NOVEXX SOLUTIONS

PROGRAMMIER-HANDBUCH

Easy Plug

Ausgabe 18 - 6/2026

Inhalt

Was ist Easy Plug? - 6

Hinweise zur Kommandobeschreibung - 7

- Gültigkeit - 7
- Aufbau eines Druckauftrags - 7
- Vereinbarungen - 9
- Variable Datenfelder - 9
- Eingabefelder - 10

Kommandoreferenz - 11

- Sofortkommandos - 11
  - #!An - Schnittstelle aktivieren - 11
  - #!CA - Easy Plug alles löschen - 11
  - #!CF - Format löschen - 11
  - #!D - Single-Start auslösen - 12
  - #!H - Kopftemperatur einstellen - 12
  - #!HP - Druckkopf-Andruck einstellen - 13
  - #!Pn - Interface deaktivieren - 15
  - #!PC - Start Offset einstellen - 15
  - #!PG - Parameter auslesen - 15
  - #!SP - Drucker stoppen - 19
  - #!SR - Drucker starten - 19
  - #!Xn - Statusrückmeldung - 19
  - #!XC - Rückmeldung Pharmacy Code - 20
  - #!XMn - Diagnose Dump / Maschinenstatus auslesen - 21
- B - 22
  - #BOF - Warteschlange Aus - 22
  - #BON - Warteschlange ein - 22
  - #BR - Druckauftrag unterbrechen - 22
- C - 25
  - #CBF - Barcode Codablock F - 25
  - #CFN - Code 49 - 26
  - #CF - Datei löschen - 28
  - #CG - Zeichenabstand verändern - 28
  - #CIM - Schnitt - 29
  - #CW - Schnittbreite - 30
- D - 30
  - #DC - Löschen aller Download-Logos - 30
  - #DF - Download einer Datei - 31
  - #DK - Download eines Logos - 32
  - #DM - Download von Monatsnamen - 32
  - #DO - Löschen eines Download-Logos - 33
- E - 34
  - #EMU - 300 dpi Emulation - 34
  - #ER - Start Etikettenformat - 34

F - 37
#FC - Materialvorschub mit Schnitt - 37
#FD - Feldausrichtung / -optionen - 37
#FF - Materialvorschub - 38
#FO - Easy Plug-Datei einlesen - 38
#FW - Folienbreite - 40
G - 40
#G - Kommandoabschluss - 40
H - 42
#HP - Druckkopf-Andruck einstellen - 42
#HV - Druckkopftemperatur einstellen - 42
I - 44
#IDM - Data Matrix Code - 44
#IM - Materialinformation - 48
J - 51
#J - Vertikale Druckposition - 51
M - 53
#M - Multiplikationsfaktor - 53
#ME - Material auswerfen - 53
#MXC - Maxicode - 54
N - 57
#N - Nationalität Zeichensatz - 57
O - 58
#OLVI - Online-Verifier initialisieren - 58
#OLVD - OLV-Grenzwert definieren - 58
P - 59
#PA - Druckbeginn verschieben - 59
#PC - Einstellen von Parameterwerten - 60
#PDF - Barcode PDF 417 - 62
#PO - Stanzen-Offset - 63
#PR - Druckgeschwindigkeit - 63
Q - 65
#Q - Druckmenge - 65
R - 66
#R - X-/Y-Offset Datenblöcke - 66
#RFC - Spezielle RFID-Befehle - 67
#RFH - Daten anfordern und zum Host senden - 67
#RFK - Transponder zerstören - 68
#RFL - Speicherbereich sperren/entsperren - 68
#RFPL - Speicherbereich permanent sperren - 69
#RFR - Daten lesen - 70
#RFW - Daten auf RFID-Etikett schreiben - 72
#RSS - GS1 DataBar & CC - 73
#RT - RFID-Daten lesen und drucken - 76
#RTC - Echtzeituhr einstellen - 78
#RX - Auswahl Stanzenerkennung - 78
S - 79
#SAZ - Aztec Definition - 79

#SB - Barcode Definition - 80
#SCF - Codablock F Definition - 81
#SDM - Data Matrix Definition - 82
#SF - Fixfont Definition - 84
#SFN - Code 49 Definition - 84
#SG - Grafik Definition - 85
#SI - Daten an Schnittstelle ausgeben - 85
#SMX - Maxicode Definition - 90
#SPF - PDF417 Definition - 92
#SQR - QR Matrix Code Definition - 93
#SRF - RFID Lese-/Schreib-Definition - 96
#SRS - GS1 Barcode Definition - 98
#SS - Speedo-Font Definition - 98
#SV - Wert einer Textvariablen ändern - 99
T - 101
#T - Horizontale Druckposition - 101
V - 102
#VC - RFID-Daten vergleichen/verifizieren - 102
#VDD - Datums-/Zeitvariable definieren - 102
#VDE - Ausdrucks-Variable definieren - 104
#VDO - Variable für OLV-Datenzugriff definieren - 106
#VDP - Zugriff auf Druckauftrags-Daten - 108
#VDS - Systemvariable definieren - 109
#VDT - Textvariable definieren - 110
#VR - RFID-Daten lesen - 112
#VTS - Standalone-Variable definieren - 112
#VW - Datenausgabe auf Ausgabeziel - 113
Y - 114
#YB - Barcode definieren und drucken - 114
#YC - Echtzeit als Text - 117
#YE - Kreis oder Ellipse definieren - 120
#YG - Ausdruck von Grafiken - 121
#YI - Logo im EPT-Format in den Bildpuffer schreiben - 122
#YIB - Logo mit Binärdaten direkt in Bildpuffer schreiben - 122
#YIR - Logo im RLE-Format direkt in Bildpuffer schreiben - 124
#YK - Logo definieren - 126
#YL - Linie definieren - 127
#YN - Textfeld - 128
#YR - Rechteck definieren - 133
#YS - Echtzeit als Barcode - 134
#YT - Textfeld definieren - 137
#YV - Variables Datenfeld - 139
Z - 142
#Z - Spiegelung - 142

Anhang - 144
Übersicht Kommandogruppen - 144
Dateioperationen - 149
Logos - 149

Hilfsprogramm „make_df.exe“ - 151
Druckerinterne Barcodes - 153
Druckerinterne Zeichensätze und Linienstile - 157
Ausdrücke - 159
Variablen - 159
Funktionen - 161
Arithmetische Funktionen - 168
Unicode - 172
Zusammenhang zwischen Ratio und Breitenfaktor von Barcodes - 175
Arabische Glyphen - 176

Was ist Easy Plug?

- Easy Plug ist eine hardwareabhängige *Kommunikations-Schnittstelle* für Etikettendrucker und Druck- & Etttikettiersysteme (nachfolgend „Drucker“) von NOVEXX Solutions. Die Kommunikation erfolgt vom Host zum Drucker über die am Drucker eingestellte Datenschnittstelle. Easy Plug ist für ein 7- oder 8-Bit Datenformat ausgelegt (ASCII).
- Easy Plug ist eine etikettenorientierte *Druckeransteuerung*, d. h. der Host-Rechner ist nach der Übertragung eines oder mehrerer Etikettenformate wieder frei für andere Aufgaben. Der Drucker produziert die gewünschte Anzahl von Etiketten selbständig.
- Easy Plug verwaltet ein Etikettenformat und speichert nachfolgende Aufträge in einem integrierten *Spooler*. Ein externer Spooler ist nicht notwendig. Durch einen zusätzlichen Spooler können Easy-Plug-Funktionen beeinträchtigt werden.
- Die Easy Plug Kommunikation ist in einzelne *Kommandos* untergliedert. Kommandoparameter können einzelne Zeichen oder beliebig lange Texte sein.

Hinweise zur Kommandobeschreibung

GÜLTIGKEIT

Die im Kapitel "Kommandoreferenz" dieses Handbuchs beschriebenen Easy Plug-Kommandos gelten für Drucker mit den aktuellen Firmwareständen gemäß folgender Tabelle:

Drucker	Firmware-Version	Bezeichnung Fw.-Gruppe
64-xx Gen. 3	6.75	64Bit Gen. 3
ALX 92x		
ALX 73x (PMA)		
DPM, PEM		
XLP 50x	7.75	
XLP 51x	MAR-V8.0	MAR
XLP 60x	BEL-V8.1 (ohne RFID) BEL-V8.T24550 (mit RFID)	BEL
XTP 604		
XPA 93x		
XDM/XPM 94x		
XPCU		

AUFBAU EINES DRUCKAUFTRAGS

Ein Druckauftrag besteht aus einem oder mehreren Easy Plug-Befehlen, die in einer Textdatei aufgelistet sind.
Im einfachsten Fall starten Sie den Druckauftrag, indem Sie die Textdatei mit den Easy Plug-Befehlen per copy-Befehl zum Drucker schicken. Verwenden Sie dazu das Terminalfenster ihres PC-Betriebssystems. Wenn sich der Drucker im Online-Modus befindet, beginnt er nach dem Interpretieren des Druckauftrags sofort zu Drucken.

Strukturbeschreibung		Kommandos (Beispiel)
1.	Drucker aktiv schalten	`#!A1`
2.	(Optional) Logo(s) senden: Falls vorgesehen, müssen die Logos vor Formatbeginn zum Drucker gesendet werden.	`#G Keine Logos vorgesehen`
3.	Druckserie vorbereiten: Materialtyp und Abmessungen, Stanzen-Offset und landesspezifischen Zeichensatz festlegen.	`#IMS70.0/85.0`
4.	Start des *Etikettenformats*	`#ERY`

Strukturbeschreibung		Kommandos (Beispiel)

5.	Format des Etiketts: Font, Barcode, Logo, Linien, Rechtecke (Art, Größe, Drehrichtung, Position).	`#J66.0#T15.0#M2/2`
		`#YT107/0///THERMO`
		`#J60.0#T20.5#M1/1`
		`#YT106/0///PRINTING-SYSTEM`
		`#J50.0#T20.5`
		`#YT104/0///The easy way`
		`#J45.0#T15.0`
		`#YT104/0///to create your labels`
		`#J25.0#T18.5`
		`#YB1/0M/7/3///123456789012`
		`#J15.0#T11.0#M1/1`
		`#YT104/0///PRICE`
		`#J15.0#T37.0#M2/2`
		`#YT106/0///120,95`
		`#J28.0#T11.0#M1/1`
		`#YT103/1///90-degree-rotation`
		`#J7.0#T51.0`
		`#YT104/2///180-degree-rotation`

6.	Abschluss des Etikettenformats (Druckmenge).	`#Q1/`

Tabelle 1: Beispiel eines Druckauftrags mit der Druckmenge 1.

Bild 1: Ausdruck des Formats aus dem Beispiel.

Zugehörige Informationen

Übersicht Kommandogruppen auf Seite 144

VEREINBARUNGEN

(leer) Leere Eingabe, keine Eingabe erforderlich, Kommandoparameter kann weggelassen werden

int Dezimalwert, ganzzahlig, absolut (Integer)

num Dezimalwert, mit Nachkommastellen, absolut (Numerisch)

name Textstring zur Benennung von Variablen – folgende Sonderzeichen dürfen nicht verwendet werden:

Zeichen	Name	ANSI	Hex
		<32	<20
	blank	32	20
!	exclam	33	21
‘’	quotedbl	34	22
#	numbersign	35	23
,	comma	44	2c
.	period	46	2e
+	plus	43	2b
-	hyphen	45	2d
*	asterisk	42	2a
/	slash	47	2f

expression Ausdruck – kann Konstanten, Variablen und Funktionen enthalten

TEXT Textstring, alphanumerisch

Kommandozeichen, kennzeichnet ein Kommando

VARIABLE DATENFELDER

Variable Datenfelder (Textfelder) werden mit Hilfe von TEXT wie folgt definiert:

```
TEXT = $n,c
```

$ Markierung

n Nummer des variablen Datenfelds [0...99]

, Trennzeichen

c Anzahl der Zeichen im variablen Datenfeld [0...255]

Beispiel:

```
#YTz/0D///$n,c#G
```

EINGABEFELDER

Eingabefelder werden erst nach dem Starten des Druckjobs mit Inhalt gefüllt.
Der Wert jedes Eingabefeldes wird nur einmal nach dem Starten des Druckjobs abgefragt. Dieser Wert gilt dann für alle Etiketten der im Druckjob festgelegten Druckmenge.

```
TEXT = $<prompt>,default
```

$	Markierung
<prompt>	Beliebiger Text, der zusammen mit dem Wert "default " im Display des Druckers angezeigt wird. Der Text kennzeichnet das zugehörige Eingabefeld und macht es von anderen Eingabfeldern unterscheidbar. Der Inhalt von "prompt" wird nicht mitgedruckt.
,	Trennzeichen
default	Legt einen Startwert für das Feld fest. Dieser Wert wird zusammen mit prompt im Drucker-Display angezeigt.

Beispiel:

```
#YT101////$<Preis:>,10,- Euro#G
```

Zur Verwendung von Eingabefeldern in Zusammenhang mit dem Standalone-Betrieb des Druckers siehe Bedienungsanleitung unter "Standalone-Betrieb".

Kommandoreferenz

SOFORTKOMMANDOS

#!An - Schnittstelle aktivieren

Das Sofortkommando #!An aktiviert die Easy Plug-Schnittstelle nach dem Einschalten des Gerätes oder nach dem Empfang eines Passiv-Kommandos #!P1.
Alle nach dem Kommando #!An ankommenden Daten werden als Easy Plug-Kommandos interpretiert.

Nach dem Einschalten muss das Kommando #!An einmal gesendet werden, sonst arbeitet der Drucker nicht.

Syntax

```
#!An
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
n	int	Eingestellte Drucker-ID-Nummer [0...31].

#!CA - Easy Plug alles löschen

Das Sofortkommando #!CA setzt Easy Plug in einen definierten Ausgangszustand zurück (CA = engl. Clear All).

- Ein laufender Druckvorgang wird nach Beendigung eines angefangenen Etiketts gestoppt und abgebrochen.
- Der Etikettenformat-Speicher wird gelöscht.
- Materialeinstellungen bleiben unverändert (Breite, Länge, Endlos/Stanze, Materialbezeichnung).
- Die Easy-Plug-Schnittstelle bleibt aktiv (d. h. die Wirkung eines vorangegangenen #!A1-Kommandos bleibt erhalten).

  Aber: Während des Löschvorgangs können für einen kurzen Zeitraum keine Daten übertragen werden (Handshake-Leitungen sind kurz deaktiviert).
- Spooler (Queue) wird gelöscht.
- Fehlernummer für #!Xn Rückgabestring wird gelöscht, falls der Fehler bereits quittiert worden ist.

Syntax

```
#!CA
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

#!CF - Format löschen

Das Sofortkommando #!CF bricht den Druck des gerade in Druck befindlichen Druckauftrags ab (CF = engl: Clear Format).

- Der laufende Druckvorgang wird nach Beendigung eines angefangenen Etiketts gestoppt.
- Der Etikettenformat-Speicher wird gelöscht. Anschließend werden die im Spooler-Buffer angefallenen Daten weiter verarbeitet.

Syntax

```
#!CF
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

#!D - Single-Start auslösen

Das Sofortkommando #!D löst genau einen Single-Start aus.

Der Startsignal-Eingang muss aktiviert sein!

Das Kommando #!D kann nicht als Startsignal im Applikatorbetrieb verwendet werden, wenn der Applikator über eine SPS gesteuert wird (64-xx mit LTSI).

Syntax

```
#!D
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

#!H - Kopftemperatur einstellen

Das Sofortkommando #!H stellt die Kopfspannung und damit die Kopftemperatur auf den gewünschten Wert n ein.

Syntax

```
#!Hn
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
n	int	Kopfspannung [0...xxx] (Werte für xxx abhängig vom Drucker, siehe Tabelle unten) 0 = niedrigste Kopftemperatur xxx = höchste Kopftemperatur

Drucker	xxx
64-xx, DPM, PEM, ALX 92x, ALX 73x	110
XLP 504/506 300 dpi	120
XLP 504 600 dpi	100

Tabelle 2: Maximalwerte für die Kopfspannung.

> VORSICHT!
> Die Druckkopftemperatur (= n-Wert) hat unmittelbaren Einfluss auf die Lebensdauer eines Druckkopfes. Es gilt: „Je höher die Temperatur, mit der der Druckkopf gefahren wird, desto niedriger ist seine Lebensdauer“. Das gilt verstärkt für HV-Werte über 100%. Deshalb beachten:
>
> ► Immer den niedrigsten n-Wert wählen, der noch ein akzeptables Druckergebnis liefert.

Zugehörige Informationen

#HV - Druckkopftemperatur einstellen auf Seite 42
Das Kommando #HV stellt die Druckkopfspannung und damit die Druckkopftemperatur auf den gewünschten Wert n ein.

#!HP - Druckkopf-Andruck einstellen

Nur anwendbar mit Maschinen die zur Firmware-Gruppe BEL gehören (siehe Kapitel Gültigkeit auf Seite 7).

Das Kommando #HP stellt den Druckkopf-Andruck auf den gewünschten Wert x ein.

Syntax

```
#!HPx
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
x	num	Druckkopf-Andruck [1.0..3.0] • 1.0 = niedriger Druckkopf-Andruck (entspricht Drehknopf-Einstellung „I“ bei älteren Maschinen) • 3.0 = hoher Druckkopf-Andruck (entspricht Drehknopf-Einstellung „III“ bei älteren Maschinen)

Beispiel

- Die Einstellung für den Druckkopf-Andruck wird aus der Parameter-Einstellung (Druck > Kopfandruck) übernommen:

```
#!A1
#IMN50/50
#ER
#T1#J1#YT109/0///HP Para#G
#Q2/
```

- Die Einstellung für den Druckkopf-Andruck wird aus dem vorangestellten HP-Kommando übernommen:

```
#!HP2.1
#ER
#T1#J1#YT109/0///HP2.1#G
#Q2/
```

- Die Einstellung für den Druckkopf-Andruck wird aus der Parameter-Einstellung (Druck > Kopfandruck) übernommen:

```
#ER
```

```
#T1#J1#YT109/0///HP Para#G
#Q1/
```

Zugehörige Informationen

#HP - Druckkopf-Andruck einstellen auf Seite 42

#!Pn - Interface deaktivieren

Das Sofortkommando #!Pn schaltet Easy Plug passiv.
Nach Empfang des Kommandos #!P1 werden alle ankommenden Kommandos ignoriert, ausgenommen das Aktiv-Kommando #!A1.

Nach #!P1 muss das Aktiv-Kommando einmal gesendet werden, sonst arbeitet der Drucker nicht.

Syntax

```
#!Pn
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
n	int	Eingestellte Drucker-ID-Nummer [0...31]

#!PC - Start Offset einstellen

Das Sofortkommando #!PC stellt den Start Offset ein (entspricht SPENDE PARAMETER > Start Offset bzw. Optionen > Spender > Start Offset).

Syntax

```
#!PC6004/<Wert>
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
<Wert>	num	[0-999.9] mm Als Dezimaltrennzeichen muss ein Punkt verwendet werden.

#!PG - Parameter auslesen

Das Sofortkommando #!PG liest Parametereinstellungen des Druckers über die eingestellte Schnittstelle aus.
Der aktuell eingestellte Parameter-Wert wird als ASCII-Text zurückgegeben, der mit <LF> (0x0a) abgeschlossen ist. Es besteht zudem die Möglichkeit, die Werte aller Parameter gesammelt auszulesen. Diese können dann auf ein anderes Gerät übertragen werden (siehe Kommando #PC).

Im Falle einer nicht vorhandenen ID wird nur <LF> zurückgegeben.

Es können auch Parameter ausgelesen werden, die nur unter bestimmten Bedingungen im Menü erscheinen.

Syntax

```
#!PG<ParameterID>#G
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
<ParameterID>		Parameter Identnummer (ID). Jeder Parameter im Menü hat eine eigene Parameter-ID.

Parameter	Wert	Bedeutung
		Alternativ kann einer der folgenden Werte eingegeben werden:
	-1	Druckerspezifische Einstellwerte (z.B. Kopfwiderstand, Sensorabgleichswerte) werden auskommentiert (durch ein vorangestelltes „*“). Druckerspezifische Einstellungen werden dadurch beim Senden an ein anderes Gerät nicht mit übertragen (Beispiel 1).
	-2	Druckerspezifische Einstellwerte (z.B. Kopfwiderstand, Sensorabgleichswerte) werden nicht auskommentiert. Die druckerspezifischen Einstellungen werden dadurch beim Senden an ein anderes Gerät zusammen mit den restlichen Einstellungen übertragen.
	-4	Zusätzliche Ausgabe von Kommentaren. Diese enthalten den Parameternamen und dessen Wert als Text. Außerdem werden zu Beginn einige Druckerdaten ausgegeben (Beispiel 2).
	-8	Zusätzliche Ausgabe von Kommentaren, die den Parametertyp und dessen zulässigen Wertebereich enthalten (Beispiel 3). Informationen zu den Parametertypen siehe unten.
	-16	Zusätzliche Ausgabe der Service Daten als Kommentar (Beisp.4).
	-30	Kombination der Werte -16, -8, -4, -2 durch Addition: (-16)+(-8)+(-4)+(-2)=(-30). Dadurch werden alle oben aufgeführten Kommentare berücksichtigt (Beispiel 5).

Beispiele:

Beispiel 1: Druckerspezifische Kommandos auskommentiert:

Kommando	Ergebnis

`#!PG-1#G`	`#!A1`
	`#PC1001/1`
	`#PC1002/8`
	`#PC1003/4.00`
	`#PC1004/4.00 *PC1024/30`
	(“*” vorangestellt)

Beispiel 2: Zusätzliche Kommentare:

Kommando	Ergebnis

`#!PG-4#G`	`#!A1`
	`#G Machine Setup for AP 5.4  300 Dpi  Version: R2.32P PE2.32C`
	`#G Serial number : 12345657890`
	`#G MAC Address : 00.0a.44.02.09.89`
	`#G Creation date : 16.06.2004 09:23`
	`#G ----------`
	`#G Printer Parameter Menu`
	`#G ----------`
	`#PC1001/1 #G Infeed no. : Nr. 1`
	`#PC1002/8 #G Inf. change spd. : 8 Inch/s`
	`#PC1003/4.00 #G Print speed : 4 Inch/s`
	`#PC1004/4.00 #G Feed speed : 4 Inch/s`
	`#PC1005/1 #G Materialtype : Punched`
	`#PC1006/35.00 #G Materiallength : 35.0 mm`
	`#PC1007/104.00 #G Materialwidth : 104.0 mm`
	`#PC1008/0.00 #G Punch offset : 0.0 mm`
	`...`

Beispiel 3: Zusätzliche Kommentare (Wertebereich):

Kommando	Ergebnis

`#!PG-8#G`	`#PC1505/4000`
	`#G <i: 1024 - 65535 (1)>`
	`#PC1506/0`
	`#G <d>`
	`#G 0 = Automatisch`
	`#G 1 = 10M Halbduplex`
	`#G 2 = 10M Vollduplex`
	`#G 3 = 100M Halbduplex`
	`#G 4 = 100M Vollduplex`
	`#PC1508/novexx`
	`#G <s: 16>`
	`...`

Beispiel 4: Zusätzlich Servicedaten als Kommentar:

Kommando	Ergebnis

`#!PG-16#G`	`…`
	`#G ----------`
	`#G Netzteil Daten`
	`#G ----------`
	`#G #G Typ : HME PSupply 450`
	`#G #G Version : H0.40 F1.32`
	`#G #G Serien Nummer : 02510012`
	`…`

Beispiel 5: Alle Kommentare:

Kommando	Ergebnis

`#!PG-30#G`	`#!A1`
	`#G Maschinen Setup für AP 5.4  300 Dpi  Version: R2.32P PE2.32C`
	`#G Serien Nummer :  12345657890`
	`#G MAC Adresse :  00.0a.44.02.09.89`
	`#G Erstellungsdatum :  16.06.2004 10:33`
	`#G ----------`
	`#G Drucker Parameter Menü`
	`#G ----------`
	`#PC1001/1 #G  Einzug-Nr. : Nr: 1`
	`#G  <i: 1 - 4 (1)>`
	`#PC1002/8 #G  Einzugs. Geschw. :  8 Inch/s`
	`#G  <i: 4 - 10 (1)>`
	`#PC1003/4.00 #G  Druckgeschwind. :  4 Inch/s`
	`#G  <f: 2.0 - 6.0 (1.0)>`
	`#PC1004/4.00 #G  Vorschubgeschw. :  4 Inch/s`
	`#G  <f: 2.0 - 6.0 (1.0)>`

Parametertypen

Die Drucker „kennen“ 4 Arten von Parametern. Der Einstellbereich bzw. die Zuordnung der Parameter kann abgefragt werden.

Parametertyp	Darstellung	Werte
Integer Parameter (Ganzahliger Wert, der zwischen Minimal- und Maximalwert liegen muss)	`<i: 1024 - 65535 (1)>`	Minimaler Wert: 1024 Maximaler Wert: 65535 Schrittweite: 1
Float Parameter (Wert mit Nachkommastellen, der zwischen Minimal- und Maximalwert liegen muss)	`<f: 0.0 - 999.9 (0.1)>`	Minimaler Wert: 0.0 Maximaler Wert: 999.9 Schrittweite: 0.1
String Parameter	`<s: 16>`	Maximale Stringlänge: 16 Zeichen
Diskreter Parameter (Ganzahliger Wert, dem eine bestimmte Bedeutung zugewiesen ist)	`<d>`	0 = RS232 1 = RS422 2 = RS485

Tabelle 3: Übersicht über Parametertypen

#!SP - Drucker stoppen

Das Sofortkommando #!SP stoppt den Druck einer Druckauftrags (SP = engl. Stop Printing).

- Im Druck befindliches Etikett wird fertig gedruckt
- Druckauftrag wird angehalten
- Spooler wird nicht weiter bearbeitet
- Druckvorgang wird erst nach Empfang eines Startkommandos #!SR wieder gestartet

Syntax

```
#!SP
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

#!SR - Drucker starten

Das Sofortkommando #!SR startet den Druck eines gestoppten Druckauftrags.
Das durch das Stoppkommando #!SP angehaltene Druckauftrag wird mit seiner Restmenge fertig gedruckt.

Syntax

```
#!SR
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

#!Xn - Statusrückmeldung

Das Sofortkommando #!Xn fordert einen Statusreport vom Drucker an.
Fordert Statusrückmeldung über die eingestellte Schnittstelle an. Für Centronics (parallele Schnittstelle) erfolgt die Übertragung im Nibble-Mode.

Hinweise zur Statusabfrage über serielle Schnittstelle:

- Rückmeldung über RS232 nur dann, wenn der CTS-Pin auf TRUE gesetzt ist
- Rückmeldung erfolgt mit gleichen Parametern wie Empfangskanal (gleiche Baudrate, Parity, Anzahl Stop-Bits, Datenbits)
- Rückmeldung erfolgt ohne Handshake und ohne "Inter-Character-Delay"
- Es kann eine Verzögerungszeit eingestellt werden
- Jede Statusnummer bleibt solange in der Rückmeldung erhalten, bis sich der Status ändert, oder ein Reset oder das Kommando #!CF durchgeführt wird

Syntax

```
#!Xn
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Parameter	Wert	Bedeutung
n	int	Verzögerungszeit der Rückmeldung [0..9]
	0	0 ms

Parameter	Wert	Bedeutung
	1	100 ms
	2	200 ms
	...	
	9	900 ms

Der Drucker antwortet auf das Sofortkommando mit einem Rückmeldestring, der sich aus folgenden Zeichen zusammensetzt:

```
SaaaaAbcdMqqqqqqFeeeeeeKxxxxxxxxxxxxxxxx
```

Der Rückmelde-String ist 40 Zeichen lang (ASCII, dezimale Zahlen, ohne Leerzeichen).

Markierung	Zeichen	Wert	Bedeutung
S	aaaa	0000	Es liegt keine Statusmeldung vor
		xxxx	Letzte, 4stellige Statusnummer
A	b	0	Status wurde schon einmal mit #!Xn abgefragt
		1	Status wurde zum ersten Mal abgefragt
	c	0	Status wurde schon am Bedienfeld quittiert
		1	Status wurde angezeigt, aber noch nicht quittiert
	d	0	Kein Etikettenformat gespeichert
		1	Ein Etikettenformat wurde eröffnet, aber noch nicht mit #Q abgeschlossen
		2	Ein gültiges Etikettenformat wird gedruckt
M	qqqqqq	int	Anzahl der noch zu druckenden Etiketten
F	eeeeee		Anzahl der freien Bytes im Spooler-Buffer
K	xxxxxxxxxxxxxxxx		Firmware Versionsnummer – entspricht der Anzeige in der unteren Zeile des Displays nach dem Einschalten des Druckers

Tabelle 4: Aufschlüsselung des Rückmeldestrings.

Beispiel

Kommando	Beschreibung
`#!X0`	Statusanforderung mit 0 ms Verzögerungszeit. Beispiel Rückmeldung: `S0000A100M000000F065536KV6.37`

#!XC - Rückmeldung Pharmacy Code

Das Sofortkommando #!XC liefert die Daten des zuletzt gedruckten Pharmacy-Barcodes zurück.

Syntax

```
#!XC
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Die Rückmeldung erfolgt in der Form `xxxxxxxyyyyyyyy`, mit:

xxxxxxx	7-stellige CNK-Nummer
yyyyyyyy	8-stellige durchlaufende Nummer

Falls seit dem letzten Druckerreset noch kein Pharmacy-Code gedruckt wurde, wird `0000000000000000` zurückgeliefert.

#!XMn - Diagnose Dump / Maschinenstatus auslesen

Das Sofortkommando #!XMn löst entweder einen Diagnose Dump aus und liest Diagnosedaten aus oder es liest verschiedener Maschinenstatus aus.

Syntax

```
#!XMn#G
```

Das Kommando funktioniert nicht im Standalone-Betrieb!

Rückgabeformat: Wert + Linefeed

Parameter	Wert	Bedeutung
n	-99	Diagnose Dump wird an der Debug-Schnittstelle (normalerweise Com1) ausgegeben und im Flash gespeichert.
	-100	Diagnose Dump wird an der aktiven Datenschnittstelle ausgegeben und nicht im Flash gespeichert.
	1004	*Rückgabewert*: 0 oder 1 0 = Online 1 = Offline
	1007	*Rückgabewert*: Anzahl Druckaufträge, die zum Drucken anstehen
	1008	*Rückgabewert*: 0 oder 1 0 = Easy-Plug-Interpreter ist *nicht* aktiv 1 = Easy-Plug-Interpreter ist aktiv
	1009	*Rückgabewert*: Rest-Druckmenge des aktiven Druckauftrags (entspricht dem Rückgabewert von #!Xn)
	1010	*Rückgabewert*: 0 oder 1 0 = Eine Fehlermeldung wird gerade angezeigt 1 = Keine Fehlermeldung
	1011	*Rückgabewert*: Zuletzt oder gerade angezeigte Fehlernummer. Weitere Eingrenzung mit #!XM1010
	1013	On-/Offline-Zustand; *Rückgabewerte*: 0 = Offline 1 = Online 2 = Online gestoppt
	1015	Druckeraktivität; *Rückgabewerte*: 0 = Drucker arbeitet nicht 1 = Drucker arbeitet (Druckauftrag wird gerade interpretiert oder gedruckt)

Parameter	Wert	Bedeutung
	1201	*Rückgabewert*: Anzeigetext der oberen Displayzeile Funktioniert nur mit Druckern mit 2zeiliger Textanzeige (z.B. 64-xx), nicht mit Grafikanzeige (z. B. XLP 504)
	1202	*Rückgabewert*: Anzeigetext der unteren Displayzeile Funktioniert nur mit Druckern mit 2zeiliger Textanzeige (z.B. 64-xx), nicht mit Grafikanzeige (z. B. XLP 504)

Beispiel

```
#!XM1004#G
```

Ausgabe: “1” gefolgt von “linefeed” wird zum Host zurückgesendet, wenn der Drucker offline ist.

B

#BOF - Warteschlange Aus

Der Befehl #BOF schaltet die Warteschlange für Druckaufträge nach dem Empfang eines Druckauftrags aus und erst nach dem Ausdruck der gewünschten Etikettenmenge wieder ein, d. h. es kann immer nur ein Druckauftrag gesendet werden.

Syntax

```
#BOF
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

#BON - Warteschlange ein

Das Kommando #BON schaltet die Warteschlange für Druckaufträge ein. Ein vorhergehendes Sperren der Warteschlange durch das Kommando #BOF wird dadurch aufgehoben. So können mehrere Druckaufträge - auch während des Druckens - in die Warteschlange übertragen werden.

Syntax

```
#BON
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

#BR - Druckauftrag unterbrechen

Das Kommando #BR stoppt den Drucker vor dem folgenden Druckauftrag.

Syntax

```
#BR
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Um das Kommando sinnvoll verwenden zu können, muss am Drucker der Einzeljob-Modus aktiviert sein (SYSTEM PARAMETER > Einzeljob Modus = „Ein“; bzw. System > Druck > Einzeljob Modus = „Ein“).

Beispiele

Beispiel 1:

```
#G ------------------------------------
#G Der Drucker stoppt vor dem Druckjob.
#G ------------------------------------
#!A1
#BR
#IMN100/20
#ERN0//
#T10#J5#YT108/0///TEXT
#Q3/
```

Beispiel 2:

```
#G ---------------------------------------------
#G Der Drucker stoppt nach dem ersten Druckjob.
#G ---------------------------------------------
#!A1
#IMN100/20
#ERN0//
#T10#J5#YT108/0///
#Q3/
#G ----------------------------------
#G Drucker geht in den Stopp-Modus
#G ----------------------------------
#BR
#!A1
#IMN100/20
#ERN0//
#T10#J5#YT108/0///
#Q3/
#BR
```

C

#CBF - Barcode Codablock F

Das Kommando #CBF druckt Barcodes vom Typ *Codablock F*.

Syntax

```
#CBF/dw/s/m/c/r/vop/a/TEXT#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Das Kommando muss mit #G abgeschlossen werden!

Parameter	Wert	Bedeutung
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.
s	int	Barcode-Breite [1...30]; Voreinstellung: 1
m	int	Höhe einer Codablock-Zeile [1...100] mm; Voreinstellung: 5 mm
c	int	Spaltenzahl [4...62]; Voreinstellung: 10
r	int	Zeilenzahl [2...44]; Voreinstellung: 0 0: Die Zeilenzahl wird vom Codablock berechnet. 1: Unzulässige Angabe, führt zu einer Fehlermeldung beim Interpretieren.
v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]
a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Alphanumerischer Text mit maximal 1024 Zeichen Stringlänge. Beachten Sie die Vorgaben für den Barcode-Typ. Das Textfeld kann auch ein Eingabefeld enthalten.

Zugehörige Informationen

Eingabefelder auf Seite 10
Druckerinterne Barcodes auf Seite 153

#CFN - Code 49

Das Kommando #CFN definiert Barcode vom Typ „Code 49“ (ANSI/AIM-BC6-2000 „Uniform Symbology Specification Code 49“)

Syntax

```
#CFNm/dkx/h/s/vop/a/TEXT#G
```

Parameter	Wert	Beschreibung
m	0	Alphanumerischer Mode
	1	Append Mode
	2	Numeric Mode
	3	Group Alphanumeric Mode
	4	Alphanumeric Mode, Shift 1
	5	Alphanumeric Mode, Shift 2
	6	Reserved
	7	Automatischer Mode (Voreinstellung, empfohlene Einstellung) Im automatischen Mode bestimmt der Drucker automatisch den Start-Modus und Kodier-Modus, indem er den Text analysiert.

d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90° gedreht
	2	Barcode um 180° gedreht
	3	Barcode um 270° gedreht
k	M	Barcode mit Klarschriftzeile Die Klarschriftzeile kann über den rechten Rand des Barcodes hinausragen.
	O	Barcode ohne Klarschriftzeile (Voreinstellung)
x	J	Klarschriftzeile unter dem Barcode
	A	Klarschriftzeile über dem Barcode

h	int	Definiert die Zeilenhöhe Zeilenhöhe = (h + 1) * DRUCK PARAMETER > Barcode Multi

s	int	Barcode-Breitenfaktor [1..30]

v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Beliebiger alphanumerischer Text. Die Vorgaben für den jeweiligen Barcode sind zu beachten. Maximal zulässige Zeichenzahl: 255. Zulässige Sonderzeichen: siehe nächster Abschnitt.

Sonderzeichen in „TEXT“

Verwendung von Sonderzeichen:

- „Automatischer Mode“: Die Sonderzeichen <FNC1>, <FNC2> und <FNC3> können an beliebiger Stelle im Benutzertext platziert werden.
- Alle anderen Modi: Der Anwender für die richtige Anwendung von Sonderzeichen verantwortlich. Entsprechende Kenntnisse über den Aufbau des Barcodes sind erforderlich. Gültige Sonderzeichen zeigt die folgende Tabelle.

Zeichen	Kürzel	Bezeichnung
<	S1	Shift 1
>	S2	Shift 2
:	FNC1	Function 1
;	FNC2	Function 2
?	FNC3	Function 3
=	NS	Numeric Shift

Tabelle 5: Gültige Sonderzeichen, wenn nicht „Automatischer Mode“ gewählt ist.

Beispiele

Automatischer Modus:

```
#!A1#IMN100.0/60.0#N9#ERN0///0
#T10#J4.5#CFN/M0/6/4///<FNC1>12345<FNC3>ABCDE#G
#Q1#G
#ERN0///0
#T7.5#J4.5#CFN/M0/6/4///MULTIPLE ROWS IN CODE 49#G
#Q1#G
#ERN0///0
#T7.5#J4.5#CFN/M0/6/4/1/1/EXAMPLE 2#G
```

```
#Q2#G
```

Manueller Startmodus:

```
#!A1
#IMN100.0/100.0
#N9
#PO0
#ERN0///0
#T15#J50#CFN2/M0A/4/4///12345=>ABCDEF<S>S#G
#Q1#G
```

Zugehörige Informationen

Druckerinterne Barcodes auf Seite 153

#CF - Datei löschen

Das Kommando #CF löscht eine Datei auf der RAM-Disk oder auf einem externen Datenspeicher.

Syntax

```
#CF/f
```

Parameter	Wert	Bedeutung
f	Datei-name	Dateiname, der gelöscht werden soll. Dateinamen immer in GROSSBUCHSTABEN angeben!

Beispiel

Die Datei “ FONT222.AFF ” soll von der RAM-Disk gelöscht werden:

```
#CF/A:\FONTS\FONT222.AFF#G
```

Zugehörige Informationen

Dateioperationen auf Seite 149
Beschreibung der korrekten Pfadangabe für Dateioperationen.

#CG - Zeichenabstand verändern

Legt den Abstand zwischen zwei gedruckten Zeichen eines Textstrings fest. Der Abstand muss in Dot angegeben werden.

Syntax

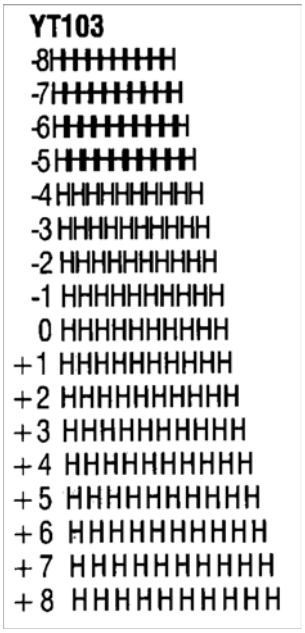
```
#CGvn
```

Parameter	Wert	Bedeutung
v	+	Zeichenabstand größer als Voreinstellung
	-	Zeichenabstand kleiner als Voreinstellung
		Der voreingestellte Zeichenabstand hängt vom verwendeten Font ab

Parameter	Wert	Bedeutung
n	int	Anzahl der Dots zwischen zwei Zeichen [-8...+8] 0: Setzt den Zeichenabstand auf den voreingestellten Wert zurück

Beispiel

Kommandofolge	Ausdruck

`#!A1#IMS82/180#ERN`
`#J175.0#T035#YT106/0///YT103`
`#J170.0#CG-8#T035#YT103/0///-8 HHHHHHHHHH#CG0`
`#J165.0#CG-7#T035#YT103/0///-7 HHHHHHHHHH#CG0`
`#J160.0#CG-6#T035#YT103/0///-6 HHHHHHHHHH#CG0`
`#J155.0#CG-5#T035#YT103/0///-5 HHHHHHHHHH#CG0`
`#J150.0#CG-4#T035#YT103/0///-4 HHHHHHHHHH#CG0`
`#J145.0#CG-3#T035#YT103/0///-3 HHHHHHHHHH#CG0`
`#J140.0#CG-2#T035#YT103/0///-2 HHHHHHHHHH#CG0`
`#J135.0#CG-1#T035#YT103/0///-1 HHHHHHHHHH#CG0`
`#J130.0#CGO#T035#YT103/0///0 HHHHHHHHHH#CG0`
`#J125.0#CG+1#T032#YT103/0///+1 HHHHHHHHHH#CG0`
`#J120.0#CG+2#T032#YT103/0///+2 HHHHHHHHHH#CG0`
`#J115.0#CG+3#T032#YT103/0///+3 HHHHHHHHHH#CG0`
`#J110.0#CG+4#T032#YT103/0///+4 HHHHHHHHHH#CG0`
`#J105.0#CG+5#T032#YT103/0///+5 HHHHHHHHHH#CG0`
`#J100.0#CG+6#T032#YT103/0///+6 HHHHHHHHHH#CG0`
`#J095.0#CG+7#T032#YT103/0///+7 HHHHHHHHHH#CG0`
`#J090.0#CG+8#T032#YT103/0///+8 HHHHHHHHHH#CG0`
`#Q1/`
	

#CIM - Schnitt

Das Kommando #CIM löst einen Schnitt des Messers bzw. des Schneidstaplers aus, sofern vorhanden.

Syntax

```
#CIM
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Das Kommando funktioniert nicht im Standalone-Betrieb!

Das Kommando funktioniert *nicht* für Drucker mit RFID-Option. Schnitte müssen hier mit dem Kommando #ER spezifiziert werden.

Kommando steht nach einem Druckauftrag: Das Etikettenmaterial wird vorgeschoben, nach dem letzten Etikett geschnitten und wieder an die Druckposition zurückgezogen.

#CW - Schnittbreite

Das Kommando #CW legt die Schnittbreite des Messers fest.

Syntax

```
#CWab
```

Gilt nur für Drucker mit angebautem Messer.

Das Kommando muss vor dem Kommando #ER stehen.

Ohne #CW-Kommando wird die maximale Schnittbreite des Messers ausgenutzt (MAX_CUT_WIDTH, siehe Tabelle).

Der Einstellwert gibt die Schnittbreite in Milimetern an. Allerdings kann es abhängig von Schnittgeschwindigkeit und Materialeigenschaften zu Abweichungen kommen. Es wird empfohlen, die Feineinstellung durch Tests zu ermitteln.

Parameter	Wert	Bedeutung
a	C	Der Drucker verwendet die in DRUCK PARAMETER > Schnittbreite bzw. in Optionen > Messer > Schnittbreite eingestellte Schnittbreite.
b	int	Wert für die Schnittbreite; Wertebereich: 0 - MAX_CUT_WIDTH

Drucker	MAX_CUT_WIDTH
64-04	106
64-05	128
64-06	160
64-08	213
XLP 504/514 mit 203 dpi	104
XLP 504/514 mit 300/600 dpi	105
XLP 506 mit 203 dpi	168
XLP 506/516 mit 300 dpi	167
XLP 604	120

Tabelle 6: Die max. Schnittbreite für unterschiedliche Drucker.

D

#DC - Löschen aller Download-Logos

Das Kommando #DC löscht alle im Download-Speicher befindlichen Logos und gibt den gesamten Speicherplatz wieder frei.

Syntax

```
#DCm
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
m	leer	Löscht sämtliche Logos auf der druckerinternen RAM-Disk.
	A	
	C	Löscht sämtliche Logos im Verzeichnis `\logos` auf der Speicherkarte gespeicherte Logos, wenn diese das Dateiformat EPT haben. Logos, die nicht im Format EPT, sondern z. B. in JPG oder BMP gespeichert sind, werden von diesem Kommando nicht gelöscht.

#DF - Download einer Datei

Laden einer Datei vom PC in den Drucker.

Syntax

```
#DF/s/f/i/b#G
```

Parameter	Wert	Bedeutung
s	leer	Datei nicht überschreiben, wenn der unter f angegebene Dateiname bereits existiert (Voreinstellung)
	N	
	O	Datei überschreiben
f		Dateiname, unter dem die geladenen Daten gespeichert werden Dateinamen immer in GROSSBUCHSTABEN angeben!
i	int	Dateigröße in Bytes [1... 2^{32}]
b		Binäre Daten der hochzuladenden Datei Um die binären Daten der zu übertragenden Datei zu erzeugen und in das Kommando einzufügen, benötigen Sie das Hilfsprogramm „make_df.exe“. make_df.exe ist auf der Dokumentations-CD im Verzeichnis „/printer/utilities“ enthalten.

Zugehörige Informationen

Dateioperationen auf Seite 149
Beschreibung der korrekten Pfadangabe für Dateioperationen.

Hilfsprogramm „make_df.exe“ auf Seite 151
Das Hilfsprogramm „Make_DF.exe“ generiert ein komplettes #DF-Kommando einschließlich der zu übertragenden binären Daten und schreibt es in eine Textdatei.

#DK - Download eines Logos

Das Kommando #DK dient zum Download eines Logos (d. h. dem Senden eines Logos vom PC zum Drucker) unter einer anzugebenden Bezugsnummer (0 bis max. 255).

Syntax

```
#DKn/m/s/s.../s#G
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Das Kommando #DK muss mit #G abgeschlossen werden.

Parameter	Wert	Bedeutung
n	int	Bezugsnummer des Logos [0...255]
m	leer	Logo wird auf die druckerinterne RAM-Disk kopiert (Voreinstellung)
	A	
	C	Logo wird auf die Speicherkarte kopiert.
		Das Logo wird jeweils in das Verzeichnis `\logos` kopiert
s	hex	Verschlüsselung einer Dot-Zeile der Logo-Matrix, hexadezimal zu je 4 Dots von links nach rechts (000 bis FFF). Zuordnung für hexadezimale Verschlüsselung: • 1 = Dot wird gedruckt • 0 = Dot wird nicht gedruckt Nicht gesetzte Dots am Zeilenende können entfallen. Verschlüsselung aller Dot-Zeilen der Logo-Matrix durch einen s-Parameter je Zeile, beginnend von unten (Zeile 1) nach oben. Für die hexadezimale Darstellung zulässig sind Großbuchstaben und Ziffern.

Beispiel

```
#DK1/A/C03/E07/F0F/FFF#G
```

Logo Nummer 1 mit nachfolgendem Aufbau wird gesendet und auf der druckerinternen RAM-Disk gespeichert.

```
Zeile 4: 1111 1111 1111 (/FFF)
Zeile 3: 1111 0000 1111 (/F0F)
Zeile 2: 1110 0000 0111 (/E07)
Zeile 1: 1100 0000 0011 (/C03)
```

Zugehörige Informationen

Logos auf Seite 149
Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

#DM - Download von Monatsnamen

Das Kommando #DM dient zum Download frei definierbarer Monatsnamen.

Syntax

```
#DM^TEXT^TEXT
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
	^	Control-Zeichen zur Trennung der einzelnen Monatsnamen
TEXT		Monatsname, frei definierbar, maximal 20 Zeichen. Der TEXT nach dem ersten Control-Zeichen ist dem Monat Januar zugeordnet, der TEXT nach dem zweiten Control-Zeichen dem Monat Februar usw. Sind weniger als 12 Monate definiert, wird der Zugriff auf den fehlenden Monat ignoriert. Das Control-Zeichen darf nicht im Text verwendet werden.

Beispiel

```
#G ----------------------------------------------------------------
#G Die Monatsnamen Jan bis Dec werden im Download Buffer
#G gespeichert (jeweils mit einem abschließenden Leerzeichen).
#G ----------------------------------------------------------------
#DM^Jan^Fev^Mar^Avr^Mai^Jui^Jul^Aou^Sep^Oct^Nov^Dec
```

#DO - Löschen eines Download-Logos

Das Kommando #DO löscht ein einzelnes, im RAM oder auf einem externen Speichermedium gespeichertes Logo.

Syntax

```
#DOn/m
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
n	int	Bezugsnummer des zu löschenden Logos [0...255]
m	leer	Logo wird auf der druckerinternen RAM-Disk gelöscht
	A	
	C	Logo wird auf dem externen Speichermedium gelöscht

Beispiel

Es wird Logo Nummer 5 gelöscht (ohne Rückfrage) und der frei werdende Speicherplatz wieder zur Verfügung gestellt:

```
#DO5
```

E

#EMU - 300 dpi Emulation

Das Kommando #EMU schaltet am XLP 504 600 dpi zwischen nativem 600dpi-Modus und 300 dpi-Emulation um.

Syntax

```
#EMUn
```

Das Kommando kann zwischen #ER und #Q beliebig oft aufgerufen werden.

Parameter	Wert	Bedeutung
n	0	*Nativer Modus*: Druckbild wird mit der Auflösung des Druckkopfes (600 dpi) generiert.
	1	*300 dpi-Emulation*: Druckbild wird so skaliert, dass der Ausdruck mit 600 dpi genauso aussieht, als wenn er mit einem 300 dpi-Druckkopf gedruckt worden wäre.

#ER - Start Etikettenformat

Das Kommando #ER markiert den Anfang eines Etikettenformats (Druckauftrags) und legt allgemeine Informationen zu diesem Druckauftrag fest.

Syntax

```
#ERxrsftw/n/b/d/TEXT
```

Parameter	Wert	Bedeutung
x	N	Kein Wechseletikett
	Y	Wechseletikett [1] nach Serie
r	int	Anzahl der gedruckten Etikettenreihen vor einem Schnitt (siehe Abb. unten).
	0	kein Schnitt (Voreinstellung)
	leer	
	1	jedes Etikett (jede Etikettenreihe) wird abgeschnitten
	2	jedes zweite Etikett (jede zweite Etikettenreihe) wird abgeschnitten
	x	jedes x-te Etikett (jede x-te Etikettenreihe) wird abgeschnitten
s	D	Dottest nach dem Ende des Druckauftrags. Nur für 64-xx, ALX 73x, ALX 92x, DPM, PEM
f		Steuert die Funktion „Foliensparen“.
	C	Thermotransfer-Druck ohne Foliensparen

1 Ein Wechseletikett ist das letzte Etikett eines Druckauftrages. Es ist etwas länger als die vorhergehenden Etiketten - das kann z.B. beim Staplereinsatz nützlich sein. Das längere Wechseletikett steht aus dem Etikettenstapel hervor und erleichtert so das Sortieren der Etiketten.

Parameter	Wert	Bedeutung
	F	Thermotransfer-Druck mit Foliensparen
	T	Thermodruck ohne Druckkopf anheben
	U	Thermodruck mit Druckkopf anheben Nur für 64-xx, ALX 73x, ALX 92x, DPM, PEM
	Z	Thermotransfer-Druck mit Turbo-Foliensparen. Zusätzlich zum normalen Foliensparen wird das Etikettenmaterial in druckfreien Bereichen mit einer höheren Geschwindigkeit als der Druckgeschwindigkeit vorgeschoben. Die Geschwindigkeit wird in DRUCK PARAMETER > Vorschubgeschw. festgelegt. Nur für 64-xx, ALX 73x, ALX 92x, DPM, PEM
t	Pnum	[P0.0-P10.0] Wartezeit in Sekunden, bevor ein neues Bitimage generiert wird.
w	leer	Standardeinstellung ohne Parameter V: Nach offline-schalten wird so lange „Bitte warten...“ angezeigt, bis die durch *Pnum* definierte Wartezeit beendet ist. Erst dann kann der Druckprozess fortgesetzt werden. Die eingestellte Verzögerung kann also nicht durch schnelles offline-online-schalten des Druckers umgangen werden.
	V	Ein Stopp durch offline-schalten des Druckers wird sofort umgesetzt, ohne das Ende der durch *Pnum* definierten Wartezeit abzuwarten. Durch schnelles offline-online-schalten kann die Wartezeit umgangen werden; in diesem Fall wird der Druck sofort fortgesetzt.

n	int	Anzahl der Etikettenbahnen die gleichzeitig gedruckt werden (siehe Abb. unten).
	0 leer	Ein Etikett pro Reihe
	1	Ein Etikett pro Reihe
	2	Zwei Etiketten pro Reihe
	x	x Etiketten pro Reihe
		Unvollständige Etikettenreihen: Falls der Druckauftrag weniger Etiketten enthält, als nötig wäre, um die letzte Reihe aufzufüllen, geschieht folgendes: • Druckauftrag wurde über Datenschnittstelle übertragen: Die Reihe wird unvollständig gedruckt, wobei mit dem Etikett an der Material-Nulllinie begonnen wird. • Druckauftrag wurde im Standalone-Betrieb gestartet: Die Reihe wird aufgefüllt und vollständig gedruckt.

b	num	Etikettenbreite (einschließlich dem Abstand zum Nachbaretikett) Der Parameter „b“ muss für mehrbahnige Etikettenmaterialien (Anzahl Etikettenbahnen > 1) angegeben werden. Die Anzahl der Etikettenbahnen mal der Etikettenbreite darf nicht größer sein, als die Materialbreite, sonst wird die überstehende Etikettenbahn nicht gedruckt.

d	num	Breite des Doppelschnitts; Einstellbereich: [0,00..5,00] mm Ohne Angabe von „d“ wird für einen Doppelschnitt der in einem vorhergehenden Druckauftrag verwendete, oder der im Parametermenü eingestellte Wert verwendet.

TEXT		Bezeichner für den Druckauftrag. Alphanumerischer Text mit maximal 255 Zeichen Länge.

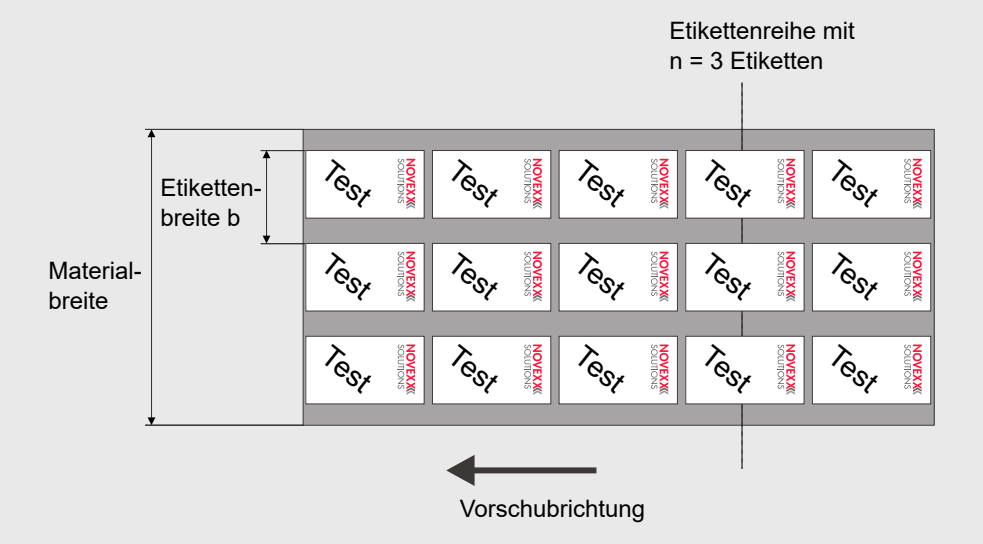


Bild 2: 3-bahniges Etikettenmaterial (n = 3).

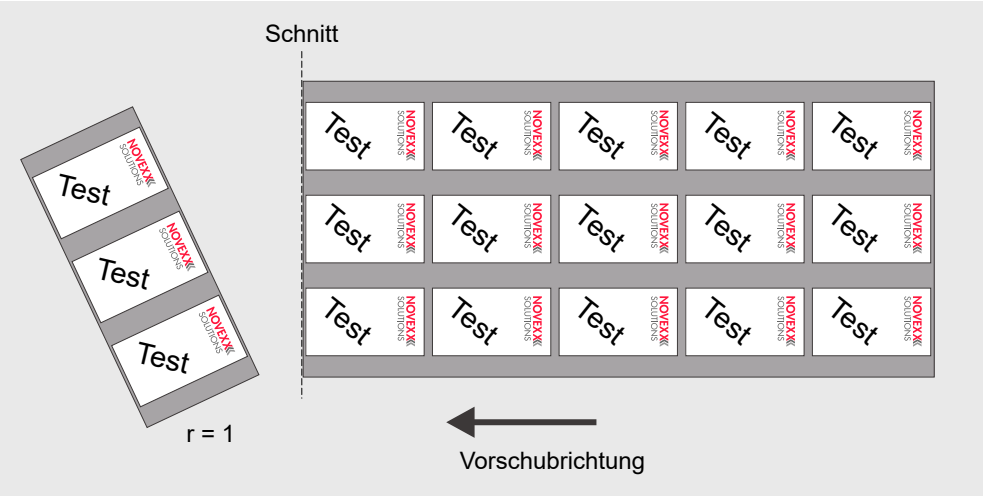


Bild 3: Beispiel: Eine gedruckte Etikettenreihe vor dem Schnitt (r = 1).

F

#FC - Materialvorschub mit Schnitt

Das Kommando #FC löst einen Materialvorschub um eine Etikettenlänge, gefolgt von einem Schnitt, aus.

```
#FC
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Das Kommando funktioniert nicht im Standalone-Betrieb!

- *Gestanztes Material*: Vorschub bis zur nächsten Stanze, Etikettenlücke oder Reflexmarke
- *Endlosmaterial*: Vorschub um definierte Etikettenlänge

#FD - Feldausrichtung / -optionen

Das Kommando #FD definiert die Ausrichtung bzw. das Erscheinungsbild eines Feldes.

Syntax

```
#FD/d/pz#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
d		Drehrichtung für *Fixfonts*:
	0	normale Schreibrichtung
	1	Text um 90° gedreht
	2	Text um 180° gedreht
	3	Text um 270° gedreht
	Dnum	[D0.0 ... D359.9] Drehrichtung für *skalierbare Fonts*
p		Ausrichtung
	L	*Linksbündig*; Das Feld wird von der Druckposition aus nach rechts aufgebaut
	M	*Mittig*; Das Feld wird beidseitig der Druckposition aufgebaut
	R	*Rechtsbündig*; Das Feld wird von der Druckposition aus nach links aufgebaut
z		Ausdruck „normal“ oder invertiert
	P	Normaler Ausdruck in Schwarz
	A	Ausdruck in weiß, d. h. der Ausdruck wird freigelassen (erfordert dunklen Hintergrund)
	E	Ausdruck mit invertiertem Bitimage, d. h. der Ausdruck wird freigelassen und der Hintergrund wird gedruckt

#FF - Materialvorschub

Das Kommando #FF löst einen Materialvorschub um eine Etikettenlänge aus. Der Vorschub erfolgt bei gestanztem Material bis zur nächsten Stanze, bei Endlosmaterial bis zur definierten Etikettenlänge.

Syntax

```
#FF
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Das Kommando funktioniert nicht im Standalone-Betrieb!

#FO - Easy Plug-Datei einlesen

Das #FO-Kommando fügt den Inhalt der angegebenen Datei an der Stelle ein, an der das Kommando steht.

Syntax

```
#FO/f#G
```

Parameter	Wert	Beschreibung
f	Datei-name	Dateiname, entsprechend den DOS-Konventionen (Laufwerk, Pfad, Dateiname, Extension) Beim Dateinamen Groß- und Kleinschreibung beachten. Die Datei muss bereits auf dem angegebenen Laufwerk gespeichert sein.

Funktionsbeschreibung

Der Vorteil dieser Aufteilung liegt darin, dass nur variable Daten über die Schnittstelle übertragen werden müssen. Die meist umfangreicheren konstanten Daten können in einer Datei auf Speicherkarte oder RAM-Disk gespeichert werden.

Wenn ein Easy-Plug-Interpreter auf ein #FO-Kommando stößt, springt er in die dort angegebene Datei, arbeitet diese ab und springt dann wieder zurück, um die restlichen Kommandos abzuarbeiten.

Es funktioniert nicht, die gleichbleibenden Daten in der Datei AUTOSTRT.FOR zu speichern, und die variablen Daten über die Schnittstelle zu senden.

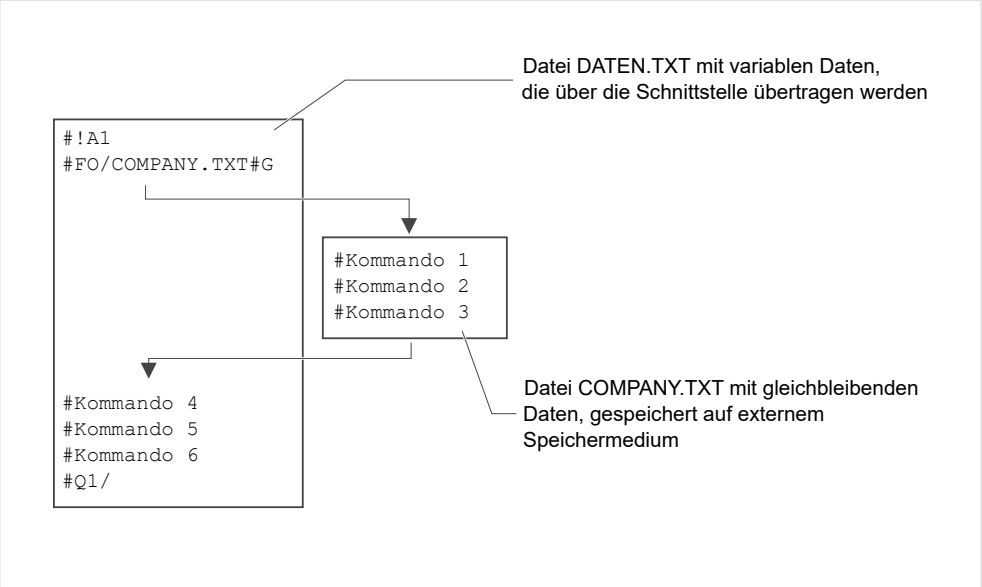


Bild 4: Einfügen von Kommandos von einer auf Datei auf einem externen Speichermedium in die eigentliche Formatdatei.

Beispiele

Kommandoaufruf:

```
#FO/C:\PFAD\...\VERZ_1\AFTER__8.FOR#G
```

Formatdatei COMPANY.TXT auf externem Speichermedium:

```
#G------ Etikettenformat 50x70 mm ------------
#IMN70/50#G
-------------- Formatkommando ----------------
#ERN0#G
#T5#J40#YT104/0///COMPANY
#T5#J30#YT104/0///located
#T5#J20#YT104/0///in GERMANY
#G---------- Feld 00 mit 15 Zeichen ----------
#T5#J20#YT104/0D///$00,15
#G-------Feld 01 (Barcode) mit 12 Zeichen ----
#T5#J02#YB1/0D/10/3///$01,12#G
```

Datei DATEN.TXT wird zum Drucker übertragen:

```
#!A1#G----------------------------------- Drucker aktivieren
#FO/COMPANY.TXT#G-------------------- Einfügen von COMPANY.TXT
#YV00/Denmark#G---------------- Feld 00 mit den Daten „Denmark“
#YV01/999333777001#G------ Feld 01 mit den Daten „999333777001“
#Q7/#G------------------------------------------ Druckmenge 7
#YV00/USA#G------------------------ Feld 00 mit den Daten „USA“
#YV01/444197666001#G------ Feld 01 mit den Daten „444197666001“
#Q5/#G------------------------------------------ Druckmenge 5
```

Zugehörige Informationen

Dateioperationen auf Seite 149
Beschreibung der korrekten Pfadangabe für Dateioperationen.

#DF - Download einer Datei auf Seite 31

#FW - Folienbreite

Nur anwendbar mit Maschinen die zur Firmware-Gruppe BEL gehören (siehe Kapitel Gültigkeit auf Seite 7).

Definiert die Breite des für den Druckauftrag verwendete Folienmaterials.

Syntax

```
#FWn
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
n	int	Folienbreite [30..107] mm

G

#G - Kommandoabschluss

Das Kommando #G kann als Kommandoabschluss oder zur Kennzeichnung einer Kommentarzeile verwendet werden.

Syntax

```
#G
```

Kommandoabschluss

Das Kommando #G wird zum Abschluss eines vorausgehenden Kommandos benutzt, falls folgende beiden Punkte zutreffen:

- Der letzte Parameter des vorausgehenden Kommandos war vom Typ num, int oder TEXT
- Das vorausgehende Kommando ist das letzte Kommando des Formats

#G sollte immer gesetzt werden zum Abschluss der Kommandos für

- Mengenangaben
- Download eines Logos
- Barcodes, für die Zeichen <20hex zugelassen sind (z. B. Code 128, EAN 128)

Kommentar

Außerdem kann #G verwendet werden, um Kommentarzeilen in einer Easy Plug Datei zu kennzeichnen. In einer Kommentarzeile dürfen, angeführt von der Zeichenfolge #G, nahezu beliebige Zeichen folgen.

Nicht erlaubt sind Zeilen, die mit folgenden Zeichenfolgen beginnen:

- `#G/`
- `#G /`

Beispiel

Anzahl der Etiketten = 100 und Abschluss des numerischen Wertes mit einem Leerkommando:

```
#Q100#G
```

H

#HP - Druckkopf-Andruck einstellen

Nur anwendbar mit Maschinen die zur Firmware-Gruppe BEL gehören (siehe Kapitel Gültigkeit auf Seite 7).

Das Kommando #HP stellt den Druckkopf-Andruck auf den gewünschten Wert x ein.

Syntax

```
#HPx
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
x	num	Druckkopf-Andruck [1.0..3.0] • 1.0 = niedriger Druckkopf-Andruck (entspricht Drehknopf-Einstellung „I“ bei älteren Maschinen) • 3.0 = hoher Druckkopf-Andruck (entspricht Drehknopf-Einstellung „III“ bei älteren Maschinen)

Beispiel

- Die Einstellung für den Druckkopf-Andruck wird aus der Parameter-Einstellung (Druck > Kopfandruck) übernommen:

```
#!A1
#IMN50/50
#ER
#T1#J1#YT109/0///HP Para#G
#Q2/
```

- Die Einstellung für den Druckkopf-Andruck wird aus dem vorangestellten HP-Kommando übernommen:

```
#HP2.1
#ER
#T1#J1#YT109/0///HP2.1#G
#Q2/
```

- Die Einstellung für den Druckkopf-Andruck wird aus der Parameter-Einstellung (Druck > Kopfandruck) übernommen:

```
#ER
#T1#J1#YT109/0///HP Para#G
#Q1/
```

Zugehörige Informationen

#!HP - Druckkopf-Andruck einstellen auf Seite 13

#HV - Druckkopftemperatur einstellen

Das Kommando #HV stellt die Druckkopfspannung und damit die Druckkopftemperatur auf den gewünschten Wert n ein.

Syntax

```
#HVn
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

> VORSICHT!
> Gefahr einer verringerten Druckkopf-Lebensdauer
> Die Druckkopftemperatur (= HV-Wert) hat unmittelbaren Einfluss auf die Lebensdauer eines Druckkopfes. Es gilt: „Je höher die Temperatur, mit der der Druckkopf gefahren wird, desto niedriger ist seine Lebensdauer“. Das gilt verstärkt für HV-Werte über 100%. Deshalb beachten:
>
> ► Immer den niedrigsten HV-Wert wählen, der noch ein akzeptables Druckergebnis liefert.

Parameter	Wert	Bedeutung
n	int	Druckkopftemperatur [0...xxx] • 0 = niedrigste Temperatur • xxx = höchste Temperatur (abhängig vom Druckertyp, siehe Tabelle unten)

Drucker	max. HV-Wert
64-xx, DPM, PEM, ALX 92x, ALX 73x	110
XLP 504/506 300 dpi	120
XLP 504 600 dpi	100

Tabelle 7: Einstellungen, mit denen die höchste Druckkopftemperatur erreicht wird (max. HV-Wert).

Zugehörige Informationen

#!H - Kopftemperatur einstellen auf Seite 12
Das Sofortkommando #!H stellt die Kopfspannung und damit die Kopftemperatur auf den gewünschten Wert n ein.

I

#IDM - Data Matrix Code

Das Kommando #IDM druckt einen Barcode vom Typ Data Matrix Code.

Syntax

```
#IDMn/idgwrck/s/vop/a/TEXT#G
```

Das Kommando #IDM muss mit #G abgeschlossen werden.

Parameter	Wert	Bedeutung
n	0	ASCII Kodiert ASCII-Daten, gepackte numerischen Daten (engl.: double density numeric data) und Steuerzeichen für Symbologien. (engl.: symbology control characters).
	1	C40 Kodiert ASCII-Daten. Packt drei alphanumerische Datenzeichen in zwei Codewörter. Wird angewendet, wenn die Daten mehr Groß- als Kleinbuchstaben enthalten.
	2	TEXT Kodiert ASCII-Daten. Packt drei alphanumerische Datenzeichen in zwei Codewörter. Wird angewendet, wenn die Daten mehr Klein- als Großbuchstaben enthalten.
	3	BASE256 Kodiert alle 8-Bit-Daten. Um alle Zeichen von 0x00 bis 0xff zu kodieren, muss SYSTEM PARAMETER > Zeichen Filter bzw. Drucker Sprache > Easy-Plug Einst. > Zeichen Filter auf “Alle Zeichen“ gestellt sein!
	4	reserviert
	5	AUTO (Voreinstellung)
i	B	EAN/UCC-Modus *mit* Klammern um den Datenbezeichner. Die Daten müssen mit Klammern gesendet werden, die Klammern werden aber nicht kodiert gedruckt. Handhabung wie EAN 128 Barcode.
	X	EAN/UCC-Modus *ohne* Klammern um den Datenbezeichner. Die Daten müssen ohne Klammern gesendet werden. Handhabung wie EAN 128 Barcode.
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
g	D	Textfeld besteht aus einem variablen Datenfeld.
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.

r	Rn	n = Reihenzahl
c	Sn	n = Spaltenzahl
└		Standardmäßig wird die Größe (Anzahl Reihen und Spalten) in Abhängigkeit von den Nutzdaten automatisch berechnet (kleinst mögliche Matrix). Mit den Parametern r und c kann die Größe vorgegeben werden (siehe Tabelle unten).
k	F	<FNC1> wird als Daten-Trennzeichen verwendet (--> Beispiel).
	G	<GS> wird als Daten-Trennzeichen verwendet.
└		Nur anwendbar mit Maschinen die zur Firmware-Gruppe BEL gehören (siehe Kapitel Gültigkeit auf Seite 7).

s	int	Anzahl der Drucker-Dots für einen Data Matrix Block [1...200]

v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Alphanumerischer Text mit maximal 1024 Zeichen Stringlänge. Beachten Sie die Vorgaben für den Barcode-Typ. Das Textfeld kann auch ein *Eingabefeld* enthalten. Das Textfeld kann auch ein *variables Datenfeld* sein. Voraussetzung: D-Flag gesetzt (siehe Parameter „g“ oben).

Reihen	Spalten	Reihen	Spalten
10	10	64	64
12	12	72	72
14	14	80	80
16	16	88	88
18	18	96	96
20	20	104	104
22	22	120	120
24	24	132	132

Reihen	Spalten		Reihen	Spalten
26	26		144	144
32	32		8	18
36	36		8	32
40	40		12	26
44	44		12	36
48	48		16	36
52	52		16	48

Tabelle 8: Zulässige Kombinationen aus Reihen (r) und Spalten (c). Beispiel: #IDM/0R12S12/16/0/1/00000A89#G (Die Daten werden in einer 12x12 Matrix codiert).

Optionale Steuerzeichen

Nicht im EAN/UCC-Modus mit BASE256-Kodierung!

~X	Wird benutzt, um Zeichen im Wertebereich von 0 bis 26 darzustellen. Das X wird wie in den folgenden Beispielen ersetzt: • ~@ = 0 • ~A = 1 • ~B = 2 • ~C = 3
~1	Ersetzt das Zeichen FNC1. Erscheint FNC1 an der ersten Stelle, zeigt das, dass die Daten dem „UCC/EAN Application Identifier“ Standard Format entsprechen. Dasselbe gilt, wenn FNC1 an der fünften Stelle des ersten Symbols eines strukturierten Anhangs (engl.: Structured Append) steht.
~2	Steht für strukturierten Anhang. Dieser wird benutzt, um Informationen aus mehreren Symbolen zu einer Sequenz zu verketten. Nach dem „~2“ müssen drei zusätzliche Byte folgen. Die ersten 4 Bit des ersten Bytes zeigen die Position dieses Symbols innerhalb der Sequenz an. Die folgenden 4 Bit zeigen die Anzahl an Symbolen innerhalb der Sequenz an. Das 2. und 3. Byte werden als Dateinamen benutzt, der Werte zwischen 1 und 254 haben kann (bis zu 254*254=64516 verschiedene Namen).
~3	Ist nur an der ersten Position des Symbols erlaubt. Es zeigt an, dass die Daten Steuerzeichen für den Barcode-Leser enthalten.
~4	Nicht zulässig
~5/~6	Sind nur an der ersten Position des Symbols erlaubt. Dadurch werden den Daten folgende Zeichen voran- bzw. nachgestellt: `[)>ASCII30 ASCII05 ASCII 29 Daten ASCII30 ACII04`
~7NNNNNN	Gibt an, dass der “Extended Channel” genutzt werden soll, wo NNNNNN ein Wert im Bereich 000000-999999 ist. *Beispiel*: `~7000010` bedeutet Extended Channel 10. Mit „Extended Channel“ können andere Zeichensätze als ASCII genutzt werden.
~dNNN	ersetzt das ASCII-Zeichen, das durch die drei Ziffern NNN kodiert ist. *Beispiel*: `~d065` ersetzt das Zeichen „A“.

Beispiel: Unterschiedliche Daten-Trennzeichen

```
#IDM0/B0/12///<FNC1>(01)08711744025670(17)181231(10)99999.E7L0185<F▶
NC1>(21)00000D0A#G
--> <FNC1> wird als Daten-Trennzeichen verwendet (erkennbar an <FNC1> in▶
nerhalb der Barcode-Daten)
```

```
#IDM0/B0G/12///
<FNC1>(01)08711744025670(17)181231(10)99999.E7L0185<GS>(21)00000D0A#G
--> <GS> wird als Daten-Trennzeichen verwendet (erkennbar an <GS> inner▶
halb der Barcode-Daten)
```

Das erste <FNC1> ist *immer* ein „FNC1“. Es legt fest, dass die nachfolgenden Daten gemäß GS1-Standard codiert sind.

Zugehörige Informationen

Eingabefelder auf Seite 10
Variable Datenfelder auf Seite 9
Druckerinterne Barcodes auf Seite 153

#IM - Materialinformation

Das Kommando #IM definiert die erforderlichen Materialinformationen und eine (optionale) Materialbezeichnung.

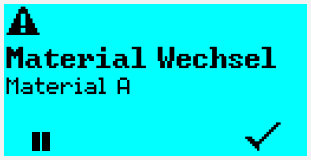
Syntax

```
#IMxyb/l/c/tg/d/ef/o/h/s
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Wenn ein Druckauftrag kein eigenes #IM-Kommando enthält, werden die Material-Einstellungen des vorhergehenden Druckauftrags übernommen.

Parameter	Wert	Bedeutung
x	N	für Endlosmaterial ohne Stanzen
	S	für Material mit Stanze
y	B	für *Batch-Modus*: Etikett vollflächig bedruckbar; hohes Ausgabevolumen.
	E	für *Normal 1:1-Modus*: Die Anfangszonen des Etiketts sind nicht bedruckbar. Alle Etiketten eines Druckauftrags werden bei programmiertem Schnitt abgeschnitten, hohes Ausgabevolumen.
	R	für *Real 1:1-Modus*: Etiketten vollflächig bedruckbar. Alle Etiketten werden bei programmiertem Schnitt ausgegeben; reduziertes Ausgabevolumen.
b	num	Materialbreite: [000.00...max. Breite] mm, abhängig vom Druckkopftyp.
l	num	Etikettenlänge: [000.00...max. Länge] mm, abhängig von der Konfiguration des Druckers. Genau genommen gibt dieser Parameter den Etikettenabstand an, gemessen von der Vorderkante des Etiketts bis zu Vorderkante des nächsten Etiketts. Die maximale Etikettenlänge des jeweiligen Druckers finden Sie auf dem Statusausdruck „Speicher Status“ oder unter Info > System > Speicher Daten > Max. Etik. Länge.

c	TEXT	Materialbezeichnung aus max. 16 alphanumerischen Zeichen. Das Zeichen # darf nicht verwendet werden. Wenn sich die neue Materialbezeichnung von der vorhergehenden unterscheidet, zeigt der Drucker die neue Bezeichnung auf dem Display an, um den Anwender auf einen evtl. notwendigen Materialwechsel hinzuweisen. Die Anzeige muss vom Benutzer durch Tastendruck quittiert werden.  Bild 5: Aufforderung zum Materialwechsel an einer XPA 93x (hier: c=„Material A“). Bild 6: Die Materialbezeichnung „Material A“ wird auch während des Druckens angezeigt. Ausnahme: Wenn die ähnliche Meldung „Nächster Job“ aktiviert wurde (System > Druck Kontrolle > Einzeljob Modus = „Ein“), wird nur diese angezeigt, es erfolgt also keine zusätzliche Aufforderung zum Materialwechsel.

t	int	Transpondertyp 0 = Automatische Erkennung (Voreinstellung) ≠ 0 (siehe Tabelle unten „Unterstützte Transpondertypen“)
g	C	Prüft Etiketten, die mit Transpondern von „Alien Technologies“ ausgestattet sind, auf Gültigkeit (siehe Beschreibung unten). Wenn der gelesene EPC nicht mit A5A5 beginnt, wird das entsprechende Etikett entwertet (d.h. diagonale Streifen werden aufgedruckt).
	D	(Voreinstellung) Prüft, ob ein Etikett mit Transponder vorhanden ist. Dieser Parameter ist nur für Transponder ohne UID/TID gedacht. Die gelesene Information wird intern als UID/TID interpretiert. Diese Einstellung nicht für Transponder verwenden, die mit Null initialisiert sind – diese können nicht erkannt werden.
	N	Keine Prüfung.

d	0	Es werden keine RFID-Operationen ausgeführt.
	int	(d>0) Entfernung in Druckrichtung (x) von der Etikettenkante bis zum Empfangsoptimum der Transponder-Antenne (näherungsweise die Mitte des Transponders). Es wird empfohlen, den Parameter y auf „R“ zu stellen, wenn Parameter d genutzt wird. Parameter d ist zwingend für die RFID-Verarbeitung erforderlich (auch wenn die Entfernung Null beträgt).

e	int	Anzahl an Wiederholversuchen mit unterschiedlichen Etiketten, wenn beim Erkennen eines Etiketts Fehler aufgetreten sind. Die Voreinstellung von e erfolgt über den Parameter RFID PARAMETER > Max Tag Stop. Einstellbereich: [0...10].
f	V	Überprüft die geschriebenen Daten: Die Prüfung erfolgt durch Lesen der Daten direkt nach dem Schreiben. Die gelesenen Daten werden daraufhin mit den geschriebenen Daten verglichen. Wenn beide Daten nicht übereinstimmen, wird ein Fehler gemeldet.

o	Oxxx oder xxx	Optimaler Transponder-Offset für statischen RFID-Betrieb. Einstellbereich: [0..???] mm. Der Optimale Transponder-Offset entspricht dem Abstand des Hotspots vom Etikettenanfang. Um den Wert zu ermitteln: Hotspot-Test durchführen (näheres siehe Handbuch „How-to-RFID“)
	FSxxEyy	Kopplungsbereich für On-the-fly-Betrieb. Sxx: Der Kopplungsbereich beginnt xx mm nach dem Etikettenanfang Eyy: Der Kopplungsbereich endet yy mm nach dem Etikettenanfang Einstellbereich: [0..???] mm

h	int	Anheben des Druckkopfes um den Transponder-Chip zu schützen. Das Maß h ist die Entfernung zwischen der Etikettenkante und der Mitte des Transponder-Chips (siehe Abb. unten). Mit dieser Einstellung werden Druckdaten nicht berücksichtigt, die im Bereich des Chips liegen.

s	int	Chipgröße in mm [5,0...15,0] mm (Voreinst.: 6,0 mm)

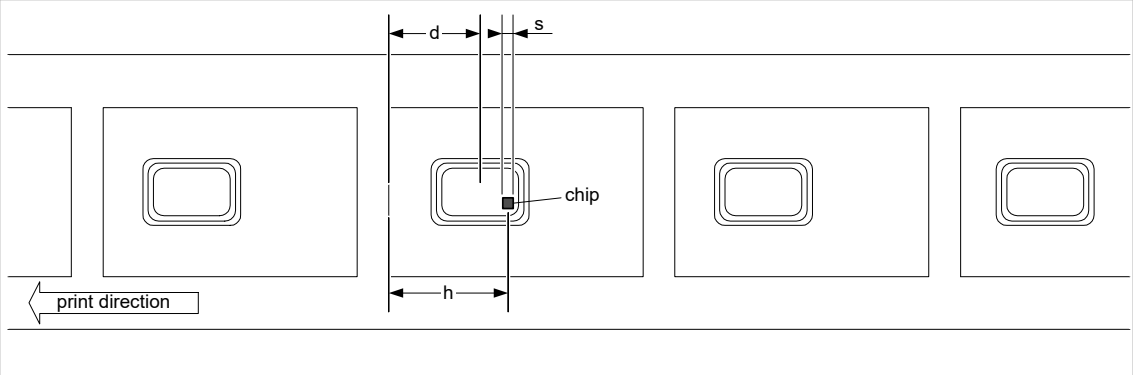


Bild 7: Schematische Darstellung der Parameter bzw. Maße d, h und s.

ID	Hersteller	HF/UHF	Etikettentyp
0	ISO 15963	HF	Autodetect ISO 15963 compliant
1	Infineon	HF	My-d
2	Philips	HF	NXP I-Code 1
3	Philips	HF	NXP I-Code SLI
4	Tag-it	HF	HF ISO

ID	Hersteller	HF/UHF	Etikettentyp
7	Philips	HF	NXP I-Code EPC
8	Philips	HF	NXP I-Code UID
15		UHF	EPC Class 1 Gen 2 / ISO 18000-6C

Tabelle 9: Unterstützte Transpondertypen.

Beispiel

Kommando	Beschreibung Materialtyp
`#IMS50/100/TYP1`	Gestanztes Material, 50 mm breit, 100 mm lang, TYP 1
`#IMN20/50/`	Endloses Material, 20 mm breit, 50 mm lang
`#IMSR98/165///70`	Gestanztes Material, 98 mm breit, 165 mm lang, mit RFID Transponder, 70 mm Abstand zwischen Etikettenkante und Mitte der Transponder-Antenne, Transpondertyp mit automatischer Erkennung (Autodetect)
`#IMSR98/165//6/70/`	Gestanztes Material, 98 mm breit, 165 mm lang, mit RFID Transponder, 70 mm Abstand zwischen Etikettenkante und Mitte der Transponder-Antenne, Transpondertyp „EPC class 1“

Transponder von Alien Technologies

Diese Transponder werden vom Hersteller mit einem EPC beschrieben. Dieser EPC beginnt mit A5A5 (Hex.), wenn es sich um einen „guten“ Transponder handelt (gute Lese-/Schreibeigenschaften). Fängt der EPC mit einer anderen Zeichenfolge an, wurde er vom Hersteller nicht mit „gut“ bewertet, was nicht zwingend bedeutet, dass man ihn nicht lesen oder beschreiben kann.

J

#J - Vertikale Druckposition

Das Kommando #J legt die vertikale Druckposition mit absolutem Wert in mm fest. Nullposition ist in allen Fällen die linke, untere Ecke des Etiketts. Ausnahme: Kommando #R ist aktiv.

Syntax

```
#Jx
```

Kommando muß zwischen #ER und #Q stehen!

Parameter	Wert	Bedeutung
x	num	Vertikale Druckposition in mm bezogen auf die linke, untere Ecke des Etiketts (000.00 mm) Text, der außerhalb der definierten Etikettenfläche liegt, wird nicht gedruckt. Bezugslinie des Textes ist die Grundlinie. Ragt die Grundlinie aus der definierten Etikettenfläche heraus, führen dies zu einer Fehlermeldung, selbst wenn der Text keine Unterlängen enthält.

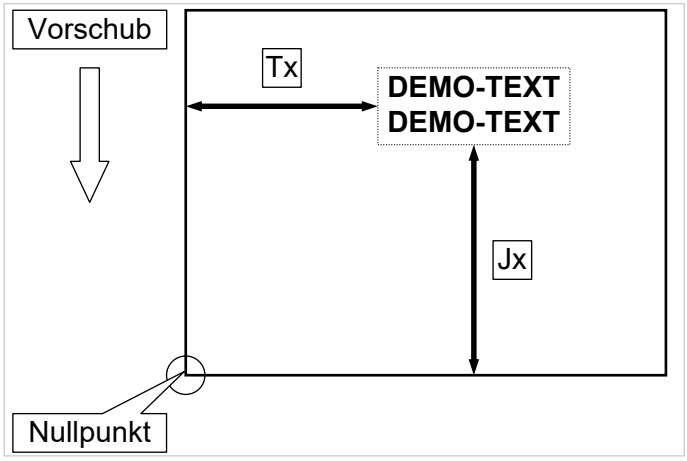


Bild 8: Vertikale (Jx) und horizontale (Tx) Druckposition auf dem Etikett.

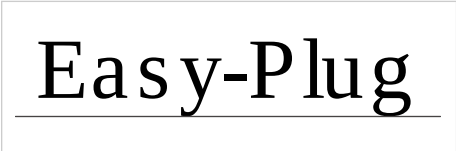


Bild 9: Grundlinie eines Textes. Auch wenn der Text keine Unterlängen (wie hier am y und g) enthielte, läge die Grundlinie doch etwas unterhalb des Textes.

Beispiel

Text beginnt 5 mm von unten:

```
#J5
```

Text beginnt 20 mm von unten:

```
#J20
```

M

#M - Multiplikationsfaktor

Das Kommando #M streckt Schriften und Logos in X/Y-Richtung um den angegebenen Faktor.

Syntax

```
#Mx/y
```

Kommando muß zwischen #ER und #Q stehen!

Alle nachfolgenden Schriften werden bis zum Senden von #M1/1, Reset oder Start einer neuen Etikettenserie um die angegebenen Faktoren vergrößert.

Parameter	Wert	Bedeutung
x	int	Vergrößerung in die Schriftbreite (Laufrichtung) [1...16]
y	int	Vergrößerung in die Schrifthöhe [1...16]

Beispiel

Schrift wird normal ausgedruckt:

```
#M1/1
```

Schrift wird in X-Richtung 2fach und in Y-Richtung 4fach gestreckt:

```
#M2/4
```

Schrift wird in X-Richtung 4fach gestreckt und in Y-Richtung normal ausgedruckt:

```
#M4/1
```

#ME - Material auswerfen

Schneidet das Etikettenmaterial ab und wirft es aus, indem der Drucker das Material rückwärts transportiert.

Syntax

```
#ME
```

Diesen Befehl nur für Geräte mit Messer (Option „Messer 2000“) verwenden.

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Das Kommando funktioniert nicht im Standalone-Betrieb!

#MXC - Maxicode

Das Kommando #MXC druckt einen Barcode vom Typ Maxicode.

Syntax

```
#MXCz/dw/x/y/vop/a/TEXT#G
```

Das Kommando muss mit #G abgeschlossen werden.

Parameter	Wert	Bedeutung
z	int	MaxiCode-Modi (2, 3, 4 oder 6)
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.
x	int	Barcodenummer bei Codeaufteilung auf mehrere Barcodes [1...8] (Voreinstellung: 1)
y	int	Anzahl der Barcodes bei Codeaufteilung auf mehrere Barcodes [1...8] (Voreinstellung: 1)
v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]
a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Je nach Modus (Parameter „z“) enthält TEXT unterschiedliche Informationen (siehe Tabelle unten): Das Textfeld kann auch ein *Eingabefeld* enthalten.

z	TEXT (Syntax)	Daten
2	`pppppppppp ccc sss MSG`	p: Numerischer Postcode (9 Zeichen) c: Zeichen ISO-Ländercode (3 Zeichen) s: Serviceklasse (3 Zeichen) MSG [2]: Codeworte (84 Zeichen)
3	`pppppp ccc sss MSG`	p: Numerischer Postcode (6 Zeichen) c: Zeichen ISO-Ländercode (3 Zeichen) s: Serviceklasse (3 Zeichen) MSG [2]: Codeworte (84 Zeichen)
Bei den Modi 2 und 3 muss zwischen Postcode, Ländercode, Serviceklasse und Nutzdaten jeweils ein Leerzeichen stehen! Die Anzahl der Zeichen im Postcode, Ländercode und Serviceklasse muss exakt eingehalten werden.		
4	`MSG`	MSG: Beliebige alphanumerische Zeichen (93)
6	`MSG`	

Tabelle 10: Aufbau und Länge des Datenstrings "TEXT" in Abhängigkeit vom gewählten Modus "z".

Einfaches Beispiel

Der String „MSG“ enthält nur alphanumerische Zeichen.

```
#!A1
#IMS100/200

#ER

#T40#J10
#MXC4/0/1/1///NOVEXX Solutions Teststring#G
#Q1#G

#!P1
```

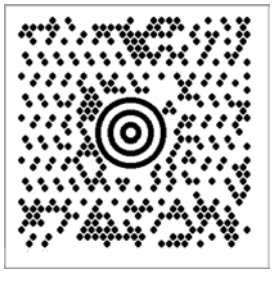


Bild 10: Ausdruck des Beispiels.

[2] Abhängig davon, welche alphanumerischen Zeichen (0x00 – 0xff) verwendet werden, ergibt sich eine unterschiedliche Anzahl von Nutzzeichen, da mehr oder weniger häufig zwischen Subsets umgeschaltet werden muss.

Beispiel mit Zeichen < 20h

Wichtig: Parameter Zeichen Filter auf „Alle Zeichen“ stellen!

Der String „MSG“ kann auch nicht darstellbare Zeichen enthalten, das sind Zeichen mit einem hexadezimalen Code <20h. Solche Zeichen sind z. B. die Steuerzeichen <RS> (0x20) und <GS> (0x1E). Spezielle Editoren machen diesen unsichtbaren Code durch Platzhalter sichtbar:

```
1  #!A1
2  #IMS100/200
3
4  #ER
5  #T40#J10
6  #MXC4/0/1/1///[)>RS01GS96123GS840GS111GS1234567890GSUPSNGS123456GS222GS1GS1/
7  2GS10GSYGSTestGSTest CGSTeRSEOT#G
8  #Q1#G
9
10 #!P1
```

Für solche Nutzdaten wird empfohlen, den alternativen Befehl #SMX zu verwenden, in dem diese Daten sichtbar dargestellt werden können.

Bild 11: Ausdruck des Beispiels.

Zugehörige Informationen

Eingabefelder auf Seite 10
Druckerinterne Barcodes auf Seite 153
#SMX - Maxicode Definition auf Seite 90
Das Kommando #SMX definiert eine Barcode vom Typ „Maxicode“. Das Drucken des Maxicodes wird durch ein nachfolgendes #VW-Kommando veranlasst.

N

#N - Nationalität Zeichensatz

Das Kommando #N schaltet zwischen den Fonts verschiedener Nationalitäten um.

Syntax

```
#Nn
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
n	0	USA
	1	UK
	2	Frankreich
	3	Deutschland
	4	Italien
	5	Schweden
	6	Spanien
	7	Norwegen
	8	Spezial
	9	IBM-ähnlich
	10	ANSI Codepage 1252 Latin 1 (entspricht ISO 8859-1 Latin 1)
	11	ANSI Codepage 1250 Central Europe
	12	ISO 8859-2 Latin 2
	13	UTF-8

Beispiel

Drucken von russischem Text mit UTF- 8 Kodierung:

```
#G ----------------------------------------------------------------------
#G Voraussetzung: Der kyrillische Zeichensatz befindet sich auf einer
#G Speicherkarte im Verzeichnis \fonts. Dateiname: „font900.xxx“.
#G ----------------------------------------------------------------------
#!A1
#IM200/100
#N13
#ER
#SS900/OV/32/#G
#T01.0#J010.0
#VW/L/“Указания по тезнике безопасности при эксплуатации машины для
 печатания этикеток“
#Q1/
```

O

#OLVI - Online-Verifier initialisieren

Das Kommando #OLVI sendet einen Initialisierungsstring an einen Online-Verifier (OLV) vom Typ RJS SV100.

Syntax

```
#OLVIn/INITTEXT
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
n	0	INITTEXT wird nur dann zum OLV gesendet, wenn sich sein Inhalt verändert hat.
	1	INITTEXT wird immer zum OLV gesendet.

INITTEXT		SV100-Kommandos, die vor dem Starten eines Druckauftrages zum OLV geschickt werden. Zulässige Initialisierungskommandos finden Sie im Handbuch des SV100.

#OLVD - OLV-Grenzwert definieren

Das Kommando #OLVD definiert Grenzwerte für die einzelnen Lesbarkeits-Kriterien. Das Kommando gilt für Online-Verifier (OLV) vom Typ RJS SV100.

Syntax

```
#OLVD/a/b/c/d/e/f/g/h/i/j/k/u
```

Kommando muß zwischen #ER und #Q stehen!

Für jeden Barcode kann ein eigener Grenzwert definiert werden.

Der gemessene Wert muss in allen Fällen größer oder gleich dem angegebenen Grenzwert sein.

Parameter	Wert	Bedeutung
a	P	Dekodierbar (Pass)
	F	Nicht dekodierbar (Fail)

b	int	Dekodierbarkeit [0...100]

c	int	Modulation [0...100]

d	int	Defekte [0...100]

e	int	Kantenkontrast [0...100]

f	int	Rmin/Rmax [0...100]

g	int	Symbolkontrast [0...100]
h	int	PCS [0...100]
i	int	R (weiß) [0...100]
j	int	R (schwarz) [0...100]
k	int	Ratio [0...99]
u	int	ANSI Symbolgrad [0...40]

Beispiel

Nur „Dekodierbarkeit“ und „Defekte“ wird ausgewertet:

```
#OLVD//45//20/////////
```

P

#PA - Druckbeginn verschieben

Das Kommando #PA verschiebt den Beginn des Druckbereichs um den angegebenen Wert.

Anheben des Druckkopfes während des Vorschiebens zum Druckbereich nur im Folienspar-Modus!

Syntax

```
#PAa/b
```

Das Kommando muss vor dem Kommando #IM gesendet werden!

Parameter	Wert	Bedeutung
a	num	Offset in mm vom physikalischen Etikettbeginn bis zum Druckbeginn
b	num	Länge des zu bedruckenden Bereichs

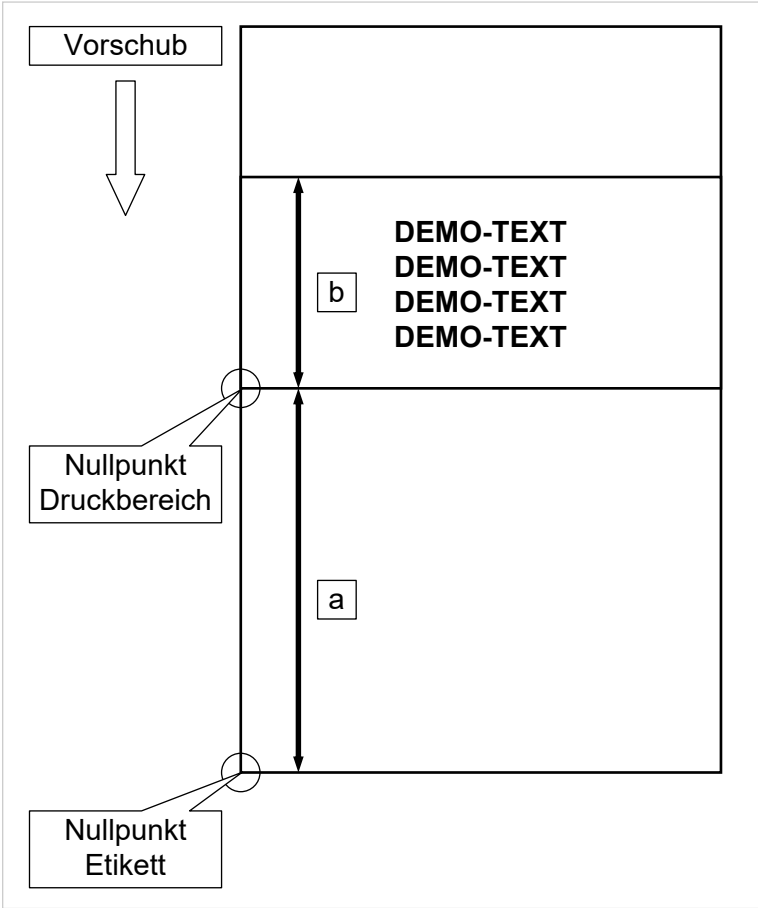


Bild 12: Verschieben des Nullpunkts des Druckbereichs mit der Länge „b“ um die Strecke „a“.

Beispiel

Länge des Druckbereichs 100 mm, Offset 400 mm vom Etikettenstart:

```
#PA400/100
```

Etikettenbreite 100 mm, Gesamtlänge 600 mm:

```
#IMSB/100/600
```

#PC - Einstellen von Parameterwerten

Das Kommando #PC stellt Parameter im Menü des Druckers auf bestimmte Werte ein.
Die Funktion ist hilfreich, wenn…

- Einstellungen von einem Gerät auf ein anderes übertragen werden sollen.
- mehrere Geräte mit denselben Einstellungen versehen werden sollen.
- die Einstellungen eines Gerätes wiederhergestellt werden sollen, z.B. nach dem Auswechseln der CPU-Platine.

Es ist zweckmäßig, die Parameterwerte des betreffenden Gerätes vorher gesammelt auszulesen. Dies kann folgendermaßen geschehen:

- Über den Easy-Plug-Befehl #!PG

- Mit Hilfe des Parameters SPEZIALFUNKTION > Param. speichern bzw. Werkzeuge > Diagnose > Param. speichern

Syntax

```
#PC<ParameterID>/<Wert>#G
```

Viele Parameter brauchen einen Systemreset, damit die geänderte Einstellung wirksam wird. Dieser kann entweder über die Bedientasten des Gerätes oder durch einen Easy Plug-Befehl ausgelöst werden.

Parameter	Wert	Bedeutung
<ParameterID>		*Parameter Identnummer* (ID) 999999 = Parameter-ID für Sonderfunktionen; diese ID muss in Verbindung mit einem der unten aufgeführten Werte verwendet werden (siehe Tabelle). Jeder Parameter im Menü hat eine eigene Parameter-ID. Parameter-ID herausfinden: SPEZIALFUNKTION > Param. speichern bzw. Werkzeuge > Diagnose > Param. speichern aufrufen. Die Zuordnung Parameter-ID/Parameter finden Sie in der so erzeugten Textdatei (Setup-Datei). Beispiele für Setup-Dateien finden Sie im Service-Handbuch des Druckers, im Themenbereich „Spezielle Anwendungen“ (64-xx) bzw. im Anhang (XLP 50x).
<Wert>		*Parameter-Wert* Die zulässigen Werte hängen von der Art des Parameters ab. Zugewiesen werden können Ganze Zahlen, Fließkommazahlen oder Zeichenketten (Strings). Die Parameter-Werte für die Parameter IP Adresse, Netzmaske und Gateway Adresse können im Format xxx.xxx.xxx.xxx oder als reine Integerwerte eingegeben werden.

Wert	Funktion
-1	Neustart des Druckers (Reset)
1	Aktuelle Parametereinstellungen als Kundenvorgabe speichern (entspricht SPEZIALFUNKTION > Kunden Vorgaben = „Verw. aktuelle“ bzw. System > Kunden Vorgaben = „Verw. aktuelle“)
2	Gespeicherte Kundenvorgabe löschen (entspricht SPEZIALFUNKTION > Kunden Vorgaben = „Löschen“ bzw. System > Kunden Vorgaben = „Löschen“)
3	Werkseinstellung wiederherstellen - falls vorhanden mit „Kundenvorgaben“ (entspricht SPEZIALFUNKTION > Werkseinstellung = „Kunden Vorgaben“ bzw. System > Werkseinstellung = „Kunden Vorgaben“), sonst mit „Werksvorgaben“ (entspricht SPEZIALFUNKTION > Werkseinstellung = „Werksvorgabe“ bzw. System > Werkseinstellung = „Werksvorgabe“)

Tabelle 11: Zulässige Werte für Sonderfunktionen – müssen mit der Parameter-ID „999999“ verwendet werden.

Beispiel

Kommandofolge	Beschreibung
`#PC1508/viper#G`	Setzt das FTP Server Passwort auf „viper“ (entspricht SCHNITTST. PARA > >NETZWERK PARAM. > FTP Passwort = „viper“ bzw. Schnittstelle > Netzwerk > Dienste > FTP Passwort = „viper“)
`#PC999999/-1#G`	Drucker wird neu gestartet (Warmstart)

#PDF - Barcode PDF 417

Das Kommando #PDF druckt einen Barcode vom Typ PDF 417.

Syntax

```
#PDFn/td/s/l/z/w/h/TEXT#G
```

Parameter	Wert	Bedeutung
n		Kompressionsart
	0	EXC-Mode (Extended alphanumeric Compaction Mode)
	1	Binary ASCII Plus Mode
t	leer	Leserichtung bidirektional (Voreinstellung)
	T	Leserichtung unidirektional
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
s	int	Security Level [0...8]
l	0	Barcodebreite wird automatisch eingestellt
	int	Barcodebreite in Codewörtern (Spaltenzahl) [1...30]: Die Breite soll so gewählt werden, dass in Zusammenhang mit der sich aus den Nutzdaten ergebenden Fläche des Barcodes die Höhe von 90 Zeilen nicht überschritten wird. Maximal sind 928 Codewörter für einen Barcode zulässig. Davon sind fest vergeben: • 1 Wort für die Längenangabe des Codes • x Worte für Checksumme (Security Level) = 2 (1 + Security Level)
z	0	Zeilenzahl wird automatisch eingestellt
	int	Zeilenzahl [3...90]
w	int	Barcodebreite [1...16]

h	num	Höhe einer PDF-Barcodezeile [1...100] mm

TEXT		Nutzdaten: Zulässige Zeichen in Abhängigkeit von der Kompressionsart. Maximale Stringlänge: 1024 Zeichen. Das Textfeld kann auch ein *Eingabefeld* enthalten.

Zugehörige Informationen

Eingabefelder auf Seite 10
Druckerinterne Barcodes auf Seite 153

#PO - Stanzen-Offset

Das Kommando #PO legt den Etikettenbeginn bei unregelmäßig ausgeformten Etiketten fest.

Syntax

```
#POva
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
v	+	Positiver Offset: Etikettenbeginn vor Stanzenende
	-	Negativer Offset: Etikettenbeginn nach Stanzenende
a	num	Offset in mm Limitationen der einzelnen Drucker beachten, siehe Bedienungsanleitung.

Beispiel

Etikettenbeginn auf Stanzenende:

```
#PO
```

Etikettenbeginn 5 mm nach Stanzenende:

```
#PO+5
```

Etikettenbeginn 20 mm vor Stanzenende:

```
#PO-20
```

#PR - Druckgeschwindigkeit

Das Kommando #PR stellt die Druck- und Vorschubgeschwindigkeit ein.

Syntax

```
#PRux/y/
```

Parameter	Wert	Bedeutung
u [3]	I	Einheit Inch/s (Voreinstellung)
	M	Einheit mm/s
x	int	Druckgeschwindigkeit; Der Einstellbereich ist abhängig vom Druckertyp; Schrittweite: 1 Inch/s
y	int	Vorschubgeschwindigkeit; Der Einstellbereich ist abhängig vom Druckertyp; Schrittweite: 1 Inch/s

Für DPM/ ALX 92x können auch Zehntelstellen eingegeben werden. Die Schrittweite beträgt hier 0,2 Inch (z.B. #PR 8.6/8.6).

Beispiel

Beide Geschwindigkeiten = 8 Inch/s (Voreinstellung):

```
#PR
```

Druckgeschwindigkeit = 4 Inch/s Vorschubgeschwindigkeit = vorher eingestellter Wert, bzw. 8 Inch/s, wenn noch kein Wert eingestellt ist:

```
#PR4/
```

Druckgeschwindigkeit = vorher eingestellter Wert, bzw. 8 Inch/s, wenn noch kein Wert eingestellt ist. Vorschubgeschwindigkeit = 6 Inch/s:

```
#PR/6/
```

Druckgeschwindigkeit = 8 Inch/s Vorschubgeschwindigkeit = 10 Inch/s:

```
#PR8/10/
```

3 Nur für XLP 51x, XLP 60x, XPA 93x, XDM/XPM 94x, XPCU

Q

#Q - Druckmenge

Das Kommando #Q schließt den Druckauftrag ab und legt die Anzahl der zu druckenden Etiketten fest. Außerdem bewirkt es, dass der Druckauftrag überprüft und in den Speicher des Druckers übernommen wird.

Nach der Übernahme kann das Format nicht mehr geändert oder ergänzt werden. Format-Kommandos werden solange nicht ausgewertet, bis der Speicher wieder leer ist. Der Speicher ist leer, wenn die Etikettenserie fertig ausgedruckt oder mit #CF abgebrochen wurde.

Syntax

```
#Qan/
```

Parameter	Wert	Bedeutung
a	A	Standalone-Betrieb: Druckmenge wird abgefragt.
	X	Standalone-Betrieb: Abfrage der Druckmenge wird unterdrückt. Es wird automatisch die Menge „n“ gedruckt.
n	leer	Speicher wird geleert. Es wird nichts gedruckt.
	0	Im Standalone-Betrieb ist die Druckmenge unendlich.
	int	Anzahl zu druckender Etiketten [1...2,2 Mrd]
	*	Druckmenge „unendlich“
/		Das Kommando #Q muss mit Slash (/) oder #G abgeschlossen werden.

Beispiel

100 Etiketten des vorhergehenden Formats werden gedruckt:

```
#Q100/
```

Sonderfall: Unvollständige Etikettenreihen bei mehrbahnigem Druck

Falls der Druckauftrag weniger Etiketten enthält, als nötig wäre, um die letzte Reihe aufzufüllen, geschieht folgendes:

- Druckauftrag wurde über Datenschnittstelle übertragen: Die Reihe wird unvollständig gedruckt, wobei mit dem Etikett an der Material-Nulllinie begonnen wird.
- Druckauftrag wurde im Standalone-Betrieb gestartet: Die Reihe wird aufgefüllt und vollständig gedruckt.

R

#R - X-/Y-Offset Datenblöcke

Das Kommando #R verschiebt alle nachfolgenden Datenblöcke in X-/Y-Richtung.
Logisch oder optisch zusammengehörende Ausdrucke (Datenblöcke) können als "Ganzes" verschoben werden.

Der Bezugspunkt #Tx (siehe #T) oder #Jx (siehe #J) für alle nachfolgenden Positionierungen wird um den eingegebenen x/y-Wert vom Nullpunkt weg verschoben.

Nullpunkt ist die linke untere Ecke des Etiketts (siehe Bild), solange diese nicht durch das #PO Kommando verschoben ist.

Syntax

```
#Rvx/vy
```

Kommando muß *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
v	+	Versatz in positive x- / y-Richtung
	-	Versatz in negative x- / y-Richtung
x	num	horizontaler Versatz der Druckposition

v	+	Versatz in positive x- / y-Richtung
	-	Versatz in negative x- / y-Richtung
y	num	vertikaler Versatz der Druckposition

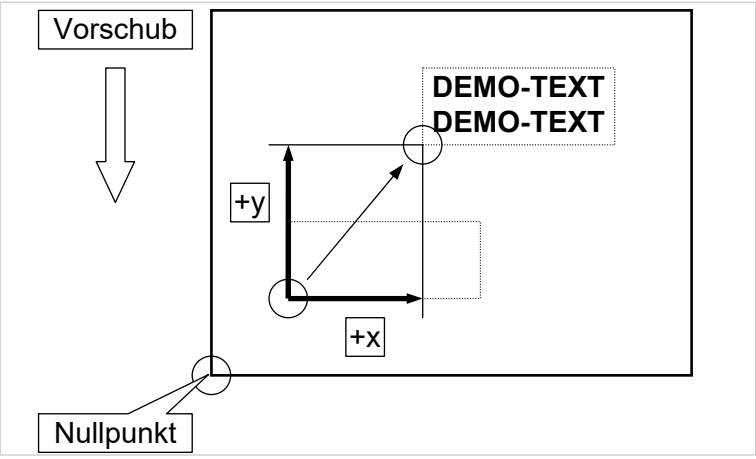


Bild 13: Verschieben eines Textblockes um +x/+y.

Beispiel

Block wird um je 5 mm nach rechts und nach oben verschoben:

```
#R5/5
```

Block wird um je 5 mm nach links und nach unten verschoben:

```
#R-5/-5
```

#RFC - Spezielle RFID-Befehle

Das Kommando #RFC sendet eine Befehlssequenz an das RFID-Lesemodul. Dadurch können spezielle Operationen veranlasst werden, die nicht durch reguläre Lese-/ Schreiboperationen abgedeckt werden.

Syntax

```
#RFC<cmd>#G
```

Der Befehl muss *zwischen* #ER und #Q stehen!

Die Befehle können nur von dem passenden RFID-Lesemodul ausgewertet werden, das eingebaut und aktiviert sein muss.

Parameter	Wert	Bedeutung
<cmd>	int	Befehlscode. ---------- HF-Technologie / FEIG Lesemodul --------- (*1) EAS Merkmale (nur von „NXP I-Code“ chips unterstützt) 16: EAS-Bit setzen 17: EAS-Bit zurücksetzen / löschen 18: EAS-Bit dauerhaft sperren

#RFH - Daten anfordern und zum Host senden

Fordert Daten aus einer RFID-Variablen an, in die mit dem #RFR-Befehl ein Wert eingelesen wurde. Diese Daten werden als Antwort über die Easy-Plug-Schnittstelle übertragen (dieselbe Schnittstelle, die für das Übertragen dieses Befehls genutzt wurde).

Syntax

```
#RFHi/m/n/HEAD
```

Nur mit eingebauter und aktivierter RFID-Option.

Der Befehl muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
i	int	ID-Nummer der Variablen, die durch den Befehl #RFR erzeugt wurde.
m	int	Anzahl der Antwortbytes; falls erforderlich, wird mit Leerzeichen (0x20) aufgefüllt.
n	int	Anzahl der Bytes in HEAD

HEAD		Zeichen, die der Antwort vorangestellt werden sollen.

Beispiel

Auf dem Transponder gespeicherte Daten (Blockgröße 4 angenommen) in hexadezimaler Darstellung:

	MSB.....LSB
Blockadresse 3:	41 42 43 44
Blockadresse 4:	33 31 32 33

Kommandofolge	Beschreibung
`#RFR1/0/I/L/2/3`	2 Blöcke werden vom Transponder gelesen und in der Variable mit der ID-Nummer 1 gespeichert. Text in Variable 1: DCBA3210
`#RFH1/14/6/BLOCK:`	Anforderung der Daten aus Variable 1: Antwort: BLOCK:DCBA3210
`#RFR3/2/B` `#RFH3/20/4/EPC=`	EPC in Variable 3 einlesen Anforderung der Daten aus Variable 3: Antwort: EPC=A5A5800F35609854
`#RFR3/1/B` `#RFH3/20/4/UID=`	UID in Variable 3 einlesen Anforderung der Daten aus Variable 3: Antwort: UID=E005000000000C5B

#RFK - Transponder zerstören

Zerstört den gerade aktiven Transponder.

Syntax

```
#RFK
```

#RFL - Speicherbereich sperren/entsperren

Das Kommando #RFL aktiviert oder deaktiviert den Schreibschutz (simple lock) verschiedener Speicherbereiche von EPC Gen 2 Tags.
Vor dem ersten Ein- oder Ausschalten des Schreibschutzes muss das Zugriffspasswort einmal mit Hilfe eines Schreibbefehls (#RFW oder #SRF + #VW/T) in den Zugriffspasswort-Speicher des Tags geschrieben worden sein. Für jeden Versuch, den Schutzstatus zu ändern, muss dasselbe Passwort als Teil des #RFL-Befehls mitgeschrieben werden.

Das Kommando #RFL selbst schreibt das Zugriffspasswort *nicht* in das Tag!

Das Kommando #RFL unterstützt *nicht* die „Permalock"-Option, wie sie im EPC Gen 2 Standard spezifiziert ist. Der Schreibschutz kann jederzeit aufgehoben werden, wenn ein entsprechender Befehl mit dem richtigen Zugriffspasswort gegeben wird.

Syntax

```
#RFLa/b/c/TEXT#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Der Befehl muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
a		Operation:
	1	Speicherbereich sperren (Schreibschutz)
	0	Speicherbereich entsperren
b		Speicherbereich wählen:
	0	Benutzerspeicher (user memory)
	2	EPC
	3	Löschpasswort (kill password)
	4	Zugriffspasswort (access password)
c	B	Hex ASCII Codierung: Für das Zugriffspasswort müssen 8 Zeichen vorgesehen werden.
	I	Rohdaten (Binärdaten): Für das Zugriffspasswort müssen 4 Zeichen vorgesehen werden.
TEXT		Zugriffspasswort: EPC Gen 2 spezifiziert das Zugriffspasswort mit einer Größe von 32 Bit (4 Byte). Je nach Codierung der Eingabedaten müssen 4 oder 8 Zeichen vorgesehen werden.

#RFPL - Speicherbereich permanent sperren

Das Kommando #RFPL aktiviert den Schreibschutz (permanent lock) verschiedener Speicherbereiche von EPC Gen 2 Tags.
Vor dem ersten Ein- oder Ausschalten des Schreibschutzes muss das Zugriffspasswort einmal mit Hilfe eines Schreibbefehls (#RFW oder #SRF + #VW/T) in den Zugriffspasswort-Speicher des Tags geschrieben worden sein.

Das Kommando #RFL selbst schreibt das Zugriffspasswort *nicht* in das Tag!

Der Vorgang kann nach der Ausführung nicht mehr rückgängig gemacht werden.

Das Kommando #RFPL setzt die „Permalock“-Option um, wie sie im EPC Gen 2 Standard spezifiziert ist.

Syntax

```
#RFPLa/b/c/TEXT#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Der Befehl muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
a		Operation:
	1	Speicherbereich permanent sperren (Schreibschutz)
	0	Speicherbereich permanent entsperren

b		Speicherbereich wählen:
	0	Benutzerspeicher (user memory)
	2	EPC
	3	Löschpasswort (kill password)
	4	Zugriffspasswort (access password)

c	B	Hex ASCII Codierung: Für das Zugriffspasswort müssen 8 Zeichen vorgesehen werden.
	I	Rohdaten (Binärdaten): Für das Zugriffspasswort müssen 4 Zeichen vorgesehen werden.

TEXT		Zugriffspasswort: EPC Gen 2 spezifiziert das Zugriffspasswort mit einer Größe von 32 Bit (4 Byte). Je nach Codierung der Eingabedaten müssen 4 oder 8 Zeichen vorgesehen werden.

#RFR - Daten lesen

Das Kommando #RFR liest Daten aus einem RFID-Transponder und weist sie einer RFID-Variablen zu. Der Inhalt der Variablen kann mit dem Befehl #!RF ausgelesen werden.

Syntax

```
#RFRi/t/ab/s/n
```

Nur mit eingebauter und aktivierter RFID-Option.

Der Befehl muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
i	int	ID-Nummer der Variablen, die für das Einlesen der Daten erstellt wurde.

t	0	BLOCK Parameter „s“ und „n“ erforderlich
	1	UID/TID Parameter „n“ optional, um den Datenumfang zu begrenzen, der bei Leseoperationen zurückgegeben wird
	2	EPC
	3	KILL password
	4	ACCESS password
	5	MEM BANK
	6	AFI
	7	MEMBLOCK Parameter „s“ und „n“ erforderlich
	8	DSFID

a	I	Lesen von ASCII-Daten (Voreinstellung): Jedes Byte wird als ASCII-Zeichen kodiert.
	B	Lesen von binären Daten: Jedes Byte wird hex. kodiert (in ASCII-Darst.).
b	L	Das niedrigstwertige Byte jedes Blocks (LSB) auf dem Transponder wird zuerst gelesen (Voreinstellung).
	M	Das höchstwertige Byte jedes Blocks (MSB) auf dem Transponder wird zuerst gelesen.

s	int	Wenn Datentyp = • BLOCK: Adresse des Startblocks • MEM BANK: Adresse/Name der Memory Bank; (Voreinstellung: 0) • MEMBLOCK: Kombination aus Bezeichner für Memory Bank und Startadresse in der Memory Bank. Schema für die Adressberechnung: `s = <EPC MemoryBankID> * 1000 + <BlockNumber>` EPC MemoryBankID: – 0: reserviert (Voreinstellung) – 1: EPC – 2: TID – 3: USER – andere Werte werden ignoriert • Anderer Datentyp: Parameter wird ignoriert

n	int	Wenn... • Datentyp = BLOCK, TID, MEM BANK oder MEMBLOCK: Anzahl der Blöcke, die gelesen werden sollen (Voreinstellung: 0). • Anderer Datentyp: Parameter wird ignoriert Wenn ein TID-Feld (t=1) gelesen wird, kann n verwendet werden, um den Umfang der gelesenen Daten zu begrenzen. Wenn keine Größe angegeben wird (n=0), wird der gesamte Inhalt der TID Memory Bank gelesen, der unerwartet viele Informationen erhalten kann (abhängig vom RFID Transponder/Chip). Typische Werte für n sind hier: • 2: Lese 32Bit Vendor-ID/Chip-ID • 4: Lese 32Bit Vendor-ID/Chip-ID + 32Bit Unique Serial Number

Beispiel

Auf dem Transponder gespeicherte Daten (Blockgröße 4 angenommen) in hexadezimaler Darstellung:

	MSB.....LSB
Blockadresse 3:	41 42 43 44
Blockadresse 4:	30 31 32 33

Kommandofolge	Beschreibung
`#RFR1/0/I/L/2/3`	2 Blöcke, beginnend mit Block 3, werden vom Transponder gelesen und in der Variablen mit der ID-Nummer 1 gespeichert. Text in Variable 1: DCBA3210
`#RFR1//B/3/2`	Einlesen binärer Daten: Text in Variable 1: 4443424133323130
`#RFR1//M/3/2`	Einlesen umgekehrter ASCII-Daten: Text in variable 1: ABCD0123
`#RFR3/2/B`	Einlesen eines EPC in Variable 3
`#RFR3/1/B`	Einlesen einer UID in Variable 3

#RFW - Daten auf RFID-Etikett schreiben

Das Kommando #RFW schreibt Daten eines definierten Typs auf ein RFID-Etikett.

Syntax

```
#RFWt/abs/r///n/DATA#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
t		Datentyp
	0	BLOCK
	1	UID/TID
	2	EPC
	3	KILL password
	4	ACCESS password
a	I	Daten in DATA werden als ASCII-kodiert angenommen (Voreinstellung). Jedes in DATA enthaltene Byte wird in der gleichen Form auf den Transponder-Chip geschrieben. Die Daten werden nicht interpretiert. Diese Einstellung eignet sich für druckbare Byte-Kodes, wie ASCII-Zeichen.
	B	Daten in DATA werden als binär kodiert angenommen. Zwei aufeinanderfolgende Bytes in DATA werden als hex. Beschreibung eines Bytekodes interpretiert. Dieser Bytekode wird auf den Chip geschrieben. Diese Einstellung eignet sich besonders für nicht druckbare Zeichensätze.
b	L	Das niedrigstwertige Byte (LSB) des Blocks wird zuerst auf den Transponder geschrieben (Voreinstellung).
	M	Das höchstwertige Byte (MSB) des Blocks wird zuerst auf den Transponder geschrieben.
s	int	Adresse des Startblocks. Voreinstellung: 0 (= Blocknr.)

r	int	Anzahl der Lese/Schreibe Wiederholversuche. Die Voreinstellung wird über den Parameter RFID PARAMETER > Nr of CMD retries festgelegt.

n	int	Anzahl der Zeichen in DATA.

DATA		Zu schreibende Daten. (Datentyp BLOCK) n muss ein Vielfaches der Blockgröße sein, sonst wir mit 0 auf ein Vielfaches der Blockgröße aufgefüllt.

Beispiel

12 Bytes als EPC 96 in hex. Darstellung schreiben:

```
#RFW2/B////24/01203D2A916E8B8719BAE03C
```

#RSS - GS1 DataBar & CC

Das Kommando #RSS druckt ein Barcode-Feld mit einem GS1 DataBar (vormals RSS) oder einem Composite Symbology Barcode.

Syntax

```
#RSSzx/dw/s/vop/a/TEXT#G
```

Das Kommando muss mit #G abgeschlossen werden!

Parameter	Wert	Bedeutung
z	int	Barcode-Nummer [4] GS1 DataBar Expanded *Stacked* wird von Easy-Plug automatisch eingesetzt, wenn folgende Bedingungen erfüllt sind: • Gewählter Barcode-Typ: GS1 DataBar Expanded (z = 6) • Breite (x) ist auf weniger als 22 Symbolzeichen pro Reihe eingestellt • Die Nutzdaten überschreiten die eingestellte Breite
x	Sn	n = [4...22] Symbolzeichen pro Reihe

d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.

s	int	Anzahl der Drucker-Dots für die Modulbreite des Barcodes.

v	+	Inkrementieren – Offset wird zu TEXT addiert

4 Siehe unten, Kapitel „Zusätzliche Informationen“.

	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Primärdaten und optional Sekundärdaten (getrennt durch „|“); alphanumerischer Code gemäß ausgewähltem Barcode-Typ, maximal 1024 Zeichen lang.

Beispiele

- GS1 DataBar Omnidirectional

  Ausgabe: 0109501101420021

```
#RSS1/0/3///0950110142002
```

- GS1 DataBar Stacked Omnidirectional

  Ausgabe: 0109501101420038

```
#RSS4/0/3///0950110142003
```

- GS1 DataBar Omnidirectional (POS und allgemeine Verteilung)

  Ausgabe: 0109501101420045

```
#RSS1/0/6///0950110142004
```

- GS1 DataBar Expanded

Ausgabe: 0109501101420052211234567 8

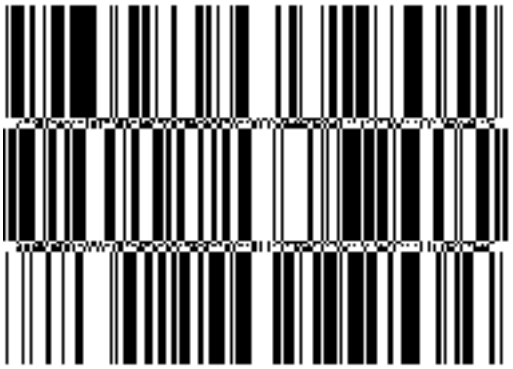
```
#RSS6/0/3///0109501101420052211234567 8
```

- GS1 DataBar Expanded Stacked

  Ausgabe: 109501101420069392299 5<GS>3202000100 17100101422123<GS>2112345678

```
#RSS6S6/0/3///0109501101420069392299 5<FNC1>3202000100 17100101422123<F▶
NC1>2112345678#G
```

ACHTUNG! Kein <cr><lf> in der Easy Plug-Sequenz oder in der Ausgabe.



- GS1 DataBar Truncated

  Ausgabe: 0109501101420076

```
#RSS2/0/3///0950110142007
```

- GS1 DataBar Limited

  Ausgabe: 0109501101420083

```
#RSS5/0/3///0950110142008
```

- GS1 DataBar Stacked

  Ausgabe: 0109501101420090

```
#RSS3/0/3///0950110142009
```

Zugehörige Informationen

Druckerinterne Barcodes auf Seite 153

#RT - RFID-Daten lesen und drucken

Das Kommando #RT liest Daten von einem RFID-Transponder und druckt sie auf das Etikett.

Syntax

```
#RTz/dbjk/t/ab/s/n
```

Nur mit eingebauter und aktivierter RFID-Option.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Zeichensatz-Nummer [100...116] (näheres siehe Link unter „Zusätzliche Informationen“
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
b	A	Text invers drucken (weiße Schrift auf dunklem Hintergrund) Der dunkle Hintergrund muss zuerst gedruckt werden (als Linie oder Rechteck).
j	L	Textausrichtung *linksbündig* (Voreinstellung): Das Schreibkommando (#T/#J) bezieht sich auf den linken Rand des Textfelds. Das Feld wird nach rechts aufgebaut.
	M	Textausrichtung *zentriert*: Das Schreibkommando (#T/#J) bezieht sich auf die Mitte des Textfelds. Das Feld wird nach beiden Seiten der Druckposition aufgebaut.
	R	Textausrichtung *rechtsbündig*: Das Schreibkommando (#T/#J) bezieht sich auf den rechten Rand des Textfelds. Das Feld wird nach links aufgebaut.
k	Snum	Maß für den festen Abstand aller Zeichen des zu druckenden Datenstrings in mm. Der Buchstabe S muss ohne Leerzeichen vor dem Zahlenwert stehen.
t	0	BLOCK Parameter „s“ und „n“ erforderlich
	1	UID/TID Parameter „n“ optional, um den Datenumfang zu begrenzen, der bei Leseoperationen zurückgegeben wird
	2	EPC
	3	KILL password
	4	ACCESS password
	5	MEM BANK
	6	AFI

	7	MEMBLOCK Parameter „s“ und „n“ erforderlich
	8	DSFID

a	I	Lesen von ASCII-Daten (Voreinstellung): Jedes Byte wird als ASCII-Zeichen kodiert.
	B	Lesen von binären Daten: Jedes Byte wird hex. kodiert (in ASCII-Darst.).
b	L	Das niedrigstwertige Byte jedes Blocks (LSB) auf dem Transponder wird zuerst gelesen (Voreinstellung).
	M	Das höchstwertige Byte jedes Blocks (MSB) auf dem Transponder wird zuerst gelesen.

s	int	Wenn Datentyp = • BLOCK: Adresse des Startblocks • MEM BANK: Adresse/Name der Memory Bank; (Voreinstellung: 0) • MEMBLOCK: Kombination aus Bezeichner für Memory Bank und Startadresse in der Memory Bank. Schema für die Adressberechnung: `s = <EPC MemoryBankID> * 1000 + <BlockNumber>` EPC MemoryBankID: – 0: reserviert (Voreinstellung) – 1: EPC – 2: TID – 3: USER – andere Werte werden ignoriert • Anderer Datentyp: Parameter wird ignoriert

n	int	Wenn... • Datentyp = BLOCK, TID, MEM BANK oder MEMBLOCK: Anzahl der Blöcke, die gelesen werden sollen (Voreinstellung: 0). • Anderer Datentyp: Parameter wird ignoriert Wenn ein TID-Feld (t=1) gelesen wird, kann n verwendet werden, um den Umfang der gelesenen Daten zu begrenzen. Wenn keine Größe angegeben wird (n=0), wird der gesamte Inhalt der TID Memory Bank gelesen, der unerwartet viele Informationen erhalten kann (abhängig vom RFID Transponder/Chip). Typische Werte für n sind hier: • 2: Lese 32Bit Vendor-ID/Chip-ID • 4: Lese 32Bit Vendor-ID/Chip-ID + 32Bit Unique Serial Number

Beispiel

Auf dem Transponder gespeicherte Daten (Blockgröße 4 angenommen) in hexadezimaler Darstellung:

	MSB.....LSB
Blockadresse 3:	41 42 43 44

Blockadresse 4:	30 31 32 33

Kommandofolge	Beschreibung
`#RT101////3/2`	2 Blöcke werden aus dem Transponder ausgelesen und gedruckt. Startadresse: Block 3; Zeichensatz: 101; Druckrichtung: normal (von links nach rechts). Ausdruck: DCBA3210
`#RT101///B/3/2`	Dito mit binären Daten: Ausdruck: 443424133323130
`#RT101///M/3/2`	Dito mit umgekehrten ASCII-Daten: Ausdruck: ABCD0123
`#RT105//2/B`	EPC in hex. Darstellung gedruckt (Font: 105; Druckrichtung: normal)

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#RTC - Echtzeituhr einstellen

Das Kommando #RTC stellt Uhrzeit und Datum der Echtzeituhr ein.

Syntax

```
#RTC/date#G
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
date		Syntax: `dd.mm.yyyy hh:mm` (dd=Tag, mm=Monat, yyyy=Jahr, hh=Stunde, mm=Minute)

Beispiel

```
#RTC/01.02.2007 01:32#G
```

#RX - Auswahl Stanzenerkennung

Das Kommando #RX wählt eine der vorhandenen Etikettenlichtschranken am Drucker aus.

Syntax

```
#RXn
```

Das Kommando muss *außerhalb* der Kommandofolge #ER bis #Q stehen!

Parameter	Wert	Bedeutung
n	0	Durchlicht
	1	Reflex
	2	Fullsize

Parameter	Wert	Bedeutung
	6	Lichtschranke für kurze Etiketten Option nur für XLP 504 und XLP 514
	7	Sensor für transparente Etiketten, an die BasicIO-Platine angeschlossen
	8	Sensor für transparente Etiketten, an die Basic-USI-Platine angeschlossen
	9	Alternativer Etikettensensor, an die APSF-Platine angeschlossen

S

#SAZ - Aztec Definition

Das Kommando #SAZ definiert einen Barcode vom Typ „Aztec“. Das eigentliche Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SAZn/etrs/k/i#G
```

Parameter	Wert	Bedeutung
n	int	Barcodegröße [0..36], siehe Barcode-Größentabelle unten
		Kein Wert: Die am besten passende Größe wird verwendet (Voreinstellung)
e	int	Fehlerkorrektur-Niveau [5..95]%; Voreinstellung: 23%
t	B	„~“ wird *nicht* als Escape-Zeichen verwendet (Voreinstellung)
	T	„~“ wird als Escape-Zeichen verwendet Gebrauch: • ~dNNN für ASCII dezimal • ~~ druckt die Tilde • ~Exxxxxx für erweiterte Kanalinterpretation
r		Initialisierungsbit für Lesegerät
	P	Kein Initialisierungsbit (Voreinstellung)
	V	Initialisierungsbit wird hinzugefügt
s	int	Anzahl Dots für einen Aztec-Block [1..20]; Voreinstellung = 1
k	int	Anzahl Symbole für strukturierten Anhang [1..26]; Voreinstellung = 1
i		ID der Datei für strukturierten Anhang (optional)
		Bis zu 23 Zeichen (nur Großbuchstaben) (Voreinstellung: no ID)

Wert	Größe	Ausgabetyp		Wert	Größe	Ausgabetyp
0	15x15	Compact		19	83x83	Full Range
1	19x19	Full Range		20	87x87	Full Range
2	19x19	Compact		21	91x91	Full Range

Wert	Größe	Ausgabetyp		Wert	Größe	Ausgabetyp
3	23x23	Full Range		22	95x95	Full Range
4	23x23	Compact		23	101x101	Full Range
5	27x27	Full Range		24	105x105	Full Range
6	27x27	Compact		25	109x109	Full Range
7	31x31	Full Range		26	113x113	Full Range
8	37x37	Full Range		27	117x117	Full Range
9	41x41	Full Range		28	121x121	Full Range
10	45x45	Full Range		29	125x125	Full Range
11	49x49	Full Range		30	131x131	Full Range
12	53x53	Full Range		31	135x135	Full Range
13	57x57	Full Range		32	139x139	Full Range
14	61x61	Full Range		33	143x143	Full Range
15	67x67	Full Range		34	147x147	Full Range
16	71x71	Full Range		35	151x151	Full Range
17	75x75	Full Range		36	11x11	Rune
18	79x79	Full Range				

Tabelle 12: Barcode Größentabelle

#SB - Barcode Definition

Das Kommando #SB definiert einen Barcode. Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SBz/kclbmre/h/s#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Barcode-Nummer (Näheres siehe Link unter „Zusätzliche Informationen“)

k	M	Barcode mit Klarschriftzeile
	O	Barcode ohne Klarschriftzeile
c	C	Es wird eine Prüfziffer nach Modulo 10 errechnet und ausgedruckt Gilt nur für 2/5 interleaved- und für ITF-Barcodes!
	N	Keine Prüfzifferberechnung
l	H	Klarschriftzeile im Blocksatz
	I	Klarschriftzeile linksbündig angeordnet
	K	Klarschriftzeile mittig angeordnet

	L	Klarschriftzeile rechtsbündig angeordnet Die Voreinstellung hängt vom Barcodetyp ab.
b	A	Positionsänderung der Klarschriftzeile. Wird die Klarschriftzeile normalerweise unter dem Barcode gedruckt, wird sie mit dieser Option über den Barcode verschoben und umgekehrt.
m	B	EAN/UCC-Mode mit Klammern um die Datenbezeichner. Die Nutzdaten müssen mit Klammern gesendet werden! Die Klammern erscheinen zwar in der Klarschriftzeile, sie werden jedoch nicht im Barcode codiert.
	X	EAN/UCC-Mode ohne Klammern um die Datenbezeichner. Die Nutzdaten müssen ohne Klammern geschickt werden!
r	Pnum	Ratio des Barcodes [2.0...3.0] Der Buchstabe P (Proportion) muß direkt vor der Ratio stehen (z.B. P2.5). Eine Ratio ohne den Buchstaben P ist ungültig. Als Dezimaltrennzeichen muss ein Punkt verwendet werden.
e	V	Der Barcode wird geprüft. Erfordert OLV-Option Nur für Barcodes mit Drehwinkel 0° oder 180°.

h	int	Die *Barcode-Höhe* berechnet sich wie folgt: Barcode-Höhe = (h + 1) mm * DRUCK PARAMETER > Barcode Multi *bzw.* Barcode-Höhe = (h + 1) mm * Druck > Format > Barcode Multi Die Voreinstellung für Barcode Multi ist 1 Damit ergeben sich z. B. folgende Werte für die Barcode-Höhe: • 1 mm Barcode-Höhe für h = 0 • 2 mm Barcode-Höhe für h = 1 • 3 mm Barcode-Höhe für h = 2

s	int	Barcode-Breitenfaktor [1...30] Dot

Zugehörige Informationen

Druckerinterne Barcodes auf Seite 153
Zusammenhang zwischen Ratio und Breitenfaktor von Barcodes auf Seite 175

#SCF - Codablock F Definition

Das Kommando #SCF definiert einen Barcode vom Typ „Codablock F“. Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SCFs/m/c/r#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
s	int	Barcode-Breite [1...30]; Voreinstellung: 1
m	int	Höhe einer Codablock-Zeile [1...100] mm; Voreinstellung: 5 mm
c	int	Spaltenzahl [4...62]; Voreinstellung: 10
r	int	Zeilenzahl [2...44]; Voreinstellung: 0 0: Die Zeilenzahl wird vom Codablock berechnet 1: Unzulässige Angabe, führt zu einer Fehlermeldung beim Interpretieren

#SDM - Data Matrix Definition

Das Kommando #SDM definiert einen Barcode vom Typ „Data Matrix“. Das eigentliche Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SDMn/irck/s#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
n	int	Kodierverfahren Näheres siehe #IDM - Data Matrix Code
	0	ASCII
	1	C40
	2	TEXT
	3	BASE256
	4	reserviert
	5	AUTO (Voreinstellung)
i	B	EAN/UCC-Modus *mit* Klammern um den Datenbezeichner. Die Daten müssen mit Klammern gesendet werden, die Klammern werden aber nicht kodiert gedruckt. Handhabung wie EAN 128 Barcode.
	X	EAN/UCC-Modus *ohne* Klammern um den Datenbezeichner. Die Daten müssen ohne Klammern gesendet werden. Handhabung wie EAN 128 Barcode.
r	Rn	n = Reihenzahl
c	Sn	n = Spaltenzahl
└		Standardmäßig wird die Größe (Anzahl Reihen und Spalten) in Abhängigkeit von den Nutzdaten automatisch berechnet (kleinst mögliche Matrix). Mit den Parametern r und c kann die Größe vorgegeben werden (siehe Tabelle unten).

k	F	<FNC1> wird als Daten-Trennzeichen verwendet (--> Beispiel)
	G	<GS> wird als Daten-Trennzeichen verwendet
└		Nur anwendbar mit Maschinen die zur Firmware-Gruppe BEL gehören (siehe Kapitel Gültigkeit auf Seite 7).

s	int	Anzahl der Drucker-Dots für einen Data Matrix Block [1...200]

Reihen	Spalten		Reihen	Spalten
10	10		64	64
12	12		72	72
14	14		80	80
16	16		88	88
18	18		96	96
20	20		104	104
22	22		120	120
24	24		132	132
26	26		144	144
32	32		8	18
36	36		8	32
40	40		12	26
44	44		12	36
48	48		16	36
52	52		16	48

Tabelle 13: Zulässige Kombinationen aus Reihen (r) und Spalten (c).

Beispiel

EAN/UCC-Modus:

```
#ER
#SDM/B/14#G
#T10#J5
#T10#J5
#VW/L/"(01)34012345123457(10)12345 <FNC1>(17)101231"#G
#Q1/
```

Unterschiedliche Daten-Trennzeichen:

```
#SDM/X/14#G    --> <FNC1> wird als Daten-Trennzeichen verwendet
#SDM/XF/14#G   --> <FNC1> wird als Daten-Trennzeichen verwendet
#SDM/XG/14#G   --> <GS> wird als Daten-Trennzeichen verwendet
```

#SF - Fixfont Definition

Das Kommando #SF legt den Zeichensatz für das Drucken von nicht skalierbarem Text fest (Fixfont). Das Drucken des Textes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SFz/k/b#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Zeichensatznummer (näheres siehe Link unter „Zusätzliche Informationen“)
k	Snum	Fester Zeichenabstand zwischen den Zeichen der Zeichenkette TEXT. Die Eingabe erfolgt in Millimetern. Der Abstand wird von Zeichenanfang zu Zeichenanfang gemessen. Das „S“ muss ohne Leerzeichen vor dem Dezimalwert „num“ stehen.
b	int	Mikroabstand in Druckkopf-Punkten. Der Mikroabstand wird zu dem Standard-Abstand der Zeichen addiert und erlaubt es, die Schriftbreite sehr genau einzupassen. Wertebereich [0…16].

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#SFN - Code 49 Definition

Das Kommando #SFN definiert einen Barcode vom Typ „Code 49“. Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SFNm/kx/s/h#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
m	0	Alphanumeric Mode
	1	Append Mode
	2	Numeric Mode
	3	Group Alphanumeric Mode
	4	Alphanumeric Mode, Shift 1
	5	Alphanumeric Mode, Shift 2
	6	Reserved
	7	Automatischer Mode (Voreinstellung und empfohlener Modus). Der Drucker bestimmt automatisch den Start-Modus und Kodier-Modus indem er den Text analysiert.

k	M	Barcode mit Klarschriftzeile Klarschriftzeile kann über den rechten Rand des Barcodes hinausgehen
	O	Barcode ohne Klarschriftzeile (Voreinstellung)
x	J	Klarschriftzeile unter dem Barcode
	A	Klarschriftzeile über dem Barcode

h	int	Berechnung der Zeilenhöhe: Zeilenhöhe = (h + 1) * DRUCK PARAMETER > Barcode Multi

#SG - Grafik Definition

Das Kommando #SG definiert eine Grafik. Das Drucken des Grafik wird durch ein nachfolgendes #VW-Kommando veranlasst.

Syntax

```
#SG#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Beispiel

Kommandofolge	Beschreibung
`#SG#G`	Grafik definieren
`#T82#J75`	Druckposition bestimmen
`#VW/L/"Grafik01.bmp" #G`	Grafik drucken

#SI - Daten an Schnittstelle ausgeben

Das Kommando #SI definiert den Auslösezeitpunkt für die Kontext-Schnittstelle (Easy Plug-Schnittstelle).
Dieser Befehl ist für viele Anwendungen nützlich, z. B. dem „Loggen“ gedruckter Etiketten oder der „Echtzeit“-Steuerung über die Schnittstelle.

Das Schreiben von Daten an die aktuelle Easy Plug-Schnittstelle wird mit dem Befehl #VW/I/Ausdruck erreicht. Der Zeitpunkt, an dem die Daten ausgegeben werden (engl. trigger point), wird mit dem #SI-Befehl festgelegt.

Syntax

```
#SIr/n/TEXT#G
```

Parameter	Wert	Bedeutung
r	int	Logischer Zeitpunkt (Auslösezeitpunkt) in der Druckablaufsteuerung (siehe Abb. auf der nächsten Seite), der festlegt, wann Daten gesendet werden sollen. Wenn ein Fehler in der Ablaufsteuerung auftritt, während ein Etikett verarbeitet wird, werden eventuell anstehende spätere Auslöser nicht berücksichtigt. RFID: Um Daten auszugeben, die vom Transponder gelesen oder zum Transponder gesendet wurden, wird r = 8 empfohlen. Dadurch wird sichergestellt, dass die RFID-Operationen und das Drucken erfolgreich abgeschlossen wurden.
	1	Start eines neuen Easy Plug-Formats
	2	Neustart eines vorhergehenden Easy Plug-Formats (#Q ohne #ER)
	3	Start der Etikettenverarbeitung
	4	Start einer RFID-Operation
	5	Ende einer *fehlerfreien* RFID-Operationen; Wenn die Lese-/ Schreib-Operation nicht erfolgreich war, werden keine Daten ausgegeben
	6	Start des tatsächlichen Druckvorgangs
	7	Am Ende eines *fehlerfreien* Druckvorgangs; Wenn das Etikett nicht fehlerfrei gedruckt wurde, werden keine Daten ausgegeben
	8	Am Ende einer *fehlerfreien* Etikettenverarbeitung
	9	Am Ende eines Easy Plug-Formats (#Qn Etiketten gedruckt)
	10	Bei Fehlern der Bewegungssteuerung
	11	Bei RFID-Fehlern (Tag Lese- oder Schreibfehler); Datenausgabe und Bad Tag Signal erfolgen gleichzeitig
	12	Gespendetes Etikett wurde abgenommen (64-xx Spender)
	13	Ende des Scanbereiches des Online-Verifiers (OLV)

n	int	Anzahl Antwortbytes
	≠ 0	Der Rückgabewert (definiert mit #VW) wird auf n Antwortbytes beschnitten. Wenn der Rückgabewert weniger als n Bytes zählt, wird er mit Leerzeichen (Hex Code 0x20) aufgefüllt.
	0	Die Länge des Rückgabewertes ist so lang wie nötig, um den Rückgabewert zu senden (Voreinstellung).

TEXT		Header, der als Präfix für die Antwortdaten verwendet wird

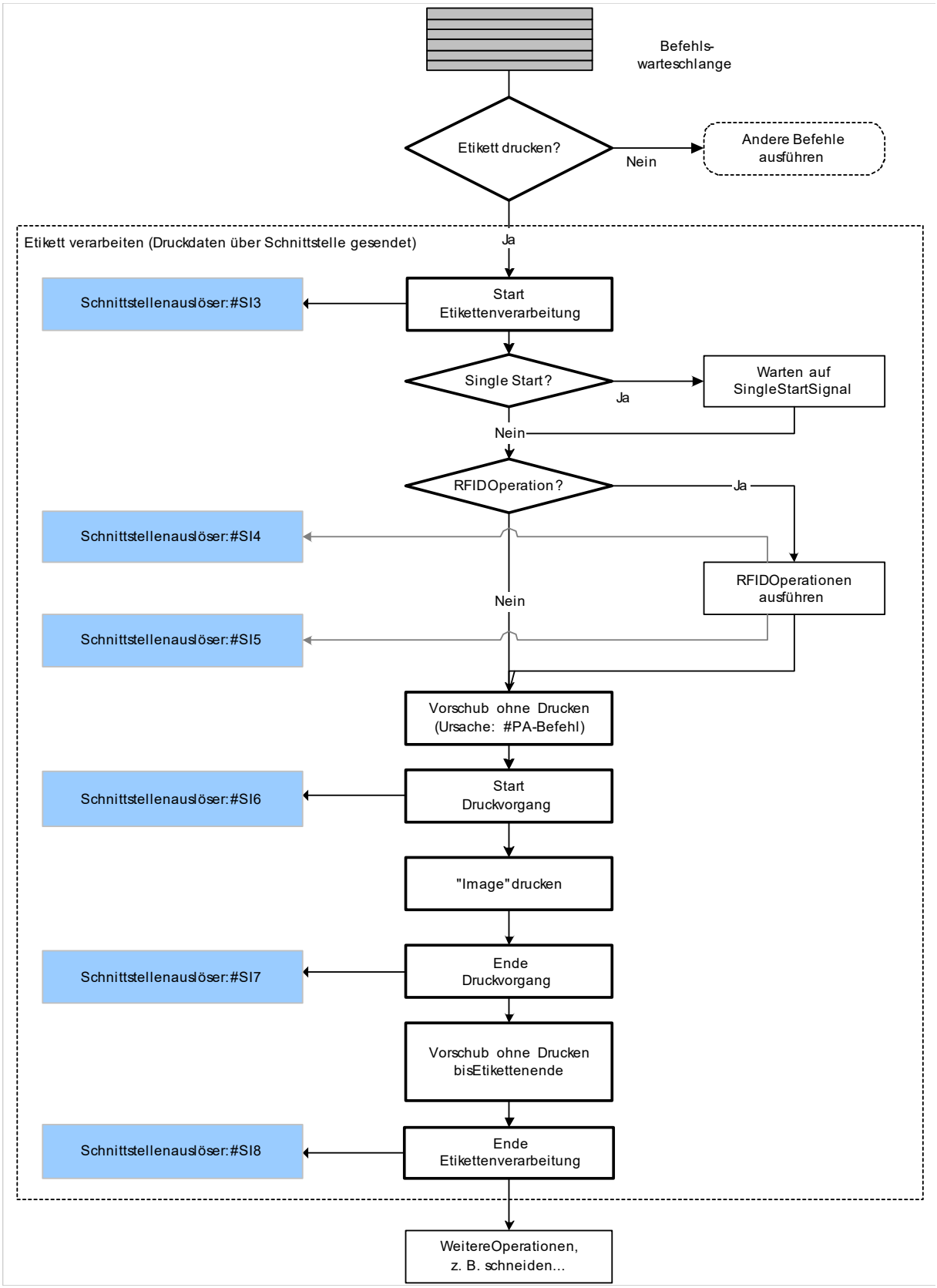


Bild 14: Das Flussdiagramm zeigt die Ausgabezeitpunkte (trigger points) #3 - #8 während des Druckens.

Beispiele

Einfaches Beispiel:

```
#G ----------------------------------------------------------------
#G Gibt den Zählerstand des Druckers zurück, wenn das Etikett fehlerfrei
#G gedruckt wurde
#G ----------------------------------------------------------------
#!A1
#IMN100/100
#ER
#G -----------------------------------
#G Zählerstand als Text definieren und drucken
#G -----------------------------------
#VDT/Counter/1/1/0000#G
#SF109/
#FD0
#T10#J20
#VW/L/Counter#G
#G -----------------------------------
#G Zählerstand an der Schnittstelle ausgeben, sobald ein Etikett
#G fehlerfrei gedruckt wurde.
#G -----------------------------------
#SI8/#G
#VW/I/chr(10) + chr(13) + "CNT: " + Counter#G
#Q3/
#G ----------------------------------------------------------------
#G Ausgabe an der aktiven Schnittstelle
#G CNT: 0000
#G CNT: 0001
#G CNT: 0002
#G ----------------------------------------------------------------
```

Komplexeres Beispiel:

```
#G ----------------------------------------------------------------
#G Zeigt den Gebrauch verschiedener Auslöseereignisse
#G ----------------------------------------------------------------
#!A1
#IMN100/100
#ER
#G -----------------------------------
#G Fehlercode an Schnittstelle ausgeben, wenn beim Drucken ein Fehler
#G aufgetreten ist.
#G -----------------------------------
#VDS/ErrorCode//M1011#G
#SI10/#G
#VW/I/chr(10) + chr(13) + "  Fehlercode: " + ErrorCode#G
#G -----------------------------------
#G Mitteilungen bezüglich des Easy-Plug-Formats
#G -----------------------------------
#SI1/#G
#VW/I/chr(10)+chr(13)+chr(10)+chr(13) + "*** Start eines neuen Easy-Plug
 Formates *** "#G
#SI2/#G
#VW/I/chr(10)+chr(13)+chr(10)+chr(13) + "*** Easy-Plug-Format Neustart
*** "#G
#SI9/#G #VW/I/chr(10) + chr(13) + "*** Easy-Plug-Format beendet *** "#G
#G -----------------------------------
#G Mitteilungen bezüglich des Etiketts
#G -----------------------------------
#SI3/#G
#VW/I/chr(10)+chr(13) + "  * Start Etikettenverarbeitung"#G
#G -----------------------------------
#G Zählerstand als Text definieren und drucken
```

```
#G ------------------------------------
#VDT/Counter/1/1/0000#G
#SF109/
#FD0
#T10#J20
#VW/L/Counter#G
#G ------------------------------------
#G Zählerstand an Schnittstelle schicken, sobald das Etikett fehlerfrei
#G gedruckt wurde.
#G ------------------------------------
#SI8/#G
#VW/I/chr(10) + chr(13) + "    Gedruckter Zählerstand: " + Counter#G
#VDT/TextVar///Text1#G
#SF109/
#FD0
#T10#J30
#VW/L/TextVar#G
#SI8/#G
#VW/I/chr(10) + chr(13) + "    Gedruckte TextVar: " + TextVar#G
#G ------------------------------------
#G Mitteilung Etikettenoperation beendet.
#G ------------------------------------
#VDD/Time/AU//^h:^m.^s#G
#SI8/#G
#VW/I/chr(10)+chr(13) + "  * Etikettenoperation beendet um: " +Time#G
#Q2/
#SV/TextVar/Neuer Text#G
#Q1/
#G ---------------------------------------------------------------------
#G Ausgabe an der aktiven Schnittstelle
#G *** Start eines neuen Easy-Plug Formates ***
#G  * Start Etikettenverarbeitung
#G    Gedruckter Zählerstand: 0000
#G    Gedruckte TextVar: Text1
#G  * Etikettenoperation beendet um: 13:07.07
#G  * Start Etikettenverarbeitung
#G    Gedruckter Zählerstand: 0001
#G    Gedruckte TextVar: Text1
#G  * Etikettenoperation beendet um: 13:07.08
#G *** Easyplug format finished ***
#G
#G *** Easy-Plug-Format Neustart ***
#G  * Start Etikettenverarbeitung
#G    Gedruckter Zählerstand: 0002
#G    Gedruckte TextVar: Neuer Text
#G  * Etikettenoperation beendet um: 13:07.09
#G *** Easy-Plug-Format beendet ***
#G ---------------------------------------------------------------------
```

#SMX - Maxicode Definition

Das Kommando #SMX definiert eine Barcode vom Typ „Maxicode“. Das Drucken des Maxicodes wird durch ein nachfolgendes #VW-Kommando veranlasst.

Syntax

```
#SMXz/x/y#G
```

Parameter	Wert	Bedeutung
z	int	MaxiCode-Modi [2,3,4,6]
x	int	Symbolnummer bei Codeaufteilung [1...8]
y	int	Anzahl der Symbole bei Codeaufteilung [1...8]

z	TEXT (Syntax)	Daten
2	`pppppppppp ccc sss MSG`	p: Numerischer Postcode (9 Zeichen) c: Zeichen ISO-Ländercode (3 Zeichen) s: Serviceklasse (3 Zeichen) MSG [5]: Codeworte (84 Zeichen)
3	`pppppp ccc sss MSG`	p: Numerischer Postcode (6 Zeichen) c: Zeichen ISO-Ländercode (3 Zeichen) s: Serviceklasse (3 Zeichen) MSG [5]: Codeworte (84 Zeichen)
	Bei den Modi 2 und 3 muss zwischen Postcode, Ländercode, Serviceklasse und Nutzdaten jeweils ein Leerzeichen stehen! Die Anzahl der Zeichen im Postcode, Ländercode und Serviceklasse muss exakt eingehalten werden.	
4	`MSG`	MSG: Beliebige alphanumerische Zeichen (93)
6	`MSG`	

Tabelle 14: Aufbau und Länge des Datenstrings „TEXT“ im zugehörigen Kommando #VDT in Abhängigkeit vom gewählten Modus „z“.

Einfaches Beispiel

Der String „MSG“ enthält nur alphanumerische Zeichen.

```
#!A1
#IMS100/200

#ER
#VDT/MaxicodeData////NOVEXX Solutions Teststring#G
#SMX4/
#T40#J45
#VW/L/MaxicodeData#G
#Q1#G
```

5 Abhängig davon, welche alphanumerischen Zeichen (0x00 – 0xff) verwendet werden, ergibt sich eine unterschiedliche Anzahl von Nutzzeichen, da mehr oder weniger häufig zwischen Subsets umgeschaltet werden muss.

```
#!P1
```

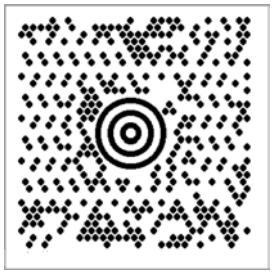


Bild 15: Ausdruck des Beispiels.

Beispiel mit Zeichen < 20h

Wichtig: Parameter Zeichen Filter auf „Alle Zeichen“ stellen!

Der String „MSG“ kann auch nicht darstellbare Zeichen enthalten, das sind Zeichen mit einem hexadezimalen Code <20h. Solche Zeichen sind z. B. die Steuerzeichen <RS> (0x20) und <GS> (0x1E). Spezielle Editoren machen diesen unsichtbaren Code durch Platzhalter sichtbar:

```
#!A1
#IMS100/200

#ER
#VDT/MaxicodeData////[)>RS01GS96123GS840GS111GS1234567890GSUPSNGS123456GS222GS1GS1
/2GS10GSYGSTestGSTest CGSTeRSEOT#G
#SMX4/
#T40#J45
#VW/L/MaxicodeData#G
#Q1#G

#!P1
```

Solche Zeichenketten machen es schwer, den Quellcode zu lesen oder zu ändern. Der unten abgebildete Quellcode zeigt eine gleichwertige Alternative, die aber leichter gelesen und geändert werden kann:

```
#!A1
#IMS100/200

#ER
#VDE/RS//Chr(30)#G
#VDE/GS//Chr(29)#G
#VDE/EOT/Chr(4)#G
#VDE/Header//"[" +Chr(41)+ ">" + RS + "01" + GS#G     !! use Chr(41) ins▶
tead of ) to prevent a known bug.

#VDE/MaxicodeData//Header + "96123" + GS + "840" + GS + "111" + GS +
 "1234567890" + GS + "UPSN" + GS + "123456" + GS + "222" + GS + "1" + GS
 + "1/2" + GS + "10" + GS + "Y" + GS + "Test" + GS + "Test C" + GS + "Te"
 + RS + EOT#G
#SMX4/
#SMX4/
#T40#J10
#VW/L/MaxicodeData#G
#Q1#G

#!P1
```

Bild 16: Ausdruck des Beispiels.

Zugehörige Informationen

#MXC - Maxicode auf Seite 54
Das Kommando #MXC druckt einen Barcode vom Typ Maxicode.

#SPF - PDF417 Definition

Das Kommando #SPF definiert einen Barcode vom Typ „PDF417“. Das Drucken des Barcodes wird durch ein nachfolgendes #VW-Kommando veranlasst.

Syntax

```
#SPFnt/s/l/z/w/h#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
n		Kompressionsart
	0	EXC-Mode (Extended alphanumeric Compaction Mode)
	1	Binary ASCII Plus Mode
t	leer	Leserichtung bidirektional (Voreinstellung)
	T	Leserichtung unidirektional
s	int	Security Level [0...8]
l	0	Barcodebreite wird automatisch eingestellt
	int	Barcodebreite in Codewörtern (Spaltenzahl) [1...30]: Die Breite soll so gewählt werden, dass in Zusammenhang mit der sich aus den Nutzdaten ergebenden Fläche des Barcodes die Höhe von 90 Zeilen nicht überschritten wird. Maximal sind 928 Codewörter für einen Barcode zulässig. Davon sind fest vergeben: • 1 Wort für die Längenangabe des Codes • x Worte für Checksumme (Security Level) = 2 (1 + Security Level)
z	0	Zeilenzahl wird automatisch eingestellt

	int	Zeilenzahl [3...90]

w	int	Barcodebreite [1...16]

h	num	Höhe einer PDF-Barcodezeile [1...100] mm

#SQR - QR Matrix Code Definition

Das Kommando #SQR definiert einen Barcode vom Type „QR Matrix Code“. Das Drucken des Barcodes wird durch ein nachfolgendes #VW-Kommando veranlasst.

Syntax

```
#SQRm/ei/s/an/d/p#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Anwendungsschema:

1. Variable definieren:

```
#VDT/QR_Data////qr code#G
```

2. Umgebungsparameter setzen:

```
#SQR2/LU/6///#G
#FD/0/L#G#T5.0#J10.0
```

3. Auf Kontextmedium “Etikett” schreiben:

```
#VW/L/QR_Data#G
```

Parameter	Wert	Bedeutung
m	int	
	1	QR Matrix Code Modell 1 (Original-Standard)
	2	QR Matrix Code Modell 2 (Erweiterter Standard [6]; Voreinstellung)

e		Fehlerkorrektur-Niveaus:
	L	7% Hohe Dichte
	M	15% Standard (Voreinstellung)
	Q	25% Hohe Zuverlässigkeit
	H	30% Sehr hohe Zuverlässigkeit
i		Zeichensätze:
	A	Automatisch (Eingabe-Datentyp wird automatisch gewählt; Voreinstellung)
	U	Benutzerdefiniert (Eingabe-Datentyp wird vom Benutzer festgelegt, siehe Tabelle unten)

s	int	Modulgröße in Pixel (Minimum: 4; Voreinstellung: 4)

6 Festgelegt in ISO/IEC 18004:2000

a	S	Einzelnes Symbol (Voreinstellung)
	A	Aktiviert “Structured Append” (engl. für “strukturiertes Anhängen”, d. h. ein QR-Code-Symbol kann in mehrere kleinere Symbole aufgeteilt werden; so kann z. B. der zur Verfügung stehende Raum besser ausgenutzt werden)
n	int	Symbol-Sequenznummer: [1...“Anzahl der Sequenzen“] (gilt, wenn “Structured Append“ aktiviert ist)

d	int	Symbolanzahl: [2...16] (gilt, wenn “Structured Append“ aktiviert ist)

p	int	Parität: [0...255] (gilt, wenn “Structured Append“ aktiviert ist)

Kennzeichen	Datentyp
N	*Numerische Daten* (Zahlen 0 – 9, z. B. **N**0123456789); jeweils 3 Ziffern werden als 10-Bit-Einheit gespeichert
A	*Alphanumerische Daten* (Zahlen 0-9; Großbuchstaben A-Z; Neun weitere Zeichen, z. B. **A**---123-ABC %$+-/:); jeweils 2 Zeichen werden als 11-Bit-Einheit gespeichert
B	*Binärdaten*. Die Anzahl Bytes muss nach dem “B” durch eine 4stellige Zahl spezifiziert werden (z. B. **B**0023)
K	*Kanji-Zeichen*; diese können auf 13 Bit komprimiert werden

Tabelle 15: Zulässige Benutzerdefinierte Datentypen. Der Buchstabe für den Datentyp muss den Eingabedaten unmittelbar vorangestellt sein.

Beispiel

```
#!A1
#IMN100.0/100.0//
#N10
#ERN//
#VDT/QR_Data_StructuredAppend_Part1////AAAAAAAA#G
#VDT/QR_Data_StructuredAppend_Part2////BBBBBBBB#G
#VDT/QR_Data_StructuredAppend_Part3////CCCCCCCC#G
#VDT/QR_Data_StructuredAppend_Part4////DDDDDDD#G
#VDT/QR_Data_Manual////N0123456789,A---123-ABC %$+-/:,B0004ÿÿÿ#G
#VDE/QR_Data_4PartsInOne/QR_Data_StructuredAppend_Part1+
QR_Data_StructuredAppend_Part2+QR_Data_StructuredAppend_Part3+
QR_Data_StructuredAppend_Part4#G
```

`#FD/0/L#G`
`#SQR2/QU/4///#G`
`#T40.0#J80.0`
`#VW/L/QR_Data_Manual#G`	

`#G --- Binärdaten "Test<cr><lf>"`
`#SQR2/QU/4///#G`
`#T20.0#J40.0`
`#VW/L/"B0006Test"+CHR(10)+CHR(13)#G`	

`#G --- Alles auf einmal`
`#SQR2/LA/4///#G`
`#T50.0#J25.0`
`#VW/L/QR_Data_4PartsInOne#G`	

`#G --- Structured Append 1..4`
`#SQR2/LA/4/A1/4/255#G`
`#T20.0#J10.0`
`#VW/L/QR_Data_StructuredAppend_Part1#G`	

`#SQR2/LA/4/A2/4/255#G`
`#T40.0#J10.0`
`#VW/L/QR_Data_StructuredAppend_Part2#G`	

`#SQR2/LA/4/A3/4/255#G`
`#T60.0#J10.0`
`#VW/L/QR_Data_StructuredAppend_Part3#G`	

`#SQR2/LA/4/A4/4/255#G`
`#T80.0#J10.0`
`#VW/L/QR_Data_StructuredAppend_Part4#G`
`#Q1#G`	

#SRF - RFID Lese-/Schreib-Definition

Das Kommando #SRF definiert Ziel (Schreiben) oder Quellfeld und Adresse (Lesen) der folgenden #VW/T oder #VR/T-Befehle.

Syntax

```
#SRFt/bfs/n/r#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
t	0	BLOCK Parameter „s“ und „n“ erforderlich
	1	UID/TID Parameter „n“ optional, um den Datenumfang zu begrenzen, der bei Leseoperationen zurückgegeben wird
	2	EPC
	3	KILL password
	4	ACCESS password
	5	MEM BANK
	6	AFI

Parameter	Wert	Bedeutung
	7	MEMBLOCK Parameter „s“ und „n“ erforderlich
	8	DSFID

b	L	Das niedrigstwertige Byte jedes Blocks (LSB) auf dem Transponder wird zuerst gelesen (Voreinstellung).
	M	Das höchstwertige Byte jedes Blocks (MSB) auf dem Transponder wird zuerst gelesen.
f	P	Schreibschutz
s	int	Wenn Datentyp = • BLOCK: Adresse des Startblocks • MEM BANK: Adresse/Name der Memory Bank; (Voreinstellung: 0) • MEMBLOCK: Kombination aus Bezeichner für Memory Bank und Startadresse in der Memory Bank. Schema für die Adressberechnung: `s = <EPC MemoryBankID> * 1000 + <BlockNumber>` EPC MemoryBankID: – 0: reserviert (Voreinstellung) – 1: EPC – 2: TID – 3: USER – andere Werte werden ignoriert • Anderer Datentyp: Parameter wird ignoriert

n	int	Wenn... • Datentyp = BLOCK, TID, MEM BANK oder MEMBLOCK: Anzahl der Blöcke, die gelesen werden sollen (Voreinstellung: 0). • Anderer Datentyp: Parameter wird ignoriert Wenn ein TID-Feld (t=1) gelesen wird, kann n verwendet werden, um den Umfang der gelesenen Daten zu begrenzen. Wenn keine Größe angegeben wird (n=0), wird der gesamte Inhalt der TID Memory Bank gelesen, der unerwartet viele Informationen erhalten kann (abhängig vom RFID Transponder/Chip). Typische Werte für n sind hier: • 2: Lese 32Bit Vendor-ID/Chip-ID • 4: Lese 32Bit Vendor-ID/Chip-ID + 32Bit Unique Serial Number

r	int	Anzahl der Lese-/Schreibversuche (Voreinstellung: Wie in RFID PARAMETER > CMD Wiederholung eingestellt).

Von RFID-Etikett lesen / Auf RFID-Etikett schreiben

Für das Lesen von Daten von einem bzw. das Schreiben von Daten auf ein RFID-Etikett sind immer eine Variable und zwei Kommandos erforderlich:

1. Variable definieren: #VDT (oder #VDE / #VDD)
2. Zielfeld oder Datentyp definieren: #SRF
3. Daten lesen oder schreiben: #VR oder #VW

Zugehörige Informationen

#VDT - Textvariable definieren auf Seite 110
#VW - Datenausgabe auf Ausgabeziel auf Seite 113

#SRS - GS1 Barcode Definition

Das Kommando #SRS definiert einen Barcode vom Type „GS1 Databar“ (vormals RSS). Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

Syntax

```
#SRSzt/s#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Barcode-Nummer (siehe Link unter „Zusätzliche Informationen“) GS1 DataBar Expanded *Stacked* wird von Easy-Plug automatisch eingesetzt, wenn folgende Bedingungen erfüllt sind: • Gewählter Barcode-Typ: GS1 DataBar Expanded (z = 6) • Breite (x) ist auf weniger als 22 Symbolzeichen pro Reihe eingestellt • Die Nutzdaten überschreiten die eingestellte Breite
t	Sn	Breite in Segmenten; Wertebereich: n = [4…22]
s	int	Anzahl der Drucker-Dots für die Modulbreite des Barcodes.

#SS - Speedo-Font Definition

Das Kommando #SS legt den Zeichensatz (Font) für das Drucken von skalierbarem Text fest (Speedo-Font). Erst ein nachfolgendes #VW-Kommando veranlasst das Drucken des Textes.

Syntax

```
#SSz/ghrkp/ex/s#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Zeichensatznummer (näheres siehe Link unter „Zusätzliche Informationen“) Bei Angabe einer ungültigen Zeichensatznummer wird automatisch mit Zeichensatz Nr. 100 gedruckt.
g	O	Voreingestellte Schrift
	B	Fettdruck (Bold)
h	V	Voreingestellte Schrift
	I	Kursivdruck (Italics)
r	K	(Voreinstellung) Verwendung von skalierbaren Fonts ohne Unicode
	U	Verwendung von Unicode-Fonts
	G	Verwendung von arabischen Unicode-Fonts (Leserichtung von rechts nach links)
k	Snum	Fester Zeichenabstand zwischen den Zeichen der Zeichenkette TEXT. Die Eingabe erfolgt in Millimetern. Der Abstand wird von Zeichenanfang zu Zeichenanfang gemessen. Der Buchstabe „S“ muss ohne Leerzeichen vor dem Zahlenwert „num“ stehen.
p	N	Standard-Zeichenalgorithmus wird verwendet.
	Q	2D-Zeichenalgorithmus wird verwendet. Das kann die Abbildungsgenauigkeit bei kleinen Zeichen erhöhen (z.B. beim Drucken chinesischer Zeichen). Der Algorithmus ist um 30-40% langsamer als der Standard-Algorithmus.
e	int	Fontgröße in Dot in y-Richtung, jeweils abhängig vom Font und dessen installierter Größe. Wertebereich: [5…6000] Dot
x	Xint	Fontgröße in Dot in x-Richtung, jeweils abhängig vom Font und dessen installierter Größe. Wertebereich: [5…6000] Dot. Der Buchstabe „X“ muss ohne Leerzeichen vor dem Zahlenwert „int“ stehen.
s	int	Mikroabstand in Druckkopf-Punkten. Der Mikroabstand wird zu dem Standard-Abstand der Zeichen addiert und erlaubt es, die Schriftbreite sehr genau einzupassen. Wertebereich [0…16].

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#SV - Wert einer Textvariablen ändern

Syntax

```
#SV/name/TEXT#G
```

Das Kommando darf *nicht* zwischen #ER und #Q stehen!

Parameter	Wert	Bedeutung
name		Name der Variablen (Muss vorher mit #VDT definiert werden!)
TEXT		Neuer Wert der Variablen (max. Länge: 4096 Zeichen)

T

#T - Horizontale Druckposition

Das Kommando #T legt die horizontale Druckposition fest.

Syntax

```
#Tx
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
x	num	Horizontale Druckposition in mm bezogen auf den horizontalen Nullpunkt. Horizontaler Nullpunkt ist stets die Position des ersten Dots des Druckkopfes. Dieses befindet sich in Vorschubrichtung gesehen am rechten Ende des Druckkopfes. Ausnahme: Der Nullpunkt wurde mit dem Kommando #R verschoben. Wenn der Druckkopf an den inneren Anschlag geschoben ist (in Vorschubrichtung ganz rechts), befindet sich das erste Dot 1 mm von der Materialnulllinie entfernt. Mit anderen Worten: In Vorschubrichtung gesehen bleibt am rechten Rand des Etiketts ein Streifen von 1 mm Breite unbedruckt.

Text, der außerhalb der definierten Etikettenfläche liegt, wird nicht gedruckt.

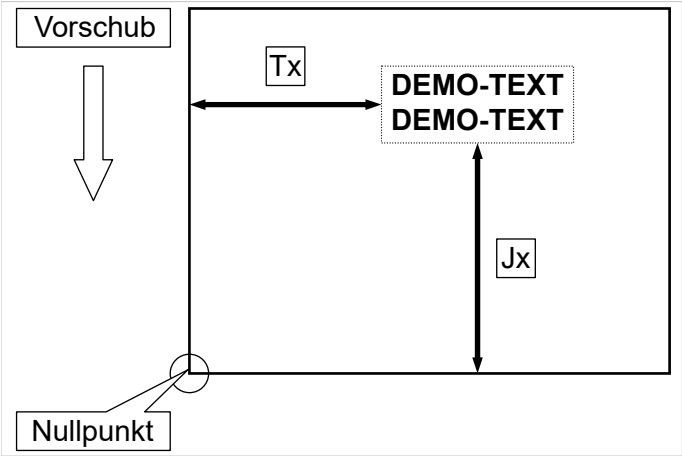


Bild 17: Vertikale (Jx) und horizontale (Tx) Druckposition auf dem Etikett.

Beispiel

Text beginnt 5 mm von links:

```
#T5
```

Text beginnt 20 mm von links:

```
#T20
```

V

#VC - RFID-Daten vergleichen/verifizieren

Syntax

```
#VC/Kontext/Ausdruck#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
Kontext	T	Daten werden vom RFID-Transponder gelesen und mit dem Inhalt einer Variablen verglichen.
Ausdruck		Siehe Link unten

Zugehörige Informationen

Ausdrücke auf Seite 159

#VDD - Datums-/Zeitvariable definieren

Das Kommando #VDD definiert eine Datums- oder Zeitvariable.

Syntax

```
#VDD/name/uv/o/TIMETEXT#G
```

Der Drucker muss mit einer Echtzeituhr ausgestattet sein.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
u	leer W	(Voreinstellung) Kein Update. Die Zeitinformation wird einmal in die Variable eingelesen. Dies geschieht, während das Layout kompiliert wird.
	U	Die Zeitinformation wird mehrmals, während des laufenden Druckjobs, eingelesen. Dies geschieht sofort nach dem Drucken des entspechenden Ausgabefeldes (#VW…), gilt aber erst für dasselbe Feld auf dem nächsten Etikett. Die Einstellung „U“ ist nicht für Spender geeignet. Begründung: Wenn das nächste Etikett erst wesentlich später gedruckt wird (z.B. im Single-Start-Betrieb mit Fußschalter), passiert folgendes: Das Ausgabefeld auf Etikett 2 enthält die Zeit, zu der dasselbe Feld auf Etikett 1 gedruckt wurde. Beispiel: Etikett 1 wird um 10.50 Uhr gedruckt, Etikett 2 erst um 11.03 Uhr. Dann enthält der Aufdruck auf Etikett 2 die Uhrzeit 10.50!

	Y	(Für Spender empfohlen) Die Zeitinformation wird mehrmals, während ein Etikett gedruckt wird, eingelesen (ca. alle 5 Sekunden).
v	A	Zeitinformation darf innerhalb eines Etikettenlayouts unterschiedlich sein (wenn U oder Y angewendet werden).

o	leer	Das aktuelle Datum wird ausgedruckt.
	0	
	int	*Tage-Offset:* Das aktuelle Datum wird um die angegebenen Anzahl Tage nach vorne gerechnet [1...65000].
	Mint	*Monats-Offset:* Das aktuelle Datum wird um die angegebene Anzahl Monate nach vorne gerechnet [1...65000]. Durch Voranstellen des Buchstaben M wird aus dem Tage- ein Monats-Offset.
	Pint	*Minuten-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Minuten nach vorne gerechnet [1...65000].
	Hint	*Stunden-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Stunden nach vorne gerechnet [1...65000].

TIMETEXT		String aus Text (ASCII-Zeichen) und Zeitangaben in beliebiger Kombination. Zeitangaben (siehe unten) werden mit dem Control-Zeichen (^) gekennzeichnet. Für den Ausdruck vorgesehenen Text als ASCII-Zeichen (ohne Control-Zeichen) in den TIMETEXT-String einfügen.

Formate für Zeitangaben:

^z	1/100 Sekunden
^s	Sekunden
^m	Minuten
^h	Stunden
^D	Tag im Monat
^d	Tag im Jahr
^W	Tag im Jahr (3-stellige Nummer des Tages im Jahr, z.B. 001 für den 1. Tag)
^w	Wochentag (1-stellige Nummer des Tages in der Woche, z. B. 1 für Montag)
^M	Monat
^Y	Jahr, zweistellig
^R	Jahr, vierstellig
^G	3-stellige Monatsangabe Deutsch
^E	3-stellige Monatsangabe Englisch
^S	Monatsname, wie er mit #DM definiert wurde
^C	Kalenderwoche immer zweistellig
^c	Kalenderwoche ohne vorangestellte 0
^K	Passendes Jahr zur Kalenderwoche, 4stellig
^k	Passendes Jahr zur Kalenderwoche, 2stellig

Hinweis zu ^K und ^k: Die Jahreszahl wird aus der Kalenderwoche bestimmt. Da das Jahr oft mitten in der ersten Woche beginnt, können die ersten Tage des Jahres noch zur letzten KW des vorangegangen Jahres zählen (siehe Abb. unten).

Bild 18: Die Jahreszahl für den 1.1.2017 wäre in diesem Beispiel noch 2016, da das Datum noch in KW 52 liegt.

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159

#VDE - Ausdrucks-Variable definieren

Das Kommando #VDE definiert eine Variable, der ein Ausdruck zugewiesen werden kann.

Syntax

```
#VDE/name/o/expression#G
```

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
o		Nicht genutzter Parameter für zukünftige Firmware-Versionen
expression		Ausdruck - kann Text, Variablen und Funktionen enthalten

Beispiel

```
#VDE/BarcodeData//"(00)"+NVE+MOD10(NVE)#G
```

mit:

" (00)"	Text
+	Operator

NVE	Variable
MOD10(NVE)	Funktion MOD10 mit Variabler NVE als Argument

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159

#VDO - Variable für OLV-Datenzugriff definieren

Das Kommando #VDO definiert eine Variable für den Zugriff auf Daten eines Online-Verifiers (OLV).

Syntax

```
#VDO/name/TEXT#G
```

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
TEXT		OLV-Datenelement (siehe unten)

Beschreibung OLV-Datenelement

Für jeden Barcode, der vom OLV gescannt wurde, wird ein Eintrag in einem Datenfeld angelegt. Auf die einzelnen Einträge kann über einen Index zugegriffen werden. Die Indexierung beginnt mit „1“. Die Nummernvergabe erfolgt in der Reihenfolge, in der die Oberkanten der Barcodes in Vorschubrichtung auf dem Etikett erscheinen (siehe Abb. unten). Jeder Eintrag enthält die in der Tabelle aufgelisteten Datenelemente:

Datenelement	Wert	Bedeutung
Barcode[idx].Data		Barcode-Daten
Barcode[idx].bLimitsOK	0	Die Grenzen des Barcodes sind aus einem der folgenden Gründen nicht in Ordnung: • Barcode wurde nicht gelesen • Die eingestellten Grenzen wurden nicht eingehalten
	1	Die Grenzen des Barcodes sind in Ordnung
Barcode[idx].bChecked	0	Der Barcode konnte vom OLV nicht gelesen werden
	1	Der Barcode wurde gelesen
OLVError	0	Alle Barcodes wurden fehlerfrei gedruckt und gelesen
	1	Keine OLV Daten verfügbar
	2	OLV-Grenze überschritten
	4	Falscher Barcode-Typ

Tabelle 16: Datenelemente und ihre möglichen Werte.

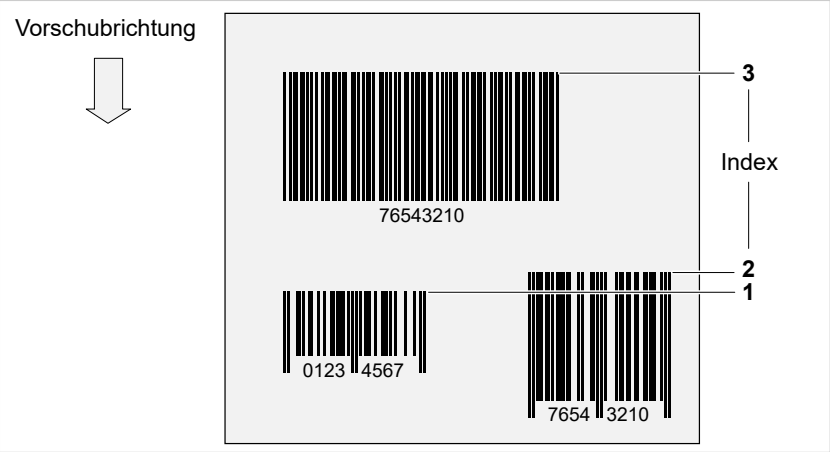


Bild 19: Die Indexierung der Barcodes erfolgt in der Reihenfolge der Barcode-Oberkanten in Vorschubrichtung auf dem Etikett.

Ein Fehler bei der Benennung der Variable führt zu der Meldung:

```
Queuestatus 2011
OLV Variable
```

Mögliche Fehlerursachen sind z. B.:

- „Barcode“ falsch geschrieben, z. B. „Baarcode[3].Data“ oder „Barcode[3].Date“
- Falsche Indexnummer angegeben, z. B: „Barcode[36].Data“ wenn nur 26 Barcodes gelesen wurden

Beispiel

```
#IMN109.0/30
#ERN0///0
#SI1/#G
#VW/I/"<SOL>"+chr(10)+chr(13)#G
#G --------------------
#G Variablendefinition
#G --------------------
#VDO/OLVD1/Barcode[1].Data#G
#VDO/OLVOK1/Barcode[1].bLimitsOK#G
#VDO/OLVD2/Barcode[2].Data#G
#VDO/OLVOK2/Barcode[2].bLimitsOK#G
#VDO/OLVD3/Barcode[3].Data#G
#VDO/OLVOK3/Barcode[3].bLimitsOK#G
#VDO/OLVErr/OLVError#G
#G ------------------------------
#G Schreiben an die Schnittstelle
#G ------------------------------
#SI13/#G Auslösepunkt am Ende des Scanvorgangs
#VW/I/OLVD1 + IfEqualThenElse(OLVOK1,"0"," NOK"," OK")+chr(10)+chr(13)#G
#VW/I/OLVD2 + IfEqualThenElse(OLVOK2,"0"," NOK"," OK")+chr(10)+chr(13)#G
#VW/I/OLVD3 + IfEqualThenElse(OLVOK3,"0"," NOK"," OK")+chr(10)+chr(13)#G
#VW/I/"Abgebrochen : "  + IfEqualThenElse(OLVErr,"0","Nein","Ja")
 +chr(10)+chr(13)#G
#T5.4#J7.4#YB13/0O/10/3///1100640210#G
#T40.4#J7.4#YB13/0O/10/3///1100640211#G
#T75.5#J7.4#YB13/0O/10/3///1100640212#G
#SI8/#G
#VW/I/"<EOL>"+chr(10)+chr(13)#G
#Q1#G
```

Datenausgabe an der Easy-Plug-Schnittstelle:

- Ohne Etikettenabbruch:

```
<SOL>
1100640210 OK
1100640211 OK
1100640212 OK
Abgebrochen: Nein
<EOL>
```

- Mit Etikettenabbruch:

```
<SOL>
1100640210 NOK
1100640211 OK
1100640212 NOK
Abgebrochen : Ja
1100640210 OK
1100640211 OK
1100640212 OK
Abgebrochen : Nein
<EOL>
```

#VDP - Zugriff auf Druckauftrags-Daten

Das Kommando #VDP ermöglicht den Zugriff auf Druckauftragsdaten wie „Druckmenge“ oder „Etikettennummer“. Mit diesen Daten können z. B. Info-Etiketten gedruckt werden (z. B. Etikett x von y, wobei x = aktuelles Etikett und y = Etikettenmenge).

Syntax

```
#VDP/name/format/JobItemID#G
```

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
format		Format der Systemvariablen. Das Format kann aus den folgenden Platzhaltern bestehen:
	leer	Keine Angabe. Das voreingestellte Format „%i“ (Zahl) wird verwendet.
	%[flag] [width]i	**flag = 0** Füllt anstelle von Leerzeichen auf der linken Seite mit „0“ auf, wenn Auffüllen spezifiziert wurde (siehe „width“) **width = Zahl** Mindestanzahl an Zeichen, die gedruckt werden soll. Wenn der zu druckende Zahlenwert kürzer ist als diese Zeichenanzahl, wird links mit Leerzeichen aufgefüllt. Wenn der Wert größer ist als die Zeichenanzahl, wird dieser trotzdem vollständig gedruckt.
JobItemID	I1	Nummer des aktuellen Etiketts im Druckauftrag
	I2	Druckmenge des aktuellen Druckauftrags
	I3	Gesamtnummer des aktuellen Etiketts (falls der Druckauftrag mit #Q ohne #ER wiederholt wurde)

	I4	Gesamtdruckmenge des aktuellen Druckauftrags (falls der Druckauftrag mit #Q ohne #ER wiederholt wurde)

Beispiel

```
#!A1
#IMN104/35
#ER
#VDP/CurrentLabel/%3i/I1#G
#VDP/CurrentQuantity//I2#G
#VDP/TotalLabel/%03i/I3#G
#VDP/TotalQuantity/%06i/I4#G
#FD/0/L#G
#SF109#G
#T5#J15
#VW/L/"Label "+CurrentLabel+" of " + CurrentQuantity#G
#T5#J5
#VW/L/"Total "+TotalLabel+" of " + TotalQuantity#G
#Q3/
#Q2/
```

Ausdruck:

```
Label 1 of 3
Total 001 of 000003
Label 2 of 3
Total 002 of 000003
Label 3 of 3
Total 003 of 000003
Label 1 of 2
Total 004 of 000005
Label 2 of 2
Total 005 of 000005
```

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159

#VDS - Systemvariable definieren

Das Kommando #VDS definiert eine Systemvariable, in die Name oder Wert jedes beliebigen Parameters aus dem Parametermenü des Druckers übernommen werden können.

Syntax

```
#VDS/name/format/RemoteID#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.

format		Format der Systemvariablen. Das Format kann aus den folgenden Platzhalter bestehen:
	%N	Name des Parameters wird übernommen.
	%S	Wert + Einheit des Parameters werden als Zeichenkette (String) übernommen (Darstellung wie auf der Anzeige des Druckers).
	%i	Wert des Parameters wird als ganzzahliger Wert (Integer) übernommen (Voraussetzung für „%i“: Der Parameter muss vom Typ Integer sein)

RemoteID	int	ID-Nr. eines Parameters Eine Liste aller ID-Nummern kann mit Hilfe von SPEZIALFUNKTION > Param. speichern bzw. Werkzeuge > Diagnose > Param. speichern generiert werden. Siehe auch Kommando #!PG.
	Mint	ID-Nr. eines Maschinenstatus Liste der ID-Nummern siehe Kommando #!XMn.

Beispiele

Beispiel Parameter:

```
#G ----------------------------------------------------------------
#G Gegebene Einstellung: SYSTEM PARAMETER > Temperaturreduz. = „20%“
#G ----------------------------------------------------------------
#VDS/my_variable/%N/2026#G
#G  ------> Inhalt von my_variable: „Temperaturreduz.“
#VDS/my_variable/%S/2026#G
#G  ------> Inhalt von my_variable: „ 20% “
#VDS/my_variable/%i/2026#G
#G  ----> Inhalt von my_variable: „ 20“
```

Beispiel Maschinenstatus:

```
#!A1
#IMN100/100
#ER
#G ----------------------------------
#G Fehlercode an Schnittstelle ausgeben, wenn beim Drucken ein Fehler
 aufgetreten ist.
#G ----------------------------------
#VDS/ErrorCode//M1011#G
#SI10/#G
#VW/I/chr(10) + chr(13) + "  Fehlercode: " + ErrorCode #G
```

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159

#VDT - Textvariable definieren

Das Kommando #VDT definiert eine Textvariablen mit Zählfeld.

Syntax

```
#VDT/name/wz/vop/a/TEXT#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
w	W	Das Zählfeld TEXT wird *ohne* Übertrag um den Offset erhöht oder verringert (lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert)
	C	Das Zählfeld TEXT wird *mit* Übertrag um den Offset erhöht oder verringert.
z	Z	Keine Unterdrückung führender Nullen im Zählfeld.
	S	Unterdrückung führender Nullen im Zählfeld. Die restlichen Ziffern werden an derselben Position gedruckt, an der sie mit führenden Nullen stehen würden.
v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]
a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].
TEXT		Beliebiger alphanumerischer Text, wobei nur numerische Werte für den Offset berücksichtigt werden. Es werden alle numerischen Werte des Textstrings zum Offset herangezogen (z. B. 10A3B56 = 10356). Maximal zulässige Zeichenzahl: 255.

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159
#SRF - RFID Lese-/Schreib-Definition auf Seite 96
#VW - Datenausgabe auf Ausgabeziel auf Seite 113

#VR - RFID-Daten lesen

Syntax

```
#VR/Kontext/Ausdruck#G
```

Nur mit eingebauter und aktivierter RFID-Option.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
Kontext	T	Daten werden vom RFID-Transponder gelesen.
Ausdruck		Siehe Link unten

Zugehörige Informationen

Ausdrücke auf Seite 159

#VTS - Standalone-Variable definieren

Das Kommando #VTS definiert eine Variable, die ihren Inhalt im Standalone-Betrieb über das Drucker-Bedienfeld zugewiesen bekommt.

Syntax

```
#VTS/name/wzy/l/vop/a/TEXT#G
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
name		Name der Variablen. Zulässig sind alle Zeichen mit Ausnahme von einigen Sonderzeichen, siehe Kapitel Vereinbarungen auf Seite 9.
w	W	Das Zählfeld TEXT wird *ohne* Übertrag um den Offset erhöht oder verringert (lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert)
	C	Das Zählfeld TEXT wird *mit* Übertrag um den Offset erhöht oder verringert.
z	Z	Keine Unterdrückung führender Nullen im Zählfeld.
	S	Unterdrückung führender Nullen im Zählfeld. Die restlichen Ziffern werden an derselben Position gedruckt, an der sie mit führenden Nullen stehen würden.
y	U	Update des Standalone-Eingabefeldes. Im Eingabefeld erscheint nach jedem Durchlauf des Druckjobs der zuletzt erreichte Zählwert aus dem vorhergehenden Durchlauf.
	N	*Kein* Update des Standalone-Eingabefeldes. Nach jedem Durchlauf des Druckjobs erscheint der voreingestellte Zählwert im Eingabefeld auf der Anzeige des Druckers.
l	int	Maximale Länge des Standalone-Eingabefeldes.

v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Text, der als Voreinstellung auf dem Display angezeigt wird. Beliebiger alphanumerischer Text. Die Textlänge sollte 16 Zeichen (= Länge einer Anzeigezeile) nicht überschreiten. Maximal zulässige Zeichenzahl: 255.

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Ausdrücke auf Seite 159

#VW - Datenausgabe auf Ausgabeziel

Syntax

```
#VW/Kontext/Ausdruck#G
```

Parameter	Wert	Bedeutung
Kontext	T	Ausgabeziel = RFID-Tag, d. h. der Wert des Ausdrucks wird an den RFID-Transponder gesendet.
	I	Ausgabeziel = Schnittstelle, d.h. der Wert des Ausdrucks wird an die in SCHNITTST. PARA > >OPTIONEN > #VW/I Schnittst. bzw. Drucker Sprache > Easy-Plug Einst. > #VW/I Schnittst. eingestellte Schnittstelle gesendet.
	L	Ausgabeziel = Etikett, d.h. der Wert des Ausdrucks wird gedruckt.

Ausdruck		Siehe Link unten

Zugehörige Informationen

Ausdrücke auf Seite 159
#SRF - RFID Lese-/Schreib-Definition auf Seite 96
#VDT - Textvariable definieren auf Seite 110

Y

#YB - Barcode definieren und drucken

Das Kommando #YB druckt ein Barcode-Feld.

Syntax

```
#YBz/dkbcejgrwlm/h/s/vop/a/TEXT#G
```

Das Kommando muß zwischen #ER und #Q stehen!

Das Kommando muss mit #G abgeschlossen werden!

Parameter	Wert	Bedeutung
z	int	Barcode-Nummer (näheres siehe Link unter „Zusätzliche Informationen“)
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
k	M	Barcode mit Klarschriftzeile
	O	Barcode ohne Klarschriftzeile
b	A	Positionsänderung der Klarschriftzeile. Wird die Klarschriftzeile normalerweise unter dem Barcode gedruckt, wird sie mit dieser Option über den Barcode verschoben und umgekehrt.
c	C	Es wird eine Prüfziffer errechnet und ausgedruckt. Gilt nur für Barcodes mit *optionaler* Prüfziffer.
e	V	Barcode wird überprüft (siehe Beispiel unten). Nur mit angeschlossenem und aktiviertem OLV. Nur für Barcodes mit 0° oder 180° Druckrichtung möglich.
j	Z	*Zentriert*: Das Feld wird links und rechts der Druckposition mittig aufgebaut.
	R	*Rechtsbündig*: Das Feld wird rechtsbündig der Druckposition aufgebaut Das Feld wird nach links aufgebaut.
g	D	Textfeld besteht aus einem variablen Datenfeld
r	Pnum	Ratio des Barcodes [2.0...3.0] Der Buchstabe P (Proportion) muß direkt vor der Ratio stehen (z.B. P2,5). Eine Ratio ohne den Buchstaben P ist ungültig. Reihenfolge beachten: Die Ratio unbedingt nach dem Parameter d (Schreibrichtung) eingeben!
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.
l	H	Klarschriftzeile im Blocksatz (Voreinstellung)

	I	Klarschriftzeile linksbündig angeordnet
	K	Klarschriftzeile mittig angeordnet
	L	Klarschriftzeile rechtsbündig angeordnet
m	B	EAN/UCC-Mode mit Klammern um die Datenbezeichner. Die Nutzdaten müssen mit Klammern geschickt werden! Die Klammern erscheinen zwar in der Klarschriftzeile, sie werden jedoch nicht im Barcode codiert.
	X	EAN/UCC-Mode ohne Klammern um die Datenbezeichner. Die Nutzdaten müssen ohne Klammern geschickt werden!

h	int	Die *Barcode-Höhe* berechnet sich wie folgt: Barcode-Höhe = (h + 1) mm * DRUCK PARAMETER > Barcode Multi *bzw.* Barcode-Höhe = (h + 1) mm * Druck > Format > Barcode Multi Die Voreinstellung für Barcode Multi ist 1 Damit ergeben sich z. B. folgende Werte für die Barcode-Höhe: • 1 mm Barcode-Höhe für h = 0 • 2 mm Barcode-Höhe für h = 1 • 3 mm Barcode-Höhe für h = 2

s	int	Barcode-Breitenfaktor [1...30] Dot

v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Beliebiger alphanumerischer Text. Die Vorgaben für den jeweiligen Barcode sind zu beachten. Maximal zulässige Zeichenzahl: 255. Das Textfeld kann auch ein *Variables Datenfeld* sein. Voraussetzung: D-Flag gesetzt. Das Textfeld kann auch ein *Eingabefeld* enthalten.

Beispiele

Kommando	Beschreibung
`#YB2/0M/2/2///1234567890123#G`	UPCA mit Klarschriftzeile, normale Druckrichtung, 3 mm hoch, Breite 2, ohne Offset.
`#YB0/0/////12345678` `#YB11/0/////54321#G`	EAN 8 und ADD on 5 werden in Reihe aneinander gestellt.
`#YB1/0D/10/3///$01,12#G`	Barcode Nummer 1 mit variablem Datenfeld Nummer 01 und 12 alphanumerischen Zeichen.
`#YB15/0B/2/3///(10)Charge1#G`	EAN/UCC-Mode mit Klammern um den Daten-bezeichner (hier: 10).
`#YB15/0X/2/3///10Charge1#G`	EAN/UCC-Mode ohne Klammern um den Daten-bezeichner (hier: 10).
`#YB27/0M/10/3/o/a/xxxxxxxyyyyyyyy#G`	Code 128 Pharmacy xxxxxxx = 7stellige CNK-Nummer (Pharmacy Produkt Ident Nummer). yyyyyyyy = 8stellige sequentielle Nummer für den ersten Code.

Barcode prüfen:

Kommandofolge	Beschreibung

`#!A1`
`#IMS100/100`
`#ER`
`#T5#J5`
`#YT109/0///OLV Test`
`#OLVD/ /70///`	

`#T5#J50 #YB0/0V/4/4///7777777`	Barcode wird überprüft (Dekodierbarkeit >= 70)
`#T5#J30#YB0/0/8/8///1234567` `#OLVD/ /60//30/`	Barcode wird nicht überprüft
`#T5#J70 #YB0/0V/8/8///8888888`	Barcode wird überprüft (Dekodierbarkeit >= 70 und Defekte <= 30)
`#Q1/`	

Zugehörige Informationen

Eingabefelder auf Seite 10
Variable Datenfelder auf Seite 9
Druckerinterne Barcodes auf Seite 153
Zusammenhang zwischen Ratio und Breitenfaktor von Barcodes auf Seite 175

#YC - Echtzeit als Text

Mit dem Kommando #YC kann eine Zeitangabe in Form von Text definiert werden. In Verbindung mit der Zeitangabe kann auch normaler Text angegeben werden.

Syntax

```
#YCz/dbqjuk/o/TIMETEXT
```

Kommando muß zwischen #ER und #Q stehen!

Es sind beliebig viele Kommandos #YC im Format möglich. Pro Druckauftrag können jedoch nur maximal 3 Kommandos (#YC oder #YS) mit der Option U (Aktualisierung während des Druckjobs) versehen werden.

Parameter	Wert	Bedeutung
z	int	Zeichensatznummer (näheres siehe Link unter „Zusätzliche Informationen“)
d	0	Text in normaler Schreibrichtung
	1	Text um 90 Grad gedreht
	2	Text um 180 Grad gedreht
	3	Text um 270 Grad gedreht
b	A	Text invers gedruckt Die Druckfläche muss vorher mit einem dunklen Hintergrund (Linie oder Rechteck) hinterlegt werden.
q	E	Bestehendes Bitimage wird invertiert (siehe Beispiel 1 unten).
j	M	*Mittig*: Das Schreibkommando (#T/#J) bezieht sich auf die Mitte des Textfelds. Das Feld wird beidseitig der Druckposition aufgebaut.
	R	*Rechtsbündig*: Das Schreibkommando (#T/#J) bezieht sich auf den rechten Rand des Textfelds. Das Feld wird nach links aufgebaut.
u	leer	(Voreinstellung) Kein Update. Die Zeitinformation wird einmal in die Variable eingelesen. Dies geschieht an dem Zeitpunkt, an dem das Layout kompiliert wird.
	W	
	U	Die Zeitinformation wird mehrmals, während des laufenden Druckjobs, eingelesen. Dies geschieht sofort nach dem Drucken des entspechenden Ausgabefeldes (#VW…), gilt aber erst für dasselbe Feld auf dem nächsten Etikett. Einstellung ist nicht für Spender geeignet. Begründung: Wenn das nächste Etikett erst wesentlich später gedruckt wird (z.B. im Single-Start-Betrieb mit Fußschalter), passiert folgendes: Das Ausgabefeld auf Etikett 2 enthält die Zeit, zu der dasselbe Feld auf Etikett 1 gedruckt wurde. Beispiel: Etikett 1 wird um 10.50 Uhr gedruckt, Etikett 2 erst um 11.03 Uhr. Dann enthält der Aufdruck auf Etikett 2 die Uhrzeit 10.50!
	Y	(Für Spender empfohlen) Die Zeitinformation wird mehrmals, während ein Etikett gedruckt wird, eingelesen (ca. alle 5 Sekunden).
k	Snum	Maß für den festen Abstand aller in TEXT enthaltenen Zeichen in mm. Der Buchstabe S muss ohne Leerzeichen vor dem Zahlenwert stehen.
o	leer	Das aktuelle Datum wird ausgedruckt.
	0	

	int	*Tage-Offset:* Das aktuelle Datum wird um die angegebenen Anzahl Tage nach vorne gerechnet [1...65000].
	Mint	*Monats-Offset:* Das aktuelle Datum wird um die angegebene Anzahl Monate nach vorne gerechnet [1...65000]. Durch Voranstellen des Buchstaben M wird aus dem Tage- ein Monats-Offset.
	Pint	*Minuten-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Minuten nach vorne gerechnet [1...65000].
	Hint	*Stunden-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Stunden nach vorne gerechnet [1...65000].

TIMETEXT		String aus Text (ASCII-Zeichen) und Zeitangaben in beliebiger Kombination. Zeitangaben (siehe unten) werden mit dem Control-Zeichen (^) gekennzeichnet. Für den Ausdruck vorgesehenen Text als ASCII-Zeichen (ohne Control-Zeichen) in den TIMETEXT-String einfügen.

Formate für Zeitangaben:

^z	1/100 Sekunden
^s	Sekunden
^m	Minuten
^h	Stunden
^D	Tag im Monat
^d	Tag im Jahr
^W	Tag im Jahr (3-stellige Nummer des Tages im Jahr, z.B. 001 für den 1. Tag)
^w	Wochentag (1-stellige Nummer des Tages in der Woche, z. B. 1 für Montag)
^M	Monat
^Y	Jahr, zweistellig
^R	Jahr, vierstellig
^G	3-stellige Monatsangabe Deutsch
^E	3-stellige Monatsangabe Englisch
^S	Monatsname, wie er mit #DM definiert wurde
^C	Kalenderwoche immer zweistellig
^c	Kalenderwoche ohne vorangestellte 0
^K	Passendes Jahr zur Kalenderwoche, 4stellig
^k	Passendes Jahr zur Kalenderwoche, 2stellig

Hinweis zu ^K und ^k: Die Jahreszahl wird aus der Kalenderwoche bestimmt. Da das Jahr oft mitten in der ersten Woche beginnt, können die ersten Tage des Jahres noch zur letzten KW des vorangegangen Jahres zählen (siehe Abb. unten).

Bild 20: Die Jahreszahl für den 1.1.2017 wäre in diesem Beispiel noch 2016, da das Datum noch in KW 52 liegt.

Beispiele

1. Der Teil der Schrift, der in die Linie hineinragt, wird weiß gedruckt, der Rest schwarz:

```
#!A1
#ER
#T5#J5#YL0/0/2.5/90#G
#T5#J5#YC109/0E//Beispiel#G
#Q1/
```

2. Ausdruck, der während des laufenden Druckjobs aktualisiert wird: Uhrzeit 08:22.18,07 Datum 23-Nov-1998:

```
#YC105/0U// Uhrzeit ^h:^m.^s,^z Datum ^D-^G-^R#G
```

3. Es sei der 31.01.2001. Das untenstehende Kommando** führt zu dem Ausdruck "28.02.2001":

```
#YC109/0/M1/^D.^M.^R#G
```

4. Es sei der 13.09.2001. Das untenstehende Kommando führt zu dem Ausdruck "Tag im Jahr : 256":

```
#YC109/0/Tag im Jahr : ^d#G
```

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#YE - Kreis oder Ellipse definieren

Das Kommando #YE druckt einem Kreis oder eine Ellipse.

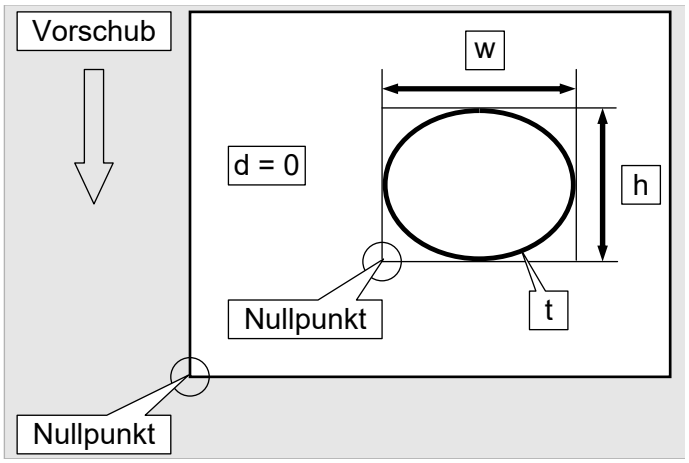
Syntax

```
#YE/d/t/w/h
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
d	0	Kreis/Ellipse in normaler Schreibrichtung
	1	Kreis/Ellipse um 90 Grad gedreht
	2	Kreis/Ellipse um 180 Grad gedreht
	3	Kreis/Ellipse um 270 Grad gedreht
t	num	Dicke der Linie in mm (min. 0,05 mm)
w	num	Breite des Kreises / der Ellipse in mm (min. 2,0 mm)
h	num	Höhe des Kreises / der Ellipse in mm (min. 2,0 mm)

Wenn w und h den gleichen Wert haben, wird ein Kreis gedruckt.



Beispiel

Ellipse 0 Grad gedreht, Linienstärke 2 mm, 25 mm breit, 15 mm hoch.

```
#YE/0/2/25/15
```

#YG - Ausdruck von Grafiken

Das Kommando #YG druckt eine Grafikdatei.
Unterstützte Grafikformate: BMP, PCX, JPG, TIF, GIF

Farb- bzw. Graustufengrafiken werden mit einem „Error Diffusion“ Algorithmus automatisch nach Schwarzweiß umgewandelt . Da dies unter Umständen sehr rechen- und speicherintensiv sein kann, wird empfohlen, Schwarzweißgrafiken zu verwenden.

Speicherbereich für das Laden und Konvertieren von Grafikdateien kann über den Parameter SYSTEM PARAMETER > Free Store Größe bzw. System > Speicher > Free Store Größe reserviert werden. Die aktuelle Aufteilung des Arbeitsspeichers finden Sie auf dem Statusausdruck INFO AUSDRUCKEN > Speicher Status bzw. Info > Status Ausdrucke > Speicher Status.

Syntax

```
#YG/djg/vo/a/Datei#G
```

Parameter	Wert	Bedeutung
d	0	Druckrichtung 0 Grad
	1	Druckrichtung 90 Grad
	2	Druckrichtung 180 Grad
	3	Druckrichtung 270 Grad
j	M	*Mittig*: Das Schreibkommando (#T/#J) bezieht sich auf die Mitte des Grafikfelds.Das Feld wird beidseitig der Druckposition aufgebaut.
	R	*Rechtsbündig*: Das Schreibkommando (#T/#J) bezieht sich auf den rechten Rand des Textfelds. Das Feld wird nach links aufgebaut.
g	D	Grafikfeld besteht aus einem variablen Feld. Definition wie beim #YT Kommado.

v	+	Inkrementieren – Offset wird zu Textfeld addiert
	-	Dekrementieren – Offset wird von Textfeld subtrahiert
o	int	Offset, der je nach Vorzeichen zum Dateinamen addiert (inkrementiert) oder vom Dateinamen subtrahiert (dekrementiert) wird

a	int	Anzahl Etiketten mit gleich bleibender Nummer [1...255]

Datei		Dateiname (incl. Laufwerksbuchstabe und Pfad in Großbuchstaben) der Grafik die gedruckt werden soll. Wird kein Pfad angegeben, wird die Datei im Default Pfad `C:\Graphics` (auf dem externen Speichermedium) gesucht.

Beispiel

Grafik um 180 Grad gedreht (Datei muß sich im `C:\Graphics` Verzeichnis auf dem externen Speichermedium befinden).

```
#YG/2///EXAMPLE.BMP
```

Zugehörige Informationen

Dateioperationen auf Seite 149
Beschreibung der korrekten Pfadangabe für Dateioperationen.

#YI - Logo im EPT-Format in den Bildpuffer schreiben

Das Kommando #YI schreibt die dem Kommando folgenden Daten in den Bildpuffer-Speicher.

Syntax

```
#YI/s/s/s.../s#G
```

Kommando muß *zwischen* #ER und #Q stehen!

Das Kommando muß mit #G abgeschlossen werden!

Da die Bildinformation ausschließlich im Bildpuffer gespeichert ist, geht sie verloren, wenn sie über den Rand des Etikettenlayouts hinausgeschoben wird.

Ab Firmware x.33 wird der Befehl auch für mehrreihiges Layout unterstützt.

Parameter	Wert	Bedeutung
s	hex	Verschlüsselung einer Dot-Zeile der Logo-Matrix, hexadezimal zu je vier Dots von links nach rechts. • Zuordnung für hexadezimale Verschlüsselung: • – 1 = Dot wird gedruckt – 0 = Dot wird nicht gedruckt • Nicht gesetzte Dots am Zeilenende können entfallen. • Verschlüsselung jeder Dot-Zeile der Logo-Matrix durch einen s-Parameter. Reihenfolge: von unten (Zeile 1) nach oben. Für die hexadezimale Darstellung zulässig sind *Großbuchstaben* und *Ziffern*.

Beispiel

Logo mit nachfolgendem Aufbau wird direkt in den Imagebuffer geschrieben:

Zeile	Dotzeile	Hex
4	1111 1111 1111	/FFF
3	1111 0000 1111 11	/F0FB
2	1110 0000	/E0
1	1100 0000 0011 0101	/C035

```
#YI/C035/E0/F0FB/FFF#G
```

Zugehörige Informationen

Logos auf Seite 149
Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

#YIB - Logo mit Binärdaten direkt in Bildpuffer schreiben

Das Kommando #YIB schreibt die auf das Kommando folgenden binären Daten direkt in den Bildpuffer.Dabei werden die Daten direkt durch die 256 ASCII-Zeichen dargestellt, d.h. 8 Dots entsprechen einem zu übertragenden Byte.

Syntax

```
#YIBc/d/bbb...b
```

Kommando muß *zwischen* #ER und #Q stehen!

Ab Firmware x.33 wird der Befehl auch für mehrreihiges Layout unterstützt.

Da die Bildinformation ausschließlich im Bildpuffer gespeichert ist, geht sie verloren, wenn sie über den Rand des Etikettenlayouts hinausgeschoben wird.

Parameter	Wert	Bedeutung
c	int	Anzahl der Dot-Zeilen der Logo-Matrix. Maximale Zeilenzahl: 65535
d	int	Anzahl der Bytes pro Zeile. Maximal: 65535 Alle Zeilen müssen gleich lang sein
b	bin	Parameter für ein Byte. Jedes Byte verschlüsselt acht Dots. Zuordnung für binäre Verschlüsselung: • 1 = Dot wird gedruckt • 0 = Dot wird nicht gedruckt

Beispiel

Mit diesem Kommando werden 3 Zeilen mit je 4 Bytes (entspricht 4 * 8 = 32 Dots je Zeile) direkt in den Image buffer geschrieben:

```
Zeile 3: 01111100 10011100 10111100 11011100
Zeile 2: 01111011 10011011 10111011 11011011
Zeile 1: 01111010 10011010 10111010 11011010
```

```
#YIB3/4/zÜ¦+{ø+_|£+_
```

Zugehörige Informationen

Logos auf Seite 149
Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

#YIR - Logo im RLE-Format direkt in Bildpuffer schreiben

Das Kommando #YIR schreibt die auf das Kommando folgenden Daten direkt in den Bildpuffer. Die Daten sind im RLE Format (Run Length Format, Lauflängenformat) codiert.

Syntax

```
#YIRc/sss...s
```

Das Kommando muß *zwischen* #ER und #Q stehen!

Da die Bildinformation ausschließlich im Bildpuffer gespeichert ist, geht sie verloren, wenn sie über den Rand des Etikettenlayouts hinausgeschoben wird.

Parameter	Wert	Bedeutung
c	int	Anzahl der Dot-Zeilen der Logo-Matrix. Maximale Zeilenzahl: 65535.
s	hex	Verschlüsselung einer Dot-Zeile der Logo-Matrix. Der Parameter s besteht aus einer Folge von Hexadezimalzahlen (Hex-Zahlen). Zulässiger Wertebereich für jede der Hex-Zahlen: 00 bis FD. Die Werte FE und FF sind für *Steuerzeichen* reserviert, wie nachfogend beschrieben.

- *Start-Code*

  Der Start-Code muß am Anfang der Daten für eine Dot-Zeile stehen. Eine Dot-Zeile, die mehrmals in Folge gedruckt werden soll, muß mit dem Start-Code FF nur einmal eingegeben werden. Die Zahl nach FF gibt an, wie oft die Zeile wiederholt werden soll. Beispiel: Der Start-Code FF04 sorgt für vier gleiche Zeilen.

  – FE (hex): Start-Code für einmalig zu druckende Zeile.

  – FF (hex): Start-Code für mehrmals zu druckende Zeile.

- *End-Code*
  Die letzte Dot-Zeile einer Logo-Matrix muß mit dem End-Code FE abgeschlossen werden.

- *RLE-Format*
  Entscheidend für das Umwandeln von Binärcode in das RLE-Format ist der Wechsel von Einsen zu Nullen und umgekehrt. Zum Umwandeln von Binärcode in RLE-Format müssen zwischen Start- und End-Code lediglich die aufeinanderfolgenden Nullen und Einsen gezählt und als Hex-Zahl notiert werden. Einstelligen Hex-Zahlen wird eine Null vorangestellt.

Binärcode:	`0001111001111111`
RLE-Code:	`FE03040207FE` (Startcode FE, 03 Nullen, 04 Einsen, 02 Null)

Tabelle 17: Beispiel: RLE-Format

Sie können den RLE-Code nur dann wie im Beispiel dargestellt eingeben, wenn a) ihr Editor über einen Hex-Modus verfügt, b) der Hex-Modus eingeschaltet ist. Mit einem Editor, der nicht über einen Hex-Modus verfügt, müssten Sie anstelle jeder Hex-Zahl das entsprechende ASCII-Zeichen einfügen, beispielsweise ♥ statt 03.

- *Dot-Zeile*
  Zwischen Start- und End-Code stehen in Paaren von Hex-Zahlen die eigentlichen Daten einer Dot-Zeile. Die erste Hex-Zahl des Paars gibt die Anzahl aufeinanderfolgender Nullen an, die zweite Hex-Zahl die Anzahl aufeinanderfolgender Einsen.

  Beispiel: `05FC` (05 Nullen, 252 Einsen)

Überschreitet die Anzahl Nullen oder Einsen des Binärcodes 253, muß ein neues Hex-Zahlen-Paar begonnen werden (die Entsprechungen der Zahlen 254 und 255 sind die Start-Codes FE und FF).

Binärcode:	`262 Einsen, 3 Nullen, eine Eins`
RLE-Code:	`FE00FD00090301` (Start-Code FE, 00 Nullen, FD Einsen, 00 Nullen, 09 Einsen, 03 Nullen, 01 Einsen)

Tabelle 18: Beispiel: Dot-Zeile

Die 262 Einsen werden auf zwei Wertepaare aufgeteilt, da sie die Anzahl von 253 überschreiten: 00FD und 0009. Da der Binärcode am Anfang keine Nullen enthält ist das erste Byte im ersten und zweiten Wertepaar Null.

- *Zeilenlänge*
  Die RLE Zeilen können beliebige Länge haben.
- *Leerzeilen*
  Eine einzelne Leerzeile hat die Form FE0000.

  Mehrere Leerzeilen werden folgendermaßen dargestellt: Start-Code FF, gefolgt von der Anzahl gewünschter Leerzeilen.

  Beispiel: `FF04` (04 Leerzeilen in Folge); `FF01 = FE0000` (eine Leerzeile)

Nach FFxx (bei xx Leerzeilen) müssen keine Nullen folgen. Der Start-Code für die nächste Zeile darf direkt anschließen.

Beispiel

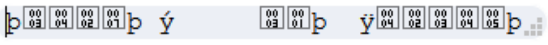
Mit diesem Kommando werden 7 Zeilen direkt in den Bildpuffer geschrieben:

- Zeile 7: 00111000011111
- Zeile 6: 00111000011111
- Zeile 5: 00111000011111
- Zeile 4: 00111000011111
- Zeile 3: leere Zeile
- Zeile 2: 111...(262 mal)...1110001
- Zeile 1: 0001111001111111

Kommandozeile:

```
#YIR7/FE03040207FE00FD00090301FE0000FF0402030405FE
```

ACHTUNG! - Diese Beispielzeile kann nicht aus dem pdf in ein Easy Plug-Format kopiert werden. Bevor diese Kommandozeile genutzt werden kann, müssen die Daten, die nach dem „/“ folgen und aus 2-Byte ASCII-Hex Code bestehen, in 1-Byte ASCII-Zeichen umgewandelt werden. Das Ergebnis sieht folgendermaßen aus:

#YIR7/ 

Zugehörige Informationen

Logos auf Seite 149
Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

#YK - Logo definieren

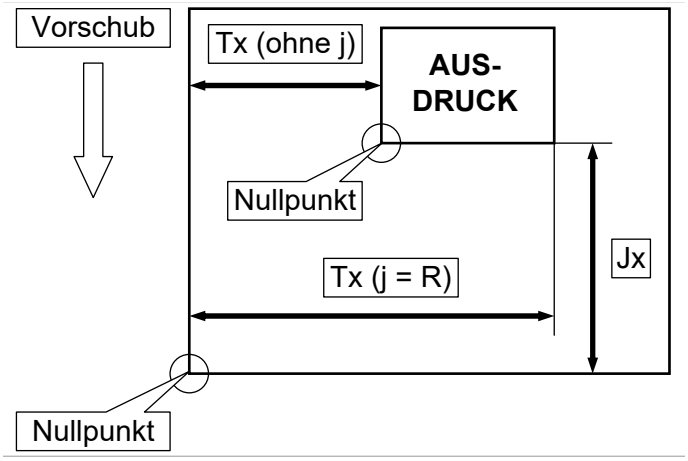
Der Befehl #YK druckt ein Logo (Klischee). Das Logo muss vorher in den Speicher geladen werden. Das Logo entspricht einem Datenblock und wird mit den Angaben zu einem Referenzpunkt (#T und #J Kommando) im Etikett positioniert.

Syntax

```
#YKn/djm
```

Kommando muß *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
n	int	Logo-Nummer (0 bis 255)
d	0	Logo in normaler Schreibrichtung
	1	Logo um 90 Grad gedreht
	2	Logo um 180 Grad gedreht
	3	Logo um 270 Grad gedreht
j	M	*Mittig*: Das Kommando #T bezieht sich auf die Mitte des Felds. Das Logo wird hierzu beidseitig aufgebaut.
	R	*Rechtsbündig*: Das Kommando #T bezieht sich auf den rechten Rand des Felds. Das Logo wird nach links aufgebaut.
m	leer	Logo wird auf der druckerinternen RAM-Disc erwartet (Voreinstellung).
	A	
	C	Logo wird auf dem externen Speichermedium erwartet.



Beispiel

Logo Nummer 10 wird ausgedruckt, falls vorhanden.

```
#YK10/0
```

Zugehörige Informationen

Logos auf Seite 149
Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

#YL - Linie definieren

Das Kommando #YL definiert Linien verschiedener Längen und Stärken.

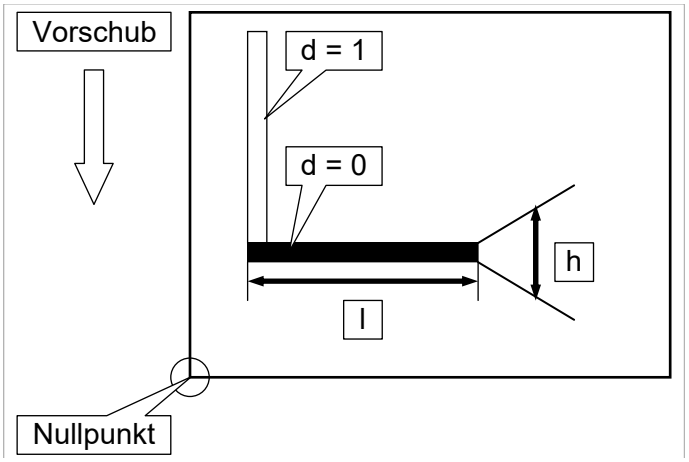
Syntax

```
#YLa/drz/h/l
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Werden Linien als Hintergrund gewünscht, müssen diese vor dem Vordergrund definiert werden.

Parameter	Wert	Bedeutung
a	int	Linienart-Nummer (näheres siehe Link unter „Zusätzliche Informationen“)
d	0	Linie in normaler Schreibrichtung
	1	Linie um 90 Grad gedreht
	2	Linie um 180 Grad gedreht
	3	Linie um 270 Grad gedreht
r	R	Die Linie wird bei jedem Etikett neu aufgebaut (z. B. als Hintergrund eines Nummernfelds). Parameter r nicht für invers gedruckte Felder verwenden! Feldinhalt wird sonst von der Linie überdruckt.
z	P	Normaler Ausdruck in schwarz
	A	„Ausdruck in weiß“, d.h. der Ausdruck wird freigelassen. Erfordert dunklen Hintergrund
	E	Ausdruck mit invertiertem Bitimage (schwarz wird freigelassen; weiß wird schwarz gedruckt)
h	num	Dicke der Linie in mm
l	num	Länge der Linie in mm



Beispiele

Linientyp 1, 0 Grad gedreht, 3 mm stark, 20 mm lang:

```
#YL1/0/3/20
```

Linientyp 4, 270 Grad gedreht, 1 mm stark, 50 mm lang

```
#YL4/3/1/50
```

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#YN - Textfeld

Das Kommando #YN druckt Text mit skalierbaren Fonts.
Das Kommando #YN definiert Größe und Font eines Textes sowie Drehrichtung und fortlaufende Nummerierung.

Bei fortlaufender Nummerierung ist Folgendes zu beachten:

- Es muss eine ausreichende Anzahl Stellen entsprechend der größten vorgesehenen Zahl definiert werden. Führende Nullen werden gedruckt.
- Nur numerische Zeichen werden inkrementiert bzw. dekrementiert.
- Alle numerischen Zeichen innerhalb eines Textstrings werden inkrementiert bzw. dekrementiert.

Syntax

```
#YNz/dbqmnjwcghrkp/e/vop/a/TEXT#G
```

Das #YN-Kommando muß *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Zeichensatznummer (näheres siehe Link unter „Zusätzliche Informationen“) Druckerintern stehen die skalierbaren Fonts 100 bis 102 zur Verfügung. Alternativ können Fonts von einem externen Speichermedium geladen werden (näheres siehe Handbuch für externe Speichermedien).

d	0	normale Schreibrichtung
	1	Text um 90 Grad gedreht
	2	Text um 180 Grad gedreht
	3	Text um 270 Grad gedreht
b	A	Text invers gedruckt Die Druckfläche muss vorher mit einem dunklen Hintergrund (Linie oder Rechteck) hinterlegt werden.
q	E	Bestehendes Bitimage wird invertiert (siehe Beispiel 1).
m	X	Textfeld besteht aus einem variablen Datenfeld
n	Y	Unterdrückung führender Nullen bei Zählfeldern. Die restlichen Ziffern werden an derselben Position gedruckt, an der sie mit führenden Nullen stehen würden (siehe Beispiel 2).
j	M	*Mittig*: Das Schreibkommando (#T/#J) bezieht sich auf die Mitte des Textfelds. Das Feld wird beidseitig der Druckposition aufgebaut.
	R	*Rechtsbündig*: Das Schreibkommando (#T/#J) bezieht sich auf den rechten Rand des Textfelds. Das Feld wird nach links aufgebaut.
f	F	Feldkennzeichnung für Datenwechsel (siehe Beschreibung zum Datenwechsel)
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.
c	Dint	Drehung von TEXT in Schritten von einem Grad [0...359] Grad; Voreinstellung: 0 Grad Der Buchstabe D muss ohne Leerzeichen vor dem Zahlenwert stehen. Parameter c hat Vorrang vor Parameter d.
g	O	Voreingestellte Schrift
	B	Fettdruck
h	V	Voreingestellte Schrift
	I	Kursive Schrift (Italics)
r	leer	Verwendung von skalierbaren Zeichensätzen ohne Unicode (Voreinstellung)
	K	
	U	Verwendung von Unicode-Zeichensätzen
	G	Verwendung von arabischen Unicode-Zeichensätzen (die Zeichen werden von rechts nach links gedruckt) Jedes Unicode-Zeichen muss innerhalb des TEXT-Bereichs über seinen Index in der folgenden Form aufgerufen werden: \u0041 (mit 0041 als Beispiel-Index für das Zeichen "A"; siehe Beispiel 2). Bei der Verwendung von Unicode-Zeichensätzen sind einige Besonderheiten zu beachten (siehe Link unten)!
k	Snum	Maß für den festen Abstand aller in TEXT enthaltenen Zeichen in mm. Der Buchstabe S muss ohne Leerzeichen vor dem Zahlenwert stehen.
p	N	Standard-Zeichenalgorithmus wird verwendet (Voreinstellung).

	Q	2D-Zeichenalgorithmus wird verwendet. Das kann die Abbildungsgenauigkeit bei kleinen Zeichen erhöhen (z.B. beim Drucken chinesischer Zeichen). Der Algorithmus ist um 30-40% langsamer als der Standard-Algorithmus.

e	int	Fontgröße [15...6000] Dot, jeweils abhängig vom Font und dessen installierter Größe. Wahlweise kann für die x-Richtung eine andere Fontgröße bestimmt werden, als für die y-Richtung (siehe Beispiel 3).

v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird
p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Beliebiger alphanumerischer Text, wobei nur numerische Werte beim Offset berücksichtigt werden. Es werden alle numerische Werte des Textstrings zum Offset herangezogen (z. B. 10A3B56 = 10356). Maximal zulässige Zeichenzahl: 255. Das Textfeld kann auch ein Variables Datenfeld sein. Voraussetzung: X-Flag gesetzt. Das Textfeld kann auch ein Eingabefeld enthalten.

Beispiele

Beispiel 1: Invertiertes Bitimage

```
#!A1
#ER
#T5#J5#YL0/0/2.5/90#G
#G ----------------------------------------------------------------
#G Der Teil der Schrift, der in die Linie hineinragt, wird weiß gedruckt,
 der Rest schwarz.
#G ----------------------------------------------------------------
#T5#J5#YN100/0E////Beispiel#G
#Q1/
```

Beispiel 2: Unterdrückung führender Nullen

```
#G ----------------------------------------------------------------
#G Ausdruck: Test  1, Test  2, …, Test 10, etc.
#G ----------------------------------------------------------------
#YN100/0Y/1/1//Test000#G
```

Beispiel 3: Speedo fonts skalieren

```
#G ----------------------------------------------------------------
#G Der Text „Speedo Test“ hat in y-Richtung Fontgröße 120, in
#G  x-Richtung Fontgröße 99.
#G ----------------------------------------------------------------
#YN100/0/120X99///Speedo Test#G
```

Beispiel 4: Speedo fonts skalieren

```
#G ----------------------------------------------------------------
#G Font 100 in Größe 10 Punkt.  Die 4-stellige Zahl wird bei jedem 5ten
#G Etikett um den Zähler "1" inkrementiert (5 x 0001, 5 x 0002).
#G ----------------------------------------------------------------
#YN100/0/10/1/5/0001#G
```

Beispiel 5: Unicode-Zeichen drucken

```
#G ----------------------------------------------------------------
#G Ausdruck: „Beispielausdruck von „A“ aus einem Unicode-Font: A“
#G ----------------------------------------------------------------
#!A1
#G ----------------------------------------------------------------
#G Endlosmaterial 100 mm breit, 30 mm lang
#G ----------------------------------------------------------------
#!IMN100/30 #ERN #T5#J10
#G ----------------------------------------------------------------
#G Font 902, Ausdruck in normaler Schreibrichtung, Unicode,
#G Fontgröße 60 Dot („\u0041“ ist der Unicode für „A“)
#G ----------------------------------------------------------------
#YN902/0U/60///Beispielausdruck von „A“ aus einem Unicode-Font: \u0041#G
#Q1#G
```

Beispiel 6:

```
#G ----------------------------------------------------------------
#G Dieses Easy Plug Programm generiert einen Ausdruck gemäß
#G der Abb. unten mit vier verschiedenen zweidimensionalen Barcodes
#G und 35-Grad-Drehung.
#G ----------------------------------------------------------------
#!A1
#G ----------------------------------------------------------------
#G Gestanztes Material 100 mm breit, 80 mm lang
#G ----------------------------------------------------------------
#IMS100/80
#ER
#J60#T15#YN100/0D15/60///Testfile "64-2DIM" 15 Degrees#G
#J48#T5#YN100/0/90///2D Barcodes as standard#G
#J10#T55#MXC4/0/2/3///1234567890abcdefghijklmnopqrstuivwxyz#G
#J10#T30#PDF0/T0/1/0/0/2/2/1234567890ABCDEFG#G
#J10#T5#IDM/0/10///1234567890abc#G
#J7#T5
#YT105/0///IDM (Data Matrix Code)   PDF-417   MXC (Maxi Code)#G
#J30#T5
#CBF/0/2/2/8/0///CODABLOCK F 345678901234567890l0040digit#G
#J26#T8#YT105/0///CODABLOCK F#G
#Q1/
```

Bild 21: Ausdruck zu Beispiel 6.

Beispiel 7:

```
#G ----------------------------------------------------------------
#G Druckt das chinesische Schriftzeichen (Abb. unten) mit dem Unicode  #G
 u50D1.
#G ----------------------------------------------------------------
#!A1
#IMN100/100/
#ERN
#T10#J10
#YN902/0U/400///\u50D1#G
#G ----------------------------------------
#G „\u50D1“ ist der Unicode für das gedruckte Zeichen
#G ----------------------------------------
#Q1#G
```

Bild 22: Ausdruck zu Beispiel 7.

Zugehörige Konzepte

Unicode auf Seite 172

Zugehörige Informationen

Eingabefelder auf Seite 10

Variable Datenfelder auf Seite 9
Druckerinterne Zeichensätze und Linienstile auf Seite 157
Arabische Glyphen auf Seite 176

#YR - Rechteck definieren

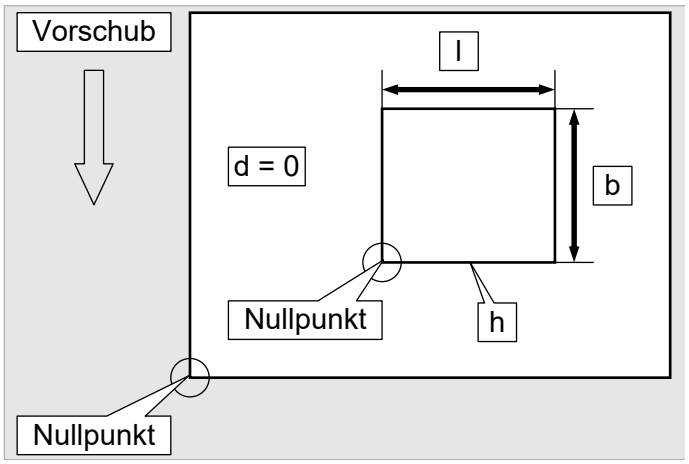
Das Kommando #YR definiert Rechtecke verschiedener Größen und Liniendicken.

Syntax

```
#YRa/dr/h/l/b
```

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
a	int	Linienart-Nummer (näheres siehe Link unter „Zusätzliche Informationen“)
d	0	Rechteck in normaler Schreibrichtung
	1	Rechteck um 90 Grad gedreht Das Rechteck rotiert um den Bezugspunkt.
	2	Rechteck um 180 Grad gedreht
	3	Rechteck um 270 Grad gedreht
r	R	Das Rechteck wird bei jedem Etikett neu aufgebaut (z. B. als Hintergrund eines Nummernfelds). Werden Rechtecke als Hintergrund gewünscht, müssen diese vor dem Vordergrund definiert werden.
h	num	Dicke der Linie in mm
l	num	Breite des Rechtecks in mm
b	num	Höhe des Rechtecks in mm



Beispiel

Rechteck mit Linienart 0, 0 Grad gedreht, Linienstärke 2 mm, 15 mm breit, 25 mm hoch.

```
#YR0/0/2/15/25
```

Zugehörige Informationen

Druckerinterne Zeichensätze und Linienstile auf Seite 157

#YS - Echtzeit als Barcode

Das Kommando #YS definiert eine Zeitangabe in Form eines Barcodes.

Syntax

```
#YSz/dkbcujlrm/h/s/o/ TIMETEXT#G
```

Der Drucker muss mit einer Echtzeituhr ausgestattet sein.

Das Kommando muss *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Barcode-Nummer (Näheres siehe Link unter „Zusätzliche Informationen“)
d	0	Barcode in normaler Schreibrichtung
	1	Barcode um 90 Grad gedreht
	2	Barcode um 180 Grad gedreht
	3	Barcode um 270 Grad gedreht
k	M	Barcode mit Klarschriftzeile
	O	Barcode ohne Klarschriftzeile
b	A	Positionsänderung der Klarschriftzeile. Wird die Klarschriftzeile normalerweise unter dem Barcode gedruckt, wird sie mit dieser Option über den Barcode verschoben und umgekehrt.
c	C	Es wird eine Prüfziffer nach Modulo 10 errechnet und ausgedruckt.
u	leer	(Voreinstellung) Kein Update. Die Zeitinformation wird einmal in die Variable eingelesen. Dies geschieht an dem Zeitpunkt, an dem das Layout kompiliert wird.
	W	
	U	Die Zeitinformation wird mehrmals, während des laufenden Druckjobs, eingelesen. Dies geschieht sofort nach dem Drucken des entspechenden Ausgabefeldes (#VW…), gilt aber erst für dasselbe Feld auf dem nächsten Etikett. Einstellung ist nicht für Spender geeignet. Begründung: Wenn das nächste Etikett erst wesentlich später gedruckt wird (z.B. im Single-Start-Betrieb mit Fußschalter), passiert folgendes: Das Ausgabefeld auf Etikett 2 enthält die Zeit, zu der dasselbe Feld auf Etikett 1 gedruckt wurde. Beispiel: Etikett 1 wird um 10.50 Uhr gedruckt, Etikett 2 erst um 11.03 Uhr. Dann enthält der Aufdruck auf Etikett 2 die Uhrzeit 10.50!
	Y	(Für Spender empfohlen) Die Zeitinformation wird mehrmals, während ein Etikett gedruckt wird, eingelesen (ca. alle 5 Sekunden).
j	Z	Feld wird mittig zur Druckposition aufgebaut

	R	Feld wird rechtsbündig zur Druckposition - d.h. von der Druckposition nach links – aufgebaut
	leer	(Voreinstellung) Feld wird linksbündig zur Druckposition – d.h. von der Druckposition nach rechts – aufgebaut
l	G	Position der Klarschriftzeile nach Voreinstellung
	H	Klarschriftzeile im Blocksatz
	I	Klarschriftzeile linksbündig angeordnet
	K	Klarschriftzeile mittig angeordnet
	L	Klarschriftzeile rechtsbündig angeordnet
r	Pnum	Ratio des Barcodes [2,0...3,0]. Die Standardeinstellung ist 2,5. Der Buchstabe P (Proportion) muß direkt vor der Ratio stehen (z.B. P2.5). Eine Ratio ohne den Buchstaben P ist ungültig. Reihenfolge beachten: Die Ratio unbedingt nach dem Parameter d (Schreibrichtung) eingeben!
m	B	EAN/UCC-Mode mit Klammern um die Datenbezeichner. Die Nutzdaten müssen mit Klammern geschickt werden! Die Klammern erscheinen zwar in der Klarschriftzeile, sie werden jedoch nicht im Barcode codiert.
	X	EAN/UCC-Mode ohne Klammern um die Datenbezeichner. Die Nutzdaten müssen ohne Klammern geschickt werden!

h	int	Barcode-Höhe: 0 = 5 mm, 1 = 10 mm, ... ab 10 = wirkliche Höhe in mm (z. B. 35 = 35 mm hoch). Dieser Wert minus 1 wird mit dem am Drucker eingestelltem Wert im Parameter BCHI multipliziert.

s	int	Barcodebreite [1...16]

o	leer	Das aktuelle Datum wird ausgedruckt.
	0	
	int	*Tage-Offset:* Das aktuelle Datum wird um die angegebenen Anzahl Tage nach vorne gerechnet [1...65000].
	Mint	*Monats-Offset:* Das aktuelle Datum wird um die angegebene Anzahl Monate nach vorne gerechnet [1...65000]. Durch Voranstellen des Buchstaben M wird aus dem Tage- ein Monats-Offset.
	Pint	*Minuten-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Minuten nach vorne gerechnet [1...65000].
	Hint	*Stunden-Offset:* Die aktuelle Zeit wird um die angegebene Anzahl Stunden nach vorne gerechnet [1...65000].

TIMETEXT		String aus Text (ASCII-Zeichen) und Zeitangaben in beliebiger Kombination. Zeitangaben (siehe unten) werden mit dem Control-Zeichen (^) gekennzeichnet. Für den Ausdruck vorgesehenen Text als ASCII-Zeichen (ohne Control-Zeichen) in den TIMETEXT-String einfügen.

Formate für Zeitangaben:

^z	1/100 Sekunden
^s	Sekunden
^m	Minuten
^h	Stunden
^D	Tag im Monat
^d	Tag im Jahr
^W	Tag im Jahr (3-stellige Nummer des Tages im Jahr, z.B. 001 für den 1. Tag)
^w	Wochentag (1-stellige Nummer des Tages in der Woche, z. B. 1 für Montag)
^M	Monat
^Y	Jahr, zweistellig
^R	Jahr, vierstellig
^G	3-stellige Monatsangabe Deutsch
^E	3-stellige Monatsangabe Englisch
^S	Monatsname, wie er mit #DM definiert wurde
^C	Kalenderwoche immer zweistellig
^c	Kalenderwoche ohne vorangestellte 0
^K	Passendes Jahr zur Kalenderwoche, 4stellig
^k	Passendes Jahr zur Kalenderwoche, 2stellig

Hinweis zu ^K und ^k: Die Jahreszahl wird aus der Kalenderwoche bestimmt. Da das Jahr oft mitten in der ersten Woche beginnt, können die ersten Tage des Jahres noch zur letzten KW des vorangegangen Jahres zählen (siehe Abb. unten).

Bild 23: Die Jahreszahl für den 1.1.2017 wäre in diesem Beispiel noch 2016, da das Datum noch in KW 52 liegt.

Beispiel

Ausdruck, der während des laufenden Druckjobs aktualisiert wird: Code 128 mit Klarschriftzeile "23-Nov-1998", normale Druckrichtung, 10 mm hoch, Breite 2.

```
#YS13/0MU/2/2//^D-^G-^R
```

Zugehörige Informationen

Druckerinterne Barcodes auf Seite 153

#YT - Textfeld definieren

Das Kommando #YT druckt Text mit Zeichensätzen in fester Größe (nicht skalierbar).

Syntax

```
#YTz/dqgnwjk/vop/a/TEXT#G
```

Das Kommando muß *zwischen* #ER und #Q stehen!

Parameter	Wert	Bedeutung
z	int	Zeichensatznummer (näheres siehe Link unter „Zusätzliche Informationen“) Bei Angabe einer ungültigen Zeichensatznummer wird automatisch mit Zeichensatz Nr. 100 gedruckt.
d	0	normale Schreibrichtung
	1	Text um 90 Grad gedreht
	2	Text um 180 Grad gedreht
	3	Text um 270 Grad gedreht
q	P	Normaler Ausdruck in schwarz
	A	„Ausdruck in weiß“, d.h. der Ausdruck wird freigelassen. Erfordert dunklen Hintergrund
	E	Ausdruck mit invertiertem Bitimage (schwarz wird freigelassen; weiß wird schwarz gedruckt); Siehe Beispiel 1
g	D	Textfeld besteht aus einem variablen Datenfeld
n	Y	Unterdrückung führender Nullen bei Zählfeldern. Die restlichen Ziffern werden an derselben Position gedruckt, an der sie mit führenden Nullen stehen würden. Siehe Beispiel 2.
w	W	Zählfeld TEXT wird ohne Übertrag inkrementiert/dekrementiert, d.h. lediglich die Einerstelle des Zahlenwertes wird erhöht/verringert.
j	M	*Mittig*: Das Schreibkommando (#T/#J) bezieht sich auf die Mitte des Textfelds. Das Feld wird beidseitig der Druckposition aufgebaut.
	R	*Rechtsbündig*: Das Schreibkommando (#T/#J) bezieht sich auf den rechten Rand des Textfelds. Das Feld wird nach links aufgebaut.
k	Snum	Maß für den festen Abstand aller in TEXT enthaltenen Zeichen in mm. Der Buchstabe S muss ohne Leerzeichen vor dem Zahlenwert stehen.
v	+	Inkrementieren – Offset wird zu TEXT addiert
	-	Dekrementieren – Offset wird von TEXT subtrahiert
o	int	Offset, der je nach Vorzeichen zu TEXT addiert oder von TEXT subtrahiert wird

p		Kennbuchstabe zur Bestimmung der Zahlenbasis des Offsets. Wird der Kennbuchstabe weggelassen, wird automatisch die dezimale Zahlenbasis verwendet.
	B	Binär [01]
	O	Oktal [01234567]
	D	Dezimal [0123456789]
	H	Hexadezimal [0123456789ABCDEF]

a	int	Anzahl Etiketten mit gleichbleibender Nummer [1...255].

TEXT		Beliebiger alphanumerischer Text, wobei nur numerische Werte beim Offset berücksichtigt werden. Es werden alle numerischen Werte des Textstrings zum Offset herangezogen (z. B. 10A3B56 = 10356). Maximal zulässige Zeichenzahl: 255. Das Textfeld kann auch ein Variables Datenfeld sein. Voraussetzung: D-Flag gesetzt. Das Textfeld kann auch ein Eingabefeld enthalten.

Beispiele

- *Beispiel 1*: Bitmimage invertieren:

```
#!A1
#ER
#T5#J5#YL0/0/2.5/90#G
#T5#J5#YT109/0E///Beispiel#G
#Q1/
```

Ausdruck: Der Teil der Schrift, der in die Linie hineinragt, wird weiß gedruckt, der Rest schwarz.

- *Beispiel 2*: Unterdrückung führender Nullen:

```
#YT106/0Y/1/1/Test000#G
```

Ausdruck: Test 1, Test 2, …, Test 10, etc.

- *Beispiel 3*: Font 100, ohne Parameter:

```
#YT100/0///DEMO#G
```

- *Beispiel 4*: Font 101, Text kursiv und um 180 Grad gedreht.

```
#YT101/2K///DEMO#G
```

- *Beispiel 5*: Font 100, Text wird alle 5 Etiketten um 1 erhöht (DEMO-10, DEMO-11, DEMO-12 usw.).

```
#YT100/0/1/5/DEMO-10#G
```

- Beispiel 6:

```
#YT109/0///T#G#YT103/0///ext#G
```

Ausdruck: Text

- *Beispiel 7*: Font 104 mit variablem Datenfeld Nummer 00 mit 15 Zeichen.

```
#YT104/0D///$00,15#G
```

- *Beispiel 8*: Zähler mit dezimaler Zahlenbasis. Offset: +1110

Kommando	Ergebnis
`#YT107/0/+1110/1/12-O.17^T#G`	12-O.17^T 23-O.27^T 34-O.37^T ...

- *Beispiel 9*: Zähler mit hexadezimaler Zahlenbasis. Offset: -1H

Kommando	Ergebnis
`#YT107/0/-1H/1/0Kbf0#G`	0Kbf0 FKbfF FKbfE ...

- *Beispiel 10*: Zähler mit binärer Zahlenbasis. Offset: +1B

Kommando	Ergebnis
`#YT107/0/+1B/1/0000000#G`	0000000 0000001 0000010 ...

- *Beispiel 11*: Zähler mit oktaler Zahlenbasis. Offset: +1O

Kommando	Ergebnis
`#YT107/0/+1O/1/0006#G`	0006 0007 0010 ...

Zugehörige Informationen

Eingabefelder auf Seite 10
Variable Datenfelder auf Seite 9
Druckerinterne Zeichensätze und Linienstile auf Seite 157

#YV - Variables Datenfeld

Dieser Befehl dient dazu, ein Feld mit variablen Daten zu füllen. Feldtypen für variable Daten sind Textfelder (#YT) und Barcodefelder (#YB). Das variable Datenfeld muss dem Drucker vor der Benutzung des Kommandos #YV bekannt sein.

Um die Rechenleistung des Druckers optimal zu nutzen, sollten Druckjobs so aufgebaut sein, dass sie die Daten für die #YV-Felder in der Druckreihenfolge schicken. Dadurch können die Felder während des Druckens aktualisiert werden. Ansonsten muss der Drucker mit dem Aktualisieren warten, bis das entsprechende #YV-Feld gedruckt ist, was den Durchsatz verringern kann!

Variable Datenfelder können nicht mit Zählfeldern kombiniert werden!

Syntax

```
#YVzn/TEXT#G
```

Parameter	Wert	Bedeutung
z	int	Datenfeld-Nummer [0...999]
n	leer	Leerzeichen am Ende von TEXT werden gelöscht.
	B	Leerzeichen am Ende von TEXT werden *nicht* gelöscht.

TEXT		Variable Daten (1 bis 255 Zeichen)

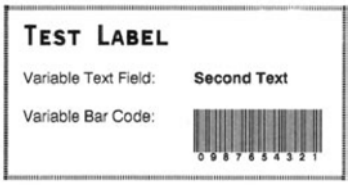
Beispiele

Kommandofolge	Ausdruck

`#!A1`
`#IMS95/50`
`#ERN`
`#T5#J5`
`#YR3//1/90/45`
`#T10#J40`
`#YT109////Test Label#G`
`#T10#J30`
`#YT104////Variable Text Field:#G`
`#T10#J20`
`#YT104////Variable Bar Code:#G`
`#T55#J30`
`#YT107/D0///$00,15`
`#T55#J10`
`#YB6/D0/10/3///$01,10`	

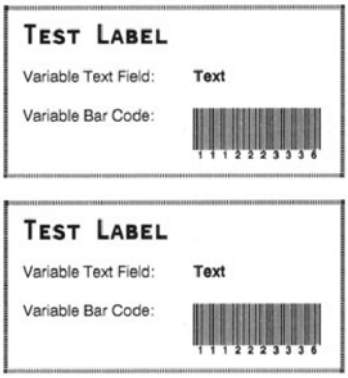
`#YV00/-- Start --#G`
`#YV01/1234567890#G`
`#Q1#G`	

`#YV00/First Text#G`
`#YV01/5555555555#G`
`#Q1#G`	

`#YV00/Second Text#G`
`#YV01/0987654321#G`
`#Q1#G`
	

Kommandofolge	Ausdruck

`#YV00/Further Text#G`
`#YV01/1234598760#G`
`#Q1#G`	

`#YV00/Text#G`
`#YV01/1112223336#G`
`#Q2#G`
	

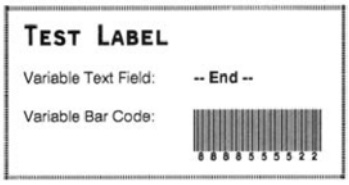
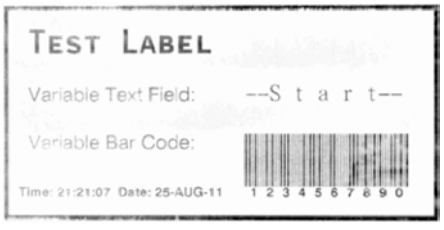
`#YV00/-- End --#G`
`#YV01/8888555522#G`
`#Q1#G`
	

Tabelle 19: Beispiel mit #YT

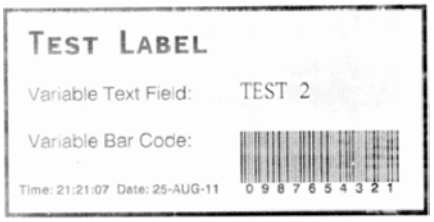
Kommandofolge	Ausdruck

`#!A1`
`#IMN95/50`
`#ER`
`#T5 #J5`
`#YR0//1/90/45`
`#T8 #J10`
`#YC106/0//Time: ^h:^m:^s  Date: ^D-^G-^R#G`
`#T10#J40`
`#YT109////Test Label#G #T10#J30`
`#YT104////Variable Text Field:#G`
`#T10#J20 #YT104////Variable Bar Code:#G`
`#T55#J30 #YN902/0X/60///$00,15`
`#T55#J10 #YB6/D0/10/3///$01,10`	

`#YV00/--S t a r t--#G`
`#YV01/1234567890#G`
`#Q1#G`
	

Kommandofolge	Ausdruck

`#YV00/TEXT 1#G`
`#YV01/5555555555#G`
`#Q1#G`	

`#YV00/TEST 2#G`
`#YV01/0987654321#G`
`#Q1#G`
	

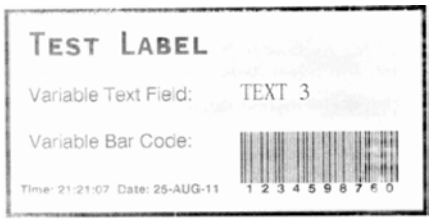
`#YV00/TEXT 3#G`
`#YV01/1234598760#G`
`#Q1#G`
	

Tabelle 20: Beispiel mit #YN

Zugehörige Informationen

Variable Datenfelder auf Seite 9

Z

#Z - Spiegelung

Das Kommando #Z spiegelt Texte, Barcodes, Logos, Linien und Rechtecke.
Gespiegelt werden alle Felder, die vor dem Kommando stehen.

Spiegelachse ist die Mittelachse des Etiketts, bezogen auf die im Kommando #IM angegebene Etikettenbreite.

Die im #IM Kommando angegeben Etikettenbreite wird auf ein Vielfaches von 2,66 mm (32/12 mm) aufgerundet. Die Spiegelachse ist die Mittelachse des Etiketts bezogen auf die aufgerundete Etikettenbreite.

Syntax

```
#Z
```

Das Kommando muß *zwischen* #ER und #Q stehen!

Bei Updatefeldern wie Nummernfeld, Clock, Linerepeat und Easyline kommt es zu Fehldrucken!

Wird das Beispiel im Standalone-Modus mehrmals gedruckt, erfolgt die Spiegelung nur beim ersten Druckjob. Die darauffolgenden Druckjobs bleiben ungespiegelt.

Beispiel

```
#!A1
#IMN64/100
#ERN
#G ---------------------------------------
#G Gespiegelter Text
#G ---------------------------------------
#T5#J5#YT106////Spiegel
#Z
#G ---------------------------------------
#G Normaler Text
#G ---------------------------------------
#T20#J50#YT107////Normaler Druck
#Q10/
```

Anhang

ÜBERSICHT KOMMANDOGRUPPEN

Allgemeine Kommandos

Kommando	Bezeichnung
#BOF	Spooler-Buffer Aus
#BON	Spooler-Buffer Ein
#CIM	Schnitt
#EMU	300 dpi-Emulation für 600 dpi-Druckkopf
#FC	Materialvorschub mit Schnitt
#FF	Materialvorschub
#G	Kommandoabschluss
#HP	Druckkopf-Andruck
#HV	Druckkopf-Temperatur einstellen
#ME	Etikettenmaterial auswerfen
#PC	Parameter einstellen
#PR	Druckgeschwindigkeit
#RTC	Echtzeituhr einstellen
#RX	Auswahl Stanzenerkennung

Download-Kommandos

Kommando	Bezeichnung
#DC	Löschen aller Download-Logos
#DF	Download einer Datei
#DK	Download eines Logos
#DM	Download von Monatsnamen
#DO	Löschen eines Download-Logos

Logo-Kommandos

Kommando	Bezeichnung
#DC	Löschen aller Download-Logos
#DK	Download eines Logos
#DO	Löschen eines Download-Logos
#YI	Logo im EPT-Format in den Bildpuffer schreiben

Kommando	Bezeichnung
#YIB	Logo mit Binärdaten direkt in Bildpuffer schreiben
#YIR	Logo im RLE-Format direkt in Bildpuffer schreiben
#YK	Logo definieren

Druckjob-Kommandos

Kommando	Bezeichnung
#BR	Druckjob unterbrechen
#ER	Start Etikettenformat
#FO	Easy Plug Datei einlesen
#IM	Materialinformation
#N	Nationalität Zeichensatz
#PA	Druckbeginn verschieben
#PO	Stanzen-Offset
#Q	Druckmenge
#S	Spende-Modus/Spendeposition

Format-Kommandos

Kommando	Bezeichnung
#CBF	Barcode Codablock F
#CG	Zeichenabstand verändern
#CFN	Code 49
#IDM	Data Matrix Code
#J	Vertikale Druckposition
#M	Multiplikationsfaktor
#MXC	Barcode MaxiCode
#PDF	Barcode PDF 417
#R	X/Y-Offset Datenblöcke
#RSS	GS1 DataBar & CC
#SQR	QR Matrix Code
#T	Horizontale Druckposition
#YB	Barcode definieren
#YC	Echtzeit als Text
#YG	Grafiken ausdrucken
#YI	Logo im EPT-Format direkt in Bildpuffer schreiben
#YIB	Logo mit Binärdaten direkt in Bildpuffer schreiben

Kommando	Bezeichnung
#YIR	Logo im RLE-Format dirket in Bildpuffer schreiben
#YK	Logo definieren
#YL	Linie definieren
#YN	Textfeld (Option Fontscale)
#YR	Rechteck definieren
#YS	Echtzeit als Barcode
#YT	Textfeld definieren
#YV	Variables Datenfeld
#Z	Spiegelung

Format-Kommandos definieren Fonts, Barcodes, Logos, Linien und Rechtecke in Art, Größe, Drehrichtung und ihre Position im Etikett anhand von Feldern. Zusätzlich besteht die Möglichkeit, alle Felder gesammelt zu versetzen.

Format-Kommandos werden im Speicher verwaltet und erst nach gültigem Formatabschluss (#Qn/) ausgeführt. Easy Plug kann nur jeweils ein Format zum Drucken aufbereiten, jedoch nachfolgende Formate in der Warteschlange zwischenspeichern.

Wurde die eingegebene Menge von Etiketten abgearbeitet, so wird dieses Format gelöscht (auch Change- und Messerinformation) und mit dem Bearbeiten des nächsten Formats begonnen.

Ein einmal gestartetes Format bleibt aktiv. Deselektieren des Gerätes unterbricht lediglich das Abarbeiten der begonnenen Etikettenserie.

Etiketten-Nullpunkt Die linke untere Ecke des Etiketts hat die horizontale (X) Position 0 (Null) und die vertikale (Y) Position 0 (Null).

Feldposition Die Kommandos #Tx und #Jx definieren die Feldposition. Diese ist die linke, untere Ecke eines Feldes bezogen auf die linke (X), untere (Y) Ecke des Etiketts (Stanzenmitte bzw. Längenposition bei Endlosmaterial).

Bei Drehung von Feldern bleibt die Feldposition konstant, die Druckposition rotiert um die Feldposition.

Variablen-Kommandos

Kommando	Bezeichnung
#SB	Definition Barcode
#SCF	Definition Codablock F
#SDM	Definition Data Matrix
#SF	Definition Fixfont
#SFN	Definition Code49
#SG	Definition Grafik
#SMX	Definition Maxicode
#SPF	Definition PDF417
#SRS	Definition GS1 DataBar

Kommando	Bezeichnung
#SS	Definition Speedo Font
#SV	Wert einer Textvariablen ändern
#VDT	Definition Textvariable
#VDD	Definition Datums-/Zeitvariable
#VDE	Definition Ausdrucks-Variable
#VDO	Definition Variable für OLV-Datenzugriff
#VDP	Definition Variable für Druckauftrags-Datenzugriff
#VDS	Definition Systemvariable
#VTS	Definition Standalone-Variable
#VR	Daten von (RFID-)Etikett einlesen
#VW	Datenausgabe auf Etikett

Um einen Text, Barcode oder eine Grafik zu drucken, muss auf die oben aufgeführten Definitions-Kommandos immer ein #VW-Kommando folgen, das die Druckausgabe veranlasst.

RFID-Kommandos

Kommandos für Geräte, die mit der RFID-Option ausgestattet sind.

Kommando	Bezeichnung
#RT	RFID-Daten lesen und drucken
#RFC	Spezielle RFID-Befehle
#RFH	Daten anfordern und an Host senden
#RFK	Transponder zerstören
#RFL	Speicherbereich sperren/entsperren
#RFPL	Speicherbereich permanent sperren/entsperren
#RFR	Daten lesen
#RFW	Daten schreiben
#SI	Daten an Schnittstelle ausgeben
#SRF	Definition Lese-/Schreibziel
#VC	RFID-Daten vergleichen
#VR	Daten von RFID-Etikett lesen
#VW	Datenausgabe auf Ausgabeziel

Sofortkommandos

Kommando	Bezeichnung
#!A1	Interface aktivieren
#!CA	Easy Plug alles löschen
#!CF	Format löschen
#!D	Single-Start auslösen
#!H	Kopftemperatur einstellen
#!P1	Interface deaktivieren
#!PC6004	Start Offset einstellen
#!PG	Parameter-Einstellung auslesen
#!SP	Drucker stoppen
#!SR	Drucker starten
#!XB	Statusrückmeldung mit Barcode
#!XC	Statusrückm. Pharmacy Code
#!Xn	Statusrückmeldung
#!Xn	Rückmeldung Barcode
#!XMn	Diagnose Dump / Maschinenstatus auslesen
#!XS	Statusrückmeldung Drucker

Kennzeichen eines Sofortkommandos ist das Ausrufezeichen „!“.

Sofortkommandos werden nicht in der Warteschlange zwischengespeichert, sondern sofort nach ihrem Eintreffen ausgeführt.

Alle anderen Kommandos werden bis nach Ausführung der Sofortkommandos zurückgestellt und dann nach der Reihenfolge ihres früheren Eintreffens wieder bearbeitet.

Sofortkommandos können nicht im Standalone-Betrieb verwendet werden.

Sofortkommandos können beliebig eingesetzt werden – auch innerhalb des Formats.

Es ist zum Beispiel folgender Ablauf möglich:

1. Druckauftrag A senden.
2. Während des Ausdrucks des ersten Druckauftrags einen zweiten Druckauftrag B zu senden, welcher in der Warteschlange gespeichert wird (#BON vorausgesetzt).
3. Mit dem Sofort-Kommando #!CF den ersten Druckauftrag A abbrechen.

> Da mit #!CF der Druckvorgang angehalten und die aktuelle Druckserie gelöscht wurde (Situation wie Serie fertig gedruckt), wird mit der Bearbeitung der in der Warteschlange angesammelten Daten fortgefahren und somit das zweite Format bearbeitet

Zugehörige Informationen

Aufbau eines Druckauftrags auf Seite 7

DATEIOPERATIONEN

Beschreibung der korrekten Pfadangabe für Dateioperationen.

Laufwerksbezeichnungen

Folgende Laufwerksbezeichnungen sind in Pfadnamen zulässig:

LW	Dateizugriff auf
A	Interne Drucker-RAM. Durch Ausschalten des Druckers gehen die im Drucker-RAM gespeicherten Daten verloren!
C	CompactFlash-Karte (Standardschacht)
D	CompactFlash-Karte (Optionaler Zusatzschacht)
E	SD-Karte
F	USB-Stick Die Laufwerksbezeichnung wird der ersten erkannten USB-Host-Schnittstelle zugewiesen.

Standalone-Betrieb ist nur mit Laufwerk C möglich.

Zuweisen von Laufwerksbezeichnungen: SCHNITTST. PARA > >LW-ZUORDNUNG > LAUFWERK x (x steht für den Laufwerksbuchstaben)

Dateinamen

Dateien, die innerhalb eines Easy-Plug-Kommandos aufgerufen werden, müssen nach den nachfolgend aufgeführten Regeln benannt sein. Dies sind dieselben Benennungsregeln, wie sie unter dem Betriebssystem MS-DOS üblich waren:

- Maximale Länge des Dateinamens: 8 Zeichen
- Zulässig sind alphanumerische Zeichen (20_{hex} bis $7E_{hex}$ der ASCII-Tabelle); dies sind:
  - Buchstaben A-Z ohne länderspezifische Sonderzeichen
  - Ziffern 0-9
- Nicht zulässig sind folgende Zeichen: . / : < > + = ; , ? [\ ] " * | sowie das Leerzeichen
- Der Dateiname kann um maximal 3 Zeichen ergänzt werden. Diese Namenserweiterung wird vom Dateinamen durch einen Dezimalpunkt getrennt

Zugehörige Informationen

#DF - Download einer Datei auf Seite 31
#CF - Datei löschen auf Seite 28
#YG - Ausdruck von Grafiken auf Seite 121

LOGOS

Ein Logo ist ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist.

Was ist ein Logo?

Im Zusammenhang mit Easy-Plug ist ein Logo ein Bild aus schwarzen und weißen Bildpunkten, das Bestandteil eines Etikettenlayouts ist . Logos werden binär kodiert im Speicher des Druckers abgelegt.

Die Verwendung von Logos ist historisch bedingt und wurde weitgehend durch die Verwendung von Grafikdateien abgelöst. Logos haben weiterhin ihre Berechtigung für ältere Druckertypen, die keine Grafikdateien verarbeiten können.

Aufbau eines Logos

Easy Plug verwaltet einen Speicherbereich, in dem vom Anwender definierte Logos unter einer Nummer (0 bis 255) gespeichert werden können. Ein Logo (Klischee, Bild) setzt sich aus in Zeilen und Spalten angeordneten Punkten (Dots) zusammen. Jedes Dot ist ein Bit. Nur auf 1 gesetzte Bits werden gedruckt, die anderen nicht.

Diese Punkte-Matrix wird in Easy Plug hexadezimal verschlüsselt wie folgt dargestellt:

- Die Verschlüsselung einer Zeile der Logo-Matrix erfolgt hexadezimal zu je 4 Dots von links nach rechts.
- Nicht gesetzte Dots am Zeilenende können entfallen.
- Die Verschlüsselung aller Zeilen der Logo-Matrix durch einen Parameter (s) je Zeile, beginnend von unten (Zeile 1) nach oben.
- Leere Zeilen werden durch 2 Slashs (//) definiert.
- Die maximale Größe eines Logos wird nur durch die Größe des Etikettenlayouts und den verfügbaren Speicherplatz begrenzt.

Beispiel

Wertigkeit	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Zeile 8: /C183	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1
Zeile 7: /C3C3	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
Zeile 6: /C7E3	1	1	0	0	0	1	1	1	1	1	1	0	0	0	1	1
Zeile 5: /CFF3	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1
Zeile 4: /DFFB	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1
Zeile 3: /FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Zeile 2: //	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zeile 1: /FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bild 24: Beispiel einer Logo-Matrix

Kurzform derselben Logo-Matrix:

```
Zeile 8: 1100 0001 1000 0011 (/C183)
Zeile 7: 1100 0011 1100 0011 (/C3C3)
Zeile 6: 1100 0111 1110 0011 (/C7E3)
Zeile 5: 1100 1111 1111 0011 (/CFF3)
Zeile 4: 1101 1111 1111 1011 (/DFFB)
Zeile 3: 1111 1111 1111 1111 (/FFFF)
Zeile 2: 0000 0000 0000 0000 (//)
Zeile 1: 1111 1111 1111 1111 (/FFFF)
```

Senden des Logos an den Drucker:

```
#G------------------------------------------------------------
```

```
Das Logo wird zum Drucker gesendet und mit der Nummer 2 im
Download-Buffer gespeichert.
-----------------------------------------------------------------
#DK2/FFFF//FFFF/DFFB/CFF3/C7E3/C3C3/C183#G
```

Zugehörige Informationen

#DK - Download eines Logos auf Seite 32
Das Kommando #DK dient zum Download eines Logos (d. h. dem Senden eines Logos vom PC zum Drucker) unter einer anzugebenden Bezugsnummer (0 bis max. 255).

#YI - Logo im EPT-Format in den Bildpuffer schreiben auf Seite 122
Das Kommando #YI schreibt die dem Kommando folgenden Daten in den Bildpuffer-Speicher.

#YIB - Logo mit Binärdaten direkt in Bildpuffer schreiben auf Seite 122

#YIR - Logo im RLE-Format direkt in Bildpuffer schreiben auf Seite 124
Das Kommando #YIR schreibt die auf das Kommando folgenden Daten direkt in den Bildpuffer. Die Daten sind im RLE Format (Run Length Format, Lauflängenformat) codiert.

#YK - Logo definieren auf Seite 126

HILFSPROGRAMM „MAKE_DF.EXE“

Das Hilfsprogramm „Make_DF.exe“ generiert ein komplettes #DF-Kommando einschließlich der zu übertragenden binären Daten und schreibt es in eine Textdatei.

Syntax

Das Easy-Plug-Kommando #DF lädt Daten vom Host-Rechner in den Drucker. Dazu müssen die Daten binär vorliegen

Rufen Sie Make_DF ohne Parameter auf, um eine Liste der erforderlichen Kommandozeilen-Parameter zu sehen.

```
MAKE_DF <SourceFilename> <DestinationFilename> <PrinterFilename> <Over▶
writeOption[N,O]>
```

Parameter	Beschreibung	Beispiel
<SourceFilename>	Laufwerks-, Pfad- und Dateiname der zu übertragenden, auf dem PC befindlichen Datei.	C:_temp\testfont.spd
<DestinationFilename>	Laufwerks-, Pfad- und Dateiname der von make_df erzeugten Textdatei. Diese Datei enthält nach erfolgreichem Programmablauf das #DF-Kommando einschließlich aller Parameter und Daten.	C:_temp\df333.txt
<PrinterFilename>	Laufwerks-, Pfad- und Dateiname der übertragenen Datei auf dem Drucker nach erfolgtem Programmablauf (siehe Parameter f des #DF-Kommandos).	C:\fonts\font333.spd
<OverwriteOption[N,O]>	Option vorhandene Datei überschreiben ja (O) / nein (N) (siehe Parameter s des #DF-Kommandos).	N

Tabelle 21: Kommandozeilen-Parameter für MAKE_DF.

Beispiel

Mit den Beispielangaben würde der Programmaufruf folgendermaßen aussehen:

```
make_df c:\_temp\testfont.spd c:\_temp\df333.txt c:\fonts\font333.spd N
```

Anzeige nach dem Programmaufruf:

```
File c:\_temp\DF333.txt created:
Printerfilename: C:\fonts\font333.spd
Filesize:   71680 Bytes
```

In der Datei df333.txt steht jetzt das gesamte #DF-Kommando einschließlich aller erforderlichen Parameter und der binären Daten der zu übertragenden Datei. #DF vorangestellt ist das Sofortkommando #!A1, das den Drucker auf das Empfangen von Easy Plug Kommandos vorbereitet.

Ausschnitt der generierten Datei:

```
#!A1#DF/N/C:\fonts\Font333.spd/78193/D1.0   _1q  _n_z_¤ + _AmericanGara▶
mond                                                  27 Oct 92 BICS
 for Speedo  23 May 88                                      BX00Copy▶
right 1989-1992 by Bitstream Inc. All rights reserved.
   _4_5    __®___hAmericanGaramondBT-Roman        AmericanGaramondRoman
   _è ø_è_ô ø_ïÿYÿ _h_…ÿ• +  ì_ö_#  ___  é˜ š_T_ _3ÿQ˜ š_]_®_f  _®_f_ __ö
  __ö Ì_\f  _\f_c __°ß ÍÖ .Ñ 6Ñ (Ç _ï_ _Ç _Ç Ã }Ä øÅ kÃ /Á gþ Çÿ gÿ ìü
 Xý Mý óý †ú _ú Úû xû õû éø Õù Èö Ž÷ «ô ×õ Jò _ó Dó íð  ð þð ½ñ {ñ ‹ï í
 qê ,ë nè \æ cç ½ä œâ 'ã à Âž êŸ Æ˜ °› ÿ˜ R™ _– _“ p• t’ _„ ã˜ òŽ _Œ a˜
 Š Æ‹ Ü^ }‰ q‰_N‰ _‰ Ý† ˜„ x„ Ñ… ¬, ×ƒ ^˜ á¿ F¼ A½ ~° ’» É¹ '· _· 'µ O²
  ³  °  _± _® -¯ ¿- lª ä« ß¨ Õ© /© ù¦ 3¦ «¤ Ì¢ É  ê¡ ‰^ ¨\ Š] _] _ÔZ zZ
 Ì[ ð[ 'X ÑY ÆW _T ^U nU 8S MQ ¿O _L ^J ÓH 6I ìF ¢D ;E |E ûB wB …C „@ »A
 (A îA ÄA ˜~ c~ =~ _~ ¹~ -˜ _˜ _˜ Ã˜ ¸| _| ÿ| Ó| ˜} `} ò} è} Þ} ´} šz `z
 Dz *z _z Òz z œ{ `{ ]{ 4{ /{ _{ á{ Ø{ ³{ ª{ …x |y _v °t _r Øp ^q 9n ×o
 #l †j Th [i ¢f ef Vf ýf Óf Œg Vg _g ³d hd úe ve \e 3e (e _e ôe ëe Àe ·e
 ¬e ƒb xb ob Db ;b _b _b üb Ób Èb ¿b “b ‹c `c Wc Lc #c _c _c äc Úc ³c ©c
 ž` c` D` &` _` ø` Þ` ‡a |a Ua Ka  a _a _a åa
```

Die generierte Datei kopieren Sie entweder in einen Easy Plug Druckauftrag oder Sie senden sie unverändert an den Drucker, um das darin enthaltene #DF-Kommando sofort auszuführen. Kopieren Sie dazu die Datei an die gewählte Drucker-Schnittstelle:

```
copy df333.txt lpt1: /b
```

Parameter /b für binäre Dateien setzen!

Jetzt ist die Datei font333.spd im Verzeichnis `\fonts` auf der CompactFlash-Karte des Druckers zu finden. Zur Kontrolle drucken Sie den Info-Ausdruck „Speicher Status“ aus – darauf finden Sie eine Liste der auf RamDisk und CompactFlash-Karte enthaltenen Font- und Logo-Dateien:

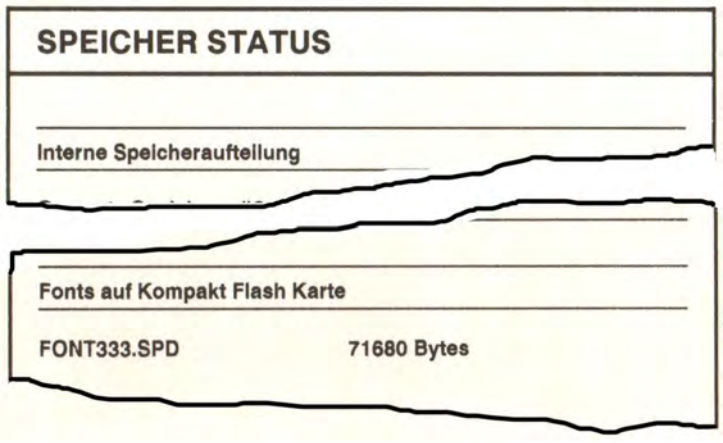


Bild 25: Info-Ausdruck „Speicher Status“ mit über- und eingetragener Font-Datei font333.spd

Zugehörige Informationen

#DF - Download einer Datei auf Seite 31

DRUCKERINTERNE BARCODES

Eindimensionale Barcodes

Eindimensionale Barcodes werden mit dem Easy Plug-Kommando #YB gedruckt.

Nr.	Barcode	Nr.	Barcode
0	EAN 8	14	MSI
1	EAN 13	15	EAN 128
2	UPCA	16	CODE 39 (3:1)
3	CODE 93	17	POSTCODE (Leit- und Identcode)
4	CODE 2/5 Interleaved	18	CODE 128 (UPS)
5	CODE 2/5 Matrix	19	CODE 39 (2,5:1)
6	CODE 2/5 5-Strich	20	CODE 2/5 Interleaved Ratio 1:3
7	CODE 39	21	CODE 2/5 Matrix Ratio 1:2,5
8	CODABAR	22	CODE 2/5 Matrix Ratio 1:3
9	UPCE	23	CODE 39 Extended
10	ADD ON 2	24	CODE 128 A
11	ADD ON 5	25	CODE 128 B
12	ITF	26	CODE 128 C
13	CODE 128	27	CODE 128 Pharmacy

Tabelle 22: Übersicht eindimensionale Barcodes.

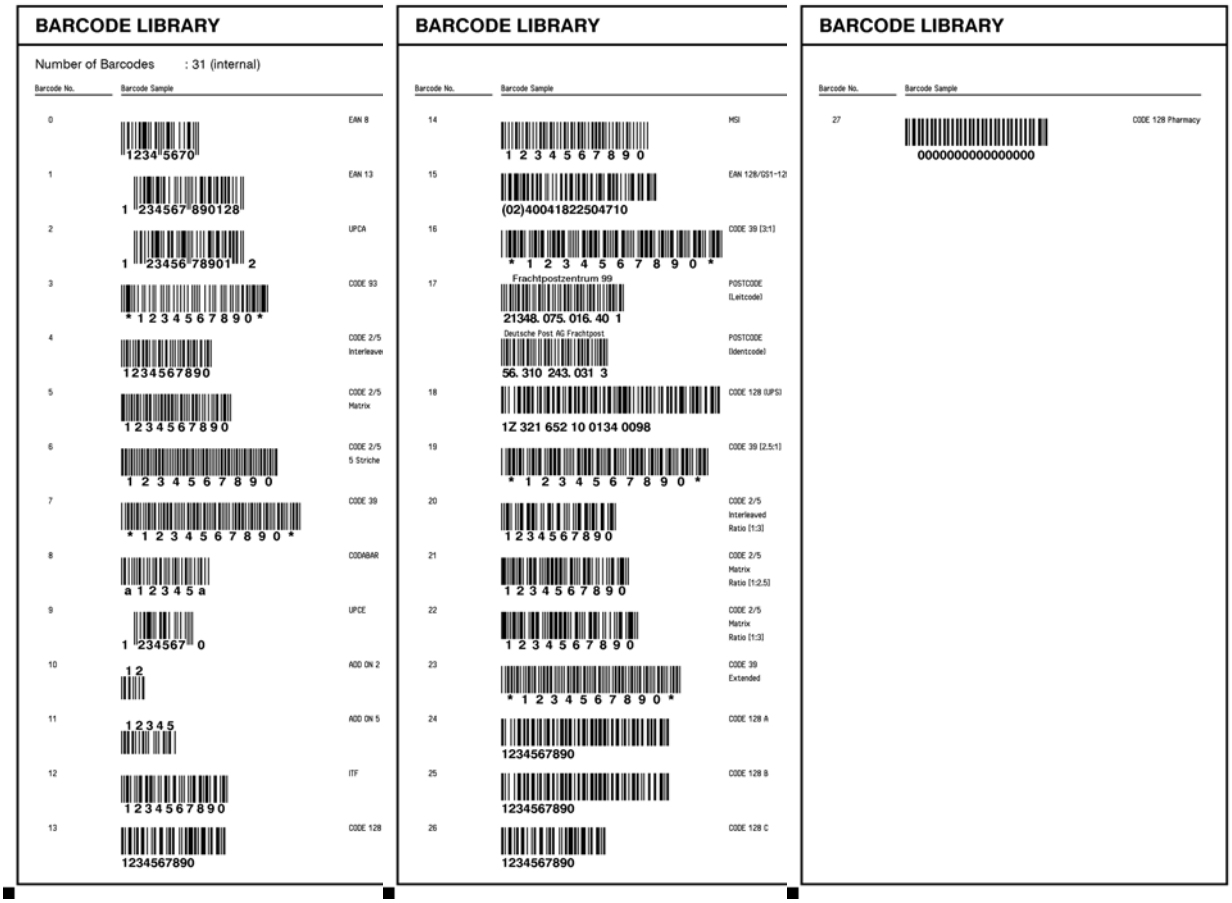


Bild 26: Druckerintern zur Verfügung stehende, eindimensionale Barcodes (Ausdruck INFO AUSDRUCKEN > Font Status bzw. Info > Status Ausdrucke > Font Status, Abschnitt „Barcode Library“).

Zweidimensionale Barcodes

Zweidimensionale Barcodes werden mit speziellen Easy Plug-Kommandos gedruckt (siehe Tabelle).

Easy Plug Kommando	Barcode
#IDM	Data Matrix Code
#MXC	Maxi Code
#PDF	PDF 417
#CBF	Codabar F
#CFN	Code 49
#SQR	QR Matrix Code

Tabelle 23: Easy Plug Kommandos für druckerintern verfügbare 2dim. Barcodes.

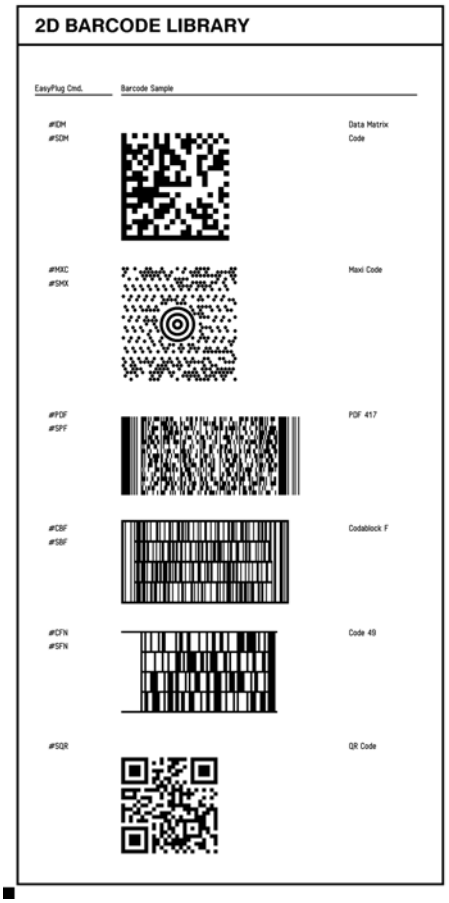


Bild 27: Druckerintern zur Verfügung stehende 2dim. Barcodes (Ausdruck INFO AUSDRUCKEN > Font Status bzw. Info > Status Ausdrucke > Font Status, Abschnitt „Barcode Library“).

GS1 Barcodes

GS1 DataBar (früher RSS) und *Composite Component* (CC) Barcodes werden mit dem Easy Plug Kommando #RSS gedruckt. Die Auswahl des Barcodes erfolgt über die Nummer in der ersten Spalte der Tabelle, die dem Befehl als Parameter hinzugefügt wird.

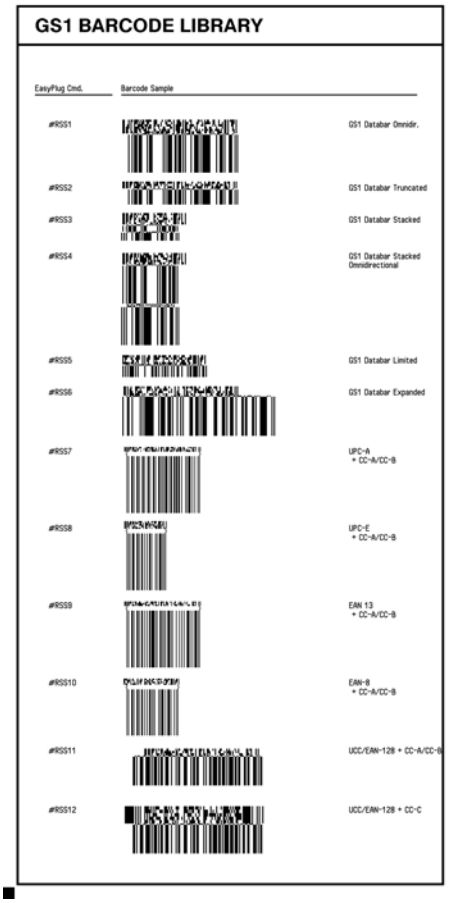


Bild 28: Druckerintern zur Verfügung stehende GS1-Codes (Ausdruck INFO AUSDRUCKEN > Font Status bzw. Info > Status Ausdrucke > Font Status, Abschnitt „Barcode Library“).

Zugehörige Informationen

#CBF - Barcode Codablock F auf Seite 25
Das Kommando #CBF druckt Barcodes vom Typ *Codablock F*.

#CFN - Code 49 auf Seite 26
Das Kommando #CFN definiert Barcode vom Typ „Code 49“ (ANSI/AIM-BC6-2000 „Uniform Symbology Specification Code 49“)

#IDM - Data Matrix Code auf Seite 44
Das Kommando #IDM druckt einen Barcode vom Typ Data Matrix Code.

#MXC - Maxicode auf Seite 54
Das Kommando #MXC druckt einen Barcode vom Typ Maxicode.

#PDF - Barcode PDF 417 auf Seite 62
Das Kommando #PDF druckt einen Barcode vom Typ PDF 417.

#RSS - GS1 DataBar & CC auf Seite 73
Das Kommando #RSS druckt ein Barcode-Feld mit einem GS1 DataBar (vormals RSS) oder einem Composite Symbology Barcode.

#YS - Echtzeit als Barcode auf Seite 134
Das Kommando #YS definiert eine Zeitangabe in Form eines Barcodes.

#SB - Barcode Definition auf Seite 80

Das Kommando #SB definiert einen Barcode. Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

#YB - Barcode definieren und drucken auf Seite 114
Das Kommando #YB druckt ein Barcode-Feld.

DRUCKERINTERNE ZEICHENSÄTZE UND LINIENSTILE

Druckerinterne Zeichensätze

Die Zeichensätze werden im entsprechenden Easy Plug Kommando über die Nummer in der Spalte „Font No.“ aufgerufen.

Font No.	Bemerkung
98-99	Zeichensätze für die Verwendung mit 600 dpi-Druckköpfen (XLP 504 600 dpi)
100-116	Zeichensätze mit fester Größe (Fixfont); Ausdruck mit Kommando #YT
100-102 (am Spaltenende)	Skalierbare (Speedo Font) Zeichensätze; Ausdruck mit Kommando #YN

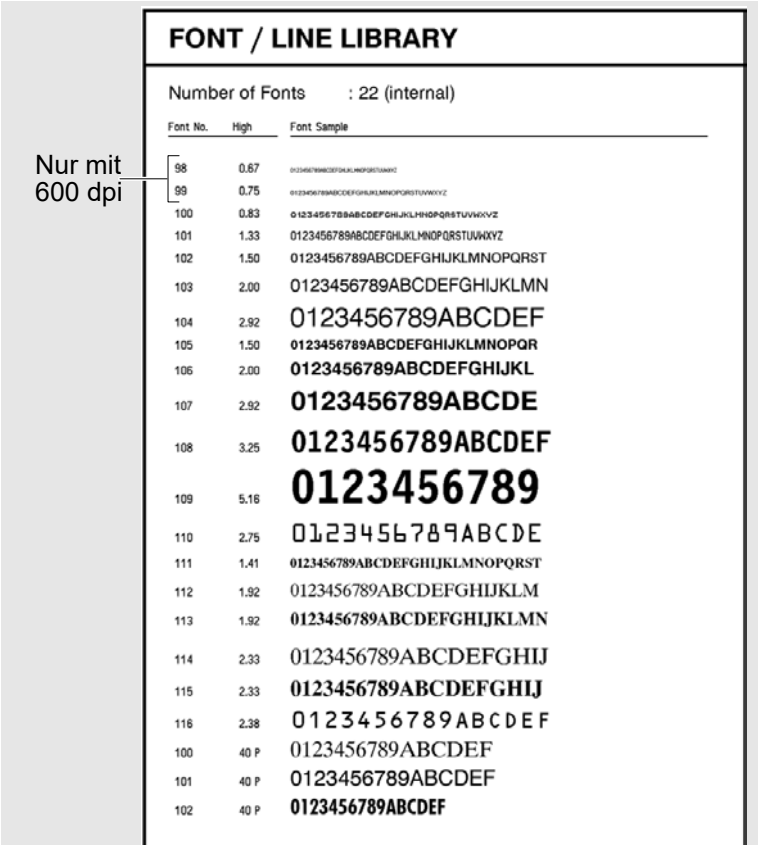


Bild 29: Druckerintern zur Verfügung stehende Schriften (Ausdruck INFO AUSDRUCKEN > Font Status bzw. Info > Status Ausdrucke > Font Status, Abschnitt „Font /Line Library“).

Druckerinterne Linienstile

Die Linienstile werden im entsprechenden Easy Plug Kommando über die Nr. in der Spalte „Line Style“ aufgerufen.

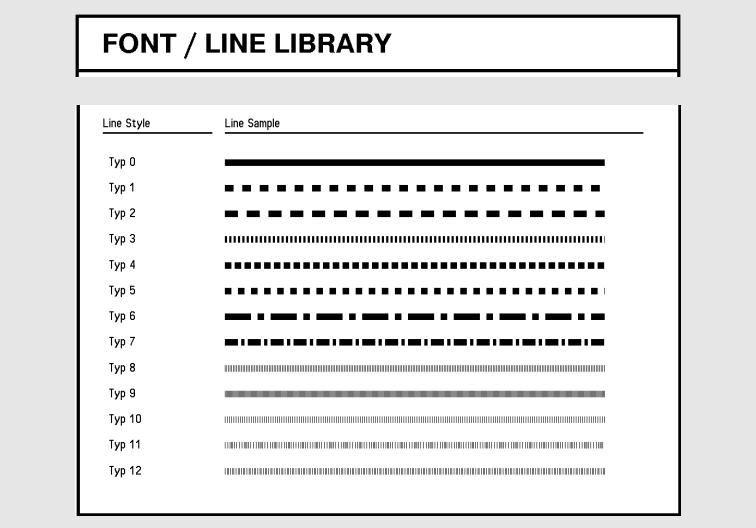


Bild 30: Druckerintern zur Verfügung stehende Linienstile (Ausdruck INFO AUSDRUCKEN > Font Status bzw. Info > Status Ausdrucke > Font Status, Abschnitt „Font /Line Library“).

Zusätzlich zu den ausgedruckten Linientypen gibt es noch die folgenden Typen:

- 13: Karomuster mit 3 Dot Kantenlänge
- 14: Karomuster mit 1 mm Kantenlänge
- 15: Karomuster mit 5 mm Kantenlänge

Die Linienbreite dieser Linientypen muss ein vielfaches der Kantenlänge der Karos betragen!

Zugehörige Informationen

#YC - Echtzeit als Text auf Seite 117
#YL - Linie definieren auf Seite 127
#YN - Textfeld auf Seite 128
#YR - Rechteck definieren auf Seite 133
#YT - Textfeld definieren auf Seite 137
Das Kommando #YT druckt Text mit Zeichensätzen in fester Größe (nicht skalierbar).

#RT - RFID-Daten lesen und drucken auf Seite 76
Das Kommando #RT liest Daten von einem RFID-Transponder und druckt sie auf das Etikett.

#SF - Fixfont Definition auf Seite 84
#SS - Speedo-Font Definition auf Seite 98

AUSDRÜCKE

Eine Kombination aus Konstanten, Werten, Variablen, Operatoren und Funktionen heißt Ausdruck.

Ein Ausdruck ist eine Kombination aus folgenden Elementen:

- Konstanten
- Variablen
- Operator „+“
- Klammern
- Funktionen

Beispiel:

Kommandofolge	Beschreibung
`#VDT/Datum////02/02/2005#G`	Textvariable „Datum“ definieren
`#VDT/Nummer////2223#G`	Textvariable „Nummer“ definieren
`Datum+":"DayOfYear(SubStr(Datum,0,1)+"2", SubStr(Datum,3,2),Substr(Datum,6,4))+"Ende"`	Ausdruck 1
`Datum+Nummer`	Ausdruck 2

Zugehörige Informationen

Variablen auf Seite 159
Funktionen auf Seite 161
Arithmetische Funktionen auf Seite 168

VARIABLEN

Easy Plug bietet die Möglichkeit, variable Daten zu berechnen und zu verwenden. Diese Daten werden in *Variablen* gespeichert.
Der Dateninhalt von Variablen heißt *Wert*. Variablenwerte können mit Hilfe von *Funktionen* und *Operatoren* verknüpft und bearbeitet werden.

Variablentypen

Variablentyp	Inhalt	Kommando
Textvariable	Alphanumerische Zeichen, wahlweise mit Zählfunktion	#VDT
Datums-/Zeitvariable	Alphanumerische Zeichen, Datum und Zeit	#VDD
Ausdrucksvariable	Ausdruck	#VDE
Systemvariable	Parametereinstellung aus dem Druckermenü	#VDS
Standalone-Variable	Alphanumerische Zeichen, Werte werden über Bedienfeld abgefragt	#VTS
Druckjobdaten-Variable	Nr. des gedruckten Etiketts, Anzahl Etiketten im Druckjob	#VDP

Tabelle 24: Übersicht über die verschiedenen Variablentypen.

Beispiel

Druckauftrag mit verschiedenen Variablentypen:

```
#!A1
#IMS105/101
#ER
#G ----------------------------------------------------------------
#G Variablen-Definitionen
#G ----------------------------------------------------------------
#VDT/Name////Hans Mueller#G
#VDT/NVE//+1/1/12345678901234567#G
#VDD/Verfall/U/M24/^D^M^Y:^s#G
#VDT/GRAFIKIDX//+1/1/00#G
#G ----------------------------------------
#G Standalone-Eingabefeld
#G ----------------------------------------
#VTS/NVEI/WZN/4/+5/1/0000#G
#G ----------------------------------------
#G Systemvariablen
#G ----------------------------------------
#VDS/PrintSpeed/%S/1003#G
#VDS/PrintSpeedName/%N/1003#G
#G ----------------------------------------
#G Ausdrucksvariablen
#G ----------------------------------------
#VDE/BarcodeData//"(00)" + NVE + Mod10(NVE) + "(12)" +SubStr(Verfal►
l,0,6)#G
#G ----------------------------------------------------------------
#G Elemente drucken
#G Drehung 0°, Ausrichtung linksbündig
#G ----------------------------------------
#FD/0/L#G
#G ----------------------------------------
#G Fixfont 106 normaler Zeichenabstand
#G ----------------------------------------
#SF106//0#G
#T5#J90 #VW/L/PrintSpeedName + " : " + PrintSpeed#G
#T5#J80 #VW/L/Verfall#G
#G ----------------------------------------
#G Speedo Font 100 (Fettdruck, 44Dot in X- und Y-Richtung)
#G ----------------------------------------
#SS100/B/44/#G
#T5#J70 #VW/L/"Name : " + Name + " NVEI:" + NVEI#G
#T5#J65 #VW/L/"Erstes Zeichen = " + SubStr(Name,0,1)#G
#T5#J60 #VW/L/"Zweites Zeichen = " + SubStr(Name,1,1)#G
#T5#J55 #VW/L/"Textlänge = " + Length(Name)#G
#G ----------------------------------------
#G EAN128 Code / Breite 3 / Höhe 20 / Klarschriftzeile zentriert
#G ----------------------------------------
#SB15/MK/20/3#G
#T5#J30 #VW/L/BarcodeData#G
#G ----------------------------------------
#G Fixfont 107 mit 16 Dot Mikroabstand
#G ----------------------------------------
#SF107//16#G #T5#J9 #VW/L/"Name : " + Name + "(*)"#G
#G ----------------------------------------
#G Grafik
#G ----------------------------------------
#SG#G
#T82#J75#VW/L/"Graph" + GRAFIKIDX + ".bmp" #G
#Q3/
```

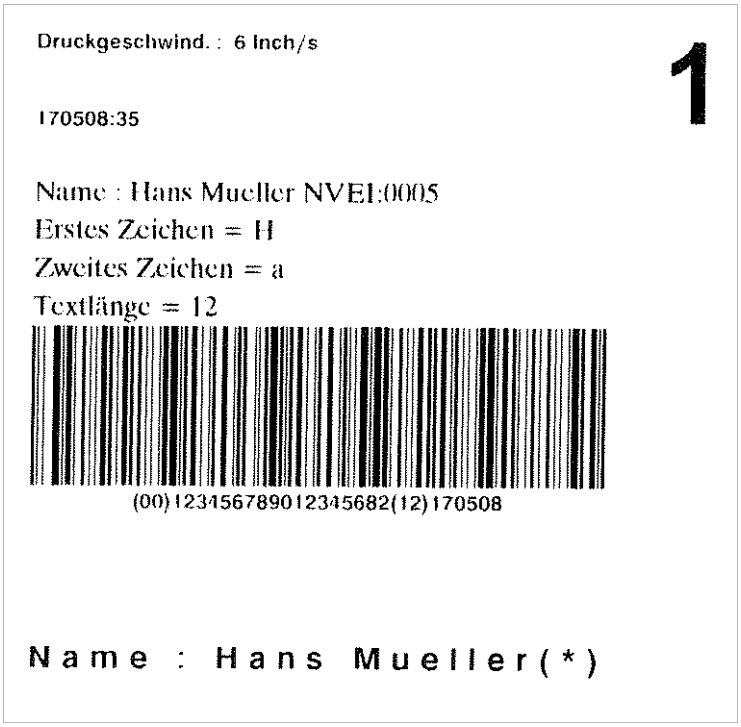


Bild 31: Ausdruck des Beispiels.

Zugehörige Informationen

Funktionen auf Seite 161
Arithmetische Funktionen auf Seite 168
Ausdrücke auf Seite 159

FUNKTIONEN

Übersicht

Funktion	Operation	Beschreibung
Mod10	Berechnet die Prüfsumme für den Wert des Ausdrucks nach Modulo 10	siehe Mod10 (Ausdruck) auf Seite 163
SubStr	Schneidet einen Teil aus dem Wert eines Ausdrucks aus	siehe SubStr (Ausdruck, a, b) auf Seite 163
Length	Ermittelt die Länge (= Anzahl der Zeichen) des Ausdrucks	siehe Length (Ausdruck) auf Seite 164
MergeRight	Zeichenketten rechtsbündig verschmelzen	siehe MergeRight (Ausdruck1, Ausdruck2) auf Seite 164
MergeLeft	Zeichenketten linksbündig verschmelzen	siehe MergeLeft (Ausdruck1, Ausdruck2) auf Seite 164
DayOfYear	Berechnet den Tag des Jahres	siehe DayOfYear (Tag,Monat,Jahr) auf Seite 164

Funktion	Operation	Beschreibung
Chr	Wandelt eine ASCII-Code Dezimalzahl in das entsprechende alphanumerische ASCII-Zeichen um	siehe Chr(a) auf Seite 165
DecToBin	Wandelt den dezimalen Teil eines Ausdrucks in eine binäre Bytekette um	siehe DecToBin (Ausdruck) auf Seite 165
BinToDec	Wandelt einen Ausdruck in eine Dezimalzahl um	siehe BinToDec (Ausdruck) auf Seite 165
HexToBin	Wandelt einen Ausdruck um, der aus einer hexadezimalen Zahl in ASCII-Schreibweise besteht; Ausgabe binär	siehe HexToBin (Ausdruck) auf Seite 165
BinToHex	Wandelt einen Ausdruck in eine hexadezimale Zeichenkette um	siehe BinToHex (Ausdruck) auf Seite 165
DualToBin	Wandelt einen dualen Ausdruck in binäre Darstellung um	siehe DualToBin (Ausdruck) auf Seite 166
BinToDual	Wandelt einen Ausdruck in duale Darstellung um	siehe BinToDual (Ausdruck) auf Seite 166
PadRight	Füllt den Wert des Ausdrucks von rechts mit dem angegebenen „Zeichen“ auf, bis die Länge der Zeichenkette der Zahl „a“ entspricht	siehe PadRight (Ausdruck, Zeichen, a) auf Seite 166
PadLeft	Füllt den Wert des Ausdrucks von links mit dem angegebenen „Zeichen“ auf, bis die Länge der Zeichenkette der Zahl „a“ entspricht	siehe PadLeft (Ausdruck, Zeichen, a) auf Seite 166
IfEqualThenElse	Wenn-Dann-Abfrage. Die Funktion vergleicht zwei Ausdrücke. Wenn die Ausdrücke gleich sind, wird „ThenAusdruck“ weitergegeben, andernfalls „ElseAusdruck“	siehe IfEqualThenElse (VergleichAusdruck1, VergleichAusdruck2, ThenAusdruck, ElseAusdruck) auf Seite 166

Tabelle 25: Übersicht über Funktionen.

Mod10 (Ausdruck)

Berechnet die Prüfsumme für den Wert des Ausdrucks nach Modulo 10. Gewichtung = 3/1.

Parameter	Beschreibung
Ausdruck	Siehe Link „Ausdrücke“ am Ende des Abschnitts

Berechnungsbeispiel:

Data	**1**	**4**	**6**	**3**	**7**	**6**	**2**	**1**
Gewichtung (3/1)	1	3	1	3	1	3	1	3
Summe	1	12	6	9	7	18	2	3
Gesamtsumme	58							
Modulo	10							
Rest	8							
Prüfsumme	10 - 8 = **2**							

SubStr (Ausdruck, a, b)

Substring - Schneidet einen Teil aus dem Wert eines Ausdrucks aus (engl. substring = Unter-Zeichenkette).

Parameter	Wert	Beschreibung
Ausdruck		Siehe Link „Ausdrücke“ am Ende des Abschnitts
a	int	Position des ersten Zeichens, das ausgeschnitten werden soll. Das Zählen der Zeichen beginnt bei "0"
b	int	Anzahl der aufeinanderfolgenden Zeichen, die ausgeschnitten werden sollen.

Beispiel:

Kommandofolge	Beschreibung
`#VDT/Name////Hans Mueller#G`	Definition der Textvariablen „Name“
`#VW/L/"Erstes Zeichen = " + SubStr(Name,0,1)#G`	Ausgabe von: „Erstes Zeichen = H“

Length (Ausdruck)

Länge - Ermittelt die Länge (= Anzahl der Zeichen) des Ausdrucks.

Beispiel:

Kommandofolge	Beschreibung
`#VDT/Name////Hans Mueller#G`	Definition der Textvariablen „Name“
`#VW/L/"Textlänge = " + Length(Name)#G`	Ausdruck: “Textlänge = 12”

MergeRight (Ausdruck1, Ausdruck2)

Zeichenketten rechtsbündig verschmelzen – “Ausdruck2“ wird rechtsbündig in „Ausdruck1“ hineinkopiert.

Beispiel:

Kommandofolge	Beschreibung
`MergeRight(„00000000“,“123“)`	Ergebnis: 00000123

MergeLeft (Ausdruck1, Ausdruck2)

Zeichenketten linksbündig verschmelzen – “Ausdruck2“ wird linksbündig in „Ausdruck1“ hineinkopiert.

Beispiel:

Kommandofolge	Beschreibung
`MergeLeft(„00000000“,“123“)`	Ergebnis: 12300000

DayOfYear (Tag,Monat,Jahr)

Berechnet den Tag des Jahres.

Parameter	Beschreibung
Tag	2stellige Zeichenkette für den Tag
Monat	2stellige Zeichenkette für den Monat
Jahr	4stellige Zeichenkette für die Jahreszahl

Kommandofolge	Beschreibung
`DayOfYear(„01“,“08“,“2005“)`	Ergebnis: 213

Chr(a)

Wandelt eine ASCII-Code Dezimalzahl in das entsprechende alphanumerische ASCII-Zeichen um.

Parameter	Wert	Beschreibung
a	int	Zu konvertierender ASCII-Code [0..255]

Beispiel:

```
Chr(65)
```

Ergebnis: A

DecToBin (Ausdruck)

Wandelt den dezimalen Teil eines Ausdrucks in eine binäre Bytekette um. Die Dezimalzahl darf nicht größer als 2^{64} sein.

Beispiel:

```
DecToBin(„100“)
```

Ergebnis: d

BinToDec (Ausdruck)

Wandelt einen Ausdruck in eine Dezimalzahl um.

Beispiel:

```
BinToDec(„d“)
```

Ergebnis: 100

HexToBin (Ausdruck)

Wandelt einen Ausdruck um, der aus einer hexadezimalen Zahl in ASCII-Schreibweise besteht; Ausgabe binär.

Beispiel:

```
HexToBin(„3161“)
```

Ergebnis: 1a

BinToHex (Ausdruck)

Wandelt einen Ausdruck in eine hexadezimale Zeichenkette um.

Beispiel:

```
BinToHex(„1a“)
```

Ergebnis: 3161

DualToBin (Ausdruck)

Wandelt einen dualen Ausdruck in binäre Darstellung um.

Beispiel:

```
DualToBin(„0011000101100001“)
```

Ergebnis: 1a

BinToDual (Ausdruck)

Wandelt einen Ausdruck in duale Darstellung um.

Beispiel:

```
BinToDual(„1a“)
```

Ergebnis: 0011000101100001

PadRight (Ausdruck, Zeichen, a)

Füllt den Wert des Ausdrucks von rechts mit dem angegebenen „Zeichen“ auf, bis die Länge der Zeichenkette der Zahl „a“ entspricht.

Parameter	Wert	Beschreibung
Ausdruck		Siehe Link „Ausdrücke“ am Ende des Abschnitts
Zeichen		ASCII-Zeichen
a	int	Länge der Zeichenkette

Beispiel:

```
PadRight(„111“,“2“,5)
```

Ergebnis: „11122“

PadLeft (Ausdruck, Zeichen, a)

Füllt den Wert des Ausdrucks von links mit dem angegebenen „Zeichen“ auf, bis die Länge der Zeichenkette der Zahl „a“ entspricht.

Parameter	Wert	Beschreibung
Ausdruck		Siehe Link „Ausdrücke“ am Ende des Abschnitts
Zeichen		ASCII-Zeichen
a	int	Länge der Zeichenkette

Beispiel:

```
PadLeft(„10101“,“0“,8)
```

Ergebnis: „00010101“

IfEqualThenElse (VergleichAusdruck1, VergleichAusdruck2, ThenAusdruck, ElseAusdruck)

Wenn-Dann-Abfrage. Die Funktion vergleicht zwei Ausdrücke. Wenn die Ausdrücke gleich sind, wird „ThenAusdruck“ weitergegeben, andernfalls „ElseAusdruck“.

Parameter	Wert	Beschreibung
VergleichAusdruck1		Erster Vergleichsausdruck (siehe Link „Ausdrücke“ am Ende des Abschnitts)
VergleichAusdruck2		Zweiter Vergleichsausdruck (siehe Link „Ausdrücke“ am Ende des Abschnitts)
ThenAusdruck		Ergebnisausdruck, wenn die Aussage „VergleichAusdruck1 = VergleichAusdruck2“ *wahr* ist
ElseAusdruck		Ergebnisausdruck, wenn die Aussage „VergleichAusdruck1 = VergleichAusdruck2“ *nicht wahr* ist

Beispiel:

```
IfEqualThenElse(TESTVAR, "0", "No", "Yes")
```

Ergebnis: Wenn die Variable „TESTVAR“ = 0 ist, wird „No“ zurückgegeben, andernfals „Yes“.

Beispiel:

```
#G ----------------------------------------------------------------
#G Das Beispiel ermittelt und druckt den Wochentag als Textstring.
#G Ausgabetext: „Day of week: Monday" (oder „Tuesday" oder…)
#G Wenn eine Wochentag-Nummer von außerhalb des Bereichs [1..7]
#G übergeben wird, „Day of week: Error" gedruckt.
#G ----------------------------------------------------------------
#!A1
#IMN100/100
#ER
#VDD/DayOfWeek///^w#G
#VDE/DayOfWeekAsString//IfEqualThenElse(DayOfWeek,"1","Monday",
IfEqualThenElse(DayOfWeek,"2","Tuesday",
IfEqualThenElse(DayOfWeek,"3","Wednesday",
IfEqualThenElse(DayOfWeek,"4","Thursday",
IfEqualThenElse(DayOfWeek,"5","Friday",
IfEqualThenElse(DayOfWeek,"6","Saturday",
IfEqualThenElse(DayOfWeek,"7","Sunday"  ,"Error")))))))#G
#SF108
#T0#J10#VW/L/"Day of week: " + DayOfWeekAsString#G
#Q1/
```

Zugehörige Informationen

Variablen auf Seite 159
Ausdrücke auf Seite 159

ARITHMETISCHE FUNKTIONEN

Allgemeine Bedingungen

Die folgenden arithmetischen Funktionen werden unterstützt:

Funktion	Operation	Beschreibung
Add	Addition	Siehe Kapitel Add(Ausdruck 1, Ausdruck 2, Ausgabeformat) auf Seite 169
Sub	Subtraktion	Siehe Kapitel Sub(Ausdruck 1, Ausdruck 2, Ausgabeformat) auf Seite 170
Mul	Multiplikation	Siehe Kapitel Mul(Ausdruck 1, Ausdruck 2, Ausgabeformat) auf Seite 170
Div	Division	Siehe Kapitel Div(Ausdruck 1, Ausdruck 2, Ausgabeformat) auf Seite 170
IfThenElse	Vergleich zweier Zahlen	Siehe Kapitel IfThenElse(Ausdruck 1, Operator, Ausdruck 2, Ausdruck 3, Ausdruck 4) auf Seite 171

Tabelle 26: Übersicht über arithmetische Funktionen

Parameter dieser Funktionen können alle Ausdrücke sein, die eine gültige Gleitkomma-Zeichenkette ergeben.

Als Trennzeichen kann ein Komma („,“) oder ein Punkt („.“) verwendet werden. Wenn ein Komma in mindestens einem Parameter verwendet wird, wird das Ergebnis automatisch mit Komma dargestellt. Andernfalls erfolgt die Darstellung mit einem Punkt.

Beispiele für gültige Gleitkomma-Zeichenketten:

- 2015
- -0.255
- 0,30
- -338.645E-1

Unterstützter Zahlenbereich: ±2.22e-308 bis ±1.80e+308

Zulässige Stellenzahl der Mantisse: 15-16

Ergebnisformat

Das Ergebnisformat wird nach folgendem Schema festgelegt:

```
%[flags][width][.precision][length]specifier
```

Wobei das “specifier”-Zeichen am Ende der wichtigste Teil ist, denn er legt Typ und Interpretation des zugrundeliegenden Arguments fest:

Specifier	Ausgabe	Beispiel
f	Dezimale Gleitkomma-Darstellung; Kleinschreibung	392.65
F	Dezimale Gleitkomma-Darstellung; Großschreibung	392.65
e	Wissenschaftliche Darstellung (Mantisse/Exponent); Kleinschreibung	3.9265e+2

Specifier	Ausgabe	Beispiel
E	Wissenschaftliche Darstellung (Mantisse/Exponent); Großschreibung	3.9265E+2
g	Benutze die kürzeste Darstellung: %e oder %f	392.65
G	Benutze die kürzeste Darstellung: %E oder %F	392.65

Das Ergebnisformat kann zusätzlich Unter-Specifier enthalten: *Flags*, *width*, *precision* und modifiers (in dieser Reihenfolge). Diese können optional angegeben werden und folgen diesen Spezifikationen:

Flags	Beschreibung
-	Linksseitige Ausrichtung innerhalb der gegebenen Feldbreite; Voreinstellung: Rechtsseitige Ausrichtung (siehe „Width“).
+	Erzwingt die Ergebnis-Darstellung mit Vorzeichen (+ oder -). Voreinstellung: Nur negative Zahlen werden mit Vorzeichen dargestellt.
(Leerzeichen)	Wenn kein Vorzeichen vorangestellt wird, wird ein Leerzeichen vor dem Ergebnis eingefügt.
0	Ersetzt links vom Ergebnis stehende Leerzeichen durch Nullen, wenn linksseitiges Auffüllen gewählt wurde (siehe „Width“).

Width	Beschreibung
(Zahl)	Minimal zu druckende Zeichenzahl. Wenn der Ergebniswert kürzer ist als diese Zeichenzahl, wird links mit Leerzeichen aufgefüllt. Sollte der Ergebniswert länger sein, wird er nicht gekürzt.
*	Die Breite (width) wird nicht im Ergebnisformat festgelegt. Stattdessen wird dem Ergebniswert ein Integer-Argument vorangestellt.
.Zahl	Für e-, E-, f- und F-Spezifier: Anzahl der Zeichen nach dem Dezimalpunkt (Voreinstellung: 6). Für g- und G-Spezifier: Max. Anzahl signifikanter Zeichen. Wenn der Punkt ohne Zahl angegeben wird, wird automatisch „0“ eingesetzt.

Beispiele:

Ergebnisformat	Beispielwert
%F	392.65
%E	3.9265E+2
%.2f	3.93 (gerundet auf 2 Dezimalstellen)
%08.4f	003.9265 (gerundet auf 2 Dezimalstellen)(8 Stellen – 1 Dezimalpunkt – 4 Stellen; links mit 2 Nullen aufgefüllt)

Add(Ausdruck 1, Ausdruck 2, Ausgabeformat)

Addition; Rückgabewert = Ausdruck 1 + Ausdruck 2

Parameter	Beschreibung
Ausdruck1	Erster Summand
Ausdruck2	Zweiter Summand
Ausgabeformat	Siehe Kapitel „Ergebnisformat“ oben

Beispiel:

```
#VDT/Value1////33,64#G
#VW/L/Add(Value1, "3,33", "%.2f")#G
```

Ergebnis: "36,97"

Sub(Ausdruck 1, Ausdruck 2, Ausgabeformat)

Subtraktion; Rückgabewert = Ausdruck 1 - Ausdruck 2

Parameter	Beschreibung
Ausdruck1	Minuend
Ausdruck2	Subtrahend
Ausgabeformat	Siehe Kapitel „Ergebnisformat“ oben

Beispiel:

```
#VW/L/Sub("20", "1.20", "%.2f")#G
```

Ergebnis: "18.80"

Mul(Ausdruck 1, Ausdruck 2, Ausgabeformat)

Multiplikation; Rückgabewert = Ausdruck 1 * Ausdruck 2

Parameter	Beschreibung
Ausdruck1	1. Faktor
Ausdruck2	2. Faktor
Ausgabeformat	Siehe Kapitel „Ergebnisformat“ oben

Beispiel:

```
#VDT/Value1////33,64#G
#VDT/Value2/C/10/1/10#G
#VW/L/Mul(Value1, Value2, "%.2f")#G
```

Ergebnis: "336,40"

Div(Ausdruck 1, Ausdruck 2, Ausgabeformat)

Division; Rückgabewert = Ausdruck 1 / Ausdruck 2

Parameter	Beschreibung
Ausdruck1	Dividend
Ausdruck2	Divisor Wenn der Divisor „0.0“ ist, gibt die Funktion den Wert „inf“ zurück, weil das Ergebnis nicht definiert ist.
Ausgabeformat	Siehe Kapitel „Ergebnisformat“ oben

Beispiel:

```
#VDT/Value2/C/10/1/3.33#G
```

```
#VW/L/Div("200.50", Value2, "%.2f")#G
```

Ergebnis: "60,21"

IfThenElse(Ausdruck 1, Operator, Ausdruck 2, Ausdruck 3, Ausdruck 4)

Parameter	Beschreibung
Ausdruck 1	Erste zu vergleichende Zahl
Operator	Vergleichsoperator; Zulässig sind „>“, „>=“, „<“, „<=“, „==“, „!=“
Ausdruck 2	Zweite zu vergleichende Zahl
Ausdruck 3	Rückgabewert, wenn das Ergebnis des Vergleichs *richtig* ist („true“)
Ausdruck 4	Rückgabewert, wenn das Ergebnis des Vergleichs *falsch* ist („false“)

Beispiel:

```
#VDT/Number/C///3.33#G
#VDE/IfResult//IfThenElse(Number, "<", "0,0", "negative number", "positi▶
ve number")#G
```

Ergebnis: "positive number"

Anwendungsbeispiel

```
#!A1
#IMN100/120
#ER
#VDT/Value0////-338,645E-1#G
#VDT/Value1////33,64#G
#VDT/Value2/C/10/1/010#G
#SF106/
#FD0
#T2#J110
#VW/L/Value1 + " * " + Value2 + " = " + Mul(Value1, Value2, "%E")
#T2#J105
#VW/L/"100.33" + " * " + Value2 + " = " + Mul("100"+".33", Value2,
 "%014.11f")
#T2#J100
#VW/L/Value1 + " * " + Value2 + " = " + Mul(Value1, Value2, "%.2f")
#T2#J95
#VW/L/Value1 + " * " + Value2 + " = " + Mul(Value1, Value2, "%.0f")
#T2#J85
#VW/L/Value1 + " / " + Value2 + " = " + Div(Value1, Value2, "%E")
#T2#J80
#VW/L/Value1 + " / " + Value2 + " = " + Div(Value1, Value2, "%014.11f")
#T2#J75
#VW/L/Value1 + " / " + Value2 + " = " + Div(Value1, Value2, "%.2f")
#T2#J70
#VW/L/Value1 + " / " + Value2 + " = " + Div(Value1, Value2, "%.0f")
#T2#J65
#VW/L/Value1 + " / " + "0.0" + " = " + Div(Value1, "0.0", "%.0f")
#T2#J60
#VW/L/Value1 + " + " + Value2 + " = " + Add(Value1, Value2, "%E")
#T2#J55
#VW/L/Value1 + " + " + Value2 + " = " + Add(Value1, Value2, "%014.11f")
#T2#J50
#VW/L/Value1 + " + " + Value2 + " = " + Add(Value1, Value2, "%.2f")
#T2#J45
#VW/L/Value1 + " + " + Value2 + " = " + Add(Value1, Value2, "%.0f")
```

```
#T2#J35
#VW/L/Value1 + " - " + Value2 + " = " + Sub(Value1, Value2, "%E")
#T2#J30
#VW/L/Value1 + " - " + Value2 + " = " + Sub(Value1, Value2, "%014.11f")
#T2#J25
#VW/L/Value1 + " - " + Value2 + " = " + Sub(Value1, Value2, "%.2f")
#T2#J20
#VW/L/Value1 + " - " + Value2 + " = " + Sub(Value1, Value2, "%.0f")
#VDE/Result//Div(Add(Value0, Value2, "%E"), Value2,"%.4f")#G
#T2#J10
#VW/L/"( " + Value0 + " + " + Value2 + " ) / " + Value2 + " = " + Re▶
sult#G
#VDE/IfResult//IfThenElse(Result, "<", "0,0", Sub(",.0",Result,"%.4f"),
 Result)#G
#T2#J5 #VW/L/"ABS( " + Result + " ) = " + IfResult#G
#T2#J1#YL0//1/90
#Q5#G
```

Zugehörige Informationen

Variablen auf Seite 159
Ausdrücke auf Seite 159

UNICODE

Unicode ist ein herstellerübergreifender Standard, der weltweit jedem Zeichen eine eindeutige Index-Codierung zuweist. Damit können z.B. asiatische oder arabische Zeichen mittels Easy Plug gedruckt werden.

Voraussetzungen

- *Zeichensatz*: Ein Unicode-codierter True Type Font muss auf einem externen Speichermedium vorliegen.

  Manche Fonts enthalten nur eine Untermenge aller Unicode-Zeichen. Stellen Sie sicher, dass der Font das betreffende Zeichen enthält.

  Der Zugriff auf externe Fonts [7] wird wesentlich schneller, wenn die Fonts in den Hauptspeicher (RAM) des Druckers geladen werden. Dies geschieht kurz nach dem Einschalten des Druckers. Voraussetzung ist eine Fontnummer aus dem Bereich [900...999] (näheres siehe „Handbuch Externe Flashspeicher“).

- *RAM*: Im Arbeitsspeicher (RAM) des Druckers muss ausreichend Platz für Fonts zur Verfügung stehen (näheres siehe „Handbuch Externe Flashspeicher“).

Limitationen

- Unicode wird nicht mit *Speedo-Fonts* unterstützt
- *Zählfelder* und *variable Felder* dürfen nicht zusammen mit Unicode-Zeichen verwendet werden.
- Weitere Limitationen für die Verwendung *arabischer Glyphen* siehe Link am Ende des Kapitels unter „Zusätzliche Informationen“.

Druckkommandos

Unicode-Zeichen können mit folgenden Easy-Plug-Befehlen gedruckt werden, wenn der Parameter "U" (für arabische Zeichen „G“) verwendet wird:

7 Fonts, die nicht fest im Drucker installiert sind (interne Fonts), z. B. Fonts, die auf einem externen Speichermedium gespeichert sind,

- #YN
- #SS + #VW

Eingabe des Textstrings

Für die Eingabe des Textstrings gibt es folgende beiden Möglichkeiten:

- Eingabe des Unicode-Indexes (\uxxxx) jedes einzelnen Zeichens
- UTF-8-Kodierung des Textes: Text in einen Texteditor mit UTF-8-Kodierung eingeben oder kopieren (z. B. Standard-Texteditor unter Windows)

Um ein Zeichen zu drucken, muss dessen Unicode Index (hex.) bekannt sein.

Unicode Indizes finden Sie unter www.unicode.org oder Sie wandeln den Text mit dem Unicode Converter um.

Kommandobeispiel:

```
#YN902/0U/60///Char A = \u0041#G
```

Wenn einer der beiden Parameter "U" oder "G" im #YN-Kommando gesetzt ist und die Zeichenfolge "\u" im String gefunden wird, gefolgt von einem gültigen Hexadezimalwert, wird das Unicode-Zeichen mit dem entsprechendem Index gedruckt. Ist das Zeichen nicht im Font enthalten, wird ein Rechteck gedruckt. Ist kein gültiger Hexadezimalwert angegeben, wird die komplette Zeichenfolge als normaler Text gedruckt.

Unicode-Zeichen können auch zusammen mit ASCII-Zeichen gedruckt werden. Alle Optionen, außer Zählfeldern und variablen Feldern können mit Unicode verwendet werden.

UTF-8

Der Textstring wird in einen UTF-8-fähigen Texteditor eingegeben und als UTF-8-kodierter Text gespeichert. Der so erstellte Easy-Plug Druckauftrag kann direkt an die Datenschnittstelle des Druckers gesendet werden, wenn der Drucker vorher auf UTF-8-Kodierung eingestellt wurde. Dies kann auf zweierlei Weise geschehen:

- Easy Plug: Kommando #N13
- Parametermenü: SYSTEM PARAMETER > Zeichensätze = „UTF-8“ bzw. Drucker Sprache > Easy-Plug Einst. > Zeichensätze = „UTF-8“.

Siehe Beispiele 4 und 7.

Beispiele

Beispiel 1:

```
#G ------------------------------------------------
#G Druckausgabe: Char A = A
#G ------------------------------------------------
#!A1
#IMN100/30
#ERN
#T5#J10
#YN902/U/60///Char A = \u0041#G
#Q1#G
```

Beispiel 2:

```
#G ------------------------------------------------
#G Der Zeichenfolge „\u“ muß ein gültiger Hexadezimalwert folgen. Ist
#G dies nicht der Fall, wird die komplette Zeichenfolge gedruckt.
```

```
#G Druckausgabe: Char A = \u004m
#G ------------------------------------------------
#!A1
#IMN100/30
#ERN
#T5#J10
#YN902/0U/60///Char A = \u004m#G
#Q1#G
```

Beispiel 3:

```
#G ------------------------------------------------
#G Druckausgabe: Char A = \u041
#G ------------------------------------------------
#!A1
#IMN100/30
#ERN
#T5#J10
#YN902/0U/60///Char A = \u041#G
#Q1#G
```

Beispiel 4: Kyrillischer Text UTF-8

```
#G ------------------------------------------------
#G Druckausgabe: Указания по тезнике безопасности при
#G эксплуатации машины для печатания этикеток
#G ------------------------------------------------
#!A1
#IMN200/100
#N13
#ER
#SS900/OV/32#G
#T01.0#J10.0
#VW/L/"Указания по тезнике безопасности при эксплуатации машины для
 печатания этикеток"
#Q1#G
```

Beispiel 5: Arabischer Text #YN (Druckausgabe siehe unten)

```
#!A1
#IMSR100/200
#ERN/1//0
#T090.50#J90.33
#YN901/RG/50///\u0627\u0644\u0643\u0645\u064A\u0629 60 \u062E
 \u0628\u0632 \u0627\u0644\u0634\u0648\u0643\u0648\u0644\u0627 \u062A
\u0629 65 \u063A#G
#Q1#G
```

Beispiel 6: Arabischer Text #SS + #VW (Druckausgabe siehe unten)

```
#!A1
#IMSR100/200
#ERN/1//0
#T090.50#J90.33
#SS901/RG/42X42/0
#VW/L/"\u0627\u0644\u0643\u0645\u064A\u0629 60 \u062E\u0628
 \u0632\u0627\u0644\u0634\u0648\u0643\u0648\u0644\u0627\u062A \u0629 65
 \u063A"#G
#Q1#G
```

Beispiel 7: Arabischer Text UTF-8 (Druckausgabe siehe unten)

```
#!A1
#IMSR100/200
#N13
#ERN/1//0
```

```
#T090.50#J90.33
#SS901/RG/42X42/0
#G ------------------------------------------------
#G UTF-8-Daten: D8 A7 D9 84 D9 83 D9 85 D9 8A D8 A9 20 36 30 20
#G D8 AE D8 A8 D8 B2 20 D8 A7 D9 84 D8 B4 D9 88 D9 83 D9 88 D9
#G 84 D8 A7 D8 AA D8 A9 20 36 35 20 D8 BA
#G ------------------------------------------------
#VW/L/"غ 65 ة تالشوكولا زالخب 60 ةيمكلا"#G
#Q1#G
```

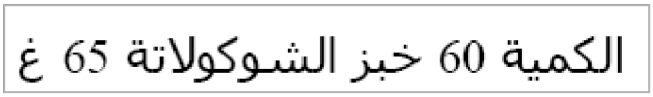


Bild 32: Ausdruck zu den Beispielen 5 bis 7.

Zugehörige Informationen

#YN - Textfeld auf Seite 128
Arabische Glyphen auf Seite 176

ZUSAMMENHANG ZWISCHEN RATIO UND BREITENFAKTOR VON BARCODES

Da nur ganze Dots gedruckt werden können, wird ein gegebener Ratio auf den nächst möglichen Wert aufgerundet, wobei wiederum der eingestellte Breitenfaktor eine maßgebliche Rolle spielt (Anzahl Dots für das schmale Element).

Ein breiterer Barcode erleichtert das Einhalten des erfoderlichen Barcode Ratios.

Beispiele

Barcode-Breitenfaktor 1 :

- Das schmale Element ist 1 Dot breit
- Das breite Element kann 2 oder 3 Dots breit sein
- Erforderliche Ratios von 1:2 bis 1:2,5 werden mit dem Ratio 1:2,0 gedruckt
- Erforderliche Ratios von 1:25 bis 1:3 werden mit dem Ratio 1:3,0 gedruckt

Barcode-Breitenfaktor 2 :

- Das schmale Element ist 2 Dots breit
- Das breite Element kann 4, 5 oder 6 Dots breit sein
- Erforderliche Ratios von 1:2,0 bis 1:2,25 werden mit dem Ratio 1:2,0 gedruckt
- Erforderliche Ratios von 1:2,5 bis 1:2,75 werden mit dem Ratio 1:2,5 gedruckt
- Erforderliche Ratios von 1:2,75 bis 1:3,0 werden mit dem Ratio 1:3,0 gedruckt

Barcode-Breitenfaktor 3 :

- Das schmale Element ist 3 Dots breit
- Das breite Element kann 6, 7, 8 oder 9 Dots breit sein
- ...

Zugehörige Informationen

#SB - Barcode Definition auf Seite 80
Das Kommando #SB definiert einen Barcode. Das Drucken des Barcodes veranlasst ein nachfolgendes #VW-Kommando.

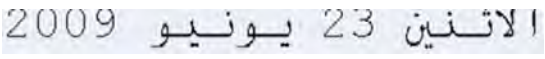
#YB - Barcode definieren und drucken auf Seite 114
Das Kommando #YB druckt ein Barcode-Feld.

ARABISCHE GLYPHEN

Limitationen für die Verwendung arabischer Glyphen mit Easy Plug

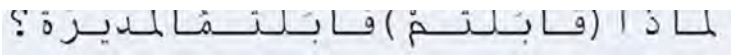
- Arabische Buchstaben im Bereich von 0x0600 – 0x06FF werden durch ihren jeweiligen kontextuellen Glyphen (Isolierte Form, Initial-, Medial-, Finalform) ersetzt. Urdu-, Sindhi- und persische Glyphen in diesem Bereich werden nur in der isolierten Form gedruckt.
- Wenn eine sehr kleine Zeichengröße benutzt wird, kann das Probleme bei der Definition der Glyphen verursachen (engl.: glyph smudging).
- Es wird nur nicht-vokalisiertes Arabisch unterstützt. Nicht alle Ligaturen und diakritischen Kombinationen des vokalisierten Arabisch werden unterstützt.
- Wenn nicht-vokalisiertes Arabisch benutzt wird, ist nur eine transparente Glyphe (Diakritische Marke, 0x064B – 0x065E) zwischen zwei nicht transparenten Glyphen erlaubt. Wenn zwei diakritische Marken hintereinander gesetzt würden, könnten diese überlappen.
- Die Ligatur Shadda (0x0651) mit Dammatan, Kasratan, Fatha, Damma und Kasra wird nur in der isolierten Form unterstützt, d. h. sie kann nur am Ende arabischer Wörter verwendet werden. Medial- und Initialformen müssen in Unicode überprüft werden.
- Wenn die Ersatzform eines Glyphen (Initial-, Medial-, Finalform, kombinierte Ligaturen) nicht unterstützt wird, wird statt dessen die isolierte Form gedruckt.
- Wenn die Glyphe im verwendeten Zeichensatz nicht vorhanden ist, erfolgt *keine* Fehlermeldung.
- Unicode Formatierungszeichen (z. B. RLM 0x200E, LRM 0x200F, LRE 0x202A, RLE 0x202B) werden nicht aus dem Datenstrom gefiltert. Diese Zeichen dürfen daher nicht für gerichtetes Kodieren von Druckdaten verwendet werden.
- Abhängig von der Position von nicht arabischen Text in einem Satz kann es erforderlich sein, ein Leerzeichen am Ende einzufügen, damit der Ausdruck korrekt erfolgt.

```
#YN900/0G/100///\u0627\u0644\u0627\u062b\u0646\u064a\u0646 23 \u064a
\u0648\u0646\u064a\u0648 2009 #G
```



- Benutzung von Klammern: Da Klammern nicht arabische Glyphen sind, werden diese innerhalb von arabischen Text in der falschen Richtung gedruckt. Um einen korrekten Ausdruck zu erreichen, müssen die entgegengesetzten Klammern verwendet werden.

```
\u0644\u0645\u0627\u0630\u0627)\u0642\u0627\u0628\u064E\u0644 \u062A
\u0640\u064F\u0645\u0652(\u0642\u0627\u0628\u064E\u0644 \u062A
\u0640\u064F\u0645\u064F\u0627\u0644\u0645\u062F\u064A \u0631\u064E
\u0629\u064E\u061F#G
```



Beispiele mit arabischem Text

Beispiel 1: Arabische und lateinische Schrift kombiniert

```
#!A1
#IM200/50
#ER
#G ------------------------------------------------
#G Variablendefinition
#G ------------------------------------------------
#VDT/User////NOVEXX Solutions 64-08
#G ------------------------------------------------
#G Text: "User Name:"
#G ------------------------------------------------
#SS900/OVG/44/#G
#T020.0 #J020.0
#VW/L/"\u0627\u0644\u0645\u0646\u062A\u062C
\u0627\u0644\u0648\u0635\u0641"+ User+" :"
#Q1/#!P1
```

Bild 33: Ausdruck zu Beispiel 1.

Beispiel 2: Variable Daten

```
#!A1
#IM200/120
#DM ^\u064A\u0646\u0627\u064A\u0631
^\u0641\u0628\u0631\u0627\u064A\u0631
^\u0627\u0644\u0645\u0633\u064A\u0631\u0629
^\u0623\u0628\u0631\u064A\u0644
^\u0631\u0628\u0645\u0627
^\u064A\u0648\u0646\u064A\u0648
^\u064A\u0648\u0644\u064A\u0648
^\u0623\u063A\u0633\u0637\u0633
^\u0633\u0628\u062A\u0645\u0628\u0631
^\u0623\u0643\u062A\u0648\u0628\u0631
^\u0646\u0648\u0641\u0645\u0628\u0631
^\u062F\u064A\u0633\u0645\u0628\u0631
#ER
#VDD/DateTime/Y//^R : ^S ^D
#SS900/OVG/44/#G
#FD/0/R#G
#T50.0 #J20.0
#VW/L/"\u0633\u062C\u0644\u062A\u0627\u0631\u064A\u062E "+DateTime#G
#Q1/#!P1
```

Bild 34: Ausdruck zu Beispiel 2.

Beispiele mit arabischem Text und Zahlen

Beispiel 1:

```
#T100.0 #J50.0
#YN900/0GR/100x60///\u0627\u0644\u0633\u0639\u0631 100\u066B00 : \u062F
\u0648\u0644\u0627\u0631#G
```

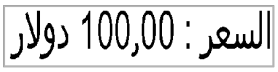


Bild 35: Ausdruck zu Beispiel 1 (Deutsch: „Preis: 100,00 Dollars“).

Beispiel 2:

```
#T100.0 #J50.0
#YN900/0GR/100x60///\u0627\u0644\u0648\u0632\u0646 20,2 :
 \u0643\u0644\u063A#G
```

Bild 36: Ausdruck zu Beispiel 2 (Deutsch: „Gewicht: 20,2 kg“)

Beispiel 3:

```
#T100.0 #J50.0
#YN900/0GR/100x60///\u0627\u0644\u0641\u0648\u0644\u0637\u064A\u0629
 14.4V :#G
```

Bild 37: Ausdruck zu Beispiel 3 (Deutsch: „Spannung: 14,4 V“)

Beispiel 4:

```
#T100.0 #J50.0
#YN900/0GR/100x60///\u0645\u0626\u0648\u064A %100.00 :#G
```

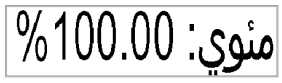


Bild 38: Ausgabe zu Beispiel 4 (Deutsch: „Prozentsatz: 100,00%“)

Beispiele mit arabischen Zahlen

Beispiel 1:

```
#YN900/0GR/100x60///\u0661\u0662\u0663\u066B\u0660\u0664\u0665\u0666#G
```

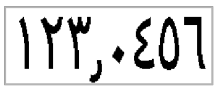


Bild 39: Ausdruck zu Beispiel 1 (Deutsch: „123,0456“)

Beispiel 2:

```
#YN900/0GR/100x60///
\u0660\u0661\u0662\u0663\u0664\u0665\u0666\u0667\u0668\u0669#G
```

Bild 40: Ausgabe zu Beispiel 2 (Deutsch: „123456789“)

Beispiel 3:

```
#YN900/0GR/100x60///
\u06F0\u06F1\u06F2\u06F3\u06F4\u06F5\u06F6\u06F7\u06F8\u06F9#G
```

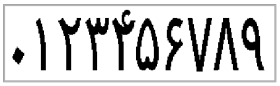


Bild 41: Ausdruck zu Beispiel 3 (Deutsch: „0123456789“)

Zugehörige Konzepte

Unicode auf Seite 172

Zugehörige Informationen

#YN - Textfeld auf Seite 128

NOVEXX SOLUTIONS

Telefon: +49-8165-925-0 | www.novexx.com

Novexx Solutions GmbH
Ohmstraße 3
85386 Eching