$(8) + CHR$(8) + CHR$(200) + CHR$(40) + CHR$(32) + <CR>
```



### 1.4.3 Barcode - Objekt

#### 1.4.3.1 Einführung

Barcode-Objekte können durch eine Vielzahl an Parametern in Form und Funktion gestaltet werden. In der nachfolgenden Skizze sind diese Parameter veranschaulicht.



Die Einbindung eines Barcodes erfolgt über folgende Objektsequenz:

```
<ESC> B STyp ; [ Parameter ; ... ] > Barcodedaten < CR >
```

S<sub>Typ</sub> = Name des Barcodetyps :

S <sub>Typ</sub>	entsprechender Barcode:
C_25_I	Code 2 of 5 Interleaved
C_39	Code 39
C_128	Code 128
EAN8	EAN-8
EAN13	EAN-13
EAN128	EAN-128

Als erster Parameter muss immer der gewünschte Barcodetyp eingegeben werden. Danach können weitere optionale Parameter folgen. Jede Parametereingabe beginnt mit einem Kennbuchstaben und wird mit einem Semikolon ';' abgeschlossen. Der Beginn der Barcodedaten wird mit dem Zeichen '>' angezeigt.

#### Optionale Parameter:

Hier können weitere barcodespezifische Parameter vorgegeben werden. Bei Nichteingabe wird ein Ersatzwert angenommen. Die Reihenfolge der Parametereingabe (Barcode, Subscript) ist beliebig. Die Angaben in den Parametern sind entweder als Dezimalzahl oder als Textstring zu tätigen. Nachfolgend eine Liste der optionalen Parameter:

**Parameter für Barcodeteil:**

<b>H d..</b>	<b>Höhe</b> des Barcodes in Anzahl Dots (1/12 mm ) Ersatzwert: H120
<b>B d..</b>	<b>Breite</b> für ein schmales Element in Anzahl Dots Ersatzwert: B3
<b>R 2   3   5</b>	<b>Ratio</b> = Verhältnis von breitem Element zu schmalen Element 2 = 2 : 1 3 = 3 : 1 5 = 5 : 2 Ersatzwert: R3
<b>K 0   1</b>	<b>Kontrolllinie:</b> 0 = OHNE, 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z 0   1   2</b>	<b>Prüfziffer:</b> 0 = OHNE, 1   2 = MIT Prüfziffer Z1: Prüfziffer erscheint nicht in der Subscriptzeile Z2: Prüfziffer wird auch in der Subscriptzeile ausgedruckt. Bei einigen Barcodes kann optional eine zusätzliche Prüfziffer generiert werden. Barcodes, in denen die Prüfziffer fester Bestandteil ist, sind hiervon nicht betroffen. Ersatzwert: Z0
<b>S 0   a   b   c</b>	<b>Startcode</b> nur bei Code 128 und EAN 128 Ersatzwert: S0

**Parameter für Subscriptzeile:**

Für jeden Barcode kann automatisch eine Subscriptzeile ausgedruckt werden. Diese wird mittig unter den Barcode gedruckt. Über folgende Parameter kann die Subscriptzeile weiter gestaltet werden:

<b>A dddd</b>	<b>Subscriptzeile: Objekt-Attribute</b> Ersatzwert: A0000
<b>T fonttyp</b>	<b>Subscriptzeile: Zeichenfont</b> Es können alle installierten Fonts benutzt werden. Ersatzwert: COURI08f
<b>C d..</b>	<b>Subscriptzeile: Y-Faktor</b> Ersatzwert: C1
<b>D d..</b>	<b>Subscriptzeile: X-Faktor</b> Ersatzwert: D1
<b>F d..</b>	<b>Subscriptzeile: Zeichenabstand</b> (Anzahl Dots) Ersatzwert: F1
<b>P d..</b>	<b>Subscriptzeile: Abstand zum Barcode</b> (Anzahl Dots) Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. P0 würde die Subscriptzeile bündig unter das Barcodesymbol drucken. Ein negativer Wert platziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
<b>P %</b>	<b>Unterdrückung der Subscriptzeile.</b> Ersatzwert: P1

Weiterhin können zur Gestaltung und Platzierung des Barcodesymbols auch folgende Objektsequenzen verwendet werden:

<ESC>A ...	Objekt-Attribute
<ESC>G ...	X-Position
<ESC>I ...	Y-Position
<ESC>Q ...	numerische Fortschaltung
<ESC>R ...	Drehung
<ESC>V...	variables Objekt

**Beispiel: Karte mit Code 39**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_39;H70;K1;B3;R2;Z1;TARIAL20f;F2;P1;>CODE39
<EOT>
```



## 1.4.3.2 Barcode: Code 2 of 5 Interleaved

Objektsequenz: <ESC> BC_2o5_I ; [ Parameter ; ... ] > Barcodedaten [ CR ]
---

<b>Parameter :</b>	<b>Barcode:</b>
<b>H</b> 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
<b>B</b> 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
<b>R</b> 2   3   5	Ratio = Verhältnis von breitem Element zu schmalen Element 2 = 2 : 1 , 3 = 3 : 1 , 5 = 5 : 2 Ersatzwert: R3
<b>K</b> 0   1	Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z</b> 0   1   2	Generierung bzw. Ausdruck des Prüfzeichens in der Subscriptzeile 0 = kein Prüfzeichen generieren 1 = Prüfzeichen im Barcode, aber nicht in der Subscriptzeile 2 = Prüfzeichen im Barcode und in der Subscriptzeile Ersatzwert: Z0
	<b>Subscriptzeile:</b>
<b>A</b> dddd	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0000
<b>T</b> fonttyp	Zeichenfont Ersatzwert: COURI08f
<b>C</b> 1 ... 255	Y-Faktor Ersatzwert: C1
<b>D</b> 1 ... 255	X-Faktor Ersatzwert: D1
<b>F</b> 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
<b>P</b> -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert platziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
<b>P</b> %	<b>Keine</b> Subscriptzeile drucken Ersatzwert: P1

**Barcodedaten:**

**gültig:** 0 - 9 (nur numerisch)  
Code 2 of 5 Interleaved verlangt eine geradzahlige Anzahl an Ziffern. Ist die eingegebene Anzahl an Ziffern ungerade, so wird automatisch eine führende Null ergänzt.

**Anzahl:** beliebig, je nach Druckfläche

**Symbolaufbau, Symbolbreite:**

Ruhezone, Startzeichen, Nutzzeichen, [Prüfzeichen], Stopzeichen, Ruhezone

**Ruhezone:** Freifeld, Breite mind. 10-fache Elementbreite

**Start-, Stopzeichen:** werden automatisch vom Programm generiert

**Prüfzeichen:** kann automatisch vom Programm generiert werden, ist aber nicht fester Bestandteil des Codes 2 of 5 Interleaved. Das Prüfzeichen errechnet sich über eine Modulo-10 - Prüfsumme mit Gewicht 3.

Randzeichen, Nutzzeichen, Prüfzeichen beinhalten je nach Ratio folgende Anzahl an Elementbreiten:

Ratio	Randzeichen zusammen	Nutzzeichen, Prüfzeichen
R 2:1	8	7
R 3:1	9	9
R 5:2	17	16



## ESC-Sequenzen

**Beispiel: Code 2 of 5 Interleaved**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_2o5_I;H70;K0;B3;R3;Z1;TARIAL20;F2;P1;>12345678
<EOT>
```



## 1.4.3.3 Barcode: Code 39

<b>Objektsequenz:</b> <ESC> BC_39; [ Parameter ; ... ] > Barcodedaten [ CR ]
--

<b>Parameter :</b>	<b>Barcode:</b>
<b>H</b> 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
<b>B</b> 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
<b>R</b> 2   3   5	Ratio = Verhältnis von breitem Element zu schmalen Element 2 = 2 : 1 , 3 = 3 : 1 , 5 = 5 : 2 Ersatzwert: R3
<b>K</b> 0   1	Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z</b> 0   1   2	Generierung bzw. Ausdruck des Prüfzeichens in der Subscriptzeile 0 = kein Prüfzeichen generieren 1 = Prüfzeichen im Barcode, aber nicht in der Subscriptzeile 2 = Prüfzeichen im Barcode und in der Subscriptzeile Ersatzwert: Z0
<b>Subscriptzeile:</b>	
<b>A</b> <i>dddd</i>	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0000
<b>T</b> <i>fonttyp</i>	Zeichenfont Ersatzwert: COURI08f
<b>C</b> 1 ... 255	Y-Faktor Ersatzwert: C1
<b>D</b> 1 ... 255	X-Faktor Ersatzwert: D1
<b>F</b> 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
<b>P</b> -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert platziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
<b>P %</b>	<b>Keine</b> Subscriptzeile drucken Ersatzwert: P1

**Barcodedaten :**

gültig: 0 - 9 , A - Z , Leerzeichen - . \$ / + %  
Anzahl: beliebig, je nach Druckfläche

**Symbolaufbau, Symbolbreite:**

Ruhezone, Randzeichen, Nutzzeichen, [Prüfzeichen], Randzeichen, Ruhezone

*Ruhezone:* Freifeld, Breite mind. 10-fache Elementbreite

*Randzeichen:* werden automatisch vom Programm generiert



**Prüfzeichen:** kann automatisch vom Programm generiert werden, ist aber nicht fester Bestandteil des Codes 39. Das Prüfzeichen errechnet sich über eine Modulo-43 - Prüfsumme.

Randzeichen, Nutzzeichen, Prüfzeichen beinhalten je nach Ratio folgende Anzahl an Elementbreiten:  
R 2:1 = 13 Elemente, R 3:1 = 16 Elemente, R 5:2 = 29 Elemente

**Beispiel: Code 39**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_39;H70;K0;B3;R3;Z1;TARIAL20;F2;P1;>CODE39
<EOT>
```



1.4.3.4 Barcode: Code 128

**Objektsequenz:** `<ESC> BC_128; [ Parameter; ... ] > Barcodedaten [CR]`

<b>Parameter :</b>	<b>Barcode:</b>
<b>H</b> 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
<b>B</b> 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
<b>K</b> 0   1	Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z</b> 1   2	Ausdruck von Startcode und Prüfzeichen in der Subscriptzeile 1 = nicht mitausdrucken , 2 = mitausdrucken Ersatzwert: Z1
<b>S</b> 0   a   b   c	Startcode: (Startcode kann auch durch die 1.Barcodedaten festgelegt werden) a = Startcode für Zeichensatz A, ansonsten keine weiteren Konvertierungen b = Startcode für Zeichensatz B, ansonsten keine weiteren Konvertierungen c = Startcode für Zeichensatz C Das Programm konvertiert die Ziffernpaare. Ist die Anzahl der Ziffern ungerade, so wird eine führende '0' ergänzt. 0 = automatische Komprimierung: Durch Umschalten auf den jeweils günstigsten Zeichensatz wird ein Barcodesymbol mit der kürzestmöglichen Breite erzeugt. Ersatzwert: S0
<b>A</b> <i>dddd</i>	<b>Subscriptzeile:</b> Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0000
<b>T</b> <i>fonttyp</i>	Zeichenfont Ersatzwert: COURI08f
<b>C</b> 1 ... 255	Y-Faktor Ersatzwert: C1
<b>D</b> 1 ... 255	X-Faktor Ersatzwert: D1
<b>F</b> 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
<b>P</b> -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert platziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.

## ESC-Sequenzen

**P %** **Keine** Subscriptzeile drucken  
Ersatzwert: P1

**Barcodedaten :**

Code 128 unterscheidet zwischen 3 verschiedenen Zeichensätzen. Die Wahl für den entsprechenden Zeichensatz kann getroffen werden mit dem 1.Barcodezeichen oder über den Parameter Sx.

Ist das 1. Zeichen kein gültiger Startcode, so gilt die Vorgabe aus Sx und der entsprechende Startcode wird eingefügt.

Gültige Startcodezeichen:  
Startcode A : ç (135 dezimal)  
Startcode B : ê (136 dezimal)  
Startcode C : ë (137 dezimal)

**nachfolgende Zeichen:**

Zeichensatz C : 0 - 9, jedoch gerade Anzahl Ziffern, sonst wird eine führende '0' ergänzt

Zeichensatz A , B : alle ASCII-Zeichen im Bereich 32 - 127 :

0 - 9, A - Z, a - z

Leerzeichen ! " # \$ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ { | } ~ DEL

**Steuerzeichen in**

Zeichensatz A	Zeichensatz B	zu sendender ASCII-Code
FNC3	FNC3	Ç ( 128 dezimal )
FNC2	FNC2	Û ( 129 )
SHIFT	SHIFT	É ( 130 )
Code C	Code C	Â ( 131 )
Code B	FNC4	Ä ( 132 )
FNC4	Code A	À ( 133 )
FNC1	FNC1	Å ( 134 )

Automatik: wie Zeichensatz B, jedoch ohne: SHIFT , Code A, Code C

Anzahl: beliebige Anzahl Zeichen, je nach Druckfläche.

**Symbolaufbau, Symbolbreite:**

Ruhezone, Startcode, Nutzzeichen, Prüfzeichen, Stopzeichen, Ruhezone

*Ruhezone:* Freifeld, Breite mind. 10-fache Elementbreite

*Startcode:* s.o.

*Stopcode:* wird automatisch vom Programm generiert

*Prüfzeichen:* wird automatisch vom Programm generiert

Startcode, Nutzzeichen, Prüfzeichen bestehen jeweils aus 11 Elementbreiten.

Das Stopzeichen hat die 13-fache Breite des schmalen Elements.

**Beispiel: Code 128**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_128;H70;K0;B3;TARIAL20;F2;P1;>Code128
<EOT>
```



Code128

## 1.4.3.5 Barcode: EAN-8

<b>Objektsequenz:</b> <ESC> <b>BEAN8</b> ; [ Parameter ; ... ] > <i>Barcodedaten</i> [CR]
---

**Parameter :** **Barcode:**

**H** 1 ... 1000 Barcodehöhe in Anzahl Dots  
Ersatzwert: H120



## ESC-Sequenzen

**B** 1 ... 4      Breite eines schmalen Elementes in Anzahl Dots  
Ersatzwert: B3  
(Zusätzlich kann das gesamte Barcodesymbol über die Objektsequenzen  
<ESC> C... und <ESC> D... in Höhe und Breite vervielfacht werden)

**K** 0 | 1      Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie  
Ersatzwert: K0

**Subscriptzeile:**

**P** 1      Subscriptzeile mitausdrucken

**P** %      **Keine** Subscriptzeile drucken  
Ersatzwert: P1

**Barcodedaten:**

gültig:      0 - 9      (nur numerisch)

Anzahl:      Der Barcode besteht aus 8 Ziffern einschließlich Prüfziffer. Bei Eingabe von 8 Ziffern wird die 8. Ziffer als Prüfziffer kontrolliert. Bei Eingabe von 7 Ziffern wird die Prüfziffer berechnet und ergänzt.

**Symbolaufbau, Symbolbreite:**

Ruhezone, Randzeichen, 4 Nutzziffern, Trennzeichen, 4 Nutzziffern, Randzeichen, Ruhezone

*Ruhezone:*      Freifeld, Breite mind. 10-fache Elementbreite

*Randzeichen:*      werden automatisch vom Programm generiert

Ein EAN-8 - Barcodesymbol besteht aus insgesamt 67 Elementbreiten.

**Beispiel: EAN-8**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN8;H70;K0;B3;>4012345
<EOT>
```



## 1.4.3.6 Barcode: EAN-13

<b>Objektsequenz:</b> <ESC> <b>BEAN13</b> ; [ <i>Parameter</i> ; ... ] > <i>Barcodedaten</i> [CR]
---

<b>Parameter :</b>	<b>Barcode:</b>
<b>H</b> 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
<b>B</b> 1 ... 4	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3 (Zusätzlich kann das gesamte Barcodesymbol über die Objektsequenzen: <ESC> C ... , <ESC> D ... in Höhe und Breite vervielfacht werden)
<b>K</b> 0   1	Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie Ersatzwert: K0

**Subscriptzeile:**

**P** 1      Subscriptzeile mitausdrucken

**P** %      **Keine** Subscriptzeile drucken  
Ersatzwert: P1

**Barcodedaten :**

gültig:      0 - 9      (nur numerisch)

Ein Blank vor der 1.Ziffer bewirkt, dass die 1. Ziffer der Subscriptzeile links neben dem Barcode gedruckt wird. Ohne Blank werden alle 13 Ziffern unterhalb des Barcodes gedruckt.



## ESC-Sequenzen

Anzahl: Der Barcode besteht aus 13 Ziffern einschließlich Prüfziffer.  
Bei Eingabe von 13 Ziffern wird die 13. Ziffer als Prüfziffer kontrolliert.  
Bei Eingabe von 12 Ziffern wird die Prüfziffer berechnet und ergänzt.

**Symbolaufbau, Symbolbreite:**

Ruhezone, Randzeichen, 7 Nutzziffern, Trennzeichen, 6 Nutzziffern, Randzeichen, Ruhezone

*Ruhezone:* Freifeld, Breite mind. 10-fache Elementbreite

*Randzeichen:* werden automatisch vom Programm generiert

Ein EAN-13 - Barcodesymbol besteht aus insgesamt 95 Elementbreiten bzw. aus 106 Elementbreiten, wenn die 1. Ziffer links neben dem Barcode gedruckt wird.

**Beispiel: EAN-13**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN13;H70;K0;B3;> 401234567890
<EOT>
```



## 1.4.3.7 Barcode: EAN-128

<b>Objektsequenz:</b> <ESC> <b>BEAN128</b> ; [ <i>Parameter</i> ; ... ] > <i>Barcodedaten</i> [CR]
--

<b>Parameter :</b>	<b>Barcode:</b>
<b>H</b> 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
<b>B</b> 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
<b>K</b> 0   1	Kontrolllinie: 0 = OHNE , 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z</b> 1   2	Ausdruck von Startcode und Prüfzeichen in der Subscriptzeile 1 = nicht mitausdrucken , 2 = mitausdrucken Ersatzwert: Z1
<b>S</b> 0   a   b   c	Startcode: (Startcode kann auch durch die 1.Barcodedaten festgelegt werden) a = Startcode für Zeichensatz A, ansonsten keine weiteren Konvertierungen b = Startcode für Zeichensatz B, ansonsten keine weiteren Konvertierungen c = Startcode für Zeichensatz C Das Programm konvertiert die Ziffernpaare. Ist die Anzahl der Ziffern ungerade, so wird eine führende '0' ergänzt. 0 = automatische Komprimierung: Durch Umschalten auf den jeweils günstigsten Zeichensatz wird ein Barcodesymbol mit der kürzestmöglichen Breite erzeugt. Statt S0 kann dieser Parameter auch entfallen. Ersatzwert: S0
	<b>Subscriptzeile:</b>
<b>A</b> <i>dddd</i>	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0000
<b>T</b> <i>fonttyp</i>	Zeichenfont Ersatzwert: COURI08f
<b>C</b> 1 ... 255	Y-Faktor Ersatzwert: C1
<b>D</b> 1 ... 255	X-Faktor Ersatzwert: D1
<b>F</b> 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1



## ESC-Sequenzen

- P** -99 .. +99      Abstand zum Barcode  
 Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert platziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
- P %**                **Keine** Subscriptzeile drucken  
 Ersatzwert: P1

**Barcodedaten :**

Der EAN-128 ist eine Variante des Code 128. Beim EAN-128 folgt nach dem Startcode immer das Codezeichen: FNC1. Dieses Codezeichen wird automatisch vom Druckerprogramm generiert. FNC1 kann aber auch als Abschlusszeichen für variable Felder innerhalb des Barcodestrings benutzt werden. In diesem Fall muss es als Daten mit eingegeben werden.

Gültige Startcodezeichen:  
 Startcode A : ç (135 dezimal)  
 Startcode B : ê (136 dezimal)  
 Startcode C : ë (137 dezimal)

**Je nach Zeichensatz ( Sx ) sind folgende Zeichen gültig:**

Zeichensatz C : (Sc) 0 - 9

Zeichensatz A/B      alle ASCII-Zeichen im Bereich 32 - 127 :

(Sa/Sb):              0 - 9 , A - Z , a - z

Leerzeichen ! " # \$ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ { | } ~ DEL

**Steuerzeichen in**

Zeichensatz A	Zeichensatz B	zu sendender ASCII-Code
FNC3	FNC3	Ç ( 128 dezimal )
FNC2	FNC2	ü ( 129            )
SHIFT	SHIFT	é ( 130            )
Code C	Code C	â ( 131            )
Code B	FNC4	ä ( 132            )
FNC4	Code A	à ( 133            )
FNC1	FNC1	å ( 134            )

autom. Komprimierung (S0) : ASCII-Zeichen 32-127 + FNC1

Anzahl:              max. 48 Nutzzeichen, jedoch darf die Anzahl der Codezeichen (Start-, Stop-, Steuer-, Prüf- und codierte Nutzzeichen) nicht größer als 35 sein.

**Symbolaufbau, Symbolbreite:**

Ruhezone, Startcode, FNC1, Nutzzeichen, Prüfzeichen, Stopzeichen, Ruhezone

*Ruhezone:*            Freifeld, Breite mind. 10-fache Elementbreite

*Startcode:*            s.o.

*FNC1:*                dieses Codezeichen wird automatisch eingefügt

*Stopcode:*            wird automatisch vom Programm generiert

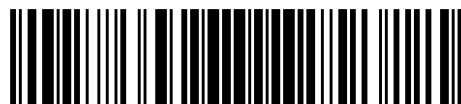
*Prüfzeichen:*        wird automatisch vom Programm generiert (Modulo 103)

Startcode, Nutzzeichen, Prüfzeichen bestehen jeweils aus 11 Elementbreiten.

Das Stopzeichen hat die 13-fache Breite des schmalen Elements.

**Beispiel: EAN 128**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN128;H70;K0;B3;TARIAL20;F2;P1;>106593â211678
<EOT>
```



10659344â211678



## 1.4.3.8 Barcode: PDF-417

<b>Objektsequenz:</b> <ESC> BPDF417; [ Parameter ; ... ] > Barcodedaten [CR]
--

**Parameter:****Bedeutung:****L****Angabe des ERROR-Levels.**

Ein PDF 417 beinhaltet einen Fehlererkennungscode, also eine Art Prüfziffer. Zusätzlich kann noch ein Fehlerkorrekturcode angehängt werden, sodass unleserliche Codewörter wiederhergestellt werden können. Je höher der ERROR-Level, desto höher der Wiederherstellungsgrad. Der Grad des ERROR-Levels sollte jedoch massvoll gewählt werden, da ein hoher Grad nicht nur das Barcodesymbol vergrößert, sondern auch die Anzahl der Nutzdaten mindert.

Empfehlenswert: 10% der Anzahl der Codewörter für die Nutzdaten sollten für die Fehlererkennung geopfert werden. Folgende ERROR-Levels sind möglich:

L0 : nur Fehlererkennung, keine Fehlerkorrektur

L1 : Fehlererkennung + 2 Codewörter für Korrektur

L2 : " + 6 "

L3 : " + 14 "

L4 : " + 30 "

L5 : " + 62 "

L6 : " + 126 "

L7 : " + 254 "

L8 : " + 510 "

(Ein PDF-Symbol kann aus max. 925 Codewörter bestehen.)

Die Angabe des ERROR-Levels kann auch in % erfolgen:

L%d : d gibt den prozentualen Anteil der Korrekturcodewörter zu den Nutzdatencodewörtern an. (Bsp.: L%10 )

Wird die L-Angabe nicht gemacht, so wird L%10 angenommen. Sollte eine feste Matrix für das Barcodesymbol vorgegeben werden, so wird der Korrekturgrad soweit erhöht, wie freie Codewörter übrig sind.

**C****Anzahl der Codespalten**

Definiert die Breite des PDF-Symbols. Ein PDF-Symbol besteht aus Randzeichen, Zeilenindikatoren und den Codewörtern für die Nutzdaten (=Codespalten).

Jedes Codewort besteht aus 17 Modulen, wobei ein Modul mindestens 1/6 mm breit sein sollte. Randzeichen und Zeilenindikatoren zusammen sind 69 Module breit.

Mögliche Angaben für PDF-Codespalten: **C1 ... C30**.

Dieser Parameter ist wahlfrei. Wird er nicht angegeben, so muss der R-Parameter definiert werden! Das Programm ermittelt die erforderliche Breite in Abhängigkeit von der Anzahl der Nutzdaten und des gewählten Korrekturgrades.

**R****Anzahl der Codezeilen**

Definiert die Höhe des PDF-Symbols. Ein PDF-Symbol besteht aus mehreren übereinander angeordneten

Barcodezeilen: 3 ... 90.

Mögliche Angaben für PDF-Codezeilen: **R3 ... R90**.

Dieser Parameter ist wahlfrei. Wird er nicht angegeben, so muss der C-Parameter definiert werden! Das Programm ermittelt die erforderliche Höhe in Abhängigkeit von der Anzahl der Nutzdaten und des gewählten Korrekturgrades.

**R, C****Definition einer Symbolmatrix**

Werden C- und R-Parameter definiert, so ist die Symbolgröße immer entsprechend der Definition gleich groß. Werden weniger Nutzdaten angegeben als im Symbol codiert werden könnten, so wird mit Füllzeichen aufgefüllt. Zu viele Nutzdaten würden allerdings einen Abbruch mit Fehler provozieren.



- T**                    **Truncated PDF**  
Darstellung des PDF-Symbols in einer verkürzten Form. Hierbei wird der rechte Zeilenindikator und das rechte Randzeichen durch ein Modul dargestellt. Die Anzahl der Module für Zeilenindikatoren und Randzeichen beträgt hier nur noch 35 Module. Diese Kurzform sollte nicht benutzt werden, wenn eine hohe Fehlerkorrektur erforderlich ist.  
T0 = normaler Modus        (= Voreinstellung)  
T1 = Truncated PDF
- W**                    **Breite eines Moduls**  
Definiert die Modulbreite in Anzahl Dots. Ein Modul sollte mindestens 1/6 mm breit sein (2 Dots bei einer 12-Dot-Thermoplastine).  
Bsp.: W2 = 2 Dots / Modul  
Voreingestellter Wert ist: W2.
- H**                    **Höhe eines Moduls (Barcodezeile)**  
Definiert die Höhe einer Barcodezeile in Anzahl Dots. Die Höhe einer Barcodezeile sollte mindestens das 3-4 fache der Modulbreite betragen.  
Bsp.: H6 = 6 Dots / Barcodezeile  
Voreingestellter Wert ist: H6.
- D**                    **PDF - Nutzdaten**  
In einem PDF-Symbol können je nach Datenart unterschiedlich viele Daten codiert werden. Hierbei werden die Daten in sogenannte Codewörter umgewandelt. Ein PDF-Symbol kann maximal 925 Codewörter beinhalten. Von diesen 925 Codewörtern muss noch die Anzahl Codewörter für die Fehlerkorrektur abgezogen werden. In einem Codewort kann codiert werden:  
2    alphanumerische Daten  
2,93 numerische Daten  
1,2   Binärdaten (Wertebereich: 0 - 255 )  
Bei einem eingestellten ERROR-Level 0 könnten also maximal in einem PDF-Symbol codiert werden:  
1850 alphanumerische Daten  
2710 numerische Daten  
1108 Binärdaten  
Die Druckersoftware sorgt für die richtige Codierung der einzelnen Datenarten. Es erfolgt eine Optimierung derart, dass die Codewortanzahl so klein wie möglich gehalten wird.

**Eingabe der Nutzdaten:**

Gültig sind alle Daten im Wertebereich: 0 - 255.

Um Konflikte mit anderen Druckersteuerzeichen auszuschließen und zur Endeerkennung der Nutzdaten müssen alle Daten, deren ASCII-Wert kleiner als 32 ist, in einer speziellen Form eingegeben werden:

  \ddd    (ddd = 3-stelliger Dezimalwert)  
  \\      (für Zeichen: "\")  
  Bsp.: \013    (= "CR" )

**PDF 417 als variables Datenfeld**

Der PDF 417 kann, wie andere Barcodes auch, als variables Barcode-Feld verwendet werden. Ein Beispiel finden Sie unter Beispiele.

**Fehlermeldungen:**

Treten Konflikte auf bei der Darstellung des PDF-Symbols infolge von fehlerhaften Parametern, so wird folgende Fehlermeldung erzeugt:

FEHLER #074  
PDF 417 - Daten

Mögliche Fehlerursachen:

- falscher ERROR-Level
- falsche Angabe für Spaltenanzahl, Barcodezeilenanzahl
- Anzahl Nutzdaten zu groß
- kein D-Parameter bzw. keine Nutzdaten
- weder C-Parameter noch R-Parameter definiert
- Anzahl Daten-Codewörter + Anzahl Korrektur-Codewörter größer 925



Nach dieser Fehlermeldung ist ein Ausdruck nicht mehr möglich.

**Beispiel:**

```
<Sx>
<ESC>G020<ESC>I0020<ESC>V1
<ESC>BPDF417;L%10;C3;T0;W2;H8;
D Dies ist ein PDF417-Barcode.\013\010
F+D Feinwerk- und Drucktechnik\013\010<CR>
<ET>
```

Hierbei handelt es sich um einen PDF417-Barcode, der zusätzlich als variables Barcodefeld ( #1 ) definiert wurde. Die Barcodebreite ist festgelegt auf 3 Codespalten. ( = 3\*17 + 69 = 120 Module ) Jedes Modul ist 2 Dots breit --> Barcodebreite = 120 \* 2 = 240 Dots.

Das Programm berechnet die einzelnen Codewörter und fügt noch 10% der Codewortanzahl für die Korrektur hinzu. So ergibt sich die Anzahl der Barcodezeilen. Die Höhe jeder Barcodezeile ist festgelegt auf 8 Dots.

**Variables Barcodefeld:**

Um einen weiteren Datensatz mit neuen PDF-Daten zu drucken, kann beispielsweise folgender variabler Steuersatz zum Drucker gesendet werden:

```
<ESC>v1; Neue PDF417-Daten.\013\010 <CR>
<ESC>#1<CR>
```

Nur die PDF-Daten sind geändert. Das Datenlayout bleibt erhalten.

Bei Eingabe des variablen Steuersatzes:

```
<ESC>v1;<CR>
```

würde kein PDF-Symbol gedruckt werden.



## 1.5 Bevorzugte Sequenzen

### 1.5.1 Statusmeldung

Die Statusanforderung ist über die V24 möglich. Der Statusreport kann jederzeit angefordert werden. Der Drucker sendet seinen aktuellen Status, sofern nicht gerade ein Ausdruck erfolgt. In diesem Fall wird der Status vor Druckbeginn der nächsten Karte gesendet. Jede Ausgabeweile wird mit <CR><LF> beendet.

**Statusanforderung**, mit <ESC>!<ENQ> (<ENQ>=05h bzw. CTRL/E):

EXP300-S02/V1.02	- Programm-Version
=00	- Drucker-Status (ss)
#0000	- Kartenanzahl
*65536	- Noch verfügbarer Eingabespeicher
/054	- Ab hier Fehlerliste falls Fehler vorliegen
/024	

Drucker-Status ss:

08	- mindestens noch 1 Karte zu drucken
10	- Karte kann entnommen werden
20	- Kartendatensatz ist gespeichert
80	- Schrittmotor für Papiertransport in Bewegung

**Kurzstatus**, mit <ESC>!<ACK> (<ACK>=06h bzw. CTRL/F):

=000/000 - Drucker-Status + Fehlercode (höchste Priorität)

Drucker-Status ss:

02	- nach Reset
08	- mindestens noch 1 Karte zu drucken
10	- Karte kann entnommen werden
20	- Kartendatensatz ist gespeichert
80	- Schrittmotor für Papiertransport in Bewegung
100	- Karte ist eingezogen

### 1.5.2 RFID-Status

Bei vorhandener RFID-Einrichtung (Option) kann die Version der Transponderelektronik abgefragt werden. z.B.: „Moby-D V4.5“

Falls keine RFID-Einrichtung vorhanden ist, erscheint die Meldung: „Not Present“.

**RFID-Status**, mit <ESC>!<BEL> (<BEL>=07h bzw. CTRL/G):

### 1.5.3 Software-Reset

Die Steuersequenz <ESC>!! bewirkt einen Drucker-Reset. Dabei wird der Drucker wieder neu gestartet. Die noch im Eingabespeicher befindlichen Daten werden gelöscht.



## 2 Anhang

### 2.1 Anhang A: Fehlermeldungen

#### 2.1.1 Fehlerebene 1 - Warnung

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #002 Steuersequenz	Falscher Wert nach <ESC>b: unzulässiges Zeichen Wert < 80	Sequenz korrigieren
WARNUNG #003 Steuersequenz	Falscher Wert nach <ESC>c: unzulässiges Zeichen Wert < 64 Wert > Dotanzahl Thermokopf	Sequenz korrigieren
WARNUNG #004 Steuersequenz	Falscher Wert nach <ESC>d: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #005 Steuersequenz	Falscher Wert nach <ESC>e: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #010 Steuersequenz	Falscher Wert nach <ESC>j: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #011 Steuersequenz	Falscher Wert nach <ESC>k: Schalter ungleich 0 oder 1 Label-Taken nicht erlaubt, da im Setup nicht angemeldet Label-Taken und Abschneider wurden gleichzeitig angemeldet Abschneider nicht erlaubt, da im Setup nicht angemeldet Transferdruck nicht erlaubt, da im Setup nicht angemeldet	<ul style="list-style-type: none"> <li>à Sequenz korrigieren</li> <li>à Service</li> <li>à Label-Taken oder Abschneider abmelden</li> <li>à Service</li> <li>à Service</li> </ul>
WARNUNG #012 Steuersequenz	Falscher Wert nach <ESC>l: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #014 Steuersequenz	Falscher Wert nach <ESC>n: Wert > 9	Sequenz korrigieren
WARNUNG #022 Steuersequenz	Falscher Wert nach <ESC>v: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #023 Steuersequenz	Falscher Wert nach <ESC>w: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #024 Steuersequenz	Falscher Wert nach <ESC>x: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #025 Steuersequenz	Falscher Wert nach <ESC>y: falsche Syntax unzulässiges Zeichen X/Y-Position = 0 Bildbreite/Bildhöhe < 16 Bildbreite > Kartenbreite	Sequenz korrigieren
WARNUNG #026 Steuersequenz	Falscher Wert nach <ESC>z: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #027 Steuersequenz	Unzulässige Steuersequenz: Falscher Kleinbuchstabe nach <ESC>	Sequenz korrigieren
WARNUNG #028 Objekt-Kennung?	Objekt-Kennung wurde nicht gefunden	Objekt mit einer Kennung über <ESC>V versehen. Objekt mit gleicher Kennung über <ESC>v neue Daten zuweisen.
WARNUNG #029 Var. Logolänge	Anzahl var. Logodaten nicht identisch mit Original-Logo	Var. Logo wird ignoriert. Var. Logo auf gleiches Format wie Original- Logo bringen.



Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #031 Objektsequenz	Falscher Wert nach <ESC>A: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #032 Objektsequenz	Falscher Wert nach <ESC>B: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #033 Objektsequenz	Falscher Wert nach <ESC>C: Y-Faktor = 0 Y-Faktor > 255	Sequenz korrigieren. Ersatzwert: einfache Zeichenhöhe
WARNUNG #034 Objektsequenz	Falscher Wert nach <ESC>D: X-Faktor = 0 X-Faktor > 255	Sequenz korrigieren. Ersatzwert: einfache Zeichenbreite
WARNUNG #036 Objektsequenz	Falscher Wert nach <ESC>F: Zeichenabstand > 255	Sequenz korrigieren. Ersatzwert: Zeichenabstand = 0
WARNUNG #037 Objektsequenz	Falscher Wert nach <ESC>G: X-Position = 0 X-Position außerhalb des Bildbereichs	Sequenz korrigieren. Die X-Position darf nicht größer als die Bildbreite sein. Ersatzwert: X-Position = 1
WARNUNG #039 Objektsequenz	Falscher Wert nach <ESC>I: Y-Position = 0 Y-Position außerhalb des Bildbereichs	Sequenz korrigieren. Die Y-Position darf nicht größer als die Bildhöhe sein. Ersatzwert: Y-Position = 1
WARNUNG #043 Objektsequenz	Falscher Wert nach <ESC>M: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #047 Objektsequenz	Falsche Angabe(n) nach <ESC>Q: Fortschaltungswert oder Fort- schaltungszyklus fehlen Fortschaltungswert > +/- 9 Fortschaltungszyklus = 0 oder >255 Schalter für Nullunterdrückung ungleich 0 oder 1 Beginn des Fortschaltungsfeldes < 1 Größe des Fortschaltungsfeldes < 0	Sequenz korrigieren
WARNUNG #048 Objektsequenz	Falscher Wert nach <ESC>R: Drehung ungleich 0/90/180/270 Grad	Sequenz korrigieren. Ersatzwert: Drehung 0 Grad
WARNUNG #050 Objektsequenz	Falscher Wert nach <ESC>T: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #052 Objektsequenz	Falsche Angabe nach <ESC>V: Die Kennung besteht aus mehr als 1 Zeichen	Sequenz korrigieren. Erlaubt ist nur 1 Zeichen als Kennung. Ersatzwert: Das Objekt erhält keine Kennung.
WARNUNG #054 Objektsequenz	Falscher Wert nach <ESC>X: unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #055 Bildzeile zuviel	Bildzeilen-Überlauf. Es wurden zuviel <ESC>Y-Sequenzen an den Drucker gesendet.	Anzahl Bild-Sequenzen überprüfen. Sie darf nicht größer als die Anzahl Bildzeilen im Bildbereich sein.
WARNUNG #056 Bildzeilenlaenge	Innerhalb <ESC>Y wurden zuviel oder zuwenig Bilddaten übertragen.	Die Anzahl Bilddaten muss mit der Bildbreite / 8 übereinstimmen.
WARNUNG #057 Objektsequenz	Unzulässige Objektsequenz: Falscher Großbuchstabe nach <ESC>	Sequenz korrigieren
WARNUNG #058 Objekt > 64Kbyte	Objekt benötigt mehr als 64KByte Speicherplatz	Objekt verkleinern oder in 2 Objekte aufteilen.
WARNUNG #060 Font-Typ	Fehlerhafte <ESC>T-Sequenz: Angewählter Font-Typ ist nicht im Drucker vorhanden.	Vorhandenen Font-Typ anwählen. Ersatztyp: COURI08F
WARNUNG #061 Barcode-Typ	Fehlerhafte <ESC>B-Sequenz: Angewählter Barcode-Typ ist nicht im Drucker vorhanden.	Anderen Barcode-Typ wählen oder Sequenz entfernen. Ersatztyp: keinen, Sequenz wird ignoriert.



## Anhang

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #062 Code 2of5 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #063 Code 39 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #064 Code 128 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #065 EAN 8 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #066 EAN 13 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #067 UPC-A - Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #068 PDF417 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #069 Barcode – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #070 Zeichen?	Zeichen kann außerhalb einer ESC-Sequenz nicht zugeordnet werden.	Zeichen wird ignoriert. Auf weitere Fehlermeldungen überprüfen und diese zuerst beheben.
WARNUNG #080 Objekt -> Bild	Objekt passt von seinen Abmes- sungen her nicht in den Bildbereich	Objekt wird ignoriert. Objekt in seiner X-/Y-Position neu plazieren
WARNUNG #081 Drehspeicher	Nicht genügend Speicher vorhanden, um das Objekt drehen zu können.	Das Objekt wird in seiner ursprünglichen Form in das Bild kopiert. Abhilfe: Bildspeicher oder Eingabespeicher verkleinern. Ansonsten Speicher erweitern (à Service)
WARNUNG #082 Arbeitsspeicher	Nicht genügend Speicher zur Aufbereitung von Objekten vorhanden.	Das Objekt wird ignoriert. Abhilfe: Bildspeicher oder Eingabespeicher verkleinern. Objektanzahl verringern oder auf var. Objekte verzichten. Ansonsten Speicher erweitern (à Service).
WARNUNG #083 Var. Bildspeicher	Nicht genügend Speicher für var. Bildspeicher vorhanden.	Alle var. Objekte werden ignoriert. Abhilfe: Bildspeicher oder Eingabespeicher verkleinern. Auf var. Objekte verzichten. Ansonsten Speicher erweitern (à Service)
WARNUNG #084 Bildspeicher	Nicht genügend Speicher für Bildbereich vorhanden.	Der Bildbereich wird in seiner Höhe entsprechend dem Speicher reduziert. Dadurch werden vermutlich nicht alle Objekte dargestellt werden. Abhilfe: Bildbereich in Abmessungen und Position manuell optimieren, Eingabespeicher verkleinern oder evtl. auf var. Objekte verzichten. Ansonsten Speicher erweitern (à Service)



## 2.1.2 Fehlerebene 2 - Fehler

Meldung	Mögliche Ursachen	Maßnahmen
FEHLER #101 Fahne drehen	Fahne steckt falsch im Eingabemodul	Fahne entfernen und mit der geraden Kante einführen
FEHLER #142 Objektsequenz	Falscher Wert nach <ESC>L: unzulässiges Zeichen	Sequenz korrigieren
FEHLER #159 Objektanzahl	Es sind max. 62 variable Objekte erlaubt.	Variable Objekte auf max. 62 reduzieren. Es ist keine weitere Druckauftrags-Bearbeitung möglich. Drucker neu initialisieren.
FEHLER #185 Eingabespeicher	Eingabespeicher ist übergelaufen.	Falls kein Schnittstellen-Handling möglich ist, Eingabespeicher vergrößern. Drucker neu initialisieren.
FEHLER #191 Logosatz...<CR>	Endekriterium der Logo <ESC>L-Sequenz nicht erkannt.	Am Ende der Logosequenz muss ein <CR> stehen. Logodaten bzw. Logoformat überprüfen. Drucker neu initialisieren.
FEHLER #192 RFID...<CR>	Endekriterium der Transponderdaten <ESC>U-Sequenz nicht erkannt.	Am Ende der Transpondersequenz muss ein <CR> stehen. Transponderdaten bzw. Transponderformat überprüfen. Drucker neu initialisieren.
FEHLER #201 Transferfolie	Transferfolie zu Ende oder Folien-Transport-Fehler.	Neue Transferfolie einlegen oder Folien- und Papiertransport prüfen. Anschließend FF-Taste betätigen. Hinweis: Die Fehlermeldung kann auch ausgelöst werden durch größere Schlaufen zwischen Druckermodul und Folienabwickler
FEHLER #213 Kartenstau	Kartenstau erkannt	Entfernen der Fahne durch betätigen der FF-Taste.
FEHLER #205 Transponder	kein Transponder vorhanden	Transponder neu einlegen oder Papiertransport überprüfen. Anschließend FF-Taste betätigen.

## 2.1.3 Fehlerebene 3 - Hardware

Meldung	Mögliche Ursachen	Maßnahmen
HARDWARE #240 Abschneider	Abschneider funktioniert nicht.	Prüfen, ob Abschneider angeschlossen ist oder blockiert.
HARDWARE #241 Druckmodul offen	Druckoberteil nicht richtig geschlossen	Druckoberteil auf korrekte Verriegelung prüfen
HARDWARE #242 Druckwalze	Abhebevorrichtung nicht funktionsfähig	Drucker neu initialisieren / Service
HARDWARE #243 IIC	Speicher IC nicht vorhanden oder defekt	Service
HARDWARE #244 WBF Eingabemodul	Eingabemodul nicht angesteckt	Drucker ausschalten, Eingabemodul anstecken, Drucker einschalten
HARDWARE #245 Transponder	Transpondereinheit nicht bereit	Service



## 2.2 Anhang B: Befehlsübersicht

### 2.2.1 Steuersequenzen

Steuerblock	= {Steuersequenz 1 [Steuersequenz n]}	n=[2...]
Steuersequenz	= {<ESC>x.... <CR>}	x=[a b ... z]

Steuersequenz	Funktion	Seite
<ESC> <b>b</b> <i>d..</i> [ <i>d..</i> ] <CR>	Bildhöhe	7
<ESC> <b>c</b> <i>d..</i> <CR>	Bildbreite	7
<ESC> <b>l</b> <i>a<sub>obj.</sub></i> ; <i>d..</i> -Breite ; <i>d..</i> -Höhe ;Daten <CR>	variables Objekt (Logo)	7
<ESC> <b>n</b> <i>d..</i> <CR>	Ländercode	8
<ESC> <b>u</b> <i>d..</i> -Offset ; <i>d..</i> -Länge ; <i>a</i> Aktion ; [ <i>HH</i> <sub>Daten</sub> ] <CR>	Transponderdaten	8
<ESC> <b>v</b> <i>a<sub>obj.</sub></i> ;Daten <CR>	variables Objekt (Text, Barcode)	8
<ESC> <b>#</b> <i>d..</i> <CR>	Druckauftrag und Kartenanzahl	7

#### Erläuterung:

*a* = 1 ASCII-Zeichen  
*d..* = Dezimalzahl  
*s* = String (max. 8-stellig)

#### Beispiele:

<ESC>Va  
 <ESC>G120  
 <ESC>Tarial18f;Textzeile



## 2.2.2 Objektsequenzen

Layoutblock	= { <STX> Objektblock 1 [Objektblock <i>n</i> ] <EOT> } <i>n</i> = [2...]
Objektblock	= { [Objektsequenz <i>n</i> ] Objekt } <i>n</i> = [1...]
Objektsequenz	= { <ESC> X ... [<CR>] } X = [A C ... Z]
Objekt	= { <ESC> B T L ... [<CR>] }

Objektsequenz	Funktion	Seite
<ESC> A <i>d..</i>	Objekt-Attribute 1: invertieren 2: um X-Achse spiegeln 4: um Y-Achse spiegeln 10: Transparenz ausschalten	9
<ESC> B <i>s<sub>Typ</sub>; Parameter ;&gt;Daten</i>	Barcode <i>s<sub>Typ</sub></i> = Barcode-Typ <i>Parameter</i> siehe ESC-Sequenzen <i>&gt;</i> = Barcode-Daten	18
<ESC> C <i>d..</i>	Y-Faktor	10
<ESC> D <i>d..</i>	X-Faktor	10
<ESC> F <i>d..</i>	Zeichenabstand	11
<ESC> G <i>d.. [;l;r;z]</i>	X-Position/Ausrichtung/Spiegelung ;l: X-Ausrichtung: linksbündig ;r: X-Ausrichtung: rechtsbündig ;z: X-Ausrichtung: zentriert	12
<ESC> I <i>d.. [;l;r;z]</i>	Y-Position/Ausrichtung/Spiegelung ;l: Y-Ausrichtung: linksbündig ;r: Y-Ausrichtung: rechtsbündig ;z: Y-Ausrichtung: zentriert	12
<ESC> L <i>d..Breite ;d..Höhe ;l ;Daten &lt;CR&gt;</i>	Logo	16
<ESC> M <i>Bildname; &lt;CR&gt;</i>	internes Logo	12
<ESC> Q <i>Parameter</i>	num. Fortschaltung	12
<ESC> R [0 90 180 270]	Drehung	13
<ESC> T <i>s<sub>Typ</sub> ;Daten</i>	Text	16
<ESC> U <i>d..Offset ; d..Länge ; HH<sub>Daten</sub> &lt;CR&gt;</i>	Transponder beschreiben	13
<ESC> V <i>a</i>	<i>a<sub>Obj</sub></i> Objekt-Name	14
<ESC> X <i>d..x1 ;d..y1 ;d..x2 ;d..y2 ;d..Breite[;d..Füllen]</i>	Linie / Rahmen	14
<ESC> Y <i>Daten &lt;CR&gt;</i>	Daten für eine Bildzeile	15
<ESC> Z <i>d</i>	Anzahl leere Bildzeilen	15

## 2.2.3 Bevorzugte Sequenzen

<ESC> ! <05h>	Status (ausführlich)	30
<ESC> ! <06h>	Kurz-Status	30
<ESC> ! <07h>	RFID-Status	30
<ESC> !!	Software-Reset	30





F+D Feinwerk- und Drucktechnik GmbH  
Kirchenstr. 38  
69239 Neckarsteinach

Tel. 06229 / 700-0  
Fax. 06229 / 700-67  
Mail: [hotline@FuDdruck.de](mailto:hotline@FuDdruck.de)  
Web: [www.FuDdruck.de](http://www.FuDdruck.de)

Copyright by F+D 3092.01.094.90.06  
Technische Änderungen vorbehalten