



Feinwerk- und Drucktechnik

Ihr kompetenter Partner für Thermodrucksysteme

# Softwarehandbuch

## Ticketdrucker

### ITP 300





# Inhaltsverzeichnis

1	ESC-Sequenzen .....	4
1.1	Erstellen von Etikettensätzen .....	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	Etikettensatzaufbau.....	5
1.1.5	Beispiel für einen Etikettensatz.....	6
1.2	Steuersequenzen.....	7
1.2.1	Kopien .....	7
1.2.2	Höhe.....	7
1.2.3	Breite.....	7
1.2.4	Transponderdaten.....	8
1.2.5	Variables Objekt: Text.....	8
1.3	Objektsequenzen .....	9
1.3.1	Objekt-Attribute .....	9
1.3.2	Y-Vergrößerung .....	9
1.3.3	X-Vergrößerung .....	10
1.3.4	Zeichenabstand .....	10
1.3.5	Positionierung.....	11
1.3.5.1	X-Koordinate.....	11
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
1.4.1.1	True Type® - Fonts .....	16
1.4.1.2	Fixed - Fonts.....	16
1.4.2	Logo - Objekt .....	17
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 2 of 5 Industrial .....	21
1.4.3.4	Barcode: Code 39 .....	21
1.4.3.5	Barcode: Code 128 .....	22
1.5	Besondere Sequenzen.....	24
1.5.1	Statusmeldung.....	24
1.5.2	RFID-Status.....	24
1.5.3	Software-Reset .....	24
1.5.4	Sensor-Status .....	24
2	Anhang.....	25
2.1	Anhang B: Befehlsübersicht .....	25
2.1.1	Steuersequenzen.....	25
2.1.2	Besondere Sequenzen.....	25
2.1.3	Objektsequenzen .....	26

Dokument: 3092.01.999.90.06

Stand: Januar 2006



## 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 Etikettensätzen

Hier wird der Aufbau eines Etikettensatzes 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 ITP 300 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 ein Etikett in verschiedener Form und Richtung gedruckt werden.

Um ein Etikett zu bedrucken, muss zunächst das gesamte Etiketten-Layout in Form von Objektblöcken definiert werden. Erst nach Erhalt des kompletten Layoutblocks kann ein Etikett 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 Etikettensatzaufbau

Der Etikettensatz hat folgenden Aufbau:

Etikettensatz	= {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>] }

#### *Etikettensatz*

Beschreibt das komplette Etiketten-Layout einschließlich Druckerkonfiguration und Druckvorgang.

#### *Steuerblock*

Beschreibt die komplette Druckerkonfiguration und Druckvorgang.

#### *Steuersequenz*

Setzt einen Steuerparameter für Druckerkonfiguration bzw. Druckvorgang.



## ESC-Sequenzen

*Objektblock*

Beschreibt das komplette Etiketten-Layout.

## Objektsequenz, Objekt

Das Etiketten-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.

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

- ⇒ kein Etikettensatz aufbereitet
- ⇒ Etikettensatz aufbereitet
- # ⇒ Anzahl Etiketten, die vom aktuellen Etikettensatz noch zu drucken sind

## 1.1.5 Beispiel für einen Etikettensatz

ESC-Sequenz	Erklärung
<STX>	<b>Beginn Layoutblock</b>
<ESC>I35<CR> <ESC>G150<CR> <ESC>R0<CR> <ESC>BC_128;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>Tarial.ttf18;Drehung 270°<CR>	Objektblock 2 (=Textdaten)  Text = ' Drehung 270° '
<EOT>	<b>Ende Layoutblock</b>
<ESC>#1<CR>	Steuersätze: (2. Steuerblock) Druckauftrag <b>Ende des Etikettensatzes</b>

Nachdem diese Daten zum Drucker übertragen worden sind, sollte der Ausdruck eines Etiketts erfolgen. Steht in der Anzeige nach dem Senden des Etikettensatzes 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]

Auf den nachfolgenden Seiten sind die einzelnen Steuersequenzen detailliert beschrieben.

### 1.2.1 Kopien

<i>Menüanwahl:</i>	LAYOUTPARAMETER / Kopien
<i>Steuersatz:</i>	<ESC>#d.<CR>
<i>Beispiel:</i>	<ESC>#10<CR>

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

Parameter:

<i>d</i>	= Kopien
<i>d +</i>	= Zwischen den Etiketten wird keine Start-/Stop-Rampe gefahren. Beisp. : <ESC>#1+, <ESC>#1+, <ESC>#1+
<i>d -</i>	= Alle Etiketten werden mit Start-/Stop-Rampe gedruckt. Beisp. : <ESC>#10-

### 1.2.2 Höhe

<i>Menüanwahl:</i>	LAYOUTPARAMETER / Höhe
<i>Steuersatz:</i>	<ESC>bd.<CR>
<i>Beispiel:</i>	<ESC>b240<CR>

Die *Höhe* legt die Höhe des Druckbereiches fest.

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

Ohne Etikettensynchronisation:

Die eingestellte Höhe bestimmt exakt den Vorschub für 1 Etikett.

Mit Etikettensynchronisation:

Die eingestellte Höhe dient zur Überwachung des Papierstaus. Eine zu klein eingegebene Höhe gegenüber dem eingelegten Etikettenformat führt zu einer unberechtigten Papierstau-Fehlermeldung. Eine zu groß eingegebene Höhe verhindert dagegen rechtzeitig eine berechtigte Papierstau-Fehlermeldung.

### 1.2.3 Breite

<i>Menüanwahl:</i>	LAYOUTPARAMETER / Breite
<i>Steuersatz:</i>	<ESC>cd.<CR>
<i>Beispiel:</i>	<ESC>c850<CR>

Die *Breite* legt die Breite des Druckbereiches fest.

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

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



## ESC-Sequenzen

## 1.2.4 Transponderdaten

<b>Steuersatz:</b>	<b>&lt;ESC&gt; u d..Offset ; d..Länge ; a Aktion ; [HH<sub>Daten</sub>] &lt;CR&gt;</b>	
	$d_{\text{Offset}}$ = 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.	

Der Drucker liest den Inhalt oder beschreibt den Transponder und gibt die entsprechenden Daten an den Host weiter.

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>u4;12;r;<CR>	Es werden ab dem 4. Block 12 Bytes ausgelesen. Rückgabe: 12 Bytes
<ESC>u4;12;w;1234567Hallo <CR>	Es werden ab dem 4. Block 12 Bytes geschrieben. Rückgabe: Anzahl der tatsächlich geschriebenen Bytes.

Transpondertyp	CODE	MFR	Anzahl Blöcke	Blockgröße in Bytes	Kapazität in Bytes
Philips Mifare Ultralight	0x04		16	4	64

## 1.2.5 Variables Objekt: Text

<b>Steuersatz:</b>	<b>&lt;ESC&gt; v a<sub>obj.</sub> ; Daten &lt;CR &gt;</b>	
	$a_{\text{obj.}}$ :	Objekt-Name
	$\text{Daten}$ :	Text-Daten
<b>Beispiel:</b>	<ESC> v a ; Text <CR> <ESC> # 1 <CR>	

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



## 1.3 Objektsequenzen

Aufbau:

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

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

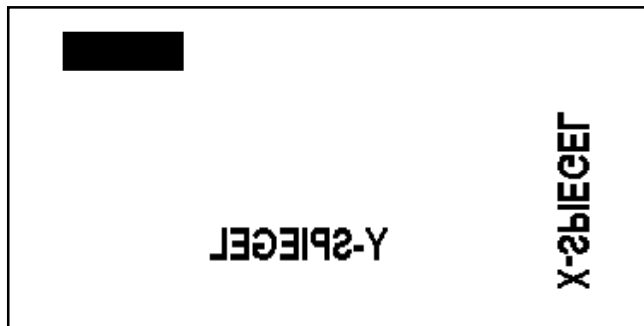
Über mehrere Objektsequenzen kann ein Objekt in Form und Position bestimmt werden. Alle Objektsequenzen, die ein Objekt beschreiben, stellen einen Objektblock dar. Die letzte Objektsequenz eines Objektblockes definiert den Objekttyp (Text, Barcode, Logo/Rahmen/Linie). Die Anzahl der Objekte ist nicht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

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

### 1.3.1 Objekt-Attribute

Objektsequenz:	<ESC> A dddd [<CR>]
ddd	= 0001 : Objekt invertieren
	= 0002 : Objekt an X-Achse spiegeln
	= 0004 : Objekt an Y-Achse spiegeln
	= 0008 : Objekt verodern
	= 0010 : Transparenz ausschalten (keine Veroderung)

Beispiel: <ESC>A0001                      Default: A0000



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

Beispiel: <ESC> A 0003                      invertiert und an X-Achse gespiegelt

### 1.3.2 Y-Vergrößerung

Objektsequenz:	<ESC> C d. [<CR>]
	$d = \underline{1} \dots 255$ : Y-Vergrößerungsfaktor

Beispiel: <ESC>C2                              Default: C1





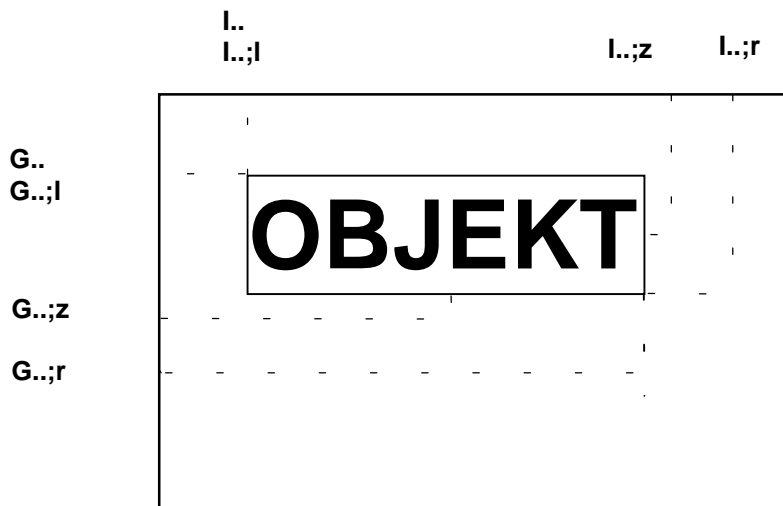
**Zeichenabstand 1**  
**Zeichenabstand 2**  
**Zeichenabstand 3**  
**Zeichenabstand 4**  
**Zeichenabstand 5**

Beispiel:

```
<STX>
<ESC>G50<ESC>I50<ESC>F1<ESC>T?arial08.sft;Zeichenabstand 1
<ESC>G50<ESC>I100<ESC>F2<ESC>T?arial08.sft;Zeichenabstand 2
<ESC>G50<ESC>I150<ESC>F3<ESC>T?arial08.sft;Zeichenabstand 3
<ESC>G50<ESC>I200<ESC>F4<ESC>T?arial08.sft;Zeichenabstand 4
<ESC>G50<ESC>I250<ESC>F5<ESC>T?arial08.sft;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 Etikettenlayouts sowohl in X- als auch in Y-Richtung zu zentrieren als auch an beiden Rändern auszurichten. Bei der Positionierung ist auf einen Mindestabstand von 3 mm 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 ... 1280	Anzahl Dots vom linken Bildrand
x-Ausrichtung =	l	linksbündig
	r	rechtsbündig
	z	zentriert

Beispiel: `<ESC>G10;z`



## 1.3.5.2 Y-Koordinate

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

Beispiel: <ESC>I10

## 1.3.6 Internes Logo

Objektsequenz:	<ESC> M <i>s</i> <sub>Logoname</sub> . <i>aaa</i> <sub>Format</sub> ; [<CR >]
Logoname:	Name des intern abgelegten Logos
Format:	PCX

Beispiel: <ESC>M fdlogo.pcx;

Mit Hilfe dieser Sequenz können Logos, die im Drucker implementiert sind, in ein Etikett 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:	<ESC> Q <i>d..w;d..z [;d..f [;d..B [;d..A]]] [&lt;CR&gt;]</i>
<i>d..w</i> = -9 ... +9	: Fortschaltungswert
<i>d..z</i> = 1 ... 255	: Fortschaltungszyklus 1-254: Anzahl Etiketten ohne Fortschaltung 255 : Fortschaltung nach jedem Auftrag
<i>d..f</i> = 0   1	: Filter für führende Nullen ( Textobjekte ) 0 : führende Nullen drucken 1 : führende Nullen unterdrücken
<i>d..B</i> = 1 ...	: Beginn des Fortschaltungsfeldes Position der 1.Ziffer des Fort.feldes
<i>d..A</i> = 0 ...	: Feldgröße: Anzahl Ziffern = 0 : bis Ende Text-, Barcodestring : > 0 : Anzahl der relevanten Ziffern

Beispiel: <ESC>Q1;1

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
<ESC>Q1;1	nach jedem Etikett wird der komplette Textstring um +1 fortgeschaltet. Führende Nullen werden gedruckt.
<ESC>Q-1;2;1	nach jeweils 2 Etiketten 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>Q1;1;0;4	nach jedem Etikett 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-Sequenzen

ESC-Sequenz	Erklärung
<ESC>Q1;1;0;4;3	nach jedem Etikett 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 Q | 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>T?arial08.sft;Drehung 0°
<ESC>G700<ESC>I240;z<ESC>R90<ESC>T?arial08.sft;Drehung 90°
<ESC>G396;z<ESC>I400<ESC>R180<ESC>T?arial08.sft;Drehung 180°
<ESC>G20<ESC>I240;z<ESC>R270<ESC>T?arial08.sft;Drehung 270°
<EOT>
```

## 1.3.9 Transponder beschreiben

**Objektsequenz:** <ESC> U *d*..*Offset* ; *d*..*Länge* ; *HH*<sub>Daten</sub> <CR>

*d*<sub>Offset</sub> = Offset in Blöcke: 0 ... Anzahl Blöcke

*d*<sub>Länge</sub> = Länge in Byte: 0 ... 32 Bytes

*HH*<sub>Daten</sub> = 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.

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 1.2.4)



## 1.3.10 Variable Objekte

**Objektsequenz:** `<ESC> V aobj [<CR>]`  
*a<sub>obj</sub>* = Objekt-Name : 0-9, A-Z, a-z (1 Zeichen)

Beispiel: `<ESC>V1`

Jedes Text-Objekt kann als variabel definiert werden. Dies eröffnet die Möglichkeit, außerhalb des Layoutblocks diesem Objekt neuen Inhalt zuzuweisen, ohne dass das komplette Etikett neu übertragen werden muss.

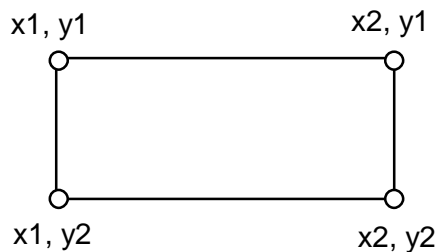
Das im Layoutblock definierte Originalobjekt gibt die maximale Feldgröße vor. Später kann dieses Objekt also maximal mit gleich vielen Daten gefüllt werden. Die Übergabe neuer Daten erfolgt mit der Steuersequenz:

`<ESC>vaobj` für Text-Objekte (siehe 1.2.5)

## 1.3.11 Linie und Rahmen

**Objektsequenz:** `<ESC> X d..x1 ; d..y1 ; d..x2 ; d..y2 ; d..Breite [ ; d..Füllen ] [<CR>]`  
*d<sub>..x1</sub>, d<sub>..y1</sub>* : Koordinate 1 : 1 ... Etikettenbreite  
*d<sub>..x2</sub>, d<sub>..y2</sub>* : Koordinate 2 : 1 ... Etikettenhöhe  
*d<sub>..Breite</sub>* : Linienbreite: Anzahl Dots : 1 ... Etikettenbreite  
*d<sub>..Füllen</sub>* : Rahmen füllen

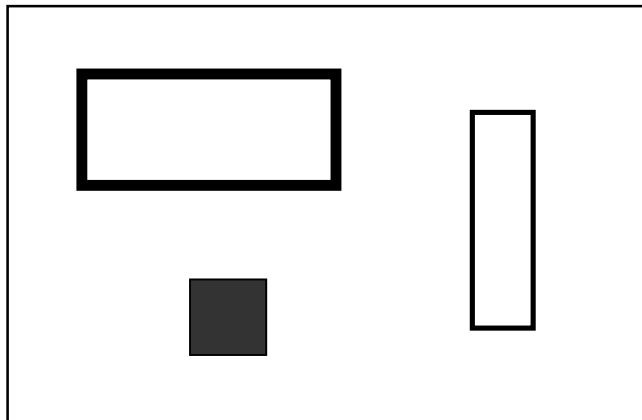
Über diese Objektsequenz können sowohl Linien als auch Rahmen erzeugt werden. Falls [x1,y1] und [x2,y2] eine diagonale Linie beschreiben, wird ein Rahmen gedruckt, ansonsten eine waagerechte oder senkrechte Linie.



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	$HH_{\text{Bilddaten}}$	<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. Bei sofortiger Bearbeitung des Etikettensatzes 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 Höhe 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 Breite ist identisch mit der Etikettenbreite, sofern explizit kein anderer Bildbereich definiert wurde. Falsche Anzahl Bilddaten führt zu unvorhersehbaren Fehlern und sollten nach Möglichkeit vermieden werden.

Objektsequenz:	<ESC> Z	$d$	=	<CR>
				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:**

Etikettenbreite = 800 Dots:

Je Bildzeile sind 100 Bilddaten zu senden.

Etikettenhöhe = 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 ITP 300 unterscheidet folgende Objekte:

&lt;ESC&gt; T ... TEXT-Objekt

&lt;ESC&gt; B ... BARCODE-Objekt

&lt;ESC&gt; L ... LOGO-Objekt

Nachfolgend sind die einzelnen Objekte detailliert beschrieben.



### 1.4.1 Text – Objekt

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

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

#### 1.4.1.1 True Type<sup>®</sup> - Fonts

<i>Objektsequenz:</i>	<b>&lt;ESC&gt; T</b> $s_{Fontname}$ . $aaa_{Typ}$ $ddd_{Größe}$ ; <i>Textdaten</i> [ <b>&lt;CR&gt;</b> ]
<i>Fontname:</i>	arial, arialbd, arialn, arialnbd,
<i>Typ:</i>	tff = True Type <sup>®</sup> - Fonts
<i>Größe:</i>	18 (2 ... 140)

Beispiel:

**<ESC>T**arial.tff18;Textstring

#### 1.4.1.2 Fixed - Fonts

<i>Objektsequenz:</i>	<b>&lt;ESC&gt; T</b> $s_{Fontname}$ [ $.aaa_{Typ}$ ]; <i>Textdaten</i> [ <b>&lt;CR&gt;</b> ]
<i>Fontname:</i>	arial08, couri08
<i>Typ:</i>	sft = Bitmap-Fonts

Beispiel:

**<ESC>T**arial08.sft;Textstring



## 1.4.2 Logo - Objekt

<b>Objektsequenz:</b>	<b>&lt;ESC&gt; L</b> <i>d..Breite</i> ; <i>d..Höhe</i> ; <b>I</b> ; <i>HH</i> <sub>Logodaten</sub> <b>&lt;CR &gt;</b>
<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</i> <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
<b>&lt;CR&gt;</b>	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
....■....	0000 1000	08	8
....■....	0000 1000	08	8
■■■.■....	1100 1000	C8	200
..■.■....	0010 1000	28	40
...■....	0001 0000	10	16

<STX>

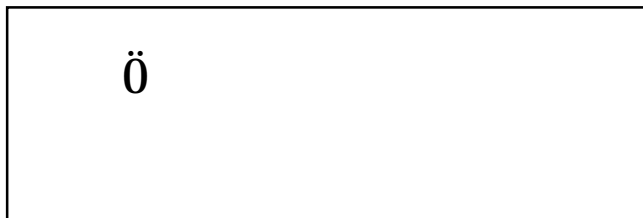
<ESC>G50<ESC>I35<ESC>R0<ESC>C4<ESC>D4

<ESC>L8;5;l;08<sub>H</sub> 08<sub>H</sub> C8<sub>H</sub> 28<sub>H</sub> 10<sub>H</sub><CR>

<EOT>

Logo-Objektsequenz in BASIC-Notation:

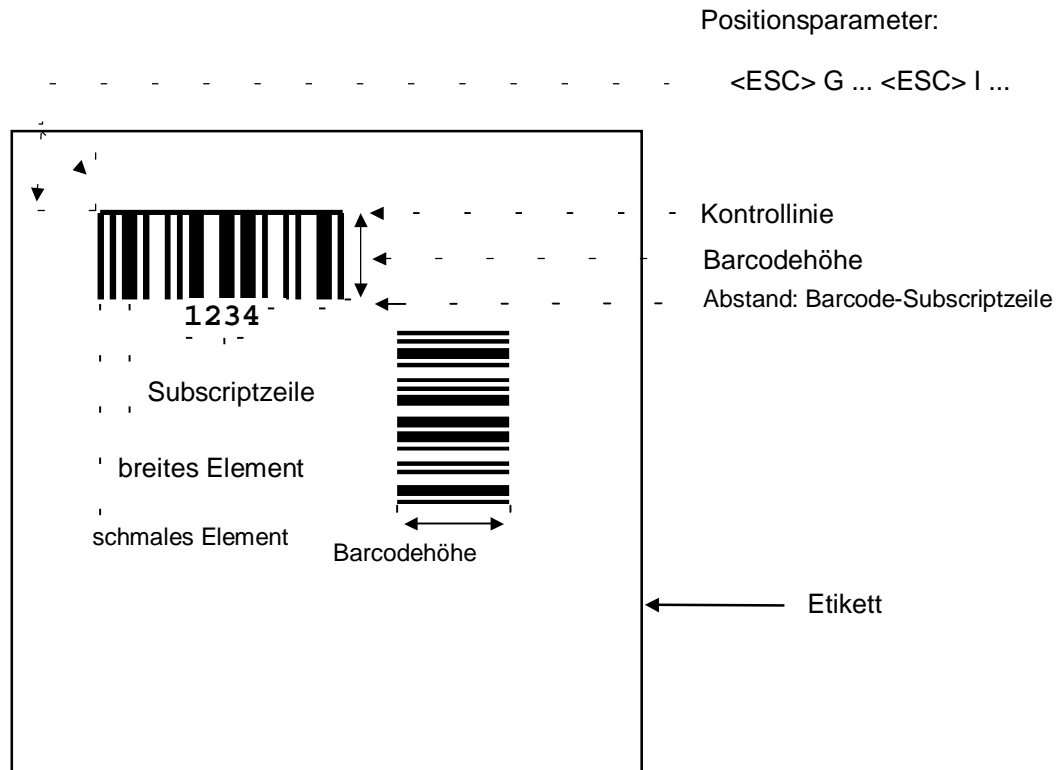
CHR\$(27)+"L?8;5;l;" + CHR\$(8) + CHR\$(8) + CHR\$(200) + CHR\$(40) + CHR\$(32) + <CR>



### 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:



## ESC-Sequenzen

**Parameter für Barcodeteil:**

<b>H</b> <i>d.</i>	<b>Höhe</b> des Barcodes in Anzahl Dots (1/12 mm ) Ersatzwert: H120
<b>B</b> <i>d.</i>	<b>Breite</b> für ein schmales Element in Anzahl Dots Ersatzwert: B3
<b>R</b> 2   3   5	<b>Ratio</b> = Verhältnis von breitem Element zu schmalen Element 2 = 2 : 1 3 = 3 : 1 5 = 5 : 2 Ersatzwert: R3
<b>K</b> 0   1	<b>Kontrolllinie:</b> 0 = OHNE, 1 = MIT Kontrolllinie Ersatzwert: K0
<b>Z</b> 0   1   2	<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</b> 0   a   b   c	<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</b> <i>dddd</i>	<b>Subscriptzeile: Objekt-Attribute</b> Ersatzwert: A0000
<b>T</b> <i>fonttyp</i>	<b>Subscriptzeile: Zeichenfont</b> Es können alle installierten Fonts benutzt werden. Ersatzwert: Courier
<b>C</b> <i>d.</i>	<b>Subscriptzeile: Y-Faktor</b> Ersatzwert: C1
<b>D</b> <i>d.</i>	<b>Subscriptzeile: X-Faktor</b> Ersatzwert: D1
<b>F</b> <i>d.</i>	<b>Subscriptzeile: Zeichenabstand</b> (Anzahl Dots) Ersatzwert: F1
<b>P</b> <i>d.</i>	<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: Etikett mit Code 39**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_39;H70;K1;B3;R2;Z1;T?arial08.sft;F2;P1;>CODE39
<EOT>
```



## 1.4.3.2 Barcode: Code 2 of 5 Interleaved

Objektsequenz: <ESC> BC_2o5_IL ; [ Parameter ; ... ] > Barcodedaten [ CR ]
--

Parameter :	Barcode:
<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: Courier
<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:



## ESC-Sequenzen

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

**Beispiel: Code 2 of 5 Interleaved**

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_2o5_I;H70;K0;B3;R3;Z1;T?arial08.sft;F2;P1;>12345678
<EOT>
```



## 1.4.3.3 Barcode: Code 2 of 5 Industrial

Objektsequenz: <ESC> BC_2o5_ID ; [ Parameter ; ... ] > Barcodedaten [ CR ]
--

Siehe Code 2 of 5 Interleaved.

## 1.4.3.4 Barcode: Code 39

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

Parameter :	Barcode:
<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: Courier
<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 :**

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;T?arial08.sft;F2;P1;>CODE39
<EOT>
```



CODE39

## 1.4.3.5 Barcode: Code 128

<b>Objektsequenz:</b> <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.Barcodaten 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>Subscriptzeile:</b>
<b>A</b> dddd	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0000
<b>T</b> fonttyp	Zeichenfont Ersatzwert: Courier



## ESC-Sequenzen

<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 :**

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	ü ( 129 )
SHIFT	SHIFT	é ( 130 )
Code C	Code C	â ( 131 )
Code B	FNC4	ä ( 132 )
FNC4	Code A	à ( 133 )
FNC1	FNC1	â ( 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;T?arial08.sft;F2;P1;>Code128
<EOT>
```



Code128



## 1.5 Besondere Sequenzen

### 1.5.1 Statusmeldung

Der Drucker sendet seinen aktuellen Status, sofern nicht gerade ein Ausdruck erfolgt. Jede Ausgabezeile wird mit <CR><LF> beendet.

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

```
ITP300-S01/V1.00  -Programm-Version
=00              - Drucker-Status (ss)
#0000           - Etikettenanzahl
*65536         - Noch verfügbarer Eingabespeicher
/054           - Ab hier Fehlerliste falls Fehler vorliegen
/024
```

Drucker-Status ss:

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

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

```
=00/000          - Drucker-Status + Fehlercode (höchste Priorität)
```

Drucker-Status ss:

```
02  - nach Reset
08  - mindestens noch 1 Etikett zu drucken
10  - Etikett kann entnommen werden
20  - Etikettendatensatz ist gespeichert
80  - Schrittmotor für Papiertransport in Bewegung
```

### 1.5.2 RFID-Status

Bei vorhandener RFID-Einrichtung (Option) kann die Version der Transponderelektronik abgefragt werden. z.B.: „CPR.M02V1.8“

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.

### 1.5.4 Sensor-Status

Anfrage: <ESC>!S                      Antwort: =0000010111000000<sub>bin</sub>

Der Status aller Sensoren wird vom Drucker an den PC gesendet.

D12-D15 sind immer 0

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Frei	Frei	Frei	Synchr oni- sation	Frei	Frei	Drucker- oberteil	Etikett entnom- men	Frei	Frei	Frei	Frei
not used	not used	not used	Index	not used	not used	printer device upper	Label- Taken	not used	not used	not used	not used



## 2 Anhang

### 2.1 Anhang B: Befehlsübersicht

#### 2.1.1 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>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</i> <sub>obj.</sub> ;Daten <CR>	variables Objekt (Text)	8
<ESC> <b>#</b> <i>d.</i> <CR> # <i>d.</i> +	Druckauftrag und Etikettenanzahl Drucken ohne Start-/Stop-Rampe	7

#### 2.1.2 Besondere Sequenzen

<ESC> ! <05h>	Status (ausführlich)	24
<ESC> ! <06h>	Kurz-Status	24
<ESC> ! <07h>	RFID-Status	24
<ESC> !!	Software-Reset	24
<ESC> ! S	Sensor-Status	24

#### Erläuterung:

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

#### Beispiele:

<ESC>Va  
 <ESC>G120  
 <ESC>Tarial.ttf18;Textzeile



## 2.1.3 Objektsequenzen

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>] }

Objektsequenz	Funktion	Seite
<ESC> A d..	Objekt-Attribute 1: invertieren 2: um X-Achse spiegeln 4: um Y-Achse spiegeln 8: verodern 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 d..	Y-Faktor	9
<ESC> D d..	X-Faktor	10
<ESC> F d..	Zeichenabstand	10
<ESC> G d.. [; ;r; z]	X-Position/Ausrichtung/Spiegelung ; : X-Ausrichtung: linksbündig ;r: X-Ausrichtung: rechtsbündig ;z: X-Ausrichtung: zentriert	11
<ESC> I d.. [; ;r; z]	Y-Position/Ausrichtung/Spiegelung ; : Y-Ausrichtung: linksbündig ;r: Y-Ausrichtung: rechtsbündig ;z: Y-Ausrichtung: zentriert	12
<ESC> L d..Breite ; d..Höhe ;   ; Daten <CR>	Logo	17
<ESC> M <i>s<sub>Logoname</sub> . aaa<sub>Format</sub> ;</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>Fontrname</sub> . aaa<sub>Typ</sub> ddd<sub>Größe</sub> ; Textdaten</i>	True Type Fonts	16
<ESC> T <i>s<sub>Fontrname</sub> [.aaa<sub>Typ</sub>] ; Textdaten</i>	Fixed Fonts	
<ESC> U d..Offset ; d..Länge ; HH <sub>Daten</sub> <CR>	Transponder beschreiben	13
<ESC> V a	<i>a<sub>Obj</sub></i> Objekt-Name	14
<ESC> X d..x1 ; d..y1 ; d..x2 ; d..y2 ; d..Breite [ ; d..Füllen ]	Linie / Rahmen	14
<ESC> Y <i>Daten</i> <CR>	Daten für eine Bildzeile	15
<ESC> Z d	Anzahl leere Bildzeilen	15





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 3092.01.999.90.06  
Technische Änderungen vorbehalten