
IDOL

Programmierhandbuch

Ausgabe 3.4L

MICRO PLEX®

Inhaltsverzeichnis

Kapitel	Seite
1. Einführung	13
1.1. IDOL - Philosophie	15
1.2. Funktionalität	16
1.3. Metrik	19
1.4. Tools, Software	21
1.5. Aufbau von IDOL - Befehlen	23
2. Funktionsgruppe JOB	27
2.1. Transparentcode ändern (temporär)	28
2.2. Auswahl der Emulation	29
2.3. Druckauflösung wählen	33
2.4. Umschalten des Papierformats	34
2.5. Anwählen des Papiereinzugs und Feeder Mapping	37
2.6. Anwählen des Papierablagefachs, Stacker Mapping und Job-Jogging	40
2.7. Mehrfachdruck einer Seite ohne Papierfach - Anwahl	44
2.8. Mehrfachdruck eines kompletten Druckjobs	45
2.9. Durchschlag - Funktion	47
2.10. Vervielfältigen von Informationsbereichen	49
2.11. Der eingeschränkte CLEAR - Befehl	51
2.12. Der CLEAR - Befehl	53
2.13. Der Reset - Befehl	54
2.14. Der FORM FEED - Befehl	56
2.15. Duplex - Druck	57
2.16. Two-Up Mode (N-Up Mode)	59
2.17. Seitennummerierung	61
2.17.1. Seitenzähler setzen	61
2.17.2. Seitenzähler erhöhen	62
2.17.3. Drucken der Seitennummer	62
2.17.4. Seitenzähler verringern	63

Kapitel	Seite
2.18. Druckfarbe wählen	64
2.18.1. Beispiel: IGP – Befehle zur Druckfarbe	65
2.18.2. Beispiel: CODE V - Befehle	67
2.18.3. Beispiel: Prescribe IIe - Befehle	68
2.18.4. Beispiel: Epson FX (ESC/P) - Befehle	69
2.19. Druckrichtung wählen	70
2.20. Schneidebefehl (inkl. Tear off und Peel off)	72
2.21. Heftbefehl (Staple)	73
2.22. Kommentar	74
3. Funktionsgruppe FONT	75
3.1. Textcursor	81
3.1.1. Absolutes Positionieren des Textcursors	82
3.1.2. Speichern der aktuellen Textcursor - Position	83
3.1.3. Anfahren der gespeicherten Textcursor - Position	84
3.1.4. Vertikale Relativpositionierung des Textcursors	85
3.1.5. Horizontale Relativpositionierung des Textcursors	86
3.1.6. Absolute Positionierung des Textcursors in x-Richtung	87
3.1.7. Absolute Positionierung des Textcursors in y-Richtung	88
3.2. Ränder	89
3.2.1. Linken Textrand auf die aktuelle Textcursor - Position setzen	90
3.2.2. Oberen Textrand auf die aktuelle Textcursor - Position setzen	91
3.2.3. Unteren Textrand auf die aktuelle Textcursor - Position setzen	92
3.2.4. Rechten Textrand auf die aktuelle Textcursor - Position setzen	93
3.2.5. Textränder auf Einschaltländer zurücksetzen	94
3.2.6. Textränder auf die Papierränder setzen	95
3.3. Textgestaltungsfunktionen	96
3.3.1. Null mit Schrägstrich	99

Kapitel	Seite
3.4. Tabulatorfunktionen	100
3.4.1. Horizontalen Tabulator an aktueller Cursorposition setzen	101
3.4.2. Vertikalen Tabulator an aktueller Cursorposition setzen	102
3.4.3. Horizontalen Tabulator ausführen	103
3.4.4. Vertikalen Tabulator ausführen	104
3.4.5. Alle horizontalen und vertikalen Tabulatoren löschen	105
3.5. Zeichenformatierung	106
3.5.1. Auswahl der Zeichentabelle	106
3.5.2. Auswahl der Schriftart	108
3.5.3. Bestimmung der Schreibrichtung	109
3.5.4. Festlegen der horizontalen Zeichenschrittweite	110
3.5.5. Festlegen der vertikalen Zeichenschrittweite (Zeilenabstand)	112
3.5.6. Inverse Zeichendarstellung (Befehl 1)	114
3.5.7. Inverse Zeichendarstellung (Befehl 2)	115
3.5.8. Rasterhinterlegung (Befehl 1)	116
3.5.9. Rasterhinterlegung (Befehl 2)	117
3.5.10. Fettdruck (Befehl 1)	118
3.5.11. Fettdruck (Befehl 2)	119
3.5.12. Unterstreichen (Befehl 1)	120
3.5.13. Unterstreichen (Befehl 2)	121
3.5.14. Sperrschrift (Befehl 1)	122
3.5.15. Sperrschrift (Befehl 2)	123
3.5.16. Hochstellen (negative Halbzeilenschaltung)	124
3.5.17. Tiefstellen (positive Halbzeilenschaltung)	125
3.6. Skalieren von Fonts	126
3.6.1. Vergrößerung von Zeichen eines Pixelfonts	127
3.6.2. Vektorfont mit Breiten- und Höhenfaktor	128
3.6.3. Skalieren von Outlinefonts und Stickfonts	129
3.6.4. Maßeinheit zum Vergrößern von Outlinefonts und Stickfonts	130
3.6.5. Skalieren der Höhe von Outlinefonts und Stickfonts	131
3.6.6. Skalieren der Breite von Outlinefonts und Stickfonts	132
3.6.7. Schreibrichtung mit Winkelangabe (Grafik-Fonts)	133
3.6.8. Zeichenneigung mit Winkelangabe (Grafik-Fonts)	134
3.6.9. Beispiele	135

Kapitel	Seite
4. Funktionsgruppe GKS (Grafisches - Kern - System)	137
4.1. Allgemeine Grafikfunktionen	138
4.1.1. Abbildungsfunktionen (Cursor und X/Y-Offsets setzen)	138
4.1.2. Definieren der internen Grafikauflösung	140
4.1.3. Setzen der Grafik - Ränder	141
4.1.4. Grafik-Reset	143
4.2. Eigenschaften grafischer Elemente	144
4.2.1. Linienbreite	144
4.2.2. Linienmode	145
4.2.3. Farbe von Grafikelementen	148
4.2.4. Raster	149
4.2.5. Graustufen-Raster	150
4.2.6. Raster laden	151
4.2.7. Hinterlegungsmodus	153
4.2.8. Hinterlegungsmodus für Zeichen und Bitmaps	155
4.2.9. Automatisches Umranden von Grafik-Flächen	156
4.3. Grafik mit absoluter Positionierung	159
4.3.1. Positionieren des Grafik-Cursors	160
4.3.2. Positionieren des Grafik-Cursors in x-Richtung	161
4.3.3. Positionieren des Grafik-Cursors in y-Richtung	162
4.3.4. Fläche	163
4.3.5. Kreis	165
4.3.6. Linienzug	166
4.3.7. Kreisbogen	167
4.3.8. Kreisfläche	168
4.3.9. Kreisflächensegment	169
4.3.10. Balken	170
4.3.11. Balken mit runden Ecken	171
4.3.12. Box mit runden Ecken	172
4.3.13. Ellipse	173
4.3.14. Ellipsenfläche	174
4.3.15. Ellipsenflächen-Segment	175
4.3.16. Ellipsenflächen-Abschnitt	177

Kapitel	Seite
4.4. Grafik mit relativer Positionierung	179
4.4.1. Verschieben des Grafikcursors	180
4.4.2. Fläche	181
4.4.3. Kreis	183
4.4.4. Linienzug	184
4.4.5. Kreisbogen	186
4.4.6. Kreisfläche	187
4.4.7. Kreisflächensegment	188
4.4.8. Balken	189
4.4.9. Balken mit runden Ecken	190
4.4.10. Box mit runden Ecken	191
4.4.11. Ellipse	192
4.4.12. Ellipsenfläche	193
4.4.13. Ellipsenflächen-Segment	194
4.4.14. Ellipsenflächen-Abschnitt	196
4.5. Grafik in Polarkoordinaten	198
4.5.1. Verschieben des Grafikcursors	199
4.5.2. Linienzug	200
4.6. Übersicht: Grafikfunktionen und Darstellungselemente	201
5. Funktionsgruppe DOWNLOAD	203
5.1. Einrichten und Ergänzen einer DOWNLOAD-Schrift	204
5.1.1. Laden eines einzelnen Zeichens	204
5.1.2. Parameter eintragen für eine DOWNLOAD-Schrift	208
5.1.3. Löschen einer DOWNLOAD-Schrift	210

Kapitel	Seite
6. Funktionsgruppe MACRO (Elektronische Formulare)	211
6.1. Speichern und Wiederherstellen von aktuellen Textparametern	214
6.2. Definieren eines MACROs	215
6.3. Aufrufen eines MACROs	217
6.4. Automatischer Aufruf eines MACROs	218
6.5. MACRO-Aufruf mit einem Schlüsselwort (Keyword)	219
6.6. Löschen eines MACROs	221
6.7. FLASH-EPROM (RFM) programmieren	222
7. Funktionsgruppe BITMAP	225
7.1. Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen (BITMAP) ohne Vergrößerungsfaktor	225
7.2. Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen (BITMAP) mit Vergrößerungsfaktor	227
7.3. Neuer BITMAP-Befehl (Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen)	229

Kapitel	Seite
8. Funktionsgruppe BARCODE	233
8.1. Übersicht: Generieren eines BARCODES	233
8.2. Eindimensionale Barcodes	242
8.2.1. 2/5 Interleaved (USD-1) (Zweibreiten-Barcode)	243
8.2.2. Code 39 (3/9) (USD-3) (Zweibreiten-Barcode)	245
8.2.3. MSI (Zweibreiten-Barcode)	246
8.2.4. Code 128 (Mehrbreiten-Barcode)	247
8.2.5. EAN 8, EAN 13 und UPC A, UPC E (Mehrbreiten-Barcode)	251
8.2.6. EAN 8 Add On und EAN 13 Add On (Mehrbreiten-Barcode)	253
8.2.7. Postnet, Planet, Kixbar und OneCode (Höhenmodulierte Barcodes)	254
8.3. Zweidimensionale Barcodes	256
8.3.1. PDF 417	256
8.3.2. USD-5	259
8.3.3. DataMatrix	260
8.3.4. MaxiCode (UPS)	262
8.3.5. Aztec	264
8.3.6. QR-Code	268
9. RFID Funktionsgruppe	271
9.1. Befehle für RFID	272
9.1.1. Übersicht	272
9.1.2. Etiketten-Identifikationsnummer lesen (Vor dem Druckvorgang)	274
9.1.3. Daten in einen Datenblock des RFID – Etiketts schreiben	275
9.2. Generierung Seiten-ID-Nummer, Abfrage RFID Daten nach dem Drucken	276
9.3. Syntax der RFID Daten (Statusmeldungen)	278

Kapitel	Seite
10. Funktionsgruppe NETZWERK	281
10.1. Jobanfang	282
10.2. Jobende	283
10.3. Jobindex	284
11. Funktionsgruppe CONTROLLER	289
11.1. Freigabeschlüssel (Softkey)	289
11.2. Einschaltwerte (Power-On Werte) festlegen	290
11.3. Verwendung von vordefinierten Konfigurationssätzen (User Configs)	291
11.3.1. Alphanumerische Zuweisungen (KonfigSatzNamen oder MAC-Adresse ändern)	292
11.4. Echtzeituhr (Real-Time Clock RTC)	294
11.4.1. Interne Variablen drucken (Seriennummer, Uhr)	295
11.5. Datei-Management Befehl	297
12. SPEZIAL - Befehle und Statusabfragen	299
12.1. Spezial - Befehle	299
12.2. Statusabfragen	300
12.2.1. Konfiguration des Status Kanals	300
12.2.2. Befehle für die Statusabfrage	301
13. DRUCKWERK – Befehle (PRINT ENGINE)	303
13.1. Druckgeschwindigkeit einstellen	303
13.2. Druckdichte (Schwärzung, Kontrast) einstellen	303
13.3. Material und Synchronisation einstellen (Thermo Mode)	304

Kapitel	Seite
14. Funktionsgruppe EIN- und AUSGABESTEuerung	305
14.1. SPS – Control (Option)	305
14.1.1. Daten an Ausgabe-Port senden (I/O-Mode 01)	306
14.1.2. Warten, bis Druckjob vollständig gedruckt ist (I/O-Mode 02)	307
14.1.3. Warten auf externe Signale (I/O-Mode 03)	308
14.1.4. Ausgabe-Port für kurze Zeit ändern (I/O-Mode 04)	309
15. ANHANG I Code - Übersetzungstabelle	311
16. ANHANG II Rasterarten	315
17. ANHANG III Befehlsübersicht	317
18. Index	323

1. Einführung

Die Seitenbeschreibungssprache IDOL (Information, Description and Organization Language) wurde speziell für MICROPLEX-Drucksysteme entwickelt. Diese Sprache ist durchgängig für alle Drucksysteme mit MPC-Controller - die Funktionsweise ist in allen Leistungs- und Geschwindigkeitsbereichen gleich. Dies ist eine Komponente für die Kompatibilität der SOLID-, RAPID- und LOGIJET-Systeme untereinander.

Mit IDOL sind die MICROPLEX-Drucksysteme sehr flexibel und lassen sich leicht an bestehende Applikationen anpassen. Die ganzen Möglichkeiten, die Drucksysteme bieten, lassen sich nur mit IDOL in vollem Umfang nutzen.

IDOL hilft, den Datentransfer zum Drucker zu minimieren und so die Effizienz von Druckvorgängen zu steigern. Die kompakte Syntax der Doppel-Escape Sequenzen gefolgt von einer Kennung und ggf. Parametern reduziert die zu übertragene Daten auf ein Minimum. Somit werden die Schnittstellen und Leitungen/Netzwerke nur mit der notwendigen Datenübertragung belegt.

IDOL ist eine lebende Sprache. Neue Funktionalitäten der Seitenbeschreibungssprache IDOL resultieren aus den Anforderungen der Kunden, die MICROPLEX-Drucker in industriellen Applikationen nutzen. Die Entwicklung neuer Befehle kann relativ schnell und unkompliziert erfolgen, da diese Seitenbeschreibungssprache in Deutschland entwickelt wurde und auch hier gepflegt wird.

Der Transparentcode ermöglicht es, die Escape-Sequenzen durch druckbare ASCII-Zeichen zu ersetzen. Dies ist nützlich, da es oft sehr schwer ist, Escape-Sequenzen per Tastatur zu erzeugen. Der Transparentcode hingegen kann von jeder Rechenanlage gesendet werden.

IDOL ist, um einen besseren Befehlsüberblick zu verschaffen, in verschiedene Funktionsgruppen strukturiert:

Job: Befehle zur Steuerung von Druckjobs, wie z.B. Anwahl der Kassette, Einschalten des Duplex-Druckmodus, Nummerierung von Seiten.

- Font:** Befehle zum Kontrollieren des Textdrucks, wie z.B. Auswahl der Zeichensätze, Setzen der Ränder, Wahl der Schreibrichtung, Festlegung des Zeilenabstandes.
- GKS:** Grafisches Kern System; es bietet Befehle zum Generieren von Vektorgrafiken, wie Linien, Flächen, Raster, Kreise etc.
- Barcode:** Mit den Barcodebefehlen lassen sich alle gängigen industriell genutzten Barcodes erzeugen (z.B. EAN 8, Code 128, PDF 417, Aztec und QR-Code). Mit Hilfe von IDOL kann der Anwender die Barcodes durch Eingabe der zugehörigen Parameter erzeugen, anstatt sie als Bild einzuscannen. Es ergibt sich ein vergleichsweise kleinerer Druckdatenstrom, daraus resultiert ein schnellerer Druckprozeß.
- Bitmap:** Funktion zum Übertragen von Bildinformationen (Pixeldaten) mit größter Effizienz und Geschwindigkeit.
- Macro:** Befehle zum Erzeugen und Verwalten von Macros, wie elektronischen Formularen, Durchschlagfunktion etc.
- Download:** Befehle zum Erzeugen von eigenen druckbaren Schriftzeichen, wie Firmenlogos, Unterschriften, Sonderschriften etc.
- SEM:** Das SEM (Simultaneous Emulation Management) ermöglicht, daß neben IDOL gleichzeitig mehrere Emulationen gültig sind. In den Anwendungen können so Befehle unterschiedlicher Emulationen gemischt werden. In den Datenstrom einer bestehenden Anwendung können IDOL-Befehle eingefügt werden, um z.B. eine Mailbox anzusteuern, während gerade Formularvordrucke mittels Prescribe erstellt werden. Da auch mehrere Emulationen gleichzeitig aktiv unterstützt werden, ist es ohne irgendeine Umkonfiguration des MICROPLEX-Druckers möglich, verschiedene Anwendungen mit unterschiedlichen Ausgabesprachen (z.B. HP PCL, Prescribe, etc.) zu betreiben.

1.1. IDOL - Philosophie

IDOL basiert auf einer Struktur von intelligenten Funktionen, die alle durch Doppel-Escape Sequenzen (<ESC><ESC>) eingeleitet werden. In diese Funktionen ist eine Vielzahl von sogenannten Emulationen eingebettet.

Eine Emulation ist das Nachahmen eines Druckers (einer Druckersprache) anderer Hersteller.

Eine Liste der Standard-Emulationen (MICROPLEX IDOL, SOLID - Standard, HP LaserJet (PCL 5e) etc.) sowie eine Liste der aktuellen optionalen Emulationen finden Sie in Abschnitt 2.2 Auswahl der Emulation.

Hinweise:

Die Reihe der zur Verfügung stehenden Emulation wird stetig erweitert.

Wie bereits beschrieben ermöglicht SEM (Simultaneous Emulation Management), daß gleichzeitig mehrere Emulationen gültig sind. In den Anwendungen können Befehle unterschiedlicher Emulationen gemischt werden. Verschiedene Schnittstellen können über SIA (Simultaneous Interface Administration) und SEM mit unterschiedlichen Emulationen bedient werden, wobei IDOL immer aktiv bleibt. Der MICROPLEX-Drucker erkennt dann automatisch, welcher Seitenbeschreibungssprache der jeweilige Befehl angehört. So sind zum Beispiel zur gleichen Zeit Befehle der Sprachen HP PCL 5e, Prescribe, IPDS und IDOL erlaubt.

1.2. Funktionalität

Im Folgenden ein kurzer Überblick über die wichtigsten IDOL - Funktionen:

Modifizierbare Zeichensätze

Die in EPROMs vorrätigen oder via Schnittstelle in den RAM - Speicher ladbaren Zeichensätze können über IDOL-Befehle modifiziert werden.

Folgende Modifikationen sind möglich:

- Skalieren von Fonts (Höhe und Breite)
- Definieren von Zeichenschrittweite und Zeilenabstand
- 4 Orientierungen: 0°, 90°, 180°, 270°
- Fett- und Inversdruck
- Unterstreichung
- Sperrschrift
- Hoch- / Tiefstellung

GKS Vektorgrafik

Der Funktionsumfang der Vektorgrafik ist an die Forderungen des Grafischen-Kern-Systems (GKS) für Drucker und Plotter angelehnt. Der IDOL - Controller empfängt relativ kurze Grafikbefehle vom HOST und setzt diese erst im Drucker in ein Bildpunkt-raster um. Solche Bilder können beinhalten:

- Linienzüge mit unterschiedlichen Linienbreiten und -arten
- Flächen
- Kreise, Kreisflächen, -segmente
- Balken
- Raster
- Ellipsen, Ellipsenflächen

Mit Hilfe der MACRO - Funktion können komplexe Grafikgebilde (zum Beispiel Formulare etc.) als Einheit zusammengefaßt und komplett im Drucker gespeichert werden.

Elektronischer Formulardruck

Die IDOL-Sprache ermöglicht die Erstellung von MACROs für elektronische Formulare. Das MACRO wird über die DOWNLOAD - Funktion in den RAM – Speicher des Druckers geladen.

MACROs können zum Beispiel enthalten:

- Grafiken wie z.B. Ellipsen und Ellipsenflächen
- Briefbögen
- Firmen - Embleme
- automatische Durchschläge (auch mit wechselnden Formularen)

Dies bewirkt, daß - solange ein MACRO im RAM - Speicher geladen ist - dessen Inhalt im Drucker präsent bleibt und nur die jeweiligen variablen Daten vom HOST übertragen werden müssen. Die festen Daten (z.B. das MACRO) werden automatisch jeder Seite hinzugefügt. Es können bis zu 99 verschiedene Formulare gleichzeitig im Drucker gespeichert sein.

All Point Addressability (APA)

Bei einer zu druckenden DIN A4 Seite sind alle 8,6 Millionen Bildpunkte (bei einer Auflösung von 300 dpi) einzeln adressierbar, so daß jede Information bildpunktgenau - absolut und/oder relativ - positioniert werden kann.

Sortier - Funktion (JOB)

Das "elektronische" Sortieren erlaubt das mehrfache Drucken ganzer Druckjobs. Bei vielen anderen Druckern ist zwar auch eine Multicopy - Funktion implementiert, diese bewirkt jedoch nur den Mehrfachdruck einzelner Seiten.

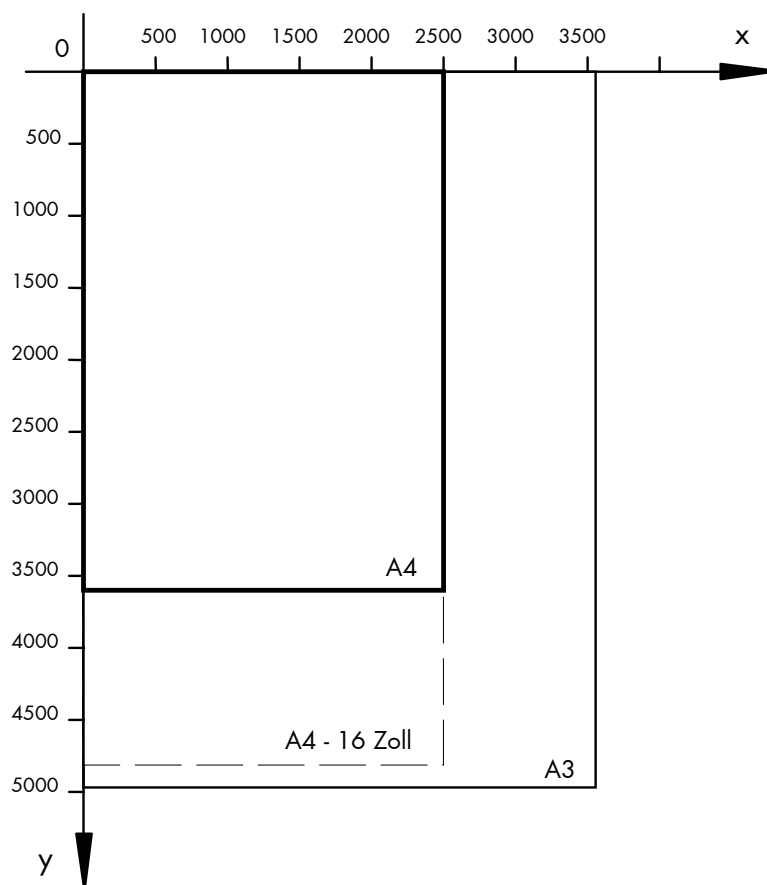
Bei MICROPLEX - Druckern können ganze Jobs (bestehend aus mehreren Seiten, wie z.B. dieses IDOL-Handbuch) in den RAM-Speicher des Druckers geladen werden, um dann komplett und seitengerecht mehrfach ausgedruckt zu werden. Die Daten werden nur einmal an den Drucker gesandt, das manuelle Sortieren entfällt.

1.3. Metrik

Um einen Bildspeicher mit Informationen beschreiben zu können, wird zur Positionierung ein Maßsystem benötigt. Bei IDOL ist dieses Maßsystem genau an die ansteuerbaren Bildpunkte (Pixel oder Dots) des Druckwerkes angelehnt. Die Standard-Auflösung eines MICROPLEX - Druckers beträgt 300 x 300 Punkte pro Zoll (dpi).

Zum Markieren von Positionen wird, ähnlich wie bei einem Bildschirm, ein sogenannter Cursor benutzt, der aber nicht sichtbar ist. Der Cursor wird in einem Koordinatensystem positioniert.

Koordinatensystem:



Für die Auflösung von 300 dpi (dots per inch) errechnen sich folgende Maximalkoordinaten:

Format	X	Y
DIN A4	2480 x 3508 Punkte	
DIN A4 (16 Zoll) Länge	2480 x 4800 Punkte	
DIN A3	3508 x 4960 Punkte	

Der Nullpunkt des Koordinatensystems befindet sich immer in der linken oberen Papierecke. Dieses Koordinatensystem ist fest. Alle Positionierungsangaben beziehen sich auf dieses Koordinatensystem.

Innerhalb dieses Koordinatensystems werden zwei voneinander unabhängige Cursor positioniert:

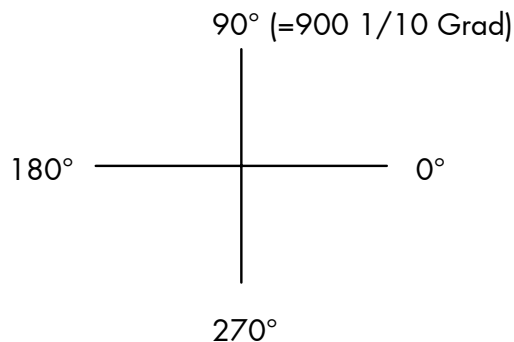
1. Textcursor
2. Grafikcursor

Der Textcursor wird bei IDOL für das Positionieren von Text und Barcodes verwendet. Der Grafikcursor wird für die GKS - Befehle verwendet.

Beachte: Die Relativpositionierung von Text (mittels der IDOL-Befehle `<ESC><ESC>S`, `<ESC><ESC>T`) ist abhängig von der Schreibrichtung!

Abhängig von den Eigenschaften des jeweiligen Druckwerks ist die bedruckbare Fläche ggfs. an den Papierkanten eingeschränkt (physikalische Ränder).

Winkel im Grafik-Koordinatensystem werden in 1/10 Grad angegeben:



1.4. Tools, Software

Um die gesamte Leistungsfähigkeit der MICROPLEX-Drucker und den Komfort der Seitenbeschreibungssprache IDOL auf einfache und schnelle Weise nutzen zu können, wurden von MICROPLEX PC-Softwareprogramme entwickelt. Mit diesen Programmen ist es möglich, auch ohne spezielle Programmierkenntnisse die IDOL-Funktionalität zu nutzen.

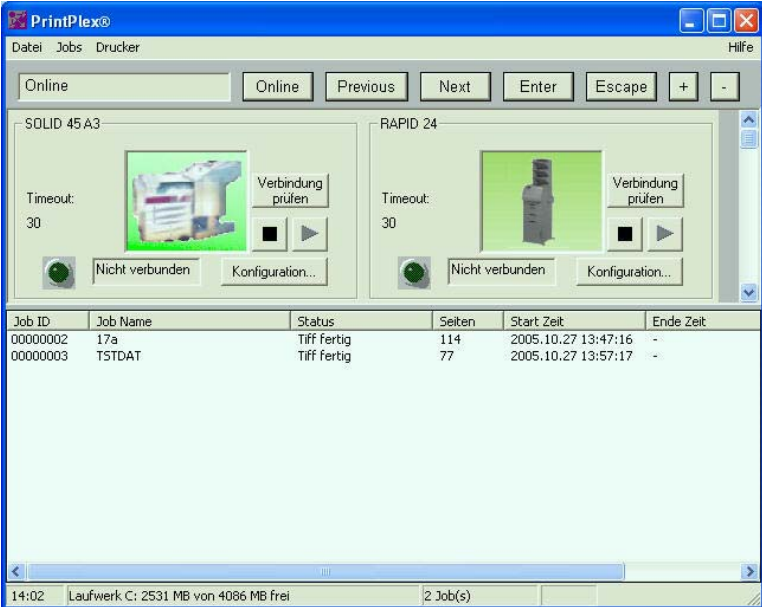
Tool:

Anwendungsgebiete:

MICROPLEX® MP-CONFIG	Druckerkonfiguration und Administration Fonts Upload, Firmware Update, RFH progr. etc.

Tool:

Anwendungsgebiete:

<p>MICROPLEX® PRINTPLEX</p>	<p>Print Management Überwachtes Spooling, Intelligentes Clustering, On-Line Archivierung etc.</p>																		
 <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>Job ID</th> <th>Job Name</th> <th>Status</th> <th>Seiten</th> <th>Start Zeit</th> <th>Ende Zeit</th> </tr> </thead> <tbody> <tr> <td>00000002</td> <td>17a</td> <td>Tiff fertig</td> <td>114</td> <td>2005.10.27 13:47:16</td> <td>-</td> </tr> <tr> <td>00000003</td> <td>TSTDAT</td> <td>Tiff fertig</td> <td>77</td> <td>2005.10.27 13:57:17</td> <td>-</td> </tr> </tbody> </table> <p style="font-size: small; margin-top: 5px;">14:02 Laufwerk C: 2531 MB von 4086 MB frei 2 Job(s)</p>		Job ID	Job Name	Status	Seiten	Start Zeit	Ende Zeit	00000002	17a	Tiff fertig	114	2005.10.27 13:47:16	-	00000003	TSTDAT	Tiff fertig	77	2005.10.27 13:57:17	-
Job ID	Job Name	Status	Seiten	Start Zeit	Ende Zeit														
00000002	17a	Tiff fertig	114	2005.10.27 13:47:16	-														
00000003	TSTDAT	Tiff fertig	77	2005.10.27 13:57:17	-														

Transparentcode als Alternative zur Escape-Sequenz:

Ein IDOL - Befehl wird durch zwei Escape - Zeichen (ESC: 1B hexadezimal, 27 dezimal) eingeleitet, um ihn von "normalen" Druckdaten zu unterscheiden.

Diese Escape - Zeichen können durch den sogenannten Transparentcode ersetzt werden. Mit diesem Transparentcode können die Befehle der Seitenbeschreibungssprache IDOL mittels **druckbarer** Zeichen eingeleitet werden. Solche Zeichen lassen sich leichter über die Tastatur eingeben. Die Voreinstellung des Transparentcodes ist 2625. Dies sind die ASCII - Zeichencodes (hexadezimal) für die Zeichen & % (vergleiche Anhang I : Code – Übersetzungstabelle).

Durch diesen Ersatz der Escape - Zeichen bestehen die IDOL - Befehle ausschließlich aus ASCII - Zeichen und können daher genau wie Text bearbeitet und verteilt werden. Sobald der IDOL - Interpreter des Druckers Transparentcodes (oder Escape - Sequenzen) findet, werden die jeweils folgenden Daten als IDOL - Befehl interpretiert und die entsprechende Funktion ausgeführt.

Angabe von Parametern:

Nach der Doppel-Escape Sequenz folgt der Funktionsname des IDOL - Befehls sowie die für diesen Befehl erforderlichen Parameter.

Die Eingabe von Parametern (z.B. n, b, x, y etc.) erfolgt - wenn nicht ausdrücklich anders beschrieben - als Dezimalzahl (1-, 2-, 3- oder 4stellig).

Ein 'h' symbolisiert dagegen einen hexadezimalen Parameter.

Zwischen den einzelnen Parametern eines IDOL - Befehls können zur besseren Übersicht sowohl Leerzeichen als auch die Zeichen "CR" und "LF" stehen.

Beispiel: Absolute Positionierung des Textcursors zu den Koordinaten
x = 245, y = 95

Syntax: <ESC><ESC>D <xxxx> <yyyy>

Befehlszeile: &%%D02450095

oder alternativ

Befehlszeile: &%%D 245 95

Beachte: Nur Parameter dürfen durch Leerzeichen getrennt (separiert) werden, zwischen den Zeichen der Kommandosequenz (z.B. <ESC><ESC>D) darf kein anderes Zeichen stehen.

Wird auf Leerzeichen zur Trennung der Parameter verzichtet, so muß jeder Parameter durch Vorstellen von Nullen auf die vorgegebene Anzahl von Stellen aufgefüllt werden (Beispiele: die x-Koordinaten 5, 10, 100 sind als x = 0005, 0010, 0100 einzugeben).

Hinweis: Vermeiden Sie die **zu häufige Verwendung von Zeilenvorschüben (Linefeed LF) und/oder Wagenrückläufen (Carriage Return CR)**. „CR“ und „LF“ werden gerne verwendet, um Befehle optisch zu trennen und um Parameter besser erkennbar zu machen (Ziel: Verbesserung der Lesbarkeit). Beachten Sie, dass jeder CR und jeder LF die Cursor-Position verändert. Sobald der Cursor die Seitengrenze überschreitet, kann ein automatischer FormFeed ausgelöst werden.

Fehlerbehandlung:

Ist die Syntax eines IDOL - Befehls nicht korrekt, wird die Interpretation abgebrochen. Die nicht interpretierten ASCII - Zeichen des fehlerhaften Befehls werden gedruckt.

Darstellung der IDOL - Befehle in diesem Handbuch:

In den folgenden Beschreibungen werden die Zeichen für die Funktionsnamen (incl. Doppel-Escape Sequenzen) als ASCII - Zeichen (vgl. Anhang I: Code - Übersetzungstabelle) und dahinter (zur eindeutigen Identifizierung) die ASCII - Zeichennummern in dezimaler Schreibweise abgebildet.

Beispiele:

Syntax: <ESC><ESC>g <n>	(27)(27)(103)...
--------------------------------	-------------------------

Syntax: <ESC><ESC>10	(27)(27)(49)(48)...
-----------------------------	----------------------------

2. Funktionsgruppe JOB

Diese Gruppe enthält im wesentlichen Befehle, die sich auf den kompletten Druckjob auswirken - insbesondere im Zusammenhang mit den Eigenschaften des Druckwerks.

Somit ist die Ausführung bestimmter Befehle nur in entsprechend ausgestatteten Druckertypen möglich (z.B. Anwahl eines Ablagefaches in einer Mailbox).

Sinnvollerweise sollten die Befehle entweder nach dem Einschalten, am Job- und/oder Seitenanfang eingegeben werden.

2.1. Transparentcode ändern (temporär)



Jeder IDOL - Befehl wird durch zwei Escape - Zeichen eingeleitet. Diese Escape - Zeichen können durch den sogenannten Transparentcode ersetzt werden. Mit diesem Transparentcode können die Befehle der Seitenbeschreibungssprache IDOL mittels **druckbarer** Zeichen eingeleitet werden. Die Voreinstellung des Transparentcodes ist 2625. Dies sind die ASCII - Zeichencodes (hexadezimal) für die Zeichen &% (vergleiche Anhang I: Code - Übersetzungstabelle).

Funktion: Mit diesem Befehl kann ein neuer Transparentcode gewählt werden. Diese Änderung ist temporär, d.h. sie ist nur so lange wirksam, bis der Drucker ausgeschaltet wird.

Syntax: <ESC><ESC>tt <hhhh> (27)(27)(116)(116)...
--

Parameter: <hhhh> : neuer Transparentcode, vierstellige Hex.-Zahl

Beispiel: Der Transparentcode soll in die ASCII-Zeichen && geändert werden:

```
&%&%tt 2626
```

Sobald der IDOL - Interpreter des Druckers diesen neuen Transparentcode (oder Escape - Sequenzen) findet, werden die jeweils folgenden Daten als IDOL - Befehl interpretiert und die entsprechende Funktion ausgeführt.

2.2. Auswahl der Emulation

Eine Emulation ist die Nachbildung der Befehlssprache anderer Drucker (Drucker anderer Hersteller). Hauptziel ist die Generierung von Ausdrucken mit möglichst ähnlichem Aussehen. Eine vollständige Nachbildung aller Druckerbefehle ist nicht möglich.

Einschränkungen ergeben sich z.B. durch

- Unterschiede in der Drucktechnologie (wie physikalische Auflösung, Ränder, Papierformate, Bedienfelder, Papierkassetten und Ablagefächer etc.)
- Nutzung von speziellen Fähigkeiten des Original-Controllers (Image Controller)

Einige Emulationen benötigen zusätzliche Fonts zur Optimierung der Ergebnisse.

Funktion: Mit diesen Befehlen wird eine der Emulationen ausgewählt und aktiviert.

Syntax:	<code><ESC><ESC>I <n></code>	<code>(27)(27)(108)...</code>
	<code><ESC><ESC>em <nn></code>	<code>(27)(27)(101)(109)...</code>

Parameter:

`<n>` = 1 bis 9 = einstellig
`<nn>` = 01 bis 99 = zweistellig

Hinweis: Da die Anzahl der Emulationen stetig anwächst und in Zukunft - dem Marktbedarf folgend - weitere hinzukommen werden, wurde der Befehl (em) mit zweistelliger Parameterangabe eingeführt.

`<nn> = 00` : schaltet wieder auf die Einschalt emulation zurück

`<nn> = 01` : MICROPLEX IDOL, SOLID - Standard
(Olympia ESW 3000)

- <nn> = 02 : Diablo 630
- <nn> = 03 : IBM ProPrinter
- <nn> = 04 : HP LaserJet (HP-PCL Level 5)
Es werden automatisch HP-kompatible Ränder
eingestellt.
- <nn> = 05 : AGFA Reno
- <nn> = 06 : PRINTRONIX IGP/PGL
- <nn> = 07 : Epson LQ Serie
- <nn> = 08 : DELPHAX S 3000 Lineprinter (auszugsweise)
- <nn> = 09 : HP 7475A (HPGL)
Bei dieser Plotter-Emulation ist normaler Textdruck
nicht möglich. Gegebenenfalls ist in eine andere
Emulation zurückzuschalten.
- <nn> = 10 : DEC LN03 + (incl. TEKTRONIX 4010/4014)
- <nn> = 11 : TEKTRONIX 4010/4014 Grafik - Terminal - Emulation
- <nn> = 12 : reserviert
- <nn> = 13 : TIFF (CCITT Fax 4)
Bei dieser Emulation ist kein Textbetrieb möglich.
Gegebenenfalls ist in eine andere Emulation
zurückzuschalten.
- <nn> = 14 : QMS (Magnum) Code V
- <nn> = 15 : Tally MT 6xx
- <nn> = 16 : Bull MP6090
- <nn> = 17 : XEROX XES

- <nn> = 18 : Datamax (Fargo FGL – Thermodrucker)
- <nn> = 19 : TEC (Thermodrucker)
- <nn> = 20 : HP DesignJet (HPGL-2)
- <nn> = 21 : Etimark MP-1220 (Thermodrucker)
- <nn> = 25 : Zebra ZPL II
- <nn> = 26 : EPL 2
- <nn> = 27 : XML
- <nn> = 28 : UBI
- <nn> = 29 : Label Point
- <nn> = 30 : IER Command
- <nn> = 31 : Kodescript+
- <nn> = 32 : CUPS Raster
- <nn> = 33 : EZPL
- <nn> = 34 : IPDS (via PPD/PPR Protocol)
Hinweis: Nach dem Erwerb und der Aktivierung dieser Option erscheint IPDS unter der Überschrift "Schnittstellen" auf dem Statusblatt des Druckers
- <nn> = 35 : AEA

Beachte:

Die erwähnten Markennamen sind Warenzeichen bzw. eingetragene Warenzeichen der genannten Unternehmen.

Hinweise: Die Emulationen IDOL und Prescribe sind immer gültig und müssen nicht besonders angewählt werden (Prescribe ist nur verfügbar, falls freigeschaltet).

Die Standardemulationen und die aktive Emulation des Druckers sind dem Statusblatt zu entnehmen.

Beispiel: Der Drucker soll in die Emulation HP-LaserJet PCL 5e geschaltet werden.

&%%em 04

2.3. Druckauflösung wählen

rs

Funktion: Mit diesem Befehl wird festgelegt, mit welcher Auflösung der Drucker standardmäßig druckt.

Es wird automatisch ein Form Feed ausgeführt und der Drucker auf die neue Auflösung geschaltet.

Der Befehl wird ignoriert, wenn der Drucker die gewählte Auflösung nicht unterstützt.

Syntax: <ESC><ESC> <i>rs</i> <nnnn> (27)(27)(114)(115) ...
--

Parameter: <nnnn> = 300, 400, 600: Kennzahl für die neue Druckauflösung des Drucker in dots per inch (dpi)

<nnnn> =	0300	:	Auflösung 300 dpi (nach Form Feed)
<nnnn> =	0400	:	Auflösung 400 dpi (nach Form Feed)
<nnnn> =	0600	:	Auflösung 600 dpi (nach Form Feed)

Beispiel: Die Auflösung des Druckers soll auf 600 dpi umgeschaltet werden:

&%&%rs 0600

Hinweis: Bei manchen Druckern (z.B. bei Thermodruckern) ist die Druckauflösung in Dots per mm definiert (metrisches System statt Zollbasiert). Wegen der erforderlichen druckerinternen Umrechnungen (300/304 und umgekehrt) kann es zu einer geringfügigen Verschlechterung der Druckqualität kommen.

2.4. Umschalten des Papierformats

Funktion: Mit diesem Befehl wird das Papierformat gewählt, das für nachfolgende Ausdrücke verwendet wird.

Syntax:	<code><ESC><ESC>g <n></code>	(27)(27)(103)...
	<code><ESC><ESC>g 7 <nn></code>	(27)(27)(103)(55)...
	<code><ESC><ESC>g 8 <bbbb> <llll></code>	(27)(27)(103)(56)...

Parameter:	<code><n> = 1</code>	: DIN A1 Format	(594mm x 841mm)
	<code><n> = 2</code>	: DIN A2 Format	(420mm x 594mm)
	<code><n> = 3</code>	: DIN A3 Format	(297mm x 420mm)
	<code><n> = 4</code>	: DIN A4 Format	(210mm x 297mm)
	<code><n> = 5</code>	: DIN A5 Format	(148mm x 210mm)
	<code><n> = 6</code>	: DIN A6 Format	(105mm x 148mm)

Weitere Papierformate:

(Befehl `<ESC><ESC>g7 <nn>`)

<code><nn> = 00</code>	: Ansi E	(34" x 44")
<code><nn> = 01</code>	: Ansi D	(22" x 34")
<code><nn> = 02</code>	: Ansi C	(17" x 22")
<code><nn> = 03</code>	: Ledger	(11" x 17")
<code><nn> = 04</code>	: Letter Format	(8,5" x 11")
<code><nn> = 05</code>	: Half-Letter	(5,5" x 8,5")
<code><nn> = 07</code>	: Legal	(8,5" x 14")
<code><nn> = 08</code>	: Executive	(7,25" x 10,5")
<code><nn> = 10</code>	: Arch B	(12" x 18")
<code><nn> = 11</code>	: Arch C	(18" x 24")
<code><nn> = 12</code>	: Arch D	(24" x 36")
<code><nn> = 13</code>	: Arch E	(36" x 48")
<code><nn> = 14</code>	: Arch F	(48" x 72")

<nn> = 17	: DIN B5	
<nn> = 18	: DIN B4	
<nn> = 19	: DIN B3	
<nn> = 20	: DIN B2	
<nn> = 21	: DIN B1	
<nn> = 25	: Ansi F	(44" x 68")
<nn> = 26	: DIN A0 Format	
<nn> = 53	: DIN A3R	
<nn> = 54	: DIN A4R	
<nn> = 55	: DIN A5R	
<nn> = 56	: DIN A6R	
<nn> = 69	: LedgerR	
<nn> = 70	: LetterR	
<nn> = 71	: HalfLetterR	
<nn> = 73	: LegalR	
<nn> = 83	: DIN B5R	
<nn> = 84	: DIN B4R	

Nicht-Standard Papierformate anwählen (Sonderformate):

(Befehl <ESC><ESC>g8 <bbbb> <llll>)

<bbbb> : Papierbreite 4-stellig in dots (300 dpi)

<llll> : Papierlänge 4-stellig in dots (300 dpi)

Hinweise:

Die Länge wird auf den nächstgrößeren Wert gesetzt, den die Druckeinheit verarbeiten kann.

Generell werden die Werte für Länge und Breite begrenzt durch die mechanischen Eigenschaften der Druckeinheit (zu große Werte werden reduziert auf den zulässigen Maximalwert – zu kleine Werte werden auf den zulässigen Minimalwert aufgerundet).

Beachte: Breite und Länge (in Transportrichtung) beschreiben immer die physikalischen Maße. Diese Formatangaben sind unabhängig von der Druckrichtung.

Hinweis: Ist die Option '**Papieranforderung**' (Formatprüfung) eingeschaltet (Konfiguration: CONFIG 32: Bit 2 der EEPROM-Sequenz auf 1 gesetzt), so geht ein Seitendrucker nach einer Formatumstellung OFF LINE, falls das neue Papierformat nicht im Drucker ist. Endlosdrucker gehen dann nach jeder Formatumstellung, nach jedem Papierende und jedem Einschalten OFF LINE.
Im Display erscheint dann die Panel-Anzeige [Papier einlegen] plus Formatangabe. Nachdem das entsprechende Papier eingelegt wurde, ist der Drucker wieder ON LINE zu schalten.

Die Grundeinstellung für das Papierformat ist <n> = 4.

Die eingestellten Ränder werden beim Umschalten des Formates automatisch transformiert.

Beachte: Die Anwahl des Papierformats hat Priorität vor der Anwahl des Papiereinzugs (siehe Befehl <ESC><ESC>G <n>).

Der Befehl wird ignoriert, wenn der Drucker nicht in dem angewählten Format drucken kann.

Beispiel: Anwahl des A3 - Formates

&%&%g 3

2.5. Anwählen des Papiereinzugs und Feeder Mapping



Funktion: Mit diesem Befehl wird der Papiereinzug festgelegt, aus dem das Papier für den Druck entnommen wird.

Syntax:	<code><ESC><ESC>G <n></code>	(27)(27)(71)...
----------------	--	------------------------

Parameter:	<code><n> = 0</code>	: Sondereinzug (z.B. Handeinlage)
	<code><n> = 1</code>	: 1. Einzug
	<code><n> = 2</code>	: 2. Einzug
	<code><n> = 3</code>	: 3. Einzug
	<code><n> = 4</code>	: teilweiser Auto-Wechsel-Modus (1-2 oder andere konfigurierbar)
	<code><n> = 5</code>	: Auto-Wechsel-Modus (alle Einzüge)
	<code><n> = 6</code>	: 4. Einzug (optional)
	<code><n> = 7</code>	: 5. Einzug (optional)

Die Grundeinstellung ist konfigurierbar.

Wenn der angewählte Einzug leer ist, schaltet der Drucker in den OFFLINE-Modus.

Ausnahme: `<n> = 4` und `<n> = 5`

Hier wird automatisch zwischen den Einzügen gewechselt, bis kein Einzug mehr Papier (von diesem Format) enthält.
Eine leere Kassette kann während des Betriebs nachgefüllt werden.

Beim Auto-Wechsel-Modus 1-2 werden (z.B.) die Kassetten 1 und 2 zu einer logischen Kassette zusammengefaßt. In Abhängigkeit vom Drucker und dessen Ausstattung sind auch andere Kombinationen möglich. Ist eine Kassette leer, wird auf die nächste Kassette umgeschaltet.

Der Drucker schaltet erst OFFLINE, wenn beide Kassetten leer sind.

Beim Auto-Wechsel-Modus 1-2-3 werden alle Kassetten des Druckers (außer der Handeinlage) zu einer logischen Kassette zusammengefaßt. Ist eine Kassette leer, wird auf die nächste Kassette umgeschaltet. Der Drucker schaltet erst OFFLINE, wenn alle Kassetten leer sind.

Beachte: Die Anwahl des Papierformats hat Priorität vor der Anwahl des Papiereinzugs. Dies bedeutet: wenn DIN A4 als Papierformat gewählt wurde und der Papiereinzug aus dem 1. Einzug, dann wird die Wahl des Einzugs ignoriert, wenn der erste Einzug z.B. DIN A3 Papier enthält.

Beispiel: Einzug aus der ersten Kassette

&%%G 1

Feederzuordnung (Feeder-Mapping):

Die Feederanwahl durch Befehle geschieht über eine sogenannte Look-Up-Tabelle. Mit Hilfe dieser Tabelle erfolgt die Zuordnung der „physikalischen“ Einzugfach-Nummern des Druckers (Engine) zu den „logischen“ Papiereinzugnummern (Parameter <n> des IDOL-Befehls „Anwählen des Papiereinzugs“).

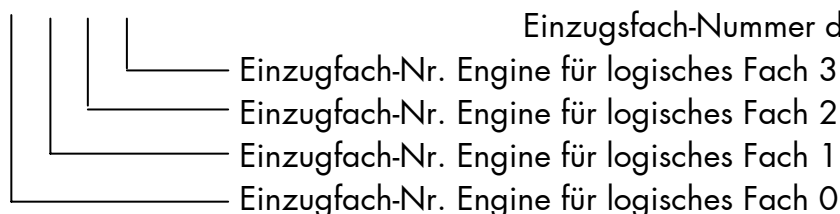
Diese Feederzuordnung (Feeder-Mapping) ist im EEPROM in CONFIG 24 und CONFIG 25 gespeichert.

Logische **Fächer 0...3:**

CONFIG 24 (Schematische Darstellung des EEPROM – Wortes)

X X X X

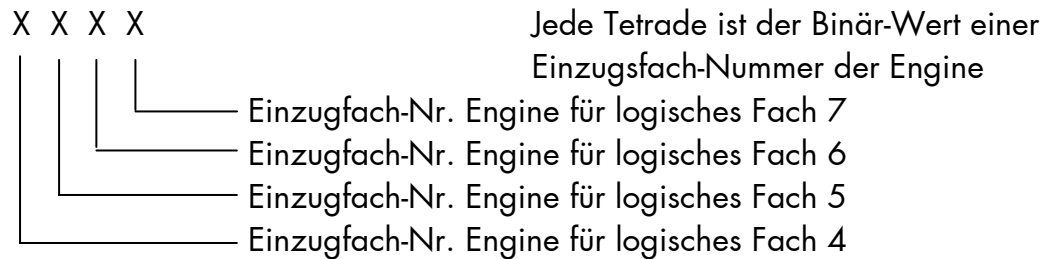
Jede Tetrade ist der Binär-Wert einer Einzugfach-Nummer der Engine



Diese Konfiguration kann z.B. mit Hilfe des IDOL – Befehls <ESC><ESC>ee geändert werden.

Logische **Fächer 4...7:**

CONFIG 25 (Schematische Darstellung des EEPROM – Wortes)



Diese Konfiguration kann z.B. mit Hilfe des IDOL – Befehls <ESC><ESC>ee geändert werden.

Im Falle der **Fabrik - Werte** (Default) entsprechen die „physikalischen“ Einzugsfachnummern des Druckers (Engine) den „logischen“ Papiereinzugsnummern.

Beispiel: In diesem Beispiel soll die Feederzuordnung von Papiereinzug 1 und 2 getauscht werden, ansonsten sollen die „physikalischen“ Einzugsfachnummern des Druckers (Engine) den „logischen“ Papiereinzugsnummern entsprechen.

Die folgende Darstellung zeigt die entsprechende Konfiguration für die logischen Papiereinzugsfächer 0 bis 3 (CONFIG 24):

Hex	0				2				1				3			
Binärwert	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1
Bit Nr.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

2.6. Anwählen des Papierablagefachs, Stacker Mapping und Job-Jogging



Funktion: Mit diesem Befehl wird das Ablagefach gewählt, in dem ein gedrucktes Blatt abgelegt wird.

Syntax:	<code><ESC><ESC>a <nn></code>	(27)(27)(97)...
----------------	---	------------------------

- Parameter:**
- `<nn> = 00` : Sonderablage
 - `<nn> = 01` : Ablage oben (z.B. face down)
 - `<nn> = 02` : Ablage unten (z.B. face up)
 - `<nn> = 03 ... 20` : Optionale Ablage (Sorter)
 - `<nn> = 81 ... 84` : Job-Jogging ohne Fachanwahl
 - `<nn> = 88 ... 89` : Face Up/Down-Steuerung (ohne Fachanwahl)
 - `<nn> = 90 ... 99` : Job-Jogging mit Fachanwahl

Die Grundeinstellung ist konfigurierbar.

Ist der Drucker mit einem **Sorter (Mailbox)** ausgestattet, haben die Fachanwahlen 03 bis 20 besondere Funktionen:

- `<nn> = 03` : Ablage in das zuletzt angewählte Sorterfach.
- `<nn> = 10` : Ablage in das zuletzt angewählte Sorterfach - 1.
(z.B. : zuletzt Fach 15 → Ablage in Fach 14)
Die Sorterfächer werden zyklisch verwendet.
...13, 12, 11, 20, 19, 18, ...
- `<nn> = 11` : Ablage in Sorterfach 1.
- `<nn> = 12` : Ablage in Sorterfach 2.
- ⋮
- `<nn> = 20` : Ablage in Sorterfach 10.

Job-Jogging ohne Fachanwahl:

<nn> = 81	: Toggle Position ("hin und her")
<nn> = 82	: vorne (Front Position)
<nn> = 83	: Mitte (Middle Position)
<nn> = 84	: hinten (Rear Position)

Face Up/Down-Steuerung ohne Fachanwahl:

<nn> = 88	: Face Down
<nn> = 89	: Face Up

Job-Jogging mit Fachanwahl:

<nn> = 90 bis 99	: Job-Jogging: Bei jedem Befehl folgt eine versetzte Ablage („Toggle Position“).
<nn> = 90	: aktuelles (bzw. letztes) Sorterfach
<nn> = 91	: Ablage Nr. 1
<nn> = 92	: Ablage Nr. 2
usw.	

Beispiel: Druck in die untere Ablage ausgeben.

&%%&%a 2

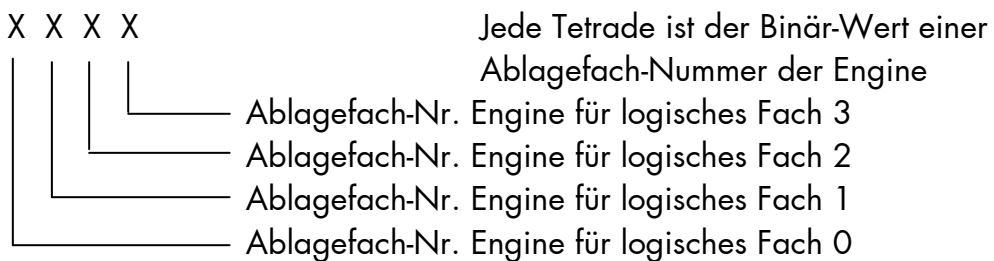
Stackerzuordnung (Stacker-Mapping):

Die Stackerzuordnung der „logischen“ Papierablagenummern 0 bis 7 (Parameter <nn> des IDOL-Befehls „Anwählen des Papierablagefachs“) ist konfigurierbar. Mit Hilfe einer sogenannten Look-Up-Tabelle können die „physikalischen“ Ablagefach-Nummern 0 bis 15 des Druckers (+ Sorter) neu zugeordnet werden.

Diese Stackerzuordnung (Stacker-Mapping) ist im EEPROM in CONFIG 40 und CONFIG 41 gespeichert. CONFIG 40: logische Fächer 0...3, CONFIG 41: logische Fächer 4...7.

Logische **Fächer 0...3:**

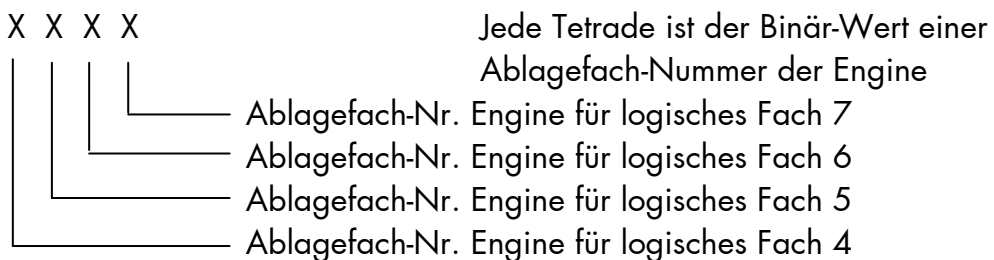
CONFIG 40 (Schematische Darstellung des EEPROM – Wortes)



Diese Konfiguration kann z.B. mit Hilfe des IDOL – Befehls <ESC><ESC>ee geändert werden.

Logische **Fächer 4...7:**

CONFIG 41 (Schematische Darstellung des EEPROM – Wortes)



Diese Konfiguration kann z.B. mit Hilfe des IDOL – Befehls <ESC><ESC>ee geändert werden.

Im Falle der **Fabrik - Werte** (Default) entsprechen die „physikalischen“ Ablagefachnummern des Druckers (Engine) den „logischen“ Papierablagefachnummern.

Beispiel: In diesem Beispiel soll die Stackerzuordnung von Papierablage 1 und 2 getauscht werden, ansonsten sollen die „physikalischen“ Ablagefachnummern des Druckers (Engine) den „logischen“ Papierablagefachnummern entsprechen.
 Die folgende Darstellung zeigt die entsprechende Konfiguration für die logischen Papierablagefächer 0 bis 3 (CONFIG 40):

Hex	0				2				1				3			
Binärwert	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1
Bit Nr.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

2.7. Mehrfachdruck einer Seite ohne Papierfach - Anwahl

ds

Funktion: Der Befehl wird zur Vervielfältigung einer Druckseite verwendet (Multicopy).
Bei diesem Befehl wird der aktuell aktive Papiereinzug verwendet.

Syntax: <ESC><ESC>ds <nnnn> (27)(27)(100)(115)...

Parameter: <nnnn> = 0001 bis 9999 : Duplikationsfaktor

Beachte: Der Bildspeicher wird während des Mehrfachdrucks nicht verändert (Seitenzähler).

Beispiel: Eine Seite soll 22mal gedruckt werden.

&%&%ds 22

2.8. Mehrfachdruck eines kompletten Druckjobs

h

Funktion: Wenn dieser Befehl am Anfang eines Druckjobs, der aus mehreren Seiten besteht, gesetzt wird, so wird der komplette Job seitengerecht <nn> - mal ausgedruckt.
Die Daten für den Druckjob werden einmal gesendet, aber <nn> mal gedruckt.

Syntax:	<pre><ESC><ESC>h <nn> <ESC><ESC>h</pre>	(27)(27)(104)...
----------------	---	-------------------------

Parameter: <nn> = 01 bis 99 :Anzahl der Vervielfältigungen

Der zu vervielfältigende Job wird mit den Befehlen
<ESC><ESC>h <nn> und <ESC><ESC>h "eingeklammert".

Beispiel: &%&%h22

```

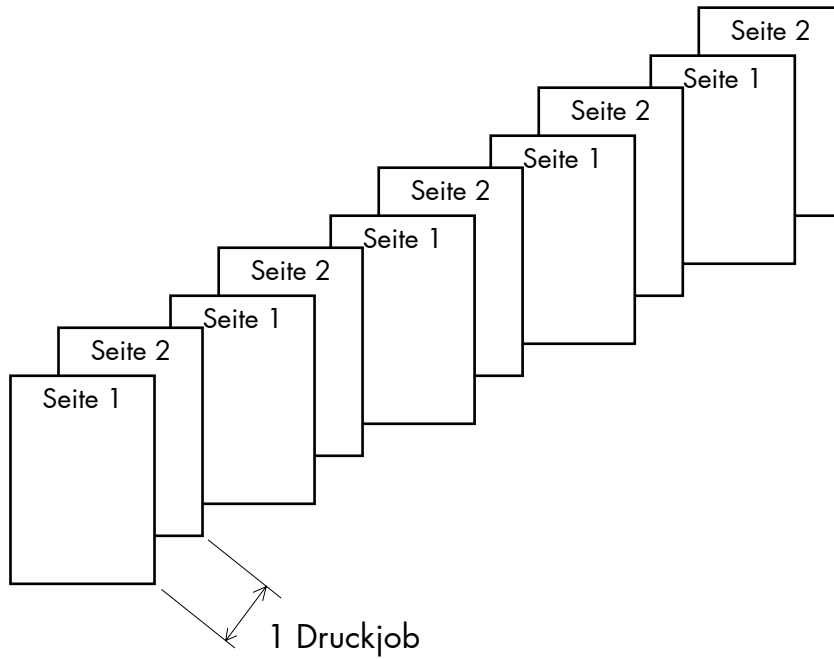
:
Zu vervielfältigender Druckjob
:
&%&%h
```

Die nachfolgenden Seiten werden dann wieder normal gedruckt.

Beachte: Der zu vervielfältigende Druckjob kann so groß wie der Eingangspuffer des Druckers sein.
Alle Befehle im Eingangspuffer werden <nn>-mal ausgeführt.

Beispiel: Ein Job, bestehend aus 2 Seiten, soll 5mal seitengerecht gedruckt werden.

```
&%&%h 05  
Seite1&%&%v  
Seite2&%&%v  
&%&%h
```



2.9. Durchschlag - Funktion

dg

Funktion: Mit diesem Befehl ist es möglich, von einer Druckseite bis zu 16 Durchschläge zu drucken, die mit jeweils einem anderen Macro gekoppelt werden können.

Syntax: <ESC><ESC>*dg* <nn> <uuuu₁> ... <uuuu_n>

(27)(27)(100)(103) ...

Parameter:

<nn> = 00 ... 16	:	Anzahl der Durchschläge (max. 16)
<nn> = 00	:	Durchschlagfunktion abschalten
<uuuu _i > = 0001 ... 9999	:	Nummer des MACROs, das für den Durchschlag Nummer „i“ ausgeführt wird
<uuuu _i > = 0000	:	kein Macro (Overlay) für diese Seite

Hinweise: Der Befehl wirkt ab dem Anfang der Seite, auf der er geschickt wird (ab Firmware Version 3.3T-01.02.1996; bei älteren Versionen wirkte der Befehl ab der Position, wo er im Datenstrom stand). Zur Zeit ist dieser Befehl nur für den Simplex Betrieb realisiert.

Beispiel: Rechnung mit verschiedenen Durchschlägen
Es werden Daten für eine Seite einer Rechnung zum Drucker übertragen. Diese Seite soll 4 mal gedruckt werden:

1. mit Rechnungs Formular, 2. ohne Formular,
3. mit Lieferschein Formular, 4. mit Archiv Formular.

Dafür werden 3 Macros generiert, z.B.:

1. Rechnungs Formular (z.B. Macro No. 0001)
2. Lieferschein Formular (z.B. Macro No. 0002)
3. Archiv Formular (z.B. Macro No. 0003)

Vor den Daten für die zu vervielfältigende Seite wird folgender Befehl geschickt:

```
<ESC><ESC>dg 04 0001 0000 0002 0003
```

<nn> = 04 : Es sollen 4 Seiten gedruckt werden.

<uuuu₁> = 0001 : Ausführen von Macro 0001 mit der 1. Seite.

<uuuu₂> = 0000 : Kein Macro mit der 2. Seite.

<uuuu₃> = 0002 : Ausführen von Macro 0002 mit der 3. Seite.

<uuuu₄> = 0003 : Ausführen von Macro 0003 mit der 4. Seite.

Nun werden alle folgenden Seiten 4 mal gedruckt und jeweils mit dem angegebenen Macro hinterlegt.

Zum Abschalten dieser Funktion wird folgender Befehl geschickt:

```
<ESC><ESC>dg 0
```


2.10. Vervielfältigen von Informationsbereichen

dx,dy

Funktion: Dieser Befehl dient zum Duplizieren des Informationsinhaltes von Teilbereichen einer Seite in x- und/oder y-Richtung.

Syntax:	<code><ESC><ESC>dx <nnnn> <vvvv></code>	(27)(27)(100)(120)...
	<code><ESC><ESC>dy <nnnn> <vvvv></code>	(27)(27)(100)(121)...

Parameter:

- `<nnnn>` = 0001 bis 9999 : Duplikationsfaktor
- `<vvvv>` = 0001 bis 9999 : Verschiebung in x- bzw. y-Richtung in Anzahl Bildpunkte (dots)

Das Ende der zu vervielfältigenden Befehlssequenz bzw. des zugehörigen Informationsinhaltes wird definiert durch

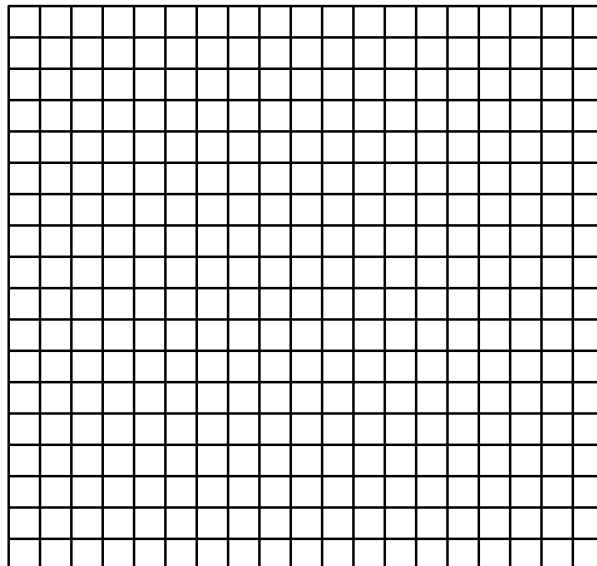
`<ESC><ESC>dx 0` und
`<ESC><ESC>dy 0`

Beachte: Die Verschiebung wirkt nur auf absolute Koordinatenangaben. In der Befehlsfolge ist die Verschachtelung einer x- und y-Vervielfältigung möglich.

Beispiel:

Erzeugung eines quadratischen Musters von 50 Punkten
Kantenlänge. Ausgabe der Seite mit einem Form Feed.

```
&%%dx 20 50  
&%%03 2 800 1500 800 2400  
&%%dx 0  
&%%dy 19 50  
&%%03 2 800 1500 1750 1500  
&%%dy 0  
&%%v
```



2.11. Der eingeschränkte CLEAR - Befehl

J

Funktion: Dieser Befehl setzt die Druckparameter auf die Grundeinstellung zurück - mit Ausnahme der Ränder und der Schreibrichtung.

Eingestellte Tabulatoren bleiben erhalten.

Ist ein MACRO vorhanden, das an ein FORM FEED (FF) gekoppelt ist, dann bleibt diese Kopplung vorhanden.

Syntax:	<code><ESC><ESC>J</code>	(27)(27)(74)
----------------	--------------------------------------	---------------------

Parameter: keine

Die Wirkungen im einzelnen:

1. Schriftparameter im Grundzustand:
 - Auto - LF : auf Einschalt-Konfiguration
 - Unterstreichen : aus
 - Inversdruck : aus
 - Sperrschrift : aus
 - Fettschrift : aus
 - Vergrößerung Höhe : 1
 - Vergrößerung Breite : 1
 - Rasterhinterlegung : aus
 - Auto - CR : auf Einschalt-Konfiguration
 - Umgekehrter "Wagenlauf" (bidirektionaler Druck, bei Zeilendrucker-Emulationen) : aus

2. Grafik-Reset wird ausgelöst
3. Schriftart
in Grundeinstellung : Einschalt-Konf. EEPROM
4. Zeichentabelle
in Grundeinstellung : Einschalt-Konf. EEPROM
5. Horizontale Grundschriftweite
in Grundeinstellung : Einschalt-Konf. EEPROM
6. Zeichenbreite
(eventuell proportional)
in Grundeinstellung : Einschalt-Konf. EEPROM
7. Vertikale Grundschriftweite
in Grundeinstellung : Einschalt-Konf. EEPROM
8. Zeilenabstand
in Grundeinstellung : Einschalt-Konf. EEPROM
9. Fonthöhenskalierung
in Grundeinstellung : Einschalt-Konf. EEPROM
10. Fontbreitenskalierung
in Grundeinstellung : Einschalt-Konf. EEPROM
11. N-Up Mode
in Grundeinstellung : Einschalt-Konf. EEPROM

2.12. Der CLEAR - Befehl



Funktion: Dieser Befehl setzt alle Druckparameter auf die Grundeinstellung zurück.
Eine eventuell vorhandene MACRO-Kopplung wird gelöst.

Syntax: <ESC><ESC>j (27)(27)(106)

Parameter: keine

Die Wirkungen im einzelnen:

1. Textränder auf Grundposition : Def. durch Einschalt-Konfiguration oder durch die Emulation
2. Tabulator-Speicher wird gelöscht
3. Schreibrichtung in Grundeinstellung : Einschalt-Konf. EEPROM
4. Textcursor auf Grundposition : Linker, oberer Textrand - bezogen auf die Schreibrichtung
5. Macro-Kopplung wird gelöst
6. Eingeschränkter Clear - Befehl (<ESC><ESC>J) wird ausgelöst

2.13. Der Reset - Befehl**re**

Funktion: Dieser Befehl setzt alle Druckparameter auf die Grundeinstellung zurück. Außerdem werden alle MACROs gelöscht und der Befehl <ESC><ESC>j ausgeführt.

Syntax: <ESC><ESC>re (27)(27)(114)(101)
--

Parameter: keine

Die Wirkungen im einzelnen:

1. Transparent Code in Grundeinstellung: Einschalt-Konf. EEPROM
2. Druckersprache in Grundeinstellung : Einschalt-Konf. EEPROM
3. Papierzufuhr und Papierablage in Grundeinstellungen : Einschalt-Konf. EEPROM
4. Emulation in Grundeinstellung : Einschalt-Konf. EEPROM
5. Druckrichtung in Grundeinstellung : Einschalt-Konf. EEPROM
6. Papierformateinstellung in Grundeinstellung : Einschalt-Konf. EEPROM
7. Drauckauflösung in Grundeinstellung : Einschalt-Konf. EEPROM

8. Temporär gespeicherte MACROs werden gelöscht (RAM - Speicher).
9. Permanent gespeicherte MACROs etc. werden aus dem RFM hochgeladen (nach dem Einschalten des Druckers).
10. Duplex Mode wird auf die Grundeinstellung gesetzt : Einschalt-Konf. EEPROM
11. Clear - Befehl (<ESC><ESC>j) wird ausgelöst.

2.14. Der FORM FEED - Befehl



Funktion: Dieser Befehl löst die Ausgabe einer Druckseite aus, sofern sich Informationen im Bildspeicher des Druckers befinden. Der Bildspeicher wird gelöscht.

Syntax: <ESC><ESC>v (27)(27)(118)
--

Parameter: keine

Beachte: Er wird ignoriert, wenn sich kein Zeichen im Speicher befindet. So wird vermieden, daß viele von Treibern generierte FORM FEED-Kommandos Leerseiten produzieren.

"Nicht druckbare" Zeichen gelten auch als Zeichen; somit löst ein FORM FEED - Befehl dann die Ausgabe einer (leeren) Druckseite aus.

2.15. Duplex - Druck

du

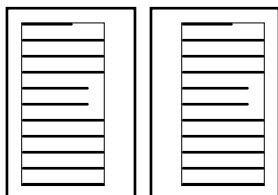
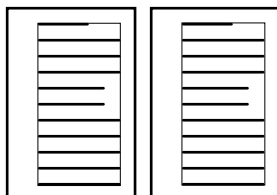
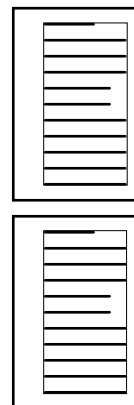
Funktion: Dieser Befehl schaltet den "intelligenten", automatischen Vorder- und Rückseitendruck ein bzw. aus, wobei er nur bei Druckern verwendet werden kann, die die entsprechenden Papierlauf-Voraussetzungen besitzen.

Syntax:	<code><ESC><ESC>du <n></code>	(27)(27)(100)(117)...
----------------	---	------------------------------

Parameter:	<code><n> = 0</code>	Duplexdruck ausschalten.
	<code><n> = 1</code>	Duplexdruck, lange Format-Seite, mit Rückseitenversatz.
	<code><n> = 2</code>	Duplexdruck, kurze Format-Seite, ohne Rückseitenversatz.
	<code><n> = 3</code>	Duplexdruck, lange Format-Seite, ohne Rückseitenversatz, für vorbedrucktes Papier.
	<code><n> = 4</code>	Duplexdruck, kurze Format-Seite, ohne Rückseitenversatz, für vorbedrucktes Papier.
	<code><n> = 5</code>	Duplexdruck, lange Format-Seite, ohne Rückseitenversatz.

Hinweis: Jeder Duplexbefehl löst einen FormFeed aus und startet mit einer Vorderseite.

Beachte: Rückseitenversatz bedeutet, daß der linke Rand einer Seite zum rechten Rand auf der Rückseite wird und umgekehrt. Dadurch bleibt beim Duplex – Printjob ein eingestellter Heftrand erhalten. Für die kurze Format-Seite kann Duplexdruck nur ohne Rückseitenversatz aktiviert werden.

Beispiele: Duplex – Drucklange Format-Seite,
mit Rückseitenversatzlange Format-Seite,
ohne Rückseitenversatzkurze Format-Seite,
ohne Rückseitenversatz**Hinweis:**

In den Duplexmodi 3 und 4 ist sichergestellt, daß zuerst die Vorderseite gedruckt wird und danach die Rückseite. Dadurch ist gewährleistet, daß vorbedrucktes Papier (z.B. Briefpapier mit Briefkopf) bei gemischtem Simplex- Duplexdruck seitenrichtig bedruckt wird.

2.16. Two-Up Mode (N-Up Mode)

tu

Funktion: Mit dieser Funktion wird die Anordnung von mehreren Druckseiten je Seitenformat (Papierbreite beziehungsweise Formatlänge) festgelegt.

<p>Syntax: <ESC><ESC> tu <nn> (27)(27)(116)(117) ...</p>
--

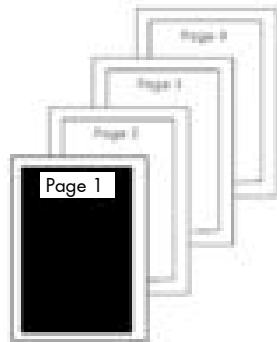
Parameter:

<nn> = 00 bis 90	:	Kennzahl (hexadezimal) für die Anzahl der Druckseiten je Seitenformat:
		1. Stelle: Anzahl quer zur Papierlaufrichtung
		2. Stelle: Anzahl in Papierlaufrichtung
<nn> = 00 : schaltet den Two-Up Mode aus		
<nn> = 01 : reserviert		
<nn> = 02 bis 09 : 2 bis 9 Druckseiten je Formatlänge (in Papierlaufrichtung)		
<nn> = 20 bis 90 : 2 bis 9 Druckseiten je Formatbreite (quer zur Papierlaufrichtung)		

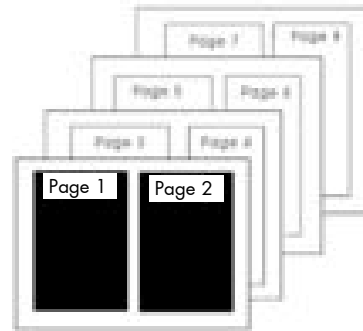
Beachte: Alle anderen Seitenparameter müssen nach Bedarf zusätzlich eingestellt werden.
Die Kombination von mehreren Druckseiten je Formatlänge und mehreren Druckseiten je Formatbreite ist zur Zeit noch nicht möglich.

Hinweis: Jeder tu - Befehl (außer die Ausschaltung des Two-Up Modes = Codenummer 00) löst einen Start in der oberen linken Ecke der Seite aus (erste Druckseite).

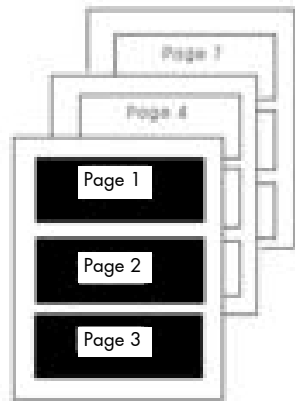
Übersicht über die Two-Up Modes (N-Up Modes)



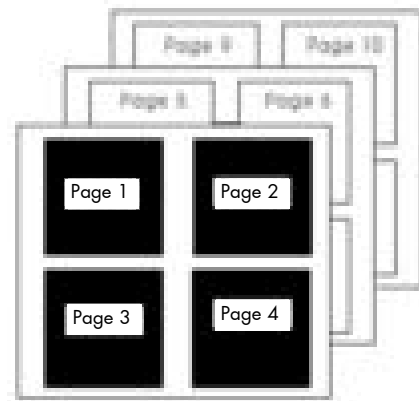
N-Up ausgeschaltet



N-Up 2 quer zur Papierlaufrichtung



N-Up 3 in Papierlaufrichtung

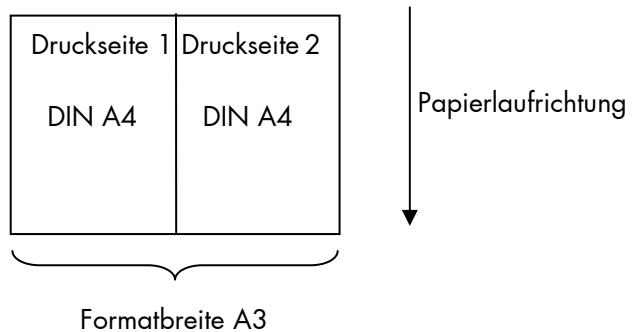


N-Up 2 in beiden Richtungen

Beispiel:

In einem Drucker wird Endlospapier mit der Formatbreite DIN A3 verwendet. Nun sollen im Two-Up Mode jeweils 2 DIN A4 Druckseiten nebeneinander ausgedruckt werden.

&%%&tu 20



2.17. Seitennummerierung

Z

Mit Hilfe von IDOL - Befehlen läßt sich ein Seitenzähler im MICROPLEX - Drucker programmieren. Dieser Seitenzähler wird direkt im Drucker verwaltet. Mit einer Reihe von Befehlen kann der Seitenzähler leicht manipuliert und ausgegeben werden.

2.17.1. Seitenzähler setzen

Z=

Funktion: Dieser Befehl setzt den Seitenzähler auf den angegebenen Wert.

Syntax: <ESC><ESC>Z= <nnnn> (27)(27)(90)(61)...

Parameter: <nnnn> = 0000 bis 9999 : Nummer der aktuellen Seite

2.17.2. Seitenzähler erhöhen**Z+****Funktion:** Dieser Befehl erhöht den Seitenzähler um den angegebenen Wert.

Syntax:	<code><ESC><ESC>Z+ <nnnn></code>	(27)(27)(90)(43)...
----------------	--	----------------------------

Parameter: `<nnnn>` = 0000 bis 9999 : Wert für die Erhöhung der Seitennummer.**2.17.3. Drucken der Seitennummer****ZP****Funktion:** Dieser Befehl bewirkt, daß die Seitennummer an der aktuellen Textcursor - Position ausgedruckt wird.

Syntax:	<code><ESC><ESC>ZP</code>	(27)(27)(90)(80)
----------------	---------------------------------------	-------------------------

Parameter: keine**Beispiel:** Es sollen drei Seiten mit Seitennummern versehen werden. Ausgehend von Seite 50 sollen die folgenden Seiten eine um jeweils 1 erhöhte Seitenzahl erhalten. Die Nummer soll an der Position $x = 2000$, $y = 3000$ gedruckt werden.

```

&Z= 50&D2000 3000&ZP&v
&Z+ 1&D2000 3000&ZP&v
&Z+ 1&D2000 3000&ZP&v

```

2.17.4. Seitenzähler verringern



Funktion: Dieser Befehl verringert den Zähler um den angegebenen Wert.

Syntax: <ESC><ESC>Z- <nnnn> **(27)(27)(90)(45)...**

Parameter: <nnnn> = 0000 bis 9999 : Wert für die Verringerung der
Seitennummer.

2.18. Druckfarbe wählen

18

Funktion: Mit diesem Befehl wird die aktive Druckfarbe gewählt. Danach werden alle Darstellungen (Text, Grafik) in der gewählten Farbe ausgegeben.

Syntax:	<code><ESC><ESC>18 <nnn></code>	(27)(27)(49)(56)...
----------------	---	----------------------------

Parameter:	<code><nnn></code>	= 001 ... 007	: aktuelle Druckfarbe
	<code><nnn></code>	= 001	: 1. Farbe (Standard: Schwarz)
	<code><nnn></code>	= 002	: Gelb
	<code><nnn></code>	= 003	: Cyan
	<code><nnn></code>	= 004	: Magenta
	<code><nnn></code>	= 005	: Rot
	<code><nnn></code>	= 006	: Grün
	<code><nnn></code>	= 007	: Blau

Beachte: Dieser Befehl wird ignoriert, wenn der Drucker nicht über mehrere Druckfarben verfügt.

Beispiel: Das Wort 'MICROPLEX' soll rot gedruckt werden, das Wort 'IDOL' soll blau gedruckt werden. Danach soll schwarz weitergedruckt werden.

```
&%%18 005MICROPLEX
&%%18 007IDOL&%%18 001
```

```
MICROPLEX
IDOL
```


2.18.1. Beispiel: IGP – Befehle zur Druckfarbe

Die Seitenbeschreibungssprache IGP verfügt von Haus aus über keine Befehle zur Farbsteuerung.

MICROPLEX hat dieser Emulation eigene Befehle in IGP Syntax hinzugefügt, die eine Farbsteuerung ermöglichen.

Eine weitere Möglichkeit, unter IGP Farbe in die Anwendung zu bringen, besteht darin, Grafiken und Logos in Farbe zu erstellen und mit den IGP eigenen Befehlen in den Drucker zu laden.

a) IGP Farbbefehle:

COLOR; <n>

<n> = 1	erste Farbe (Standard: Schwarz)
<n> = 2	Gelb
<n> = 3	Cyan
<n> = 4	Magenta
<n> = 5	Rot
<n> = 6	Grün
<n> = 7	Blau

Beispiel:

```

^CREATE;AB1
COLOR;5           (Farbumschaltung auf "rot")
ALPHA
02;40;2;2;' ORDER CONFIRMATION'
C12;03;63;0;0;'Normalauftrag   '
STOP
SCALE;CHAR;10;10
COLOR;1           (Farbumschaltung zurück auf erste Farbe „schwarz“)
BOX
2;22;4;27;80
STOP

```

Eine Farbumschaltung wirkt ab der Position, an der sie gesetzt wird bis zum nächsten Farbbefehl.

b) Farbige Logos:

Farbige Logos können mit Hilfe des IGP Befehls
^LOGO;LOGO NAME;VL;HL
definiert werden.

Unterstützt werden folgende Formate:

PCX: 1 bit Schwarz/weiß
24 Bit RGB
TIFF: 1 bit Schwarz/weiß

2.18.2. Beispiel: CODE V - Befehle

Die Seitenbeschreibungssprache CODE V verfügt von Haus aus über keine Befehle zur Farbsteuerung.

MICROPLEX hat dieser Emulation eigene Befehle in CODE V Syntax hinzugefügt, die eine Farbsteuerung ermöglichen.

Befehl zum Einschalten einer Farbe:

^|COLOR; <n>

<n> = 1	erste Farbe (Standard: Schwarz)
<n> = 2	Gelb
<n> = 3	Cyan
<n> = 4	Magenta
<n> = 5	Rot
<n> = 6	Grün
<n> = 7	Blau

2.18.3. Beispiel: Prescribe IIe - Befehle

Zur Zeit sind folgende Farbbefehle verfügbar:

ACLI index, color1, color2, color3;	Add Color by Index
Index: 1 .. 255	
color: 0 .. 255	
SCOL number;	Select Color
number: 0 .. 255	
CID color-space, encoding;	Configure Color Image Data
color-space:	
0 = Device-dependent RGB (Default)	
1 = Device-dependent CMY	
2 = Void	
3 = CIE L*a*b	
encoding (Pixel encoding):	
1 = Indexed by Pixel (Default)	
3 = Direct by Pixel	

Farbige Bitmaps (24 Bit RGB, unkomprimiert) können mit Hilfe des folgenden Befehls definiert werden:

RVCL length, data;	Receive Color Raster Data
length: bytes of raster data	
data: binary raster data 0H to FFH	

2.18.4. Beispiel: Epson FX (ESC/P) - Befehle

Diese Farbbefehle sind Epson Stylus kompatibel.

Befehl zum Einschalten einer Farbe:

<ESC>r <nn>

<nn> = 00	Schwarz
<nn> = 01	Magenta
<nn> = 02	Cyan
<nn> = 03	Violett
<nn> = 04	Gelb
<nn> = 05	Rot
<nn> = 06	Grün

2.19. Druckrichtung wählen

Funktion: Mit diesem Befehl wird die aktive Druckrichtung gewählt, die Einstellung ist in 90° - Schritten möglich.
Die Festlegung der Druckrichtung bestimmt die Ausrichtung des gesamten Druckbildes incl. Grafiken etc. auf dem Papier.
Im Vergleich dazu beeinflusst der Befehl <ESC><ESC>C nur die Schreibrichtung.

Syntax:	<ESC><ESC>rt <n>	(27)(27)(114)(116) ...
----------------	------------------	-------------------------------

Parameter:

<n> =	0 bis 3	: Befehlstyp
<n> =	0	: Portrait (Hochformat)
<n> =	1	: Landscape (Querformat)
<n> =	2	: Portrait Kopf
<n> =	3	: Landscape Kopf

Hinweis: Jeder rt Befehl löst einen FormFeed aus und die Papierformat-einstellung wird auf die Grundeinstellung zurückgesetzt.

Beachte: Die gesetzten Ränder bleiben erhalten, die logische Beziehung (oben, unten, rechts, links) wird neu zugeordnet.

Beispiel: Auf der ersten Seite soll ein Beispieltext plus Grafik mit der Ausrichtung Portrait Kopf gedruckt werden. Auf der folgenden Seite soll das gleiche Druckbild mit der Richtung Portrait (Hochformat) gedruckt werden:

```
&%%D10502760&%%E
&%%F&%%rt2Richtung 2 – Portrait Kopf&%%E
&%%49 01&%%13 09
&%%07 0900 3000 1800 3200
&%%v
&%%F&%%rt0Richtung 0 —— Portrait&%%E
```

&%%49 01&%%13 09
&%%07 0900 3000 1800 3200
&%%v

2.20. Schneidebefehl (inkl. Tear off und Peel off)**CU**

Funktion: Mit diesem Befehl wird der Ablauf von Materialtransport und Schnitt festgelegt. Auch die Vorschub-Modi Abziehen (Peel-Off) und Abreißen (Tear off) werden unterstützt.

Der Befehl wird ignoriert, wenn der Drucker diese Funktionen nicht unterstützt.

Syntax:	<code><ESC><ESC>cu <n></code>	(27)(27)(99)(117) ...
----------------	---	------------------------------

Parameter:	<code><n> =</code>	0 bis 7	: Befehlstyp
	<code><n> =</code>	0	: Abschalten des Schneidebefehls cu 3 (automatisches Schneiden beenden)
	<code><n> =</code>	1	: Schneiden sofort (am Anfang der Seite)
	<code><n> =</code>	2	: Schneiden beim Seitenauswurf (Form Feed) (am Ende der Seite)
	<code><n> =</code>	3	: Automatisches Schneiden bei jedem Seitenauswurf (Form Feed)
	<code><n> =</code>	4	: Abreißen (Tear off) sofort
	<code><n> =</code>	5	: Abreißen (Tear off) nach Seitenauswurf
	<code><n> =</code>	6	: Spenden (Peel off) sofort
	<code><n> =</code>	7	: Spenden (Peel off) nach Seitenauswurf

Beispiel: Nach der Bedruckung soll die fertige Seite abgeschnitten werden.
 &%%&cu2
 ... Befehle zur Bedruckung der Seite ... gefolgt von einem Form Feed:
 &%%&v

Nach der Bedruckung der Seite erfolgt ein zusätzlicher Materialvorschub, um das Material zum Cutter zu transportieren und abzuschneiden. (Das noch unbedruckte Material wird vor der weiteren Bedruckung wieder zurück transportiert).

2.21. Heftbefehl (Staple)



Funktion: Mit diesem Befehl wird festgelegt, wie die Seiten eines Jobs mit Heftklammern zusammengeheftet werden. Nach dem Startbefehl (Jobanfang) werden die folgenden Seiten gesammelt und dann nach dem Befehl Job beenden zusammengeheftet.

Der Befehl wird ignoriert, wenn der Drucker die Heftfunktion nicht unterstützt.

<p>Syntax: <ESC><ESC><i>tk</i> <n> (27)(27)(116)(107) ...</p>

Parameter:	<n> =	0 bis 3	: Befehlstyp
	<n> =	0	: abschalten, Job beenden
	<n> =	1	: Jobanfang, Klammerposition einfach oben
	<n> =	2	: Jobanfang, Klammerposition einfach unten
	<n> =	3	: Jobanfang, Klammerposition doppelt seitlich

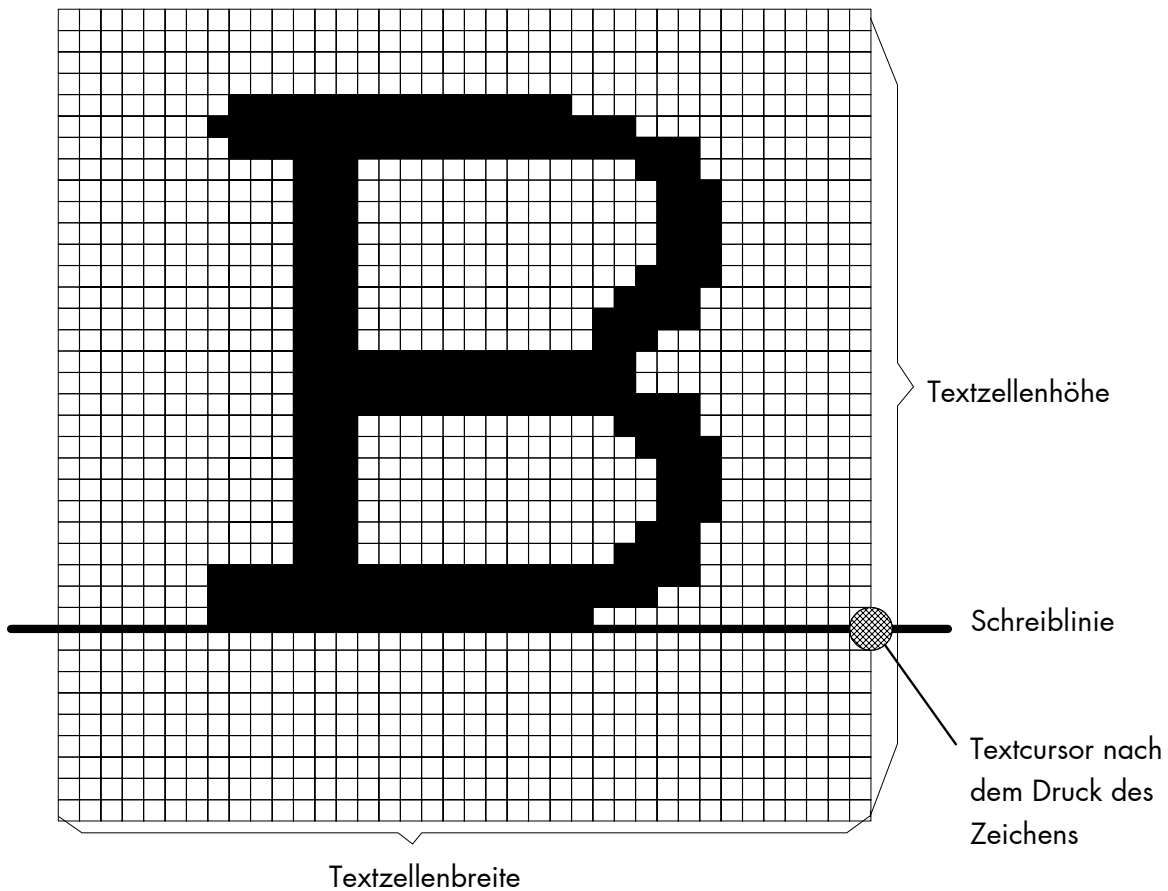
3. Funktionsgruppe FONT

Die Befehle der Funktionsgruppe FONT (Zeichensatz) beziehen sich auf das Layout eines Textes im weiteren Sinne. Mit den Befehlen dieser Funktionsgruppe werden die Schriftart, Schreibrichtung, Ränder, Tabulatoren sowie die Ausrichtung des Textes (Blocksatz, linksbündig etc.) bestimmt. Ferner enthält die Funktionsgruppe FONT Befehle zum Positionieren des Textcursors.

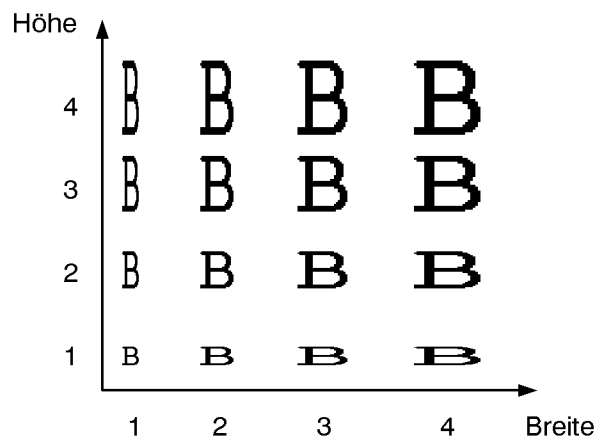
Es werden 4 Fonttypen unterschieden:

- Pixelfonts (z.B. Font 0600)
- Stickfonts (z.B. Font 0060)
- Vektorfonts (z.B. Font 0050)
- Outlinefonts (Bézier) (z.B. Font 2000)

Pixelfonts: Jedes Zeichen ist als Bitmuster in der Fontbank gespeichert.

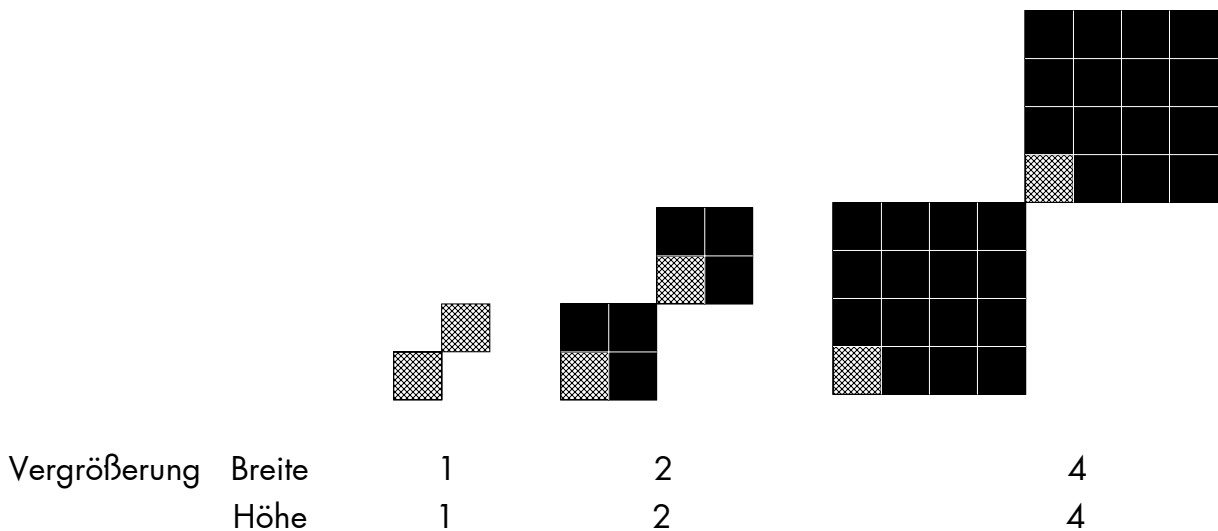


Pixelfonts können in ganzzahligen Stufen vergrößert werden (1 bis 4). Beim Vergrößern von Zeichen eines Pixelfonts entsteht ein "Treppeneffekt". Die Schriftqualität wird geringer.



Das Diagramm zeigt das Aussehen von Zeichen eines Pixelfonts abhängig von der Vergrößerung in Breite und Höhe.

Beim Skalieren eines Pixelfonts werden die einzelnen Pixel gemäß des Vergrößerungsfaktors neu berechnet. Es werden nur die im Font vorhandenen Pixel vergrößert. Dies ist der Grund für den "Treppeneffekt" bei starker Vergrößerung.

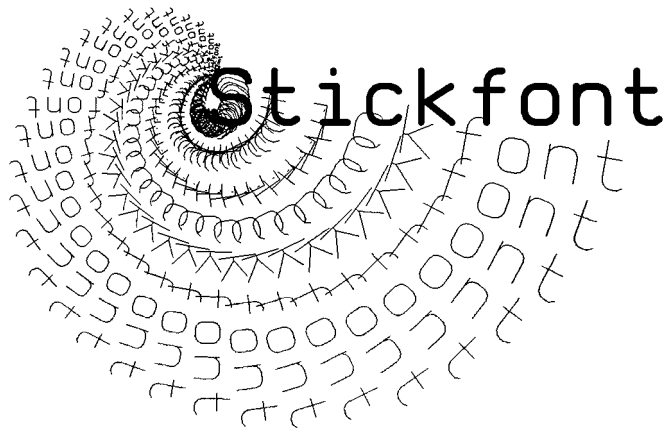


Stickfonts: Stickfonts sind Plotterschriftarten. Ein Zeichen eines Stickfonts besteht nur aus einer Linie. Dieser Font ist skalierbar.
Die Zeichen eines Stickfonts werden mit der aktuellen Zeichenfarbe, Linienbreite und dem aktuellen Linienmode gezeichnet.



Hier als Beispiel der Font 0060 mit der Linienbreite 4, 10 und 20 Punkten.

Stickfonts lassen sich stufenlos in der Breite und Höhe skalieren.
Hier einige Beispiele:



Vektorfonts: Zeichen von Vektorfonts bestehen aus ausgefüllten Flächen. Die Flächen werden von vertikalen, horizontalen und diagonalen Linien (Vektoren) begrenzt und beschrieben. Vektorfonts sind skalierbar. Hier zum Beispiel der Font 0050. Flächen werden mit dem aktuellen Raster gefüllt.



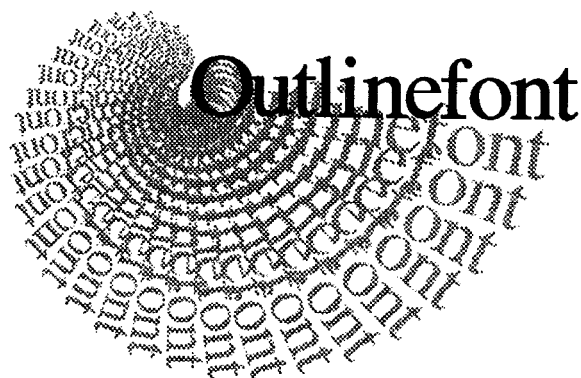
Vektorfonts lassen sich in der Höhe und Breite skalieren.
Hier einige Beispiele:



Outlinefonts: Zeichen von Outlinefonts bestehen (wie bei Vektorfonts) aus ausgefüllten Flächen. Flächen werden mit dem aktuellen Raster gefüllt. Diese Flächen werden - anstatt durch Vektoren - durch Bézierkurven (Polynome) beschrieben. Dadurch sind geschwungene Kurvenverläufe möglich.
Hier zum Beispiel der Font 2000:

B

Outlinefonts lassen sich stufenlos in der Breite und Höhe skalieren.
Hier einige Beispiele:



3.1. Textcursor

Der Textcursor beschreibt die aktuelle Position, an der zum Beispiel der nächste Buchstabe oder ein Barcode positioniert/gedruckt wird.

Die **Standard Auflösung** beträgt 300 dpi (Bildpunkte je Zoll) in horizontaler und in vertikaler Richtung.

Beachte: **Verwenden Sie keine Zeilenvorschübe (Linefeed LF) und/oder Wagenrückläufe (Carriage Return CR) direkt nach den Befehlen zum Positionieren des Textcursors,** weil „CR“ und „LF“ den Cursor ebenfalls bewegen.

Hinweis: Generell gilt: Vermeiden Sie die zu häufige Verwendung von Zeilenvorschüben (Linefeed LF) und/oder Wagenrückläufen (Carriage Return CR). „CR“ und „LF“ werden gerne verwendet, um Befehle optisch zu trennen und um Parameter besser erkennbar zu machen (Ziel: Verbesserung der Lesbarkeit). Beachten Sie, dass jeder CR und jeder LF die Cursor-Position verändert. Sobald der Cursor die Seitengrenze überschreitet, kann ein automatischer FormFeed ausgelöst werden.

3.1.1. Absolutes Positionieren des Textcursors

Funktion: Mit diesem Befehl wird der Textcursor innerhalb des Koordinatensystems absolut positioniert. Der Koordinatenursprung befindet sich in der linken oberen Ecke der Seite. Eine absolute Positionierung ist unabhängig von der aktuellen Schreibrichtung (vergl. Abschnitt 1.3. Metrik).

Syntax: <ESC><ESC>D <xxxx> <yyyy> (27)(27)(68)...

Parameter: <xxxx> = 0000 bis x_{\max} : x - Koordinate in Bildpunkten
 <yyyy> = 0000 bis y_{\max} : y - Koordinate in Bildpunkten

Beachte: Die Anzahl der Bildpunkte hängt vom maximalen Papierformat des Druckers ab.
 Bei A4 gilt zum Beispiel: $x_{\max} = 2480$, $y_{\max} = 3508$
 (vergl. Abschnitt 1.3. Metrik)
 Wird versucht, den Textcursor außerhalb der Maximalgrenzen zu positionieren, so wird der Positionierbefehl ignoriert. Der Textcursor verbleibt an der aktuellen Position!

Beispiel: Das Wort "IDOL" soll an der Position $x = 0700$, $y = 3150$ gedruckt werden.

```
&%%D0700 3150IDOL
```

```
IDOL
```

3.1.2. Speichern der aktuellen Textcursor - Position

E

Funktion:

Bei der Bearbeitung einer Druckseite im Bildspeicher, z.B. bei Formulardruck, ist es manchmal erforderlich, eine bestimmte Cursorposition immer wieder anfahren zu können. Der Befehl "Cursorposition speichern" speichert die aktuelle Textcursor - Position.

Wenn einmal eine Textcursorposition gespeichert wurde, können alle anderen Positionierungen relativ zu diesem Punkt geschehen. Durch einfaches Verändern dieser einen Koordinate kann das gesamte Druckbild auf einer Seite verschoben werden, da sich alle anderen Koordinaten mit der relativen Positionierung verschieben, ohne dass diese verändert werden müssen. Somit ist die Pflege von elektronischen Formularen sehr leicht.

Syntax:

<ESC><ESC>E

(27)(27)(69)...

Parameter:

keine

Beispiel:

Der Textcursor wird auf die Koordinaten $x = 1000$, $y = 500$ gesetzt. Diese Textcursor - Position soll gespeichert werden.

&%%D1000 0500

&%%E

3.1.4. Vertikale Relativpositionierung des Textcursors

S

Funktion: Der Befehl bewirkt einen vertikalen Versatz des Textcursors gegenüber der aktuellen Position.

Syntax: <ESC><ESC>S <nnnn> (27)(27)(83)...

Parameter: <nnnn> = -9999 bis 9999 : Versatz in Anzahl Bildpunkten

Beachte: Bei negativem n erfolgt ein Versatz nach oben, bei positivem n nach unten bezogen auf die aktuelle Textcursor - Position.

"Vertikal" bezieht sich auf die aktuelle Schreibrichtung. Der maximal mögliche Versatz ergibt sich aus dem Abstand der gespeicherten Position zum absoluten oberen oder unteren Textrand. Wird dieser maximale Versatz überschritten, so wird der Befehl ignoriert.

Beispiel: Es soll der Text "EFG" und "KLM" nach oben verschoben sein.

```
&%%&%D 0900 3000ABCD&%%&%S -40EFG&%%&%S 40HIJ
&%%&%S -40KLM&%%&%S 40NOP
```

```
      EFG   KLM
     ABCD  HIJ  NOP
```


3.1.6. Absolute Positionierung des Textcursors in x-Richtung **CX**

Funktion: Mit diesem Befehl wird der Textcursor absolut in x-Richtung (horizontal) positioniert. Die aktuelle y-Position bleibt erhalten.

Syntax: <ESC><ESC>cx <nnnn> (27)(27)(99)(120)...

Parameter: <nnnn> = -9999 bis 9999 : x-Position in Anzahl Bildpunkten

Beachte: "Horizontal" bezieht sich immer auf Schreibrichtung "0" (Portrait). Der Maximalwert für <nnnn> ergibt sich aus der maximalen Anzahl Bildpunkte in der jeweiligen Schreibrichtung und des Papierformates. Wird dieser maximale Wert überschritten, so wird der Befehl ignoriert.

Beispiel: Die Worte "MICROPLEX" und "IDOL" sollen bei den absoluten x-Koordinaten 900 und 1500 geschrieben werden.

```
&%%cx0900MICROPLEX&%%cx1500IDOL
```

SOLID

IDOL

3.1.7. Absolute Positionierung des Textcursors in y-Richtung

cy

Funktion: Mit diesem Befehl wird der Textcursor absolut in y-Richtung (vertikal) positioniert. Die aktuelle x-Position bleibt erhalten.

Syntax: <ESC><ESC>cy <nnnn> (27)(27)(99)(121)...
--

Parameter: <nnnn> = -9999 bis 9999 : y-Position in Anzahl
Bildpunkten

Beachte: "Vertikal" bezieht sich auf die aktuelle Schreibrichtung.
Der Maximalwert für <nnnn> ergibt sich aus der maximalen Anzahl Bildpunkte für die jeweiligen Schreibrichtung und des Papierformates. Wird dieser maximale Wert überschritten, so wird der Befehl ignoriert.

Beispiel: Das Wort "IDOL" soll bei den absoluten y-Koordinaten 2700 und 2900 geschrieben werden. Der aktuelle x-Wert soll beibehalten werden.

&%%cy2700IDOL&%%cy2900IDOL

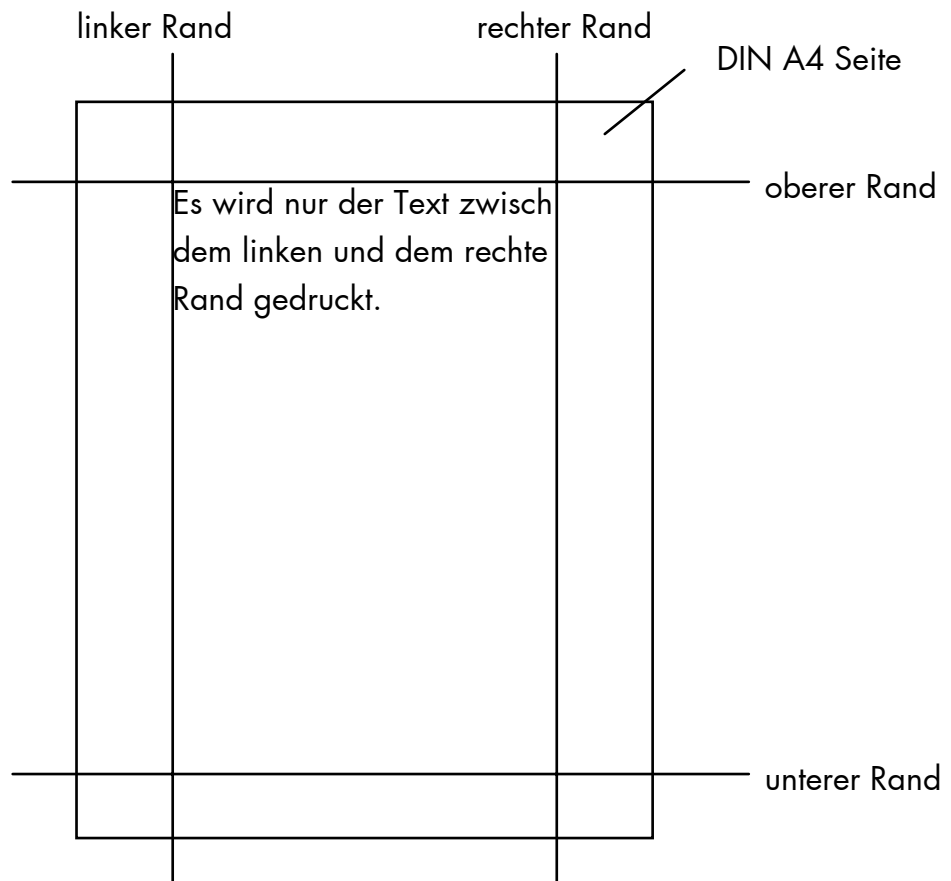
IDOL

IDOL

3.2. Ränder

Mit den folgenden Befehlen können die Text - Ränder gesetzt werden. Dies ist jedoch nur möglich im Zusammenhang mit dem Positionieren des Textcursors. Hierbei handelt es sich also nicht um die absoluten Ränder. Die Zuordnung der Ränder ist abhängig von der Schreibrichtung. Die eingestellten Textränder bleiben erhalten, bis sie nach einem Aus-/Einschalten oder durch einen CLEAR - Befehl durch die im EEPROM gespeicherten Anfangswerte ersetzt werden.

Beachte: Es muß zuerst der Textcursor an die zukünftige Randposition gesetzt werden. Erst dann kann der Rand auf die aktuelle Textcursor - Position gesetzt werden.



Hinweis: Die neuen Text - Ränder begrenzen den Bereich, in dem anschließend der Textcursor positioniert werden kann. Nur Befehle zur absoluten Positionierung (siehe vorige Abschnitte) können den Cursor außerhalb dieses Bereiches positionieren (wobei natürlich die absoluten Ränder des Papierformats nicht überschritten werden können).

3.2.1. Linken Textrand auf die aktuelle Textcursor - Position setzen

rl

Funktion: Dieser Befehl setzt den linken Textrand auf die aktuelle Textcursor - Position.

Syntax: <ESC><ESC> <i>rl</i> (27)(27)(114)(108)

Parameter: keine

Beachte: Nach einem CR-Befehl ist der linke Rand die aktuelle Textcursor-Position.

Beispiel: Der linke Rand soll auf die Koordinate $x = 0100$ gesetzt werden.

1. Absolutpositionierung des Textcursors:

$x = 0100$, $y =$ beliebig, hier = 1000

2. Setzen des linken Textrandes

```
&%%D0100 1000&%%rl
```

3.2.2. Oberen Textrand auf die aktuelle Textcursor - Position setzen

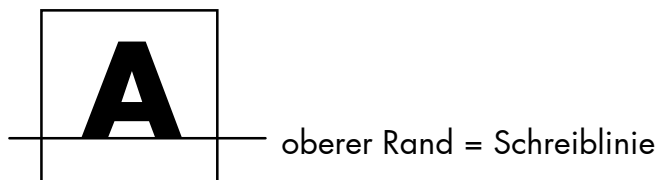
ro

Funktion: Dieser Befehl setzt den oberen Textrand auf die aktuelle Textcursor - Position.

Syntax:	<code><ESC><ESC>ro</code>	(27)(27)(114)(111)
----------------	---------------------------------------	---------------------------

Parameter: keine

Beachte: Der obere Rand bezieht sich auf die Schreiblinie der Textzellen.



Beispiel: Der obere Rand soll auf die Koordinate $y = 0100$ gesetzt werden.

1. Absolutpositionierung des Textcursors:
 $x =$ beliebig, hier: $x = 0100$, $y = 0100$

2. Setzen des oberen Textrandes

`&%%&%D0100 0100&%%&%ro`

3.2.4. Rechten Textrand auf die aktuelle Textcursor - Position setzen

rr

Funktion: Dieser Befehl setzt den rechten Textrand auf die aktuelle Textcursor - Position.

Syntax:	<code><ESC><ESC>rr</code>	(27)(27)(114)(114)
----------------	---------------------------------------	---------------------------

Parameter: keine

Beachte: Wenn eine Textzelle nicht vollständig gedruckt werden kann, wird das Zeichen nicht gedruckt. Es können keine halben Zeichen (Clipping) entstehen.

Beispiel: Der rechte Rand soll auf die Koordinate x = 2400 gesetzt werden.

1. Absolutpositionierung des Textcursors:
x = 2400, y = beliebig, hier: y = 0100

2. Setzen des rechten Textrandes

`&%%D2400 0100&%%rr`

3.2.5. Textränder auf Einschaltländer zurücksetzen

Funktion: Dieser Befehl setzt die Textränder auf die im EEPROM gespeicherten Werte zurück.

Syntax: <code><ESC><ESC>ri</code> (27)(27)(114)(105)
--

Parameter: keine

Beispiel: Die aktuellen Ränder sollen auf die Einschaltländer zurückgesetzt werden.

`&%&%ri`

3.2.6. Textränder auf die Papierränder setzen

rm

Funktion: Dieser Befehl setzt die Textränder auf die absoluten Ränder des aktuell angewählten Papierformats. (Der linke Textrand, der obere Textrand, der untere Textrand und der rechte Textrand werden auf Null gesetzt.)

Syntax: <code><ESC><ESC>rm</code> (27)(27)(114)(109)
--

Parameter: keine

3.3. Textgestaltungsfunktionen

Funktion: Mit dem folgenden Befehl lassen sich im Drucker nachträglich Texte gestalten, die mittels einfachster Editoren eingegeben werden können.

Syntax:	<code><ESC><ESC>pf <n></code>	(27)(27)(112)(102)...
----------------	---	------------------------------

Parameter:

- `<n> = 0` : Gestaltungsfunktionen ausschalten
- `<n> = 1` : Blocksatz einschalten
- `<n> = 2` : Zentrieren um die aktuelle Cursorposition
- `<n> = 3` : Zentrieren zwischen den aktuellen Texträndern
- `<n> = 4` : Rechtsbündig drucken
- `<n> = 5` : Rechtsbündig drucken an der aktuellen Cursorposition

Beachte: Beim Blocksatz gibt es keine automatische Silbentrennung. Der Ausgleich der unterschiedlichen Zeilenlängen erfolgt in Bildpunkten zwischen den Wörtern. Das heißt, der Blocksatz sieht nur gut aus, wenn die Zeile von der Länge her bereits in etwa paßt.

Beispiel: Die Textgestaltungsfunktionen sollen anhand einiger Beispiele demonstriert werden.

```

&%%B0610&%%cx0650&%%r&%%4012
&%%27-0031600&%%cx1700&%%rr&%%4012
&%%27-00031600
&%%ty042Das ist Blocksatz:
&%%pf1

```


Beim Blocksatz gibt es keine automatische Silbentrennung. Der Ausgleich der unterschiedlichen Zeilenlängen erfolgt in Bildpunkten zwischen den Wörtern. Das heißt, der Blocksatz sieht nur gut aus, wenn die Zeile von der Länge her bereits in etwa paßt.

&%%pf0&%%pf3

Zentrieren

zwischen den Rändern

&%%pf0

Zentrieren an der aktuellen

Cursorposition:

&%%pf2

&%%cx09250&%%cx13757777

&%%cx092511&%%cx13756666

&%%cx0925222&%%cx1375555

&%%cx09253333&%%cx137544

&%%cx092544444&%%cx13753

&%%pf0

Die Positionen müssen mit einer Absolutpositionierung oder echten Tabulatoren angefahren werden!

Rechtsbündig:

&%%pf4

12.999.877,02 DM

1.278,09 DM

45,89 DM

23.893,01 DM

&%%pf0

Rechtsbündig an der aktuellen Position:

&%%pf5

&%%cx105023.56&%%cx152066666.99

&%%cx10501.78&%%cx152023.33

&%%cx105098234.18&%%cx1520222.66

&%%pf0

Das ist Blocksatz:

Beim Blocksatz gibt es keine automatische Silbentrennung. Der Ausgleich der unterschiedlichen Zeilenlängen erfolgt in Bildpunkten zwischen den Wörtern. Das heißt, der Blocksatz sieht nur gut aus, wenn die Zeile von der Länge her bereits in etwa paßt.

Zentrieren
zwischen den Rändern

Zentrieren an der aktuellen
Cursorposition:

0	77777
11	6666
222	555
3333	44
44444	3

Die Positionen müssen mit einer Absolutpositionierung oder echten Tabulatoren angefahren werden!

Rechtsbündig:

12.999.877,02	DM
1.278,09	DM
45,89	DM
23.893,01	DM

Rechtsbündig an der aktuellen Position:

23.56	66666.99
1.78	23.33
98234.18	222.66

3.3.1 Null mit Schrägstrich

zI

Funktion: Dieser Befehl wählt das Druckzeichen für die Null aus. Falls dieses Zeichen beim aktuellen Font keinen Schrägstrich besitzt, wird automatisch ein Schrägstrich über das Zeichen der Null gedruckt. (Diese Funktion ist bislang nur für Fix-Fonts optimiert).

Syntax: <ESC><ESC>zI <n> (27)(27)(122)(108)...
--

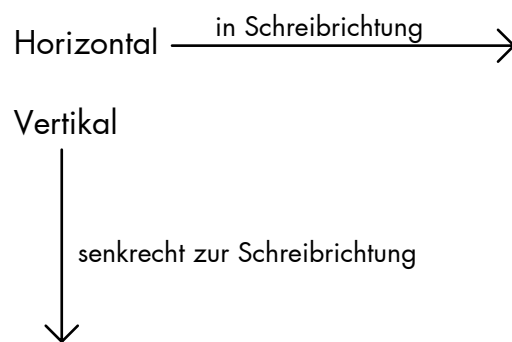
Parameter: <n> = 0 Null mit Schrägstrich
 <n> = 1 Null wird automatisch mit Schrägstrich gedruckt.

3.4. Tabulatorfunktionen

Mit den folgenden Befehlen können Tabulatoren gesetzt, angefahren oder insgesamt gelöscht werden. Beim Setzen muß vorher der Textcursor auf die gewünschte Position gestellt werden.

Die Textcursor - Positionen beziehen sich in x- und y - Richtung auf das in Abschnitt 1.3 beschriebene Koordinatensystem.

Die Richtungen horizontal und vertikal beziehen sich auf die aktuelle Schreibrichtung.



Es gibt zwei Gruppen von Tabulatoren, die unabhängig voneinander gesetzt werden können.

Die erste Gruppe von Tabulatoren gilt für die Schreibrichtungen Portrait und Portrait Kopf. Die zweite Gruppe von Tabulatoren gilt für die Schreibrichtungen Landscape und Landscape Kopf.

Horizontale und vertikale Tabulatoren können auf derselben Seite verwendet werden, sie arbeiten unabhängig voneinander. Das bedeutet, Tabulatoren werden für die beiden Gruppen von Schreibrichtungen unabhängig voneinander gesetzt und sind auch nur für diese Schreibrichtungen aktiv.

3.4.2. Vertikalen Tabulator an aktueller Cursorposition setzen

Funktion: Dieser Befehl setzt einen vertikalen Tabulator an der aktuellen Textcursor - Position.

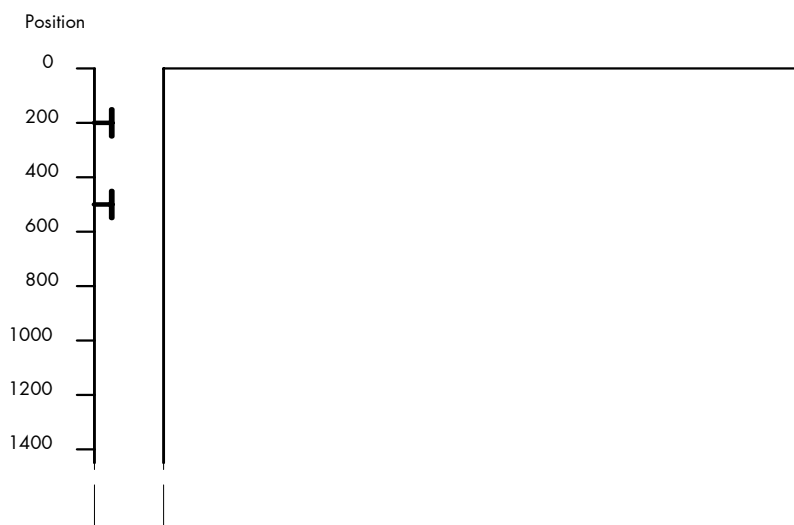
Syntax: `<ESC><ESC>tv` **(27)(27)(116)(118)**

Parameter: keine

Beachte: Bevor ein vertikaler Tabulator gesetzt werden kann, muß der Textcursor an die gewünschte Position gesetzt werden.

Beispiel: Es sollen vertikale Tabulatoren an die y-Positionen 200 und 500 gesetzt werden. (Die x-Koordinate spielt bei der Positionierung keine Rolle!)

```
&%%D1000 0200&%%tv&%%D1000 0500&%%tv
```



3.4.3. Horizontalen Tabulator ausführen

ph

Funktion: Nach Aufrufen dieses Befehls wird ein horizontaler Tabulator ausgeführt. Das heißt, der Textcursor wird an die Position des nächsten (in Schreibrichtung) horizontalen Tabulators gesetzt.

Syntax:	<code><ESC><ESC>ph</code>	(27)(27)(112)(104)
----------------	---------------------------------------	---------------------------

Parameter: keine

Beispiel: Es sollen horizontale Tabulatoren an den x-Positionen 800 und 1300 gesetzt werden. An beiden Positionen soll das Wort "IDOL" gedruckt werden.

```
&%%&D0800 1000&%%&%th&%%&D1300 1000&%%&%th
&%%&%phIDOL&%%&%phIDOL
```

IDOL

IDOL

3.4.4. Vertikalen Tabulator ausführen

Funktion: Nach Aufrufen dieses Befehls wird ein vertikaler Tabulator ausgeführt. Das heißt, der Textcursor wird an die Position des nächsten vertikalen Tabulators gesetzt (also unterhalb der aktuellen Cursorposition).

Syntax:	<code><ESC><ESC>pv</code>	(27)(27)(112)(118)
----------------	---------------------------------------	---------------------------

Parameter: keine

Beispiel: Es sollen vertikale Tabulatoren an den y-Positionen 2500 und 3000 gesetzt werden. An beiden Positionen soll das Wort "IDOL" gedruckt werden.

```
&%%D0700 2500&%%tv&%%D0700 3000&%%tv
&%%pvIDOL
&%%pvIDOL
```

```
IDOL
```

```
IDOL
```


3.4.5. Alle horizontalen und vertikalen Tabulatoren löschen



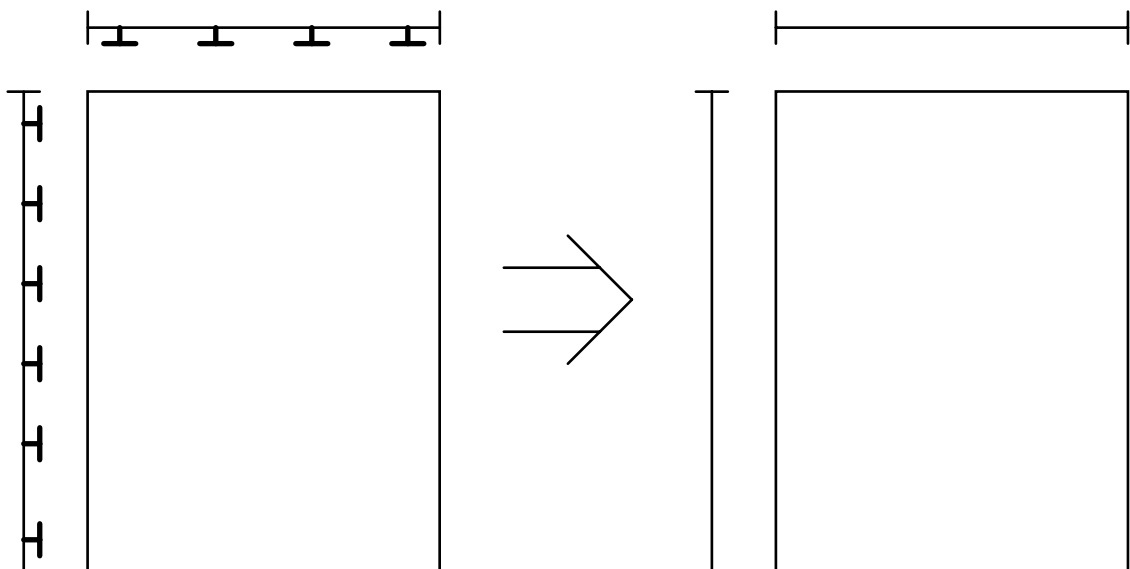
Funktion: Dieser Befehl löscht alle gesetzten Tabulatoren unabhängig von der Schreibrichtung.

Syntax: `<ESC><ESC>t1` **(27)(27)(116)(108)**

Parameter: keine

Beispiel: Es werden alle Tabulatoren gelöscht.

`&%&%t1`



3.5. Zeichenformatierung

Die folgenden Befehle der Funktionsgruppe FONT beeinflussen das Aussehen der Zeichen in Größe, Schreibrichtung, Zeichensatz etc.

3.5.1. Auswahl der Zeichentabelle



Funktion: Die überwiegende Zahl der Drucker - Schriftarten enthält einen umfassenden Zeichenvorrat nach gängigen Marktstandards oder internationalen Normen.

Die Auswahl der applikationsspezifischen Untermenge (=Tabelle) erfolgt durch diesen Befehl.

Syntax:	<code><ESC><ESC>L <nnn></code>	(27)(27)(76)...
----------------	--	------------------------

Parameter: `<nnn>` : dreistellige Kennzahl

<code><nnn> = 063</code>	: Frankreich	(ISO 25)
<code><nnn> = 410</code>	: Roman Extensions	
<code><nnn> = 420</code>	: Microsoft Publishing	
<code><nnn> = 430</code>	: Ventura Math	
<code><nnn> = 440</code>	: Math-8	
<code><nnn> = 558</code>	: Großbritannien	(ISO 4)
<code><nnn> = 610</code>	: US-Legal	
<code><nnn> = 710</code>	: DEC Linedraw	
<code><nnn> = 720</code>	: DEC Technical	
<code><nnn> = 730</code>	: Pi Font	

<nnn> = 760	: Italien	(ISO 15)
<nnn> = 763	: Frankreich	(ISO 69)
<nnn> = 766	: Deutsch	(ISO 21)
<nnn> = 769	: Norwegen	(ISO 61)
<nnn> = 804	: ASCII International	
<nnn> = 811	: Spanien	(ISO 17)
<nnn> = 812	: Portugal	(ISO 16)
<nnn> = 857	: Swedish names	(ISO 11)
<nnn> = 901	: IBM PC-1	
<nnn> = 902	: IBM PC-2	
<nnn> = 903	: IBM PC-1 (D/N)	
<nnn> = 904	: IBM PC-2 (D/N)	
<nnn> = 905	: Mazovia (Polen) IBM-PCII mit poln. Z.	
<nnn> = 906	: IBM-PCII mit slowakischen Zeichen	
<nnn> = 910	: DEC Multinational	
<nnn> = 920	: IGP Multinational (DIN 66303) = ECMA 94	
<nnn> = 926	: Latin 1 (Windows 3.1)	
<nnn> = 929	: Latin 9 (ISO 8859/15) [Euro]	
<nnn> = 930	: ASCII + Roman-8	
<nnn> = 940	: PC-850 (Latin 1)	
<nnn> = 950	: PC-852 (Latin 2)	
<nnn> = 955	: Latin 2 (ISO 8859/2)	
<nnn> = 956	: Latin 2 (Windows 3.1)	
<nnn> = 960	: Desk Top Symbol Set	
<nnn> = 970	: Ventura International Symbol Set	
<nnn> = 990	: XEROX ISO-Tabelle	

Hinweis: Die erwähnten Markennamen sind Warenzeichen bzw. eingetragene Warenzeichen der genannten Unternehmen.

Beispiel: Es soll die Zeichentabelle "ASCII International" eingeschaltet werden.

&%&%L 804

3.5.2. Auswahl der Schriftart

Funktion: Jeder MICROPLEX-Drucker verfügt über eine Anzahl von Standard-Schriftarten. Optional können weitere hinzukommen (siehe MICROPLEX - Fontkatalog).
Die Auswahl der Schriftart erfolgt über den Befehl:

<p>Syntax: <ESC><ESC>B <nnnn> (27)(27)(66)...</p>

Parameter: <nnnn> = : vierstellige Kennzahl, die z.B. für die im Drucker vorhandenen Schriftarten dem Statusblatt entnommen werden kann.

Beachte: Bei Vektor- und Outlinefonts sind u.U. noch die Parameter für skalierbare Schriften zu setzen.

Beispiel: Es soll die Schriftart Nummer 600, Courier 10 cpi, eingeschaltet werden. (cpi = Zeichenschrittweite; siehe Kapitel 3.5.4)

```
&%&%B 0600
```

Die im Drucker vorhandenen Schriftarten werden auf dem Statusblatt des Druckers folgendermaßen aufgelistet:

Zeichensätze:

```
01  - no Bank-Id-
" 0600 10.0 12.0  !" $ % & . / 0 1 2 ? A B C D E [ \ ] . f g h i . . .
" 4506  P      6.0  !" $ % & . / 0 1 2 ? A B C D E [ \ ] . f g h i . . .
" xxxx  x      xxx  !" $ % & . / 0 1 2 ? A B C D E [ \ ] . f g h i . . .
. . .
```

3.5.3. Bestimmung der Schreibrichtung

C

Funktion: Mit diesem Befehl sind Drehungen der Schreibrichtung in 90° - Schritten möglich, ohne daß die Textcursorposition oder andere Einstellungen verändert werden.

Syntax:	<code><ESC><ESC>C <n></code>	(27)(27)(67)...
----------------	--	------------------------

Parameter: `<n> = 0, 1, 2, 3` : einstellige Kennzahl

`<n> = 0` : Portrait (Hochformat)

`<n> = 1` : Landscape (Querformat)

`<n> = 2` : Portrait - Kopf

`<n> = 3` : Landscape - Kopf

Beachte: Die gesetzten Ränder bleiben erhalten, die logische Beziehung (oben, unten, rechts, links) wird neu zugeordnet.

Beispiel: Demonstration der vier Schreibrichtungen

```
&%%D10502760&%%E
```

```
&%%F&%%C0Richtung 0 ——— Portrait&%%E
```

```
&%%F&%%C3Richtung 3 - Landscape Kopf&%%E
```

```
&%%F&%%C2Richtung 2 - Portrait Kopf&%%E
```

```
&%%F&%%C1Richtung 1 ——— Landscape&%%E
```

```
&%%F&%%C0
```

```

Richtung 0 ——— Portrait
Richtung 3 - Landscape Kopf
Richtung 2 - Portrait Kopf
Richtung 1 ——— Landscape

```

3.5.4. Festlegen der horizontalen Zeichenschrittweite



Funktion: Dieser Befehl legt die aktuell zu verwendende Zeichenschrittweite in Bildpunkten (1 Bildpunkt = 1/300 Zoll) fest. Basis ist die Standard-Auflösung 300 dpi (Bildpunkte je Zoll). Die horizontale Schrittweite ist abhängig von der gewählten Anzahl Zeichen pro Zoll. Bei einer kleinen Anzahl von Zeichen pro Zoll wird die Zeichenschrittweite relativ groß sein. Die Formel zur Ermittlung der Zeichenschrittweite lautet:

$$\text{Zeichenschrittweite} = 300 \text{ dpi} / \text{Anzahl Zeichen pro Zoll}$$

Die folgende Tabelle zeigt einige gebräuchliche Werte:

Zeichen pro Zoll (characters per inch)	Zeichenschrittweite (Anzahl Bildpunkte = dots)
10 cpi	30 dots
12 cpi	25 dots
15 cpi	20 dots
20 cpi	15 dots

Syntax:	<code><ESC><ESC>tx <nnn></code>	(27)(27)(116)(120)...
----------------	---	------------------------------

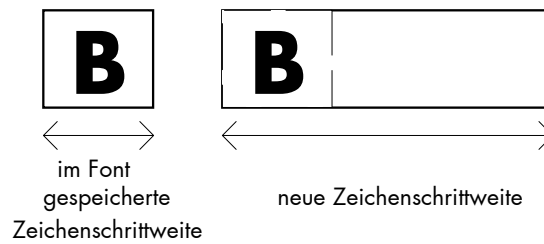
Parameter: `<nnn> = 000 ... 999` : Anzahl der Bildpunkte, dreistellig

`<nnn> = 000` : Die Schrittweiten werden auf die im Font gespeicherten Werte der eingeschalteten Schriftart gesetzt.

<nnn> = 001...999 : Die Schrittweite wird auf die angegebene Anzahl Bildpunkte gesetzt. Der im Font gespeicherte Wert wird ignoriert.

Beachte:

Das Zeichen wird linksbündig in den festgelegten Zeichenschritt positioniert.



Beispiel:

Das Wort "IDOL" soll zuerst mit der im Font festgelegten Zeichenschrittweite gedruckt werden. Dann soll das Wort "IDOL" mit einem festen Zeichenabstand von 100 Bildpunkten gedruckt werden.

Danach soll in der zur Schriftart gehörigen Zeichenschrittweite weitergedruckt werden.

```
IDOL&%%&%tx100IDOL&%%&%tx000IDOL
IDOLI D O L IDOL
```

3.5.5. Festlegen der vertikalen Zeichenschrittweite (Zeilenabstand)

Funktion: Dieser Befehl legt den Zeilenabstand in Bildpunkten (1 Bildpunkt = 1/300 Zoll) fest. Basis ist die Standard-Auflösung 300 dpi (Bildpunkte je Zoll). Die vertikale Zeichenschrittweite ist abhängig von der gewählten Anzahl Zeilen pro Zoll. Bei einer kleinen Anzahl von Zeilen pro Zoll wird der Zeilenabstand relativ groß sein. Die Formel zur Ermittlung der vertikalen Zeichenschrittweite lautet:

$$\text{Zeilenabstand} = 300 \text{ dpi} / \text{Anzahl Zeilen pro Zoll}$$

Die folgende Tabelle zeigt einige gebräuchliche Werte:

Zeilen pro Zoll (lines per inch)	Zeilenabstand (Anzahl Bildpunkte = dots)
4 lpi	75 dots
6 lpi	50 dots
10 lpi	30 dots
12 lpi	25 dots

Syntax:	<code><ESC><ESC>ty <nnn></code>	(27)(27)(116)(121)...
----------------	---	------------------------------

Parameter: `<nnn> = 000 ... 999` : Anzahl der Bildpunkte, dreistellig

`<nnn> = 000` : Die Zeilenabstände werden gemäß den im Font gespeicherten und zur jeweils eingeschalteten Schriftart gehörenden Werten festgelegt. Gibt es verschiedene Schriftarten in einer Zeile, wird der größte Zeilenabstandswert ausgeführt.

<nnn> = 001...999 : Der Zeilenabstand wird auf die angegebene Anzahl Bildpunkte gesetzt. Der im Font gespeicherte Wert wird ignoriert.

Beispiel:

Ein mehrzeiliger Text soll zwischenzeitlich mit dem Zeilenabstand $\frac{1}{4}$ Zoll gedruckt werden.

&%%&B0600

Der Zeilenabstand entspricht dem im Font gespeicherten Wert.

&%%&ty075 Dieser Text wird mit einem Zeilenabstand von $\frac{1}{4}$ Zoll gedruckt.

&%%&ty000Es wird wieder der im Font gespeicherte Wert aktiviert.

Der Zeilenabstand entspricht dem im Font gespeicherten Wert.

Dieser Text wird mit einem Zeilenabstand von $\frac{1}{4}$ Zoll gedruckt.

Es wird wieder der im Font gespeicherte Wert aktiviert.

3.5.6. Inverse Zeichendarstellung (Befehl 1)

Funktion: Der Befehl schaltet die Darstellungsform "invers" ein, d.h. alle nachfolgenden Zeichen werden weiß auf schwarzem Hintergrund gedruckt.

Mit demselben Befehl wird der Normalzustand wiederhergestellt:

Syntax: <ESC><ESC>*I*

(27)(27)(73)...

Parameter: keine

Beispiel: Das Wort "MICROPLEX" soll invers dargestellt werden.

```
&%&%IMICROPLEX&%&%I
```

MICROPLEX

3.5.7. Inverse Zeichendarstellung (Befehl 2)

pc

Funktion: Dieser Befehl wird verwendet für ein definiertes Ein- oder Ausschalten der "inversen" Zeichendarstellung.

Zum Vergleich: der alternative Befehl 1 (`<ESC><ESC>I` ; siehe vorigen Abschnitt) schaltet nur "hin und her", er basiert auf einem Toggle-Effekt.

Syntax:	<code><ESC><ESC>pc <n></code>	(27)(27)(112)(99)...
----------------	---	-----------------------------

Parameter:

<code><n> = 1</code>	: "inverse" Zeichendarstellung einschalten
<code><n> = 0</code>	: "inverse" Zeichendarstellung ausschalten

Beispiel: Das Wort "MICROPLEX" soll invers dargestellt werden.

`&%%&%pc 1MICROPLEX&%%&%pc 0`

MICROPLEX

3.5.8. Rasterhinterlegung (Befehl 1)



Funktion: Mit diesem Befehl kann der Text durch das aktuell eingeschaltete Raster hinterlegt werden.
Der gleiche Befehl schaltet die Funktion aus.

Syntax:	<code><ESC><ESC>i</code>	(27)(27)(105)
----------------	--------------------------------------	----------------------

Parameter: keine

Beispiel: Das Wort "MICROPLEX - IDOL" soll mit dem Raster Nr. 51 hinterlegt werden (siehe auch Kapitel 4.2.4 Raster).

`&%%13 51&%%iMICROPLEX - IDOL&%%i`

`MICROPLEX - IDOL`

3.5.10. Fettdruck (Befehl 1)

Funktion: Mit diesem Befehl wird zwischen den Betriebsarten "Fettdruck" und "Normaldruck" gewechselt.

Befindet sich der Drucker in der Betriebsart "Normaldruck", kann mit dem Befehl zur Betriebsart "Fettdruck" gewechselt werden. Befindet sich der Drucker in der Betriebsart "Fettdruck", wird zur Betriebsart "Normaldruck" gewechselt.

Der Fettschriftversatz ist im aktuellen Font festgelegt.

Syntax:	<code><ESC><ESC>f</code>	(27)(27)(102)
----------------	--------------------------------------	----------------------

Parameter: keine

Beispiel: Das Wort "IDOL" soll normal, fett und dann wieder normal gedruckt werden.

```
IDOL&%%&%f IDOL &%%&%fIDOL
```

```
IDOL IDOL IDOL
```

3.5.11. Fettdruck (Befehl 2)

bo

Funktion: Mit diesem Befehl wird der Fettdruck gezielt ein- oder ausgeschaltet, dies ist vorteilhaft im Falle von Reset.
Im Gegensatz dazu bewirkt der Befehl `<ESC><ESC>f` (siehe vorheriges Kapitel) immer nur einen Wechsel der Betriebsart.

Syntax:	<code><ESC><ESC>bo <n></code>	(27)(27)(98)(111)...
----------------	---	-----------------------------

Parameter:

<code><n> = 1</code>	: Fettdruck einschalten
<code><n> = 0</code>	: Fettdruck ausschalten

Beispiel: Das Wort "IDOL" soll normal, fett und dann wieder normal gedruckt werden.

IDOL&%%&%bo 1 IDOL &%%&%bo 0IDOL

IDOL **IDOL** IDOL

3.5.12. Unterstreichen (Befehl 1)

Funktion: Mit diesem Befehl wird zwischen den Betriebsarten "Unterstreichen" und "Nicht Unterstreichen" gewechselt.

Zu beachten ist, daß jede Cursorbewegung zum Unterstreichen führt, z.B. Leerzeichen, Relativpositionierung des Textcursors.

Befindet sich der Drucker in der Betriebsart "Unterstreichen", kann man mit dem Befehl zur Betriebsart "Nicht Unterstreichen" wechseln. Befindet sich der Drucker in der Betriebsart "Nicht Unterstreichen", wird zur Betriebsart "Unterstreichen" gewechselt.

Syntax: <ESC><ESC>u

(27)(27)(117)

Parameter: keine

Beispiel: Das Wort "IDOL" soll unterstrichen werden.

&%%&%uIDOL&%%&%u

IDOL

3.5.13. Unterstreichen (Befehl 2)**pu**

Funktion: Mit diesem Befehl wird das Unterstreichen gezielt ein- oder ausgeschaltet, dies ist vorteilhaft im Falle von Reset.
Im Gegensatz dazu bewirkt der Befehl <ESC><ESC>u (siehe vorheriges Kapitel) immer nur einen Wechsel der Betriebsart.

Syntax:	<ESC><ESC>pu <n>	(27)(27)(112)(117)...
----------------	------------------	------------------------------

Parameter: <n> = 1 : Unterstreichen einschalten
<n> = 0 : Unterstreichen ausschalten

Beispiel: Das Wort "IDOL" soll unterstrichen werden.

&%&%pu1IDOL&%&%pu0

IDOL

3.5.14. Sperrschrift (Befehl 1)

Funktion: Mit diesem Befehl wird die Betriebsart "Sperrschrift" eingestellt, wobei die halbe Leerschritt - Breite zwischen den einzelnen Buchstaben eingefügt wird.

Befindet sich der Drucker in dieser Betriebsart, kann man mit demselben Befehl den Mode "Sperrschrift" wieder verlassen.

Syntax:	<ESC><ESC> <i>s</i>	(27)(27)(115)
----------------	---------------------	---------------

Parameter: keine

Beispiel: Das Wort "IDOL" soll gesperrt gedruckt werden.

&%&%sIDOL&%&%s

I D O L

3.5.15. Sperrschrift (Befehl 2)**ps**

Funktion: Mit diesem Befehl wird der Mode Sperrschrift gezielt ein- oder ausgeschaltet. Im Gegensatz dazu bewirkt der Befehl `<ESC><ESC>s` (siehe vorheriges Kapitel) immer einen Wechsel der Betriebsart.

Syntax: <code><ESC><ESC>ps <n></code> (27)(27)(112)(115)...

Parameter: `<n> = 1` : Sperrschrift einschalten
 `<n> = 0` : Sperrschrift ausschalten

Beispiel: Das Wort "IDOL" soll gesperrt gedruckt werden.

```
&%&%ps1IDOL&%&%ps0
```

```
I D O L
```

3.5.16. Hochstellen (negative Halbzeilenschaltung)

zn

Funktion: Dieser Befehl löst einen halben Zeilenvorschub aus, und zwar von der aktuellen Zeile aus nach oben. Der folgende Text wird "hochgestellt". Das Hochstellen ist abhängig vom aktuell eingestellten Zeilenabstand (vergl. Abschnitt 3.5.5 "Festlegen der vertikalen Zeichenschrittweite"). Um normal weiterzuschreiben, wird der Befehl zum "Tiefstellen" verwendet (siehe nächster Befehl).

Syntax: <ESC><ESC>zn (27)(27)(122)(110)

Parameter: keine

Beispiel: Es soll "MPC²" gedruckt werden.

MPC&%%&%zn2&%%&%zp

MPC²

3.5.17. Tiefstellen (positive Halbzeilenschaltung)

zp

Funktion: Dieser Befehl löst einen halben Zeilenvorschub aus, und zwar von der aktuellen Zeile aus nach unten. Der folgende Text wird "tiefgestellt". Um normal weiterzuschreiben, wird der Befehl zum "Hochstellen" verwendet (siehe vorheriger Befehl). Auch das Tiefstellen ist abhängig vom aktuell eingestellten Zeilenabstand (vergl. Abschnitt 3.5.5 "Festlegen der vertikalen Zeichenschrittweite").

Syntax: <ESC><ESC>zp (27)(27)(122)(112)

Parameter: keine

Beispiel: Es soll "MPC₂" gedruckt werden.

```
MPC&%&%zp2&%&%zn
```

```
MPC2
```

3.6. Skalieren von Fonts

Fonts lassen sich in ihrer Größe verändern (skalieren). Dies geschieht mit Hilfe von IDOL-Befehlen.

Mit den folgend beschriebenen Befehlen lassen sich die Schriftarten beeinflussen. Die Befehle gelten nicht für alle Fontarten gleichermaßen. In der Beschreibung der Befehle ist vermerkt, für welche Fontart der Befehl gültig ist.

Zum Füllen der Grafik-Fonts (Stick-, Vektor- und Outlinefonts) wird das aktuelle Raster verwendet.

Beachte: Um die in den folgenden Kapiteln beschriebenen Befehle (50, 51, 52, 57, 60) nutzen zu können, muß vorher eine Grafik-Schriftart eingeschaltet worden sein.

Beim Abweichen von den Standard-Schreibrichtungen (0, 1, 2, 3) werden Funktionen wie "Unterstreichen", "Invers" etc. nicht korrekt ausgeführt.

Die Eigenschaften der Grafikfonts können unabhängig voneinander eingestellt werden.

3.6.1. Vergrößerung von Zeichen eines Pixelfonts

A

Funktion: Der Befehl leitet eine Vergrößerung der nachfolgenden Zeichen ein. Dieser Befehl wird für Pixelfonts verwendet.

<p>Syntax: <ESC><ESC>A <h> (27)(27)(65)...</p>
--

Parameter: b = 1, 2, 3, 4 : einstelliger Vergrößerungsfaktor in der Breite
 h = 1, 2, 3, 4 : einstelliger Vergrößerungsfaktor in der Höhe

Grundzustand: b = 1, h = 1

Beachte: Hierbei wird einfach jeder Bildpunkt eines Zeichens 2-, 3- oder 4mal ausgedruckt. Das bedeutet, daß es zu einer Stufenstruktur in den äußeren Konturen des Zeichens kommt.
 Die Vergrößerung wirkt auch auf die Steuerzeichen CR, LF, SP .
 Siehe auch Kapitel „Funktionsgruppe FONT“.

Beispiel: Das Wort "IDOL" soll doppelt breit und in dreifacher Höhe ausgegeben werden.
 Danach soll wieder normal weitergedruckt werden.

IDOL&%%&%A 2 3IDOL&%%&%A 1 1IDOL

IDOL **IDOL** IDOL

3.6.2. Vektorfont mit Breiten- und Höhenfaktor

Funktion: Mit diesem Befehl wird ein Vektorfont in der Breite und Höhe vergrößert.

Syntax: <ESC><ESC>50 <bbb> <hhh> (27)(27)(53)(48)...
--

Parameter: <bbb> = 001 bis 999 : Vergrößerungsfaktor Breite
 <hhh> = 001 bis 999 : Vergrößerungsfaktor Höhe

Beispiel: Das Wort IDOL soll mit dem Font Nr. 0050 geschrieben und in der Breite um den Faktor 5, in der Höhe um den Faktor 3 vergrößert werden.

```
&%&%B 0050 &%&%50 005 003IDOL
```

IDOL

3.6.3. Skalieren von Outlinefonts und Stickfonts

60

Funktion: Zum Skalieren von Outlinefonts bzw. Stickfonts werden zwei IDOL-Befehle zusammen verwendet.
Der erste Befehl (Funktionsname: 60) gibt den Größenwert in der Breite und Höhe an, der zweite Befehl (siehe nächste Seite) gibt die "Maßeinheit" an (z.B. "Punkt", dpi, ...).
Mit diesen Befehlen werden Outlinefonts und Stickfonts stufenlos in der Breite und Höhe vergrößert.

Syntax: <ESC><ESC>60 <bbb> <hhh> (27)(27)(54)(48)...
--

Parameter: <bbb> = 000 bis 999 : Größenwert Breite
 <hhh> = 000 bis 999 : Größenwert Höhe

Beispiel: Siehe nächste Seite; Befehl <ESC><ESC>57.

Beachte: Zusätzlich kann die Breite und die Höhe von Outlinefonts und Stickfonts auch separat skaliert werden. Dazu stehen zwei weitere Befehle zur Verfügung (Funktionsname 61 und 62, siehe Folgeseiten).

3.6.5. Skalieren der Höhe von Outlinefonts und Stickfonts

61

Funktion:

Mit diesem Befehl kann die Höhe von Skalierfonts separat eingestellt werden. (Vergleiche Skalieren von Outlinefonts und Stickfonts, Funktionsname 60).

Dieser Befehl (Funktionsname: 61) gibt den Größenwert in der Höhe vierstellig an, der zugehörige Befehl (Funktionsname 57) gibt die "Maßeinheit" an (z.B. "Punkt", dpi, ...).

Es wird das aktuelle Raster zum Füllen der Zeichen verwendet.

Syntax:`<ESC><ESC>61 <nnnn>`**(27)(27)(54)(49)...****Parameter:**`<nnnn> = 0000 bis 9999`

: Größenwert Höhe

3.6.7. Schreibrichtung mit Winkelangabe (Grafik-Fonts)

51

Funktion: Mit diesem Befehl kann ein Winkel zwischen Schreiblinie und X - Achse eingestellt werden. Die Schreibrichtung kann in Schritten von 1/10 Grad eingestellt werden. Dieser Befehl kann auf alle skalierbaren Fonts angewendet werden. Es wird das aktuelle Raster zum Füllen der Zeichen verwendet.

Syntax:	<ESC><ESC>51 <nnnn>	(27)(27)(53)(49)...
----------------	---------------------	----------------------------

Parameter: <nnnn> = 0000 bis 3600 : Winkel zwischen Schreiblinie und x-Achse in Einheiten von 1/10 Grad

Beachte: Wurde Text mit geneigter Schreibrichtung gedruckt, so kann der Textcursor nicht mittels CR an den Anfang der nächsten Zeile positioniert werden (CR wird nicht transformiert). Deshalb muß die Cursorpositionierung im Zusammenhang mit diesem Befehl (Texte mit geneigter Schreibrichtung) immer absolut oder relativ erfolgen.

Beispiel: Das Wort "I D O L" soll in einem Winkel von 30 Grad zur Schreiblinie gedruckt werden. Als Schrift wird der Font 2000 in 40 Punkt Größe verwendet.

```
&%%&B2000&%%&%60 040 040&%%&%57 072
&%%&%51 0300I D O L
```

IDOL

3.6.9. Beispiele

Auf dieser Seite sehen Sie einige Beispiele für verschiedene Kombinationen der in diesem Abschnitt beschriebenen Befehle.

Font 2000, Schreibrichtung 20 Grad, Zeichenneigung 30 Grad:

&%%B2000&%%60 040 040&%%57 072
&%%51 0200&%%52 0300MICROPLEX

MICROPLEX

Grafikschriften, wie Outlinefonts werden mit dem aktuellen Raster gefüllt. Hier ein Beispiel mit Raster 51:

&%%13 51
&%%B2000&%%60 040 040&%%57 072&%%51 0450 KOPIE

KOPIE

Werden Zeichen eines Grafikfonts mit einem Raster gefüllt, geht die Kantenstruktur teilweise verloren. Das führt dazu, dass diese Zeichen etwas "ausgefranst" aussehen. Grafik-Flächen, und nichts anderes sind Grafikzeichen, können durch einen IDOL-Befehl automatisch umrandet werden. Der Befehl zum automatischen Umranden von Grafikflächen ist im Kapitel 4 Funktionsgruppe GKS (Grafisches – Kern - System) ausführlich beschrieben.

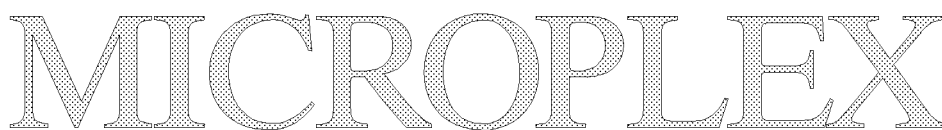
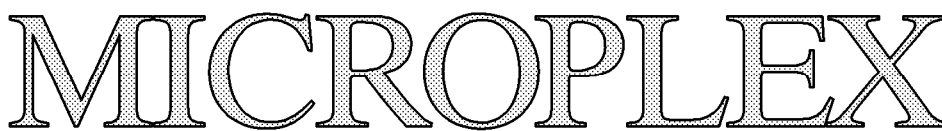
Beispiel ohne automatisches Umranden:

```
&B2000&60 060 060&57 072&13 51MICROPLEX
```



Beispiel mit automatischem Umranden in verschiedenen Strichstärken (IDOL-Befehl `<ESC><ESC>49`):

```
&49 01&B2000&60 060 060&57 072&13 51MICROPLEX
```



4. Funktionsgruppe GKS (Grafisches - Kern - System)

Für die Grafikfunktionen gilt ebenfalls das in Abschnitt 1.3 (Metrik) beschriebene Koordinatensystem.

Bei Grafik ist es im Gegensatz zum Textmode generell möglich und sinnvoll, den vollen Positionierungsbereich der Befehlsparameter zu nutzen, also auch außerhalb des "Blattes". (max. Parameter -9999 bis +9999)

Dies ist zum Beispiel bei der Programmierung von Kreisbögen notwendig, deren Kreismittelpunkte außerhalb des "Blattes" liegen.

Notwendige Winkelangaben (Kreisbogen, Kreisflächensegment, Polarkoordinaten) stützen sich auf das übliche mathematische Koordinatensystem (mathematisch positive Winkel = entgegen dem Uhrzeigersinn). Es sind nur positive Winkelangaben (in 1/10 Grad) zulässig.

Der Grafikkursor ist unabhängig vom Textcursor.

Beachte: Bei Linienzügen und Flächen wird der Grafikkursor automatisch zur letzten Koordinate verschoben. Alle anderen Grafikfunktionen beeinflussen den Grafikkursor nicht.

Nach dem Einschalten des Druckers oder durch einen Grafik-Reset (Befehl <ESC><ESC>43) gelten folgende Grundeinstellungen:

- Linienbreite = 4 = Punkte (4/300 Zoll)
- Linienmode = 0 = durchgezogen
- Auflösung = 300 = Punkte pro Zoll (dpi)
- X/Y-Offset = 0
- Raster = 0 = geschlossene Fläche (schwarz)
- Hinterlegungsart = 0 = replace
- Farbe = 0 = schwarz
- Grafikränder = max. = alle Grafikränder auf die Blattkanten
- Grafikkursor = X = Y = 0 = linke, obere Ecke des Blattes

4.1. Allgemeine Grafikfunktionen

4.1.1. Abbildungsfunktionen (Cursor und X/Y-Offsets setzen)

40

Funktion: Diese Funktionen bieten die Möglichkeit, Text- und Grafikkursor sowie den Nullpunkt des Grafik - Koordinatensystems aufeinander abzubilden.

<p>Syntax: <ESC><ESC>40 <a> (27)(27)(52)(48)...</p>

Parameter: <a> = 1, 2, 3 : Ziel („Wohin wird verschoben?“)
 = 1, 2, 3 : Objekt („Was wird verschoben?“)

Dabei bedeuten:

- 1 = Textcursor
- 2 = Grafikkursor
- 3 = Nullpunkt des Grafik - Koordinatensystems

Aus den Parametern ergeben sich folgende zulässige Kombinationen:

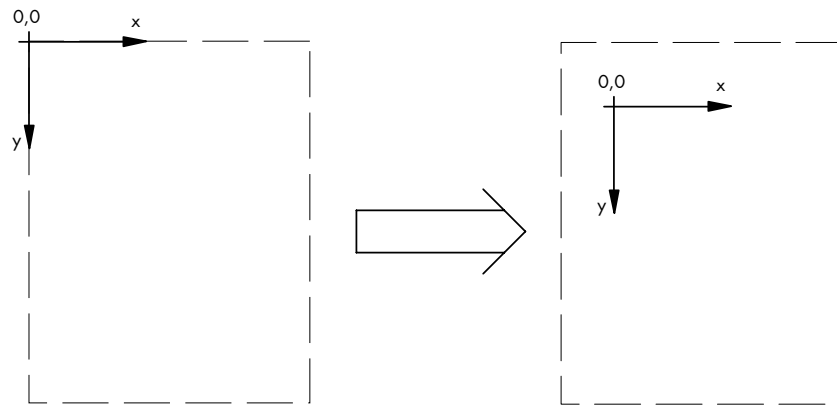
<ESC><ESC>40 1 2	: Grafikkursor auf die aktuelle Textcursor - Position
<ESC><ESC>40 2 1	: Textcursor auf die aktuelle Grafikkursor - Position
<ESC><ESC>40 1 3	: Grafik - Nullpunkt auf die aktuelle Textcursor - Pos.
<ESC><ESC>40 2 3	: Grafik - Nullpunkt auf die akt. Grafikkursor - Pos.
<ESC><ESC>40 3 2	: Grafikkursor auf den Grafik - Nullpunkt
<ESC><ESC>40 3 1	: Textcursor auf den Grafik - Nullpunkt

Beachte: Der Textcursor kann nicht außerhalb der Seite positioniert werden.

Beispiel:

Verschieben des Nullpunktes des Grafik - Koordinatensystems (= Quelle) auf die Position $x = y = 1000$ (= Ziel).

Der Textcursor wird zunächst auf diese Position gesetzt. Dann wird der Nullpunkt auf die Textcursor - Position abgebildet.



&%%&%D 1000 1000&%%&%40 1 3

Tip:

Um in elektronischen Formularen den Fließtext auf die Grafik (Linien, Raster ...) abzustimmen, ist die Kombination `<ESC><ESC>E` (Textcursor-Position speichern), `<ESC><ESC>F` (gespeicherte Textcursor-Position anfahren) und `<ESC><ESC>40 12` (Grafikcursor auf aktuelle Textcursor-Position) sinnvoll, wenn die folgenden Grafik-Befehle als relative Befehle verwendet werden.

Bei der Pflege der Formulare lassen sich einzelne Elemente (relative Koordinaten) oder das gesamte Formular (gespeicherte Textcursor-Position) verschieben.

4.1.2. Definieren der internen Grafikauflösung

Funktion: Mit diesem Befehl wird die interne Grafikauflösung des Druckers festgelegt. Alle relativen und absoluten Längenangaben bei Grafikfunktionen beziehen sich auf die eingestellte Auflösung.

<p>Syntax: <ESC><ESC>41 <nnnn> (27)(27)(52)(49) ...</p>

Parameter: <nnnn> = 0001 bis 9999 : Auflösung in Punkten pro Zoll

Die interne Grafikauflösung ist in der Grundeinstellung 300 Punkte pro Zoll, was den physikalischen Gegebenheiten der meisten Druckwerke entspricht.

Beachte: Änderungen der internen Grafikauflösung wirken nur auf Vektorgrafik und Grafikpositionierungen. Hingegen beziehen sich Textangaben und Bitmap-Daten nach wie vor auf eine Auflösung von 300 Punkten pro Zoll.

Beispiel: Die interne Auflösung soll auf 254 Punkte pro Zoll gesetzt werden. Dadurch entspricht ein Punkt 0,1 mm. Damit können Angaben in 0,1 mm Einheiten vorgenommen werden (z.B. Cursorpositionierung).

&%%41 254

Beispiel:

Die Grafik-Ränder sollen auf die Positionen $x = 500$, $y = 1300$ (oben, links) und $x = 2100$, $y = 3200$ (unten, rechts) gesetzt und durch dünne Linien (Raster 50) sichtbar gemacht werden. Der darin positionierte Kreis (Raster 56) und die mit Raster 51 hinterlegte Fläche werden an den Rändern abgeschnitten.

```
&%&%13 50
```

```
&%&%08 500 1300&%&%27 -2 1900&%&%27 1600 2
```

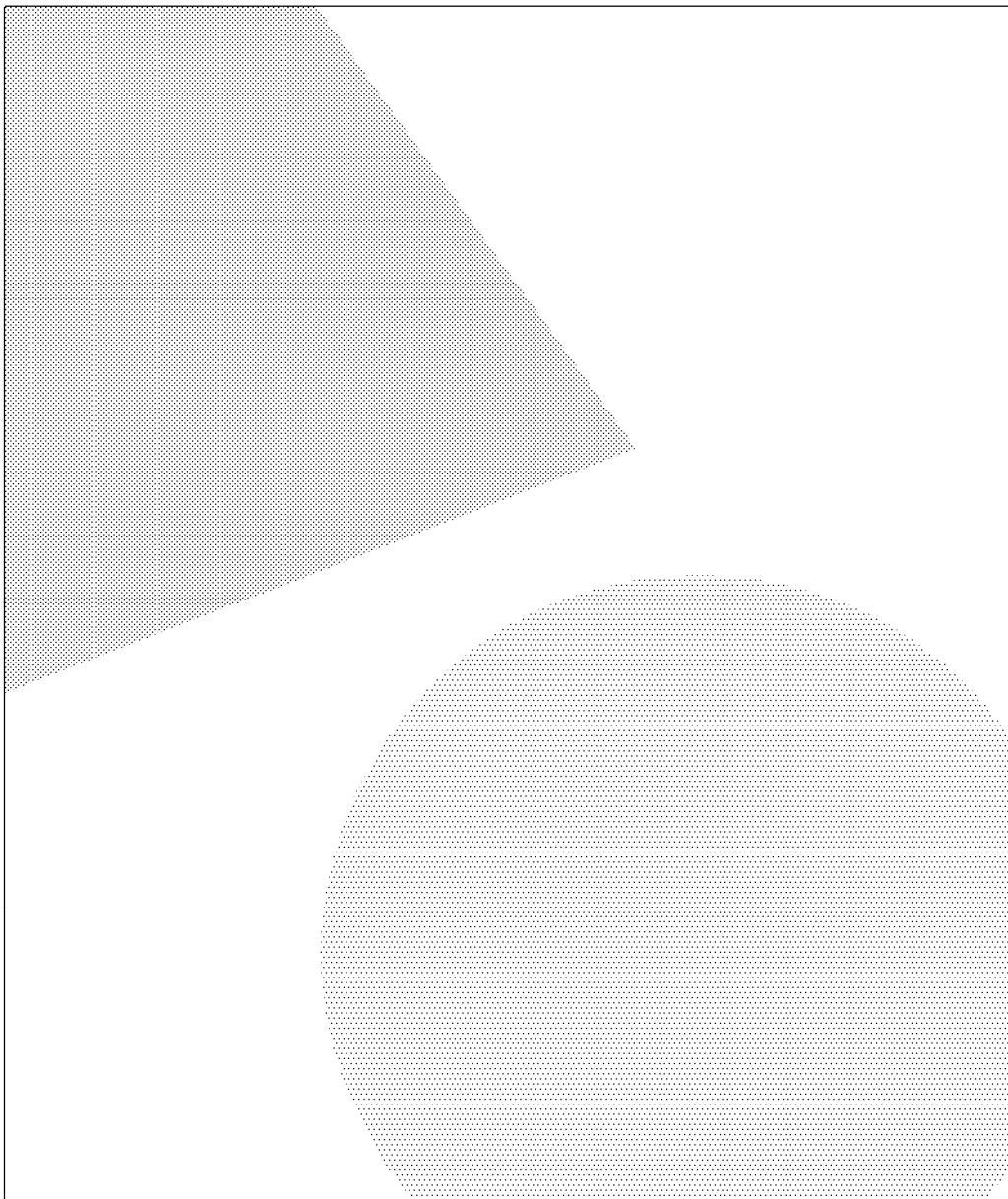
```
&%&%421&%&%424
```

```
&%&%08 2100 3200&%&%27 2 -1900&%&%27-1600 2
```

```
&%&%422&%&%423
```

```
&%&%13 56&%&%05 600 1600 2900
```

```
&%&%13 51&%&%01 3 200 200 1500 2000 -300 2700
```



4.2.2. Linienmode

11

Funktion: Mit diesem Befehl ist es möglich, periodische Unterbrechungen in Linienzügen zu programmieren. Die 8 Hexadezimal-Ziffern werden als eine Kette von Bits interpretiert. Ein auf "0" gesetztes Bit steht dabei für eine Unterbrechung.

Syntax: <ESC><ESC>11 <hhhhhhhh> (27)(27)(49)(49)...

















Parameter: <hhhhhhhh> = 00000000 bis FFFFFFFF : Kennzahl für die Art des Linienmusters.

■ ■ ■ ■ □ □ □ □ ■ ■ ■ ■ □ □ □ □ ■ ■ ■ ■ □ □ □ □ ■ ■ ■ ■ □ □ □ □ gewünscht Linienmuster
 F 0 F 0 F 0 F 0 Kennzahl

Die Kennzahl ist eine achtstellige ASCII-kodierte Hexadezimalzahl.

Beachte: Die Werte 00000000 und FFFFFFFF sind Sonderfälle und in der Wirkung identisch. In beiden Fällen werden geschlossene Linien gezeichnet.
 Im Einschaltzustand ist n = 00000000. Das heißt, der Linienmode ist auf geschlossene Linie programmiert.
 Der Linienmode beeinflusst die Grafikfunktionen Kreis, Linienzug und Kreisbogen.
 Es müssen immer 8 Hexadezimal-Ziffern angegeben werden. Voranstehende Nullen dürfen nicht weggelassen werden.

Ein HEX-Zeichen entspricht 4 aufeinander folgende Dots:

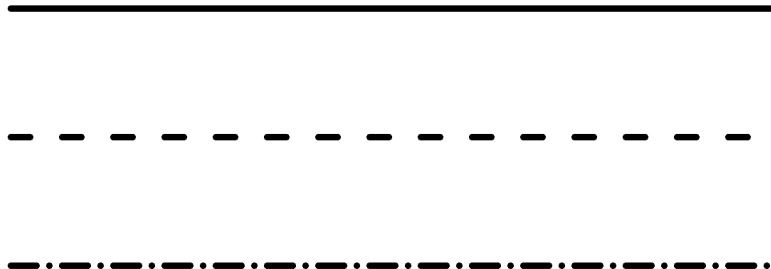
HEX 0: 	HEX 8: 
HEX 1: 	HEX 9: 
HEX 2: 	HEX A: 
HEX 3: 	HEX B: 
HEX 4: 	HEX C: 
HEX 5: 	HEX D: 
HEX 6: 	HEX E: 
HEX 7: 	HEX F: 

Beispiel:

10 Bildpunkte breite Linien mit verschiedenen Linienmustern

```

&%&%10 10
&%&%1100000000
&%&%03 2 600 2400 1800 2400
&%&%11F0F0F0F0
&%&%03 2 600 2600 1800 2600
&%&%11FAFAFAFA
&%&%03 2 600 2800 1800 2800
    
```



Beachte: Der Linienmode wird automatisch an die eingestellte Linienbreite angeglichen. Wird zum Beispiel eine unterbrochene Linie verbreitert, so vergrößern sich automatisch die Unterbrechungslängen.

Beim Einsatz der Befehle für Linienmode und Linienbreite ist die Reihenfolge entscheidend für das Ergebnis. Dazu einige Beispiele:

Einschaltzustand: Linienbreite = 4 dot, Linienmode: durchgezogene Linie

```
&%%23 1 1300 0000
```



Nun wird zuerst die Linienbreite verändert und danach der Linienmode.

Linienbreite: 15 dot

Linienmode: FAFAFAFA

```
&%%10 15
&%%11 FAFAFAFA
&%%23 1 1300 0000
```



Jetzt wird zuerst der Linienmode gesetzt und danach die Linienbreite.

Linienmode: FAFAFAFA

Linienbreite: 15 dot

```
&%%11 FAFAFAFA
&%%10 15
&%%23 1 1300 0000
```



4.2.3. Farbe von Grafikelementen

Funktion: Mit diesem Befehl wird die aktuelle Zeichenfarbe festgelegt. Nach dem Einschalten ist die Farbe "Schwarz" eingestellt. Die Farbe beeinflusst alle Grafikfunktionen.

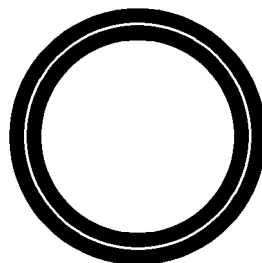
Syntax:	<code><ESC><ESC>12 <n></code>	(27)(27)(49)(50)...
----------------	---	----------------------------

Parameter:

<code><n> =</code>	0, 1	: Kennzahl für eingestellte Farbe
<code><n> =</code>	0	: Farbe Schwarz
<code><n> =</code>	1	: Farbe Weiß

Beispiel: Es sollen 2 schwarze, konzentrische Kreisringe dargestellt werden, indem man in eine schwarze Fläche eine weiße Fläche und einen weißen Kreis einzeichnet.

```
&%%05 0200 1250 2700
&%%12 1
&%%05 0150 1250 2700
&%%02 0175 1250 2700
&%%12 0
```



4.2.4. Raster

13

Funktion: Mit diesem Befehl wird das aktuelle Raster eingestellt. Nach dem Einschalten wird das Raster n = 00 geladen (geschlossene Fläche). Der Befehl "Raster" beeinflusst die Grafikfunktionen Fläche, Kreisfläche, Kreisflächensegment und Balken sowie die Funktion Texthinterlegung.

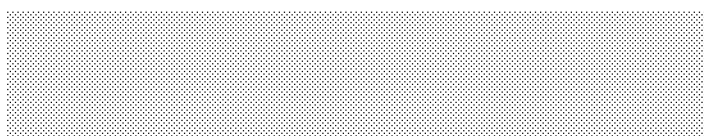
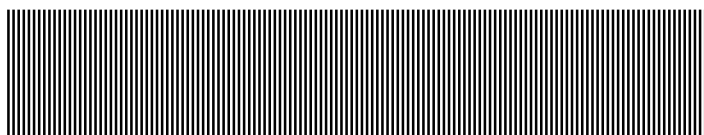
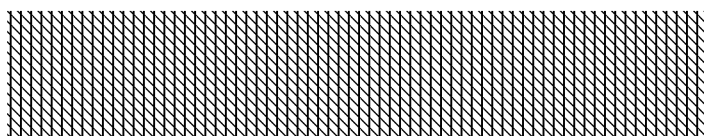
Die derzeit verfügbaren Raster sind im **Anhang II Rasterarten** dargestellt.

Syntax: <ESC><ESC>13 <nn> **(27)(27)(49)(51)...**

Parameter: <nn> = 00 bis 25, 50 bis 67 : Kennzahl für die Rasterart

Beispiel: Es soll ein Balken mit den Rastern 7, 25 und 51 gedruckt werden.

```
&%%13 07 &%%07 700 2300 1800 2500  
&%%13 20 &%%07 700 2600 1800 2800  
&%%13 51 &%%07 700 2900 1800 3100
```



4.2.5. Graustufen-Raster

59

Funktion: Mit diesem Befehl wird die aktuelle Graustufe gewählt. Zur Zeit sind 32 optimierte Graustufen-Raster zugeordnet. Der Befehl "Graustufen-Raster" beeinflusst die Grafikfunktionen Fläche, Kreisfläche, Kreisflächensegment und Balken sowie die Funktion Text hinterlegung.

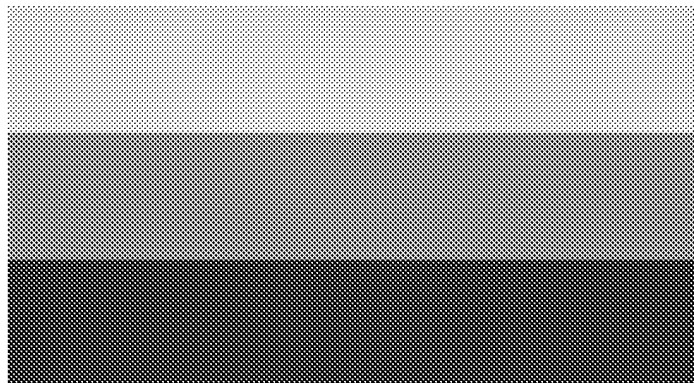
Syntax: <ESC><ESC>59 <nnn> (27)(27)(53)(57)...
--

Parameter: <nnn> = 000 bis 255 : Kennzahl für die Graustufe

Wählbar sind Graustufen zwischen „Weiß“ (kein Raster, Kennzahl 000) und „Schwarz“ (geschlossene Fläche, Kennzahl 255).

Beispiel: Zuerst wird ein hellgrauer Balken gedruckt (Graustufen-Kennzahl 050). Danach wird die Kennzahl 127 verwendet (entspricht 50 % Grau), der dritte Balken wird mit der Graustufe 200 gedruckt.

```
&%%59 050 &%%07 700 2500 1800 2700
&%%59 127 &%%07 700 2700 1800 2900
&%%59 200 &%%07 700 2900 1800 3100
```



4.2.6. Raster laden

16

Funktion: Mit diesem Befehl können selbst definierte Raster heruntergeladen werden (temporär). Die 8 Hexadezimal-Ziffern werden als eine Kette von Bits interpretiert. Ein auf "0" gesetztes Bit steht dabei für eine Unterbrechung.

Das so definierte Raster kann mit dem Befehl <ESC><ESC>13 nn mit nn = 99 aktiviert werden. Nach einem Aus- und Einschalten oder einem Reset ist das mit der Kennzahl 99 ladbare Raster wieder mit schwarz (0) definiert.

Syntax:	<ESC><ESC>16 <nn> <hhhhhhh>	(27)(27)(49)(54)...
----------------	-----------------------------	----------------------------

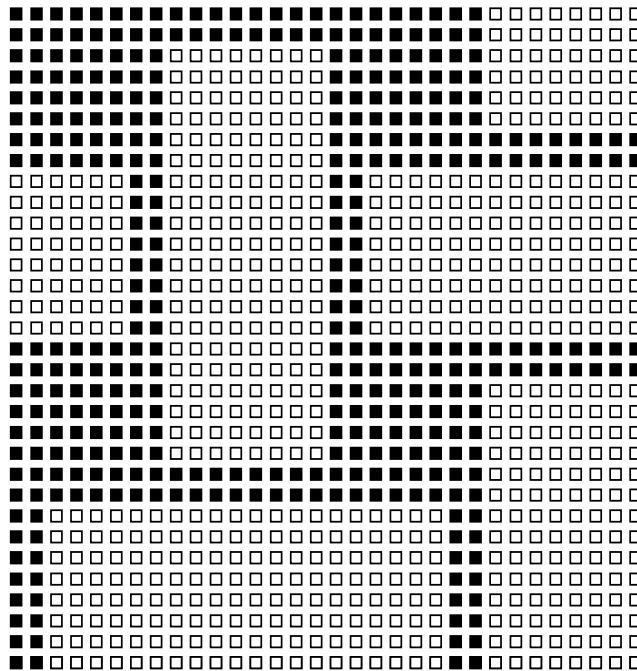
Parameter: <nn> = 01 bis 32 : Anzahl der Rasterzeilen (Höhe)
Zulässig sind 1,2,4,8,16 oder 32 Rasterzeilen.

<hhhhhhh> = 00000000 bis FFFFFFFF : Kennzahl für das selbst definierte Raster (Rastermuster).

Die Kennzahl ist eine achtstellige ASCII-kodierte Hexadezimalzahl, sie beschreibt eine 32 Bit breite Rasterzeile. Jedes HEX-Zeichen beschreibt also 4 aufeinander folgende Dots.

Beispiel: Es soll eine Dreieckfläche mit einem selbst definierten Raster gedruckt werden.

gewünschtes Rastermuster:



Kennzahlen:

```

FFF FFF F00
FFF FFF F00
FF00 FF00
FF00 FF00
FF00 FF00
FF00 FF00
FF00 FF00
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
0300 C000
FF00 FF00
FF00 FF00
FF00 FF00
FF00 FF00
FF00 FF00
FF00 FF00
C000 0300
C000 0300
C000 0300
C000 0300
C000 0300
C000 0300
C000 0300

```

Hinweis:

Zwischen den einzelnen Parametern eines IDOL - Befehls können zur Verbesserung der Übersichtlichkeit die Zeichen "CR" und "LF" verwendet werden.

```

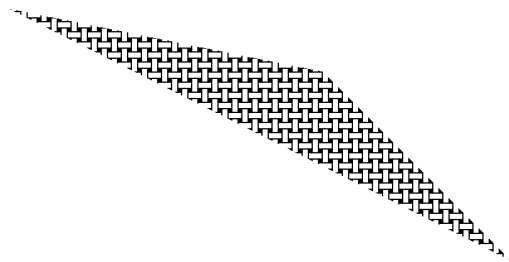
&&&16 32 <CR><LF>
FFFFFF00 FFFFFFF00 FF00FF00 FF00FF00<CR><LF>
FF00FF00 FF00FF00 FF00FFFF FF00FFFF<CR><LF>
0300C000 0300C000 0300C000 0300C000<CR><LF>
0300C000 0300C000 0300C000 0300C000<CR><LF>
FF00FFFF FF00FFFF FF00FF00 FF00FF00<CR><LF>
FF00FF00 FF00FF00 FFFFFFF0 FFFFFFF0<CR><LF>
C0000300 C0000300 C0000300 C0000300<CR><LF>
C0000300 C0000300 C0000300 C0000300<CR><LF>

```

```

&&&13 99
&&&21 3 100 50 500 100 300 300

```



4.2.7. Hinterlegungsmode

14

Funktion: Der Hinterlegungsmode gibt an, in welcher Form Druckinformationen in den Bildspeicher eingetragen werden. Mit diesem Befehl wird der aktuelle Speichermodus eingestellt.

Syntax:	<code><ESC><ESC>14 <n></code>	(27)(27)(49)(52)...
----------------	---	----------------------------

Parameter:

<code><n> =</code>	0, 1, 2, 3	: Kennzahl für die Hinterlegungsart
<code><n> =</code>	0	: ersetzen (replace)
<code><n> =</code>	1	: überschreiben (overstrike)
<code><n> =</code>	2	: invertieren (complement)
<code><n> =</code>	3	: löschen (erase)

Der Hinterlegungsmode gibt an, in welcher Form Druckinformationen in den Bildspeicher eingetragen werden. Zusammen mit der Funktion "Farbe" ergeben sich folgende Varianten:

- Schwarz - replace : Das Raster wird komplett abgelegt, der alte Inhalt wird überschrieben.
- Schwarz - overstrike : Nur die schwarzen Teile des Rasters werden abgelegt.
- Schwarz - complement : An den schwarzen Teilen des Rasters wird der Inhalt des Bildspeichers invertiert.
- Schwarz - erase : An den schwarzen Teilen des Rasters wird der Bildspeicher gelöscht.

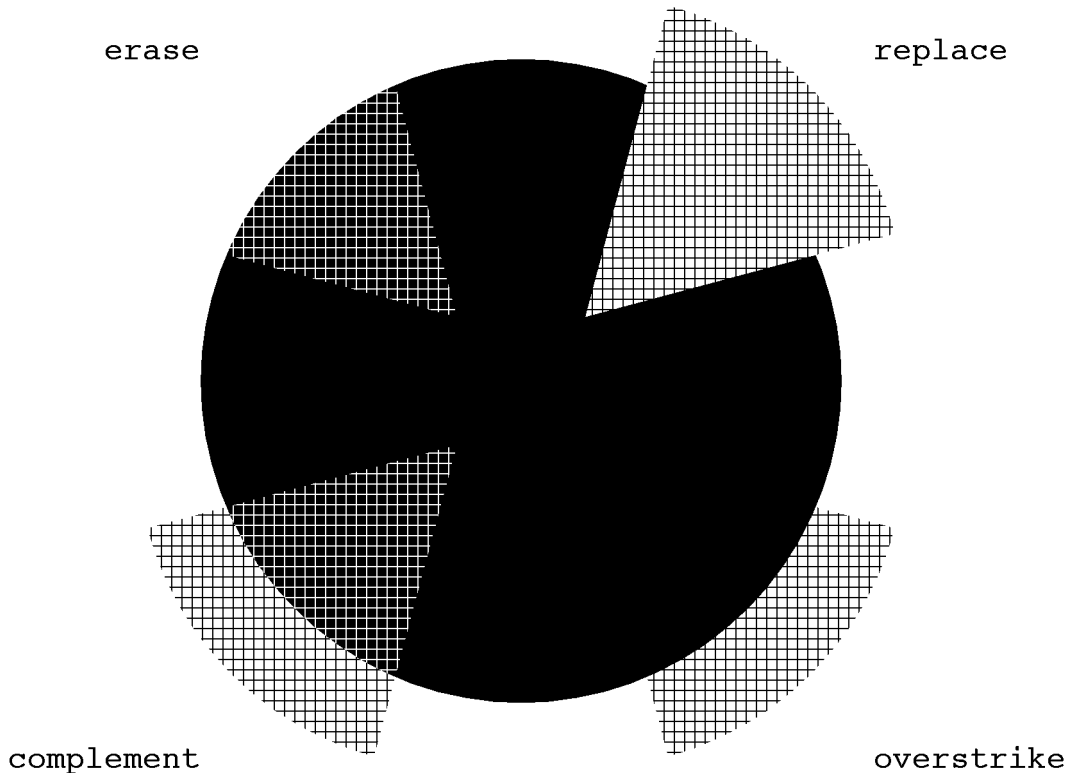
- Weiß - replace : Das Raster wird invertiert und komplett abgelegt.
- Weiß - overstrike : wie Schwarz-erase
- Weiß - complement : wie Schwarz-complement
- Weiß - erase : wie Schwarz-overstrike

Beispiel:

Das Beispiel zeigt eine Kreisfläche, der Kreisflächensegmente in verschiedenen Hinterlegungsmodi überlagert sind.

```

&%%13 0
&%%05 500 1250 2400
&%%1305
&%%140 &%%06 150 750 500 1350 2300
&%%141 &%%06 2850 3450 0500 1350 2500
&%%142 &%%06 1950 2550 0500 1150 2500
&%%143 &%%06 1050 1650 0500 1150 2300
    
```



4.2.8. Hinterlegungsmodus für Zeichen und Bitmaps

19

Funktion: Der Hinterlegungsmodus für Bitmaps gibt an, in welcher Form die Bitmap-Daten mit den schon vorhandenen Informationen im Bildspeicher verknüpft werden. Die Wirkungsweise dieses Befehls ist vergleichbar mit dem Befehl <ESC><ESC>14.

<p>Syntax: <ESC><ESC>19 <nnnn> (27)(27)(49)(57)...</p>
--

Parameter:

<nnnn> =	0, 1, 2, 3, 4	: Vierstellige Kennzahl für die Hinterlegungsart
<nnnn> =	0	: ersetzen (replace)
<nnnn> =	1	: überschreiben (overstrike) (Standard/default, z.B. nach Grafik-Reset)
<nnnn> =	2	: invertieren (complement)
<nnnn> =	3	: löschen (erase)
<nnnn> =	4	: weiß - replace

Zusammen mit der Funktion "Farbe" ergeben sich folgende Varianten des Hinterlegungsmodus für Bitmaps:

- Schwarz - replace : Die Bitmap wird komplett abgelegt, der alte Inhalt wird überschrieben.
- Schwarz - overstrike : Nur die schwarzen Teile der Bitmap werden abgelegt.
- Schwarz - complement : An den schwarzen Teilen der Bitmap wird der Inhalt des Bildspeichers invertiert.
- Schwarz - erase : An den schwarzen Teilen der Bitmap wird der Bildspeicher gelöscht.

4.2.9. Automatisches Umranden von Grafik-Flächen

Funktion: Mit diesem Befehl wird das automatische Umranden von Grafik-Flächen ein- bzw. ausgeschaltet.

Syntax: <ESC><ESC>49 <nn> (27)(27)(52)(57)...

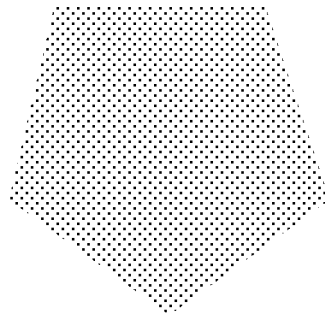
Parameter: <nn> = 1 : Automatisches Umranden einschalten.
<nn> = 0 : Automatisches Umranden ausschalten.

Beachte: Die Grafik-Flächen werden mit Linien im aktuellen Linienmode und der aktuellen Linienbreite umrandet.

Beispiel: Eine Fünfeckfläche, gefüllt mit Raster 9, wird mit verschiedenen Linien automatisch umrandet.

1. Ohne Umrandung

```
&%%%13 09&%%%01 005 1080 2700 1410 2700 1510
3000 1260 3180 1010 3000
```



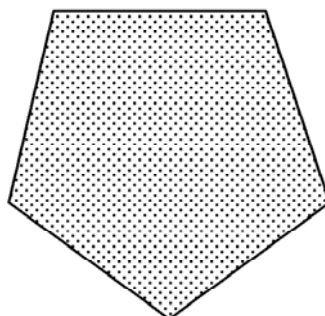
2. Umrandung mit durchgezogener Linie, 4 dot breit.

&%%10 04

&%%11 FFFFFFFF

&%%49 1

&%%13 09&%%01 005 1080 950 1410 950 1510 1250
1210 1430 1010 1250



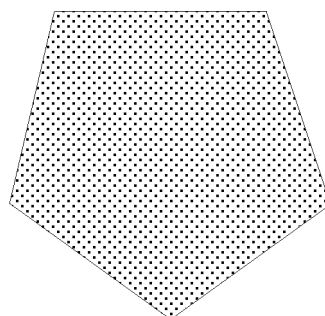
3. Umrandung mit durchgezogener Linie, 1 dot breit.

&%%10 01

&%%11 FFFFFFFF

&%%49 1

&%%13 09 &%%01 005 1080 2150 1410 2150 1510
2450 1260 2630 1010 2450



4. Umrandung mit gestrichelter Linie, 4 dot breit.

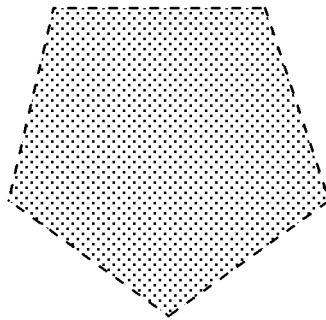
&%%%10 04

&%%%11 FOFOF0FO

&%%%49 1

&%%%13 09&%%%01 005 1080 950 1410 950 1510 1250

1210 1430 1010 1250



4.3. Grafik mit absoluter Positionierung

Für die Grafikfunktionen mit absoluten Parametern gilt das schon beschriebene Koordinatensystem in Einheiten von Bildpunkten (1 Bildpunkt = 1/300 Zoll).

Per Definition geht die positive y-Achse von oben nach unten.

(Vergleiche Abschnitt 1.3 Metrik)

4.3.1. Positionieren des Grafik-Cursors

Funktion: Mit diesem Befehl wird der Grafikcursor absolut positioniert. Die Position wird in absoluten Koordinaten als Parameter angegeben.

Syntax: <ESC><ESC>08 <xxxx> <yyyy> (27)(27)(48)(56)...
--

Parameter: <xxxx> = -9999 bis 9999 : x - Koordinate des Grafik-Cursors
 <yyyy> = -9999 bis 9999 : y - Koordinate des Grafik-Cursors

Beachte: Der Grafik-Cursor ist vom Text - Cursor völlig getrennt zu setzen.

Beispiel: Der Grafik-Cursor wird auf die obere linke Ecke gesetzt.

&%%08 0000 0000

4.3.2. Positionieren des Grafik-Cursors in x-Richtung

44

Funktion: Mit diesem Befehl wird der Grafikcursor in x-Richtung absolut positioniert. Die Position wird in absoluten Koordinaten als Parameter angegeben. Die y-Position des Grafikcursors bleibt dabei erhalten.

Syntax: <ESC><ESC>44 <xxxx> (27)(27)(52)(52)...

Parameter: <xxxx> = -9999 bis 9999 : x - Koordinate des Grafik-Cursors

Beispiel: Der Grafikcursor soll auf die x - Koordinate 1200 gesetzt werden. Die aktuelle y - Koordinate soll beibehalten werden.

&%&%44 1200

4.3.3. Positionieren des Grafik-Cursors in y-Richtung

45

Funktion: Mit diesem Befehl wird der Grafikcursor in y-Richtung absolut positioniert. Die Position wird in absoluten Koordinaten als Parameter angegeben. Die x-Position des Grafikcursors bleibt dabei erhalten.

<p>Syntax: <ESC><ESC>45 <yyyy> (27)(27)(52)(53)...</p>
--

Parameter: <yyyy> = -9999 bis 9999 : y - Koordinate des Grafik-Cursors

Beispiel: Der Grafikcursor soll auf die y - Koordinate 1200 gesetzt werden. Die aktuelle x - Koordinate soll beibehalten werden.

&%%&%45 1200

4.3.4. Fläche

01

Funktion: Mit diesem Befehl wird eine absolut positionierte Fläche gezeichnet.
Die Punkte werden durch Linien in der eingegebenen Reihenfolge miteinander verbunden. Dabei ist es möglich, daß sich Verbindungslinien überschneiden. Die eingeschlossene Fläche wird mit der aktuellen Zeichenfarbe ausgefüllt.

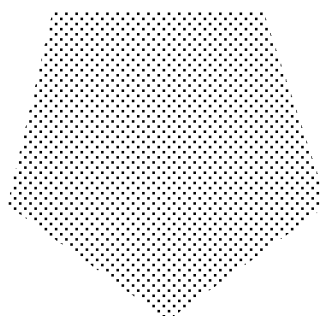
Syntax: `<ESC><ESC>01 <nnn> <xxxx1><yyyy1> ...
... <xxxxn><yyyyn>`

(27)(27)(48)(49)...

Parameter: `<nnn>` = 001 bis 255 : Anzahl der Koordinatenpaare
`<xxxxi>` = -9999 bis 9999 : X - Koordinaten
`<yyyyi>` = -9999 bis 9999 : Y - Koordinaten

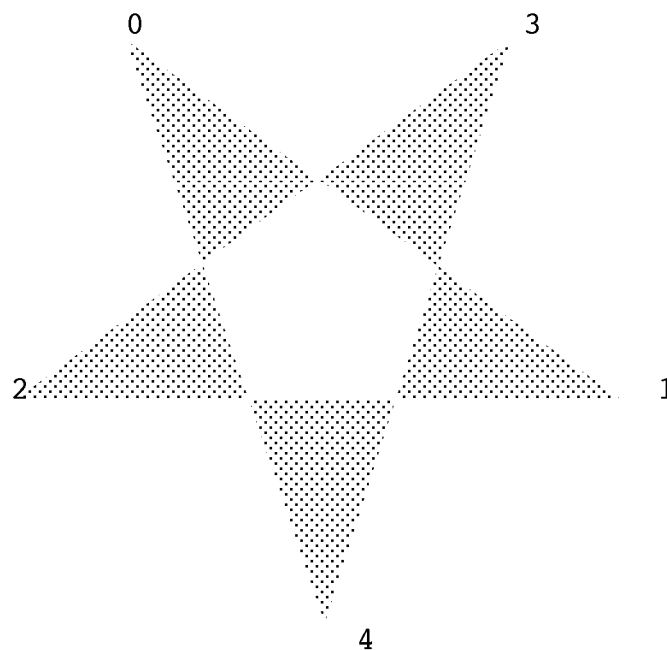
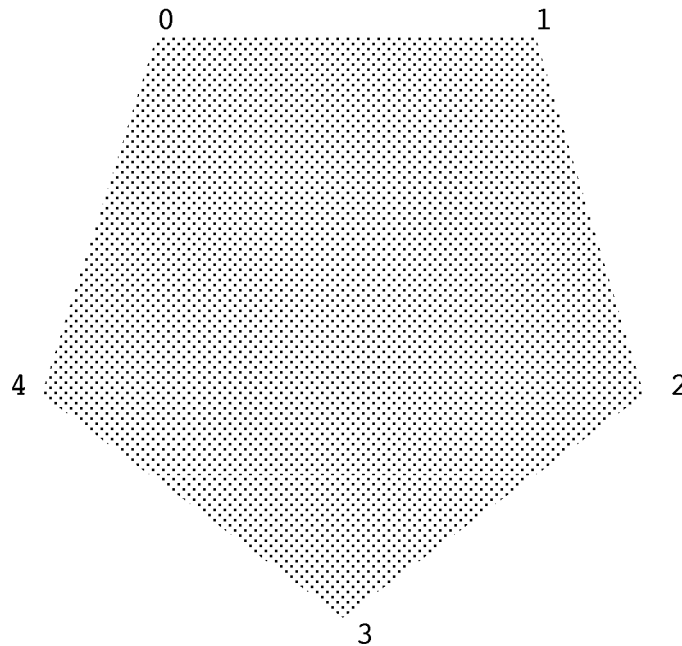
Beispiel: Zeichnen einer Fünfeckfläche mit Raster 9
(siehe auch Kapitel 4.2.4 Raster).

```
&%%&13 09&%%&01 005 1080 2700 1410 2700 1510  
3000 1260 3180 1010 3000
```



Beachte:

Beim Flächenfüllen werden die angegebenen Koordinaten quasi durch unsichtbare Linien verbunden; danach wird von oben her Zeile für Zeile mit dem aktuellen Raster gefüllt. Die unsichtbaren Linien schalten den Füllvorgang ein und aus. Die Reihenfolge der Koordinatenpaare hat Einfluß auf das Aussehen der Fläche (siehe Beispiel).

Beispiel:

4.3.5. Kreis

02

Funktion: Mit diesem Befehl wird ein absolut positionierter Kreis gezeichnet. Vorher vereinbarte Parameter wie Linienbreite, Farbe, Linienmode und Speichermode werden berücksichtigt.

Syntax: <ESC><ESC>02 <rrrr> <xxxx> <yyyy>

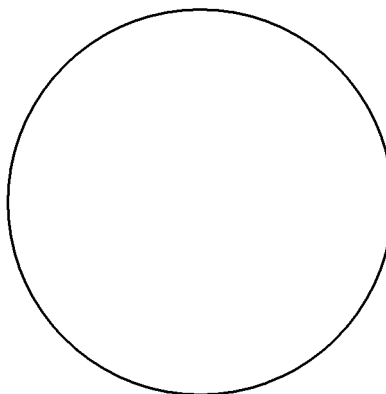
(27)(27)(48)(50)...

Parameter:

r =	0000 bis 9999	: Radius
x =	-9999 bis 9999	: X - Koordinate des Kreismittelpunktes
y =	-9999 bis 9999	: Y - Koordinate des Kreismittelpunktes

Beispiel: Es soll ein Kreis mit einem Radius von 300 Bildpunkten um den Punkt mit den Koordinaten x = 1250, y = 2700 gezeichnet werden.

&%&%02 0300 1250 2700



4.3.6. Linienzug

Funktion: Mit diesem Befehl wird ein Linienzug gezeichnet. Die Punkte (x_n, y_n) werden durch Linien in der eingegebenen Reihenfolge miteinander verbunden. Vorher vereinbarte Parameter wie Linienbreite, Farbe, Linienmode und Speichermodus werden berücksichtigt.

Syntax:

```
<ESC><ESC>O3 <nnnn> <xxxx1> <YYYY1>
... <xxxxn> <YYYYn>
```

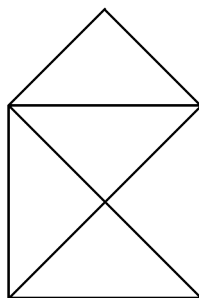
(27)(27)(48)(51)...

Parameter:

<nnnn> =	0001 bis 9999	: Anzahl der Koordinatenpaare
<xxxx _i > =	-9999 bis 9999	: X - Koordinaten
<yyyy _i > =	-9999 bis 9999	: Y - Koordinaten

Beispiel: Linienzug zwischen 9 Punkten

```
&%&%O3 0009 1100 3100 1100 2800 1400 2800 1400
3100 1100 3100 1400 2800 1250 2650 1100 2800 1400
3100
```



4.3.7. Kreisbogen

04

Funktion: Der Befehl bewirkt das Zeichnen eines Kreisbogens mit dem Kreismittelpunkt (x, y) und dem Radius r. Vorher vereinbarte Parameter wie Linienbreite, Farbe, Linienmode und Speichermodus werden berücksichtigt. Start- und Endwinkel beziehen sich jeweils auf das IDOL-Koordinatensystem.

Syntax:	<pre><ESC><ESC>O4 <nnnn1> <nnnn2> <rrrr> <xxxx> <yyyy> (27)(27)(48)(52)...</pre>
----------------	---

Parameter:	<table> <tr> <td><nnnn1></td> <td>= 0000 bis 3600</td> <td>: Bogenbeginn in Einheiten von 1/10 Grad</td> </tr> <tr> <td><nnnn2></td> <td>= 0000 bis 3600</td> <td>: Bogenende in Einheiten von 1/10 Grad</td> </tr> <tr> <td><rrrr></td> <td>= 0000 bis 9999</td> <td>: Radius</td> </tr> <tr> <td><xxxx></td> <td>= -9999 bis 9999</td> <td>: x - Koordinate des Kreismittelpunktes</td> </tr> <tr> <td><yyyy></td> <td>= -9999 bis 9999</td> <td>: y - Koordinate des Kreismittelpunktes</td> </tr> </table>	<nnnn1>	= 0000 bis 3600	: Bogenbeginn in Einheiten von 1/10 Grad	<nnnn2>	= 0000 bis 3600	: Bogenende in Einheiten von 1/10 Grad	<rrrr>	= 0000 bis 9999	: Radius	<xxxx>	= -9999 bis 9999	: x - Koordinate des Kreismittelpunktes	<yyyy>	= -9999 bis 9999	: y - Koordinate des Kreismittelpunktes
<nnnn1>	= 0000 bis 3600	: Bogenbeginn in Einheiten von 1/10 Grad														
<nnnn2>	= 0000 bis 3600	: Bogenende in Einheiten von 1/10 Grad														
<rrrr>	= 0000 bis 9999	: Radius														
<xxxx>	= -9999 bis 9999	: x - Koordinate des Kreismittelpunktes														
<yyyy>	= -9999 bis 9999	: y - Koordinate des Kreismittelpunktes														

Beispiel: Es soll ein kleiner Halbkreis (r = 100; nach oben offen) an einen größeren Halbkreis (r = 200; nach unten offen) angehängt werden.

```
&%&%04 1800 0000 0100 1000 3100&%&%04 0000 1800
0200 1300 3100
```



4.3.8. Kreisfläche

Funktion: Es wird eine Kreisfläche mit dem Radius r und dem Kreismittelpunkt (x, y) ausgegeben.
Vorher vereinbarte Parameter wie Farbe, Rastermode, automatisches Umranden und Speichermodi werden berücksichtigt.

Syntax: <ESC><ESC>05 <rrrr> <xxxx> <yyyy>

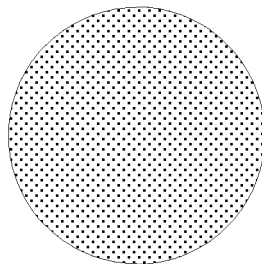
(27)(27)(48)(53)...

Parameter:

<rrrr>	=	0000 bis 9999	: Radius
<xxxx>	=	-9999 bis 9999	: x - Koordinate des Kreismittelpunktes
<yyyy>	=	-9999 bis 9999	: y - Koordinate des Kreismittelpunktes

Beispiel: Kreisfläche mit 200 Bildpunkten Radius (Raster 9, automatisch umrandet)

```
&%&%49 01&%&%13 09
&%&%05 0200 1250 2600
```



4.3.9. Kreisflächensegment

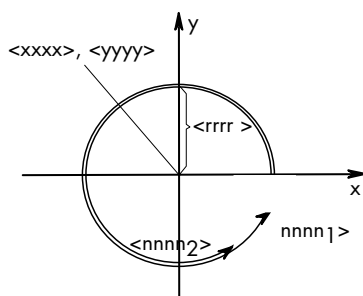
06

Funktion: Der Befehl bewirkt das Zeichnen eines Kreisflächensegmentes mit dem Kreismittelpunkt (x, y) und dem Radius r. Sind Bogenbeginn und Bogenende gleich, wird ein Vollkreis gezeichnet. Vorher vereinbarte Parameter wie Farbe, Rastermode und Speichermode werden berücksichtigt.

Syntax: <ESC><ESC>O6 <nnnn1> <nnnn2> <rrrr> <xxxx> <yyyy>
(27)(27)(48)(54)...

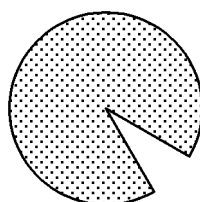
Parameter:

<nnnn1>	= 0000 bis 3600	: Bogenbeginn in Einheiten von 1/10 Grad
<nnnn2>	= 0000 bis 3600	: Bogenende in Einheiten von 1/10 Grad
<rrrr>	= 0000 bis 9999	: Radius
<xxxx>	= -9999 bis 9999	: x - Koordinate des Kreismittelpunktes
<yyyy>	= -9999 bis 9999	: y - Koordinate des Kreismittelpunktes



Beispiel: Kreisflächensegment mit 30 Grad "Tortenausschnitt" und 300 Grad Verdrehung. (Raster 9, automatisch umrandet)

```
&%&%49 01&%&%13 09
&%&%06 3300 3000 0150 1250 3050
```



4.3.10. Balken

07

Funktion: Mit diesem Befehl wird ein Balken gezeichnet. Der Balken wird durch die Punkte x_1, y_1 (= oberer linker Eckpunkt) und x_2, y_2 (= unterer rechter Eckpunkt) definiert.
Vorher vereinbarte Parameter wie Farbe, Rastermode und Speichermoder werden berücksichtigt.

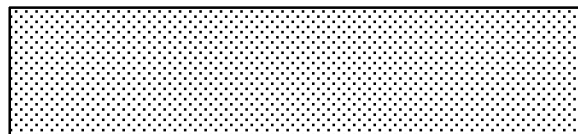
Syntax: `<ESC><ESC>O7 <xxxx1> <yyyy1> <xxxx2> <yyyy2>`
(27)(27)(48)(55)...

Parameter:

<code><xxxx₁></code> =	- 9999 bis 9999	: x - Koordinate des oberen linken Eckpunktes
<code><yyyy₁></code> =	- 9999 bis 9999	: y - Koordinate des oberen linken Eckpunktes
<code><xxxx₂></code> =	- 9999 bis 9999	: x - Koordinate des unteren rechten Eckpunktes
<code><yyyy₂></code> =	- 9999 bis 9999	: y - Koordinate des unteren rechten Eckpunktes

Beispiel: Balken mit den Eckpunkten (900/3000) und (1800/3200)

```
&%%49 01&%%13 09
&%%07 0900 3000 1800 3200
```



4.3.11. Balken mit runden Ecken

36

Funktion: Mit diesem Befehl wird ein Balken mit runden Ecken gezeichnet. Der Balken wird durch die Punkte x_1, y_1 (= oberer linker Eckpunkt) und x_2, y_2 (= unterer rechter Eckpunkt) definiert. Vorher vereinbarte Parameter wie Farbe, Rastermode und Speichermode werden berücksichtigt.

Syntax: <ESC><ESC>36 <xxxx₁> <yyyy₁> <xxxx₂> <yyyy₂> <rrrr>

(27)(27)(51)(54)...

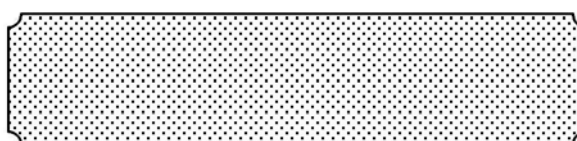
Parameter:

<xxxx ₁ > =	- 9999 bis 9999	: x – Koordinate Eckpunkt oben links
<yyyy ₁ > =	- 9999 bis 9999	: y – Koordinate Eckpunkt oben links
<xxxx ₂ > =	- 9999 bis 9999	: x – Koordinate Eckpunkt unten rechts
<yyyy ₂ > =	- 9999 bis 9999	: y – Koordinate Eckpunkt unten rechts
<rrrr> =	0000 bis 9999	: Radius des Bogens (in dot), der an den Ecken eingefügt wird. Es sind positive und negative Werte möglich.

Beispiel: Balken mit den Eckpunkten (900/2800) und (1800/3000),
Radius der Bögen: -20 pixels

```
&%%49 01&%%13 09
```

```
&%%36 0900 2800 1800 3000 -20
```



4.3.12. Box mit runden Ecken

Funktion: Mit diesem Befehl wird eine absolut positionierte Box (mit runden Ecken) gezeichnet. Die Box wird durch die Punkte x_1, y_1 (= oberer linker Eckpunkt) und x_2, y_2 (= unterer rechter Eckpunkt) definiert. Vorher vereinbarte Parameter wie Linienbreite, Farbe, Linienmode und Speichermode werden berücksichtigt.

Syntax: `<ESC><ESC>34 <xxxx1> <yyyy1> <xxxx2> <yyyy2> <rrrr>`
(27)(27)(51)(52)...

Parameter:

- `<xxxx1>` = -9999 bis 9999 : x - Koordinate des oberen linken Eckpunktes
- `<yyyy1>` = -9999 bis 9999 : y - Koordinate des oberen linken Eckpunktes
- `<xxxx2>` = -9999 bis 9999 : x - Koordinate des unteren rechten Eckpunktes
- `<yyyy2>` = -9999 bis 9999 : y - Koordinate des unteren rechten Eckpunktes
- `<rrrr>` = 0000 bis 9999 : Radius des Bogens (in dot), der an den Ecken eingefügt wird. Es sind positive und negative Werte möglich.

Beispiel: Box mit Radien $r = 70$ Punkte. Linker oberer Eckpunkt $x = 700$, $y = 2700$, rechter unterer Eckpunkt $x = 1500$, $y = 2900$.
`&%%34 700 2700 1500 2900 70`



4.3.13. Ellipse

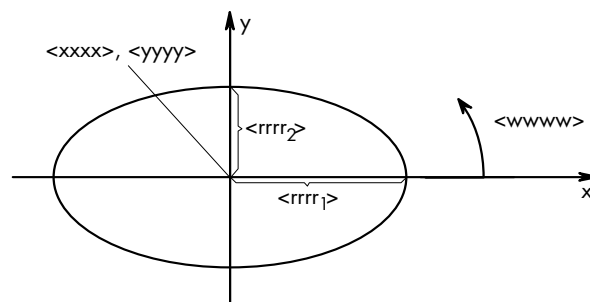
56

Funktion: Mit diesem Befehl wird eine Ellipse in der aktuell gültigen Zeichenfarbe und mit dem aktuellen Linienmode gezeichnet.

Syntax: `<ESC><ESC>56 <rrrr1> <rrrr2>
<xxxx> <yyyy> <wwwwww>
(27)(27)(53)(54)...`

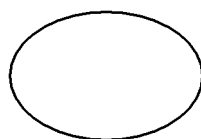
Parameter:

<code><rrrr1></code>	=	0000 bis 9999	: Radius der Ellipse in x - Richtung
<code><rrrr2></code>	=	0000 bis 9999	: Radius der Ellipse in y - Richtung
<code><xxxx></code>	=	- 9999 bis 9999	: x - Koordinate des Ellipsen-Mittelpunktes
<code><yyyy></code>	=	- 9999 bis 9999	: y - Koordinate des Ellipsen-Mittelpunktes
<code><wwwwww></code>	=	0000 bis 3600	: Drehwinkel der Ellipsenachsen in Einheiten von 1/10 Grad



Beispiel: Eine Ellipse soll gezeichnet werden.

```
&%&%56 0150 0100 1150 3050 0000
```



4.3.14. Ellipsenfläche

Funktion: Dieser Befehl bewirkt das Zeichnen einer Ellipsenfläche. Die Fläche wird mit dem aktuell gültigen Raster gefüllt.

Syntax: <ESC><ESC>55 <rrrrr₁> <rrrrr₂>
 <xxxx> <yyyy> <www>
(27)(27)(53)(53)...

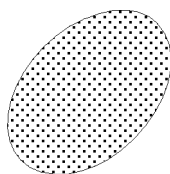
Parameter:

<rrrrr ₁ > =	0000 bis 9999	: Radius der Ellipsenfläche in x - Richtung
<rrrrr ₂ > =	0000 bis 9999	: Radius der Ellipsenfläche in y - Richtung
<xxxx> =	-9999 bis 9999	: x - Koordinate des Ellipsenflächen - Mittelpunktes
<yyyy> =	-9999 bis 9999	: y - Koordinate des Ellipsenflächen - Mittelpunktes
<www> =	0000 bis 3600	: Drehwinkel der Ellipsenflächen- achsen in Einheiten von 1/10 Grad

Beachte: Die Fläche wird mit dem aktuellen Raster aufgefüllt und mit dem aktuellen Linienmode gezeichnet.

Beispiel: Mit dem Raster 9 soll eine kleine, schrägliegende Ellipsenfläche gezeichnet werden.

```
&%%4901&%%13 09
&%%55 150 100 1150 3050 450
```



4.3.15. Ellipsenflächen-Segment

63

Funktion: Dieser Befehl bewirkt das Zeichnen eines Ellipsenflächen-Segments. Die Fläche wird mit dem aktuell gültigen Raster gefüllt.

Syntax: <ESC><ESC>63 <nnnn1> <nnnn2> <rrrr_x> <rrrr_y>
<xxxx> <yyyy> <www>

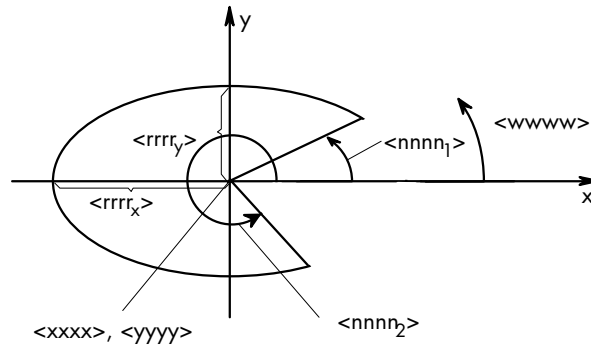
(27)(27)(54)(51)...

Parameter:

<nnnn1> =	0000 bis 3600	: Anfangswinkel des Flächenelements in 1/10 Grad
<nnnn2> =	0000 bis 3600	: Endwinkel des Flächenelements in 1/10 Grad
<rrrr _x > =	0000 bis 9999	: Radius der Ellipse in x - Richtung
<rrrr _y > =	0000 bis 9999	: Radius der Ellipse in y - Richtung
<xxxx> =	-9999 bis 9999	: x - Koordinate des Ellipsen - Mittelpunktes
<yyyy> =	-9999 bis 9999	: y - Koordinate des Ellipsen - Mittelpunktes
<www> =	0000 bis 3600	: Drehwinkel der Ellipsenachsen in 1/10 Grad

Beachte: Die Fläche wird mit dem aktuellen Raster aufgefüllt.

Das Ellipsenflächen-Segment wird durch die Parameter wie folgt beschrieben:

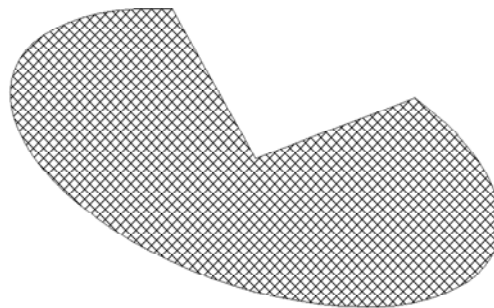


Beispiel:

Mit dem Raster 6 soll ein Ellipsenflächen-Segment gezeichnet werden.

&%%49 01&%%13 06

&%%63 300 3300 200 400 1250 2200 700



4.3.16. Ellipsenflächen-Abschnitt

67

Funktion: Dieser Befehl bewirkt das Zeichnen eines Ellipsenflächen-Abschnitts. Die Fläche wird mit dem aktuell gültigen Raster gefüllt.

Syntax: <ESC><ESC>67 <nnnn1> <nnnn2> <rrrrx> <rrrry>
<xxxx> <yyyy> <www>

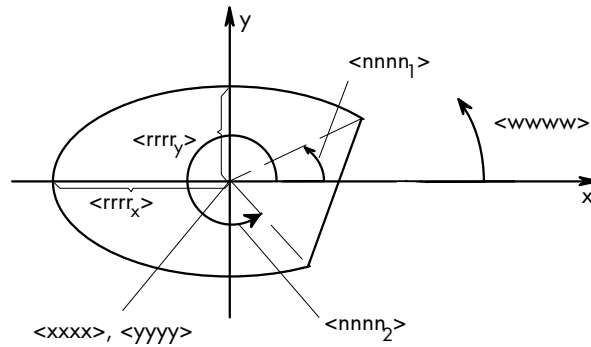
(27)(27)(54)(55)...

Parameter:

<nnnn1> =	0000 bis 3600	: Anfangswinkel des Flächenelements in 1/10 Grad
<nnnn2> =	0000 bis 3600	: Endwinkel des Flächenelements in 1/10 Grad
<rrrrx> =	0000 bis 9999	: Radius der Ellipse in x - Richtung
<rrrry> =	0000 bis 9999	: Radius der Ellipse in y - Richtung
<xxxx> =	-9999 bis 9999	: x - Koordinate des Ellipsen - Mittelpunktes
<yyyy> =	-9999 bis 9999	: y - Koordinate des Ellipsen - Mittelpunktes
<www> =	0000 bis 3600	: Drehwinkel der Ellipsenachsen in 1/10 Grad

Beachte: Die Fläche wird mit dem aktuellen Raster aufgefüllt.

Der Ellipsenflächen-Abschnitt wird durch die Parameter wie folgt beschrieben:

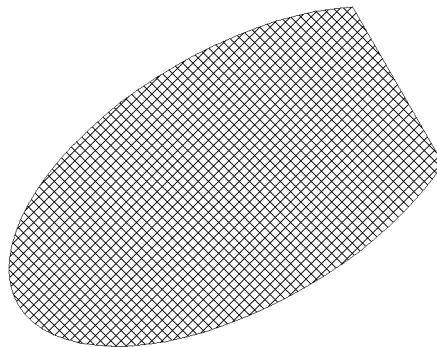


Beispiel:

Mit dem Raster 6 soll ein Ellipsenflächen-Abschnitt mit einem Drehwinkel von 30 Grad gezeichnet werden.

&%%49 01&%%13 06

&%%67 450 3150 400 200 1250 2200 300



4.4. Grafik mit relativer Positionierung

Relative Positionierungen beziehen sich immer auf die aktuelle Position des Grafikcursors. Dieser wird - von einem absolut positionierten Startpunkt ausgehend - nach jeder Parameter-Eingabe auf die dadurch definierte neue Position gesetzt. Die nachfolgende Parameter-Eingabe bezieht sich dann auf diese aktuelle Position.

Beachte: Bei der Angabe der Anzahl von Koordinaten, mit denen das jeweilige Grafikelement beschrieben werden soll, wird der Startpunkt nicht mitgezählt. Dieser muß vor dem Start der relativen Positionierung durch einen Positionierungsbefehl gesetzt werden. Vorher definierte Gestaltungsparameter wie Raster, Linienbreite und -mode etc. werden berücksichtigt.

4.4.1. Verschieben des Grafikcursors

Funktion: Mit diesem Befehl wird der Grafikcursor relativ zu seiner aktuellen Position verschoben.

Syntax: <ESC><ESC>28 <xxxx> <yyyy> (27)(27)(50)(56)...

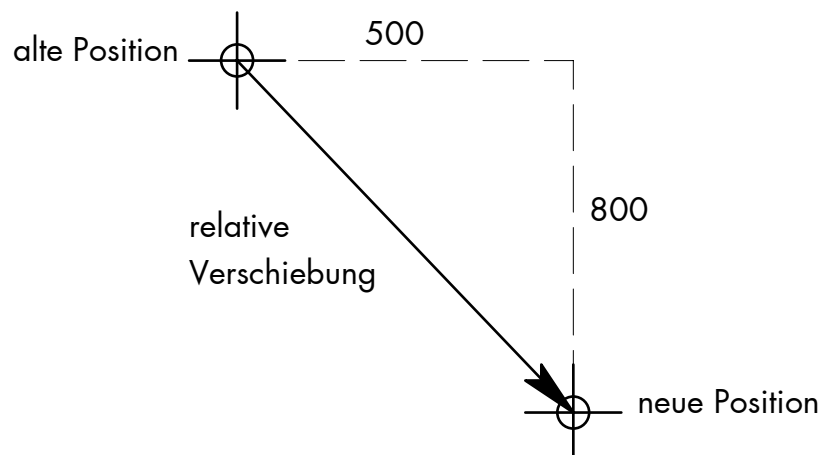
Parameter:

<xxxx> = -9999 bis 9999 : Anzahl Bildpunkte, um die der Grafikcursor in x - Richtung verschoben werden soll.

<yyyy> = -9999 bis 9999 : Anzahl Bildpunkte, um die der Grafikcursor in y - Richtung verschoben werden soll.

Beispiel: Der Grafikcursor soll von seiner aktuellen Position um 500 Punkte in x - Richtung und um 800 Punkte in y - Richtung verschoben werden.

&%%28 500 800



4.4.2. Fläche**21**

Funktion: Mit diesem Befehl wird eine relativ positionierte Fläche gezeichnet.

Syntax: <ESC><ESC>21 <nnn> <xxxx₁><yyyy₁>
 ... <xxxx_n><yyyy_n>

(27)(27)(50)(49)...

Parameter: <nnn> = 000 bis 255 : Anzahl der Eckpunkte
 ohne den Startpunkt

 <xxxx_i> = -9999 bis 9999 : Angabe der x - Koordinate für
 die jeweiligen Eckpunkte

 <yyyy_i> = -9999 bis 9999 : Angabe der y- Koordinate für
 die jeweiligen Eckpunkte

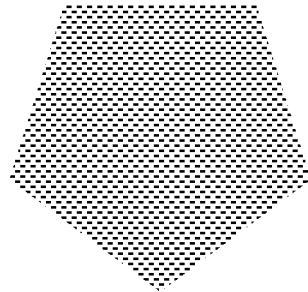
Beachte: Die Eckpunkte werden durch imaginäre Linien miteinander verbunden, die sich auch überschneiden können. Der n -te Eckpunkt wird automatisch mit dem Ausgangspunkt verbunden. Die eingeschlossene Fläche wird aufgefüllt.

Beispiel:

Es soll relativ zum absolut gesetzten Startpunkt $x = 1100$,
 $y = 1300$ ein Fünfeck gezeichnet werden.

%%08 1100 1300

%21 4 300 0 85 270 -235 175 -235 -175



4.4.3. Kreis

22

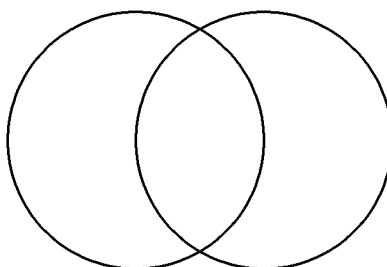
Funktion: Mit diesem Befehl wird ein Kreis an der aktuellen Grafikkursor - Position gezeichnet.

Syntax: <ESC><ESC>22 <rrrr> **(27)(27)(50)(50)...**

Parameter: <rrrr> = 0000 bis 9999 : Radius

Beispiel: Es sollen zwei sich überschneidende Kreise gezeichnet werden, wobei der Abstand zwischen den Kreismittelpunkten gleich dem Radius von 200 Bildpunkten ist.

```
&%&%08 1150 2500  
&%&%22 200  
&%&%28 200 0  
&%&%22 200
```



4.4.4. Linienzug

Funktion: Mit diesem Befehl wird ein Linienzug ausgehend von der aktuellen Grafikkursor - Position gezeichnet.

Syntax: `<ESC><ESC>23 <nnnn> <xxxx1><yyyy1>
... <xxxxn><yyyyn>`

(27)(27)(50)(51)...

Parameter: `<nnnn> = 0001 bis 9999` : Anzahl der zu verbindenden Punkte ohne den Startpunkt.

`<xxxxi> = -9999 bis 9999` : Angabe der x - Koordinate für die jeweiligen Eckpunkte

`<yyyyi> = -9999 bis 9999` : Angabe der y - Koordinate für die jeweiligen Eckpunkte

Beachte: Der Grafikkursor bleibt auf dem letzten Punkt n stehen.

Beispiel:

Zeichnen eines offenen Quadrates mit 200 Bildpunkten
Seitenlänge.

Linker oberer Eckpunkt an der Position $x = 1150$, $y = 1200$.

```
&%&%08 1150 1200
```

```
&%&%23 3 200 0 0 200 -200 0
```



4.4.5. Kreisbogen

Funktion: Mit diesem Befehl wird ein Kreisbogen um die aktuelle Grafikcursor - Position gezeichnet.

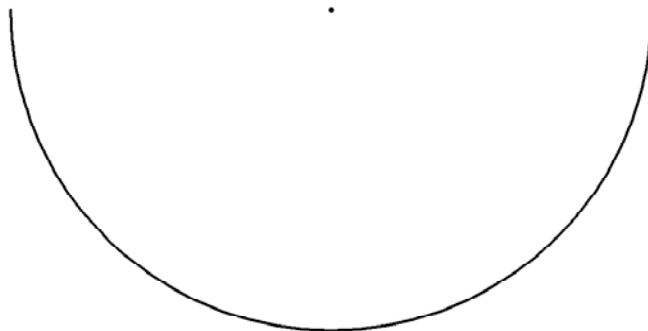
Syntax: <ESC><ESC>24 <nnnn₁> <nnnn₂> <rrrr>
(27)(27)(50)(52)...

Parameter:

<nnnn ₁ > =	0000 bis 3600	: Bogenbeginn in Einheiten von 1/10 Grad
<nnnn ₂ > =	0000 bis 3600	: Bogenende in Einheiten von 1/10 Grad
<rrrr> =	0000 bis 9999	: Radius

Beispiel: Zeichnen eines Halbkreises mit $r = 500$ um die Position $x = 1200$, $y = 2600$.

```
&%%08 1200 2600  
&%%24 1800 0 500
```



4.4.6. Kreisfläche

25

Funktion: Mit diesem Befehl wird eine Kreisfläche gezeichnet. Die Fläche wird mit dem aktuellen Raster ausgefüllt.

Syntax: <ESC><ESC>25 <rrrr> (27)(27)(50)(53)...

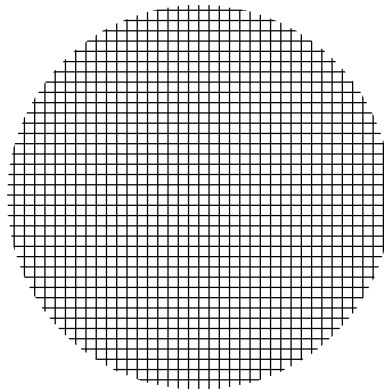
Parameter: <rrrr> = 0000 bis 9999 : Radius

Beispiel: Zeichnen einer Kreisfläche mit $r = 300$ um die Position $x = 1250$, $y = 2500$.
Die Fläche soll mit dem Raster 5 ausgefüllt sein.

```
&%%08 1250 2500
```

```
&%%13 05
```

```
&%%25 300
```



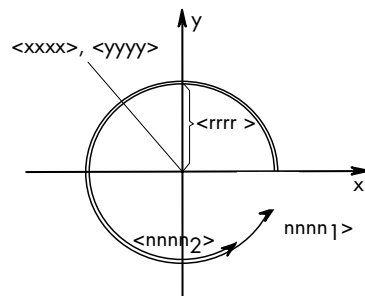
4.4.7. Kreisflächensegment

Funktion: Mit diesem Befehl wird ein Kreisflächensegment an der aktuellen Grafikkursor - Position gezeichnet. Die Fläche wird mit dem aktuellen Raster ausgefüllt.

Syntax: `<ESC><ESC>26 <nnnn1> <nnnn2> <r rrr>`
(27)(27)(50)(54)...

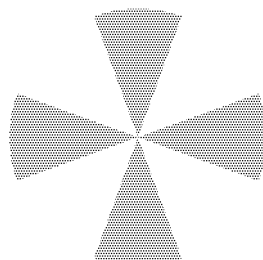
Parameter:

- `<nnnn1> = 0000 bis 3600` : Bogenbeginn in Einheiten von 1/10 Grad
- `<nnnn2> = 0000 bis 3600` : Bogenende in Einheiten von 1/10 Grad
- `<r rrr> = 0000 bis 9999` : Radius



Beispiel: Zeichnen von 4 aufeinander senkrecht stehenden "Tortenstücken" (Öffnungswinkel 40 Grad) mit den Spitzen am Punkt $x = 1250, y = 2900$.

```
&%%08 1250 2900
&%%26 700 1100 200 &%%26 1600 2000 200
&%%26 2500 2900 200 &%%26 3400 200 200
```



4.4.8. Balken

27

Funktion: Mit diesem Befehl wird ein Balken gezeichnet. Die linke, obere Ecke befindet sich an der aktuellen Grafikkursor - Position. Der Balken wird mit dem aktuellen Raster ausgefüllt.

Syntax: <ESC><ESC>27 <bbbb> <hhhh> **(27)(27)(50)(55)...**

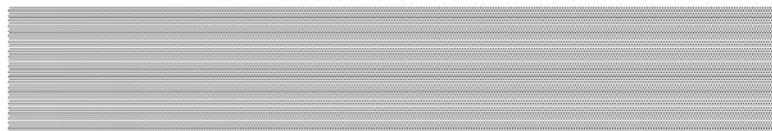
Parameter: <bbbb> = -9999 bis 9999 : Breite des Balkens, ausgehend vom Grafikkursor, positiv nach rechts

<hhhh> = -9999 bis 9999 : Höhe des Balkens, ausgehend vom Grafikkursor, positiv nach unten

Beispiel: Waagrecht liegender Balken, 1200 Punkte breit, 200 Punkte hoch. Linker oberer Eckpunkt x = 700, y = 2700.

```
&%%08 700 2700
```

```
&%%27 1200 200
```



4.4.9. Balken mit runden Ecken

Funktion: Mit diesem Befehl wird ein Balken mit runden Ecken gezeichnet (relativ positioniert). Die linke, obere Ecke befindet sich an der aktuellen Grafikkursor - Position. Der Balken wird mit dem aktuellen Raster ausgefüllt.

Syntax: <ESC><ESC>37 <bbbb> <hhhh> <rrrr> **(27)(27)(51)(55)...**

Parameter:

<bbbb> =	-9999 bis 9999	: Breite des Balkens, ausgehend vom Grafikkursor, <u>positiv nach rechts</u>
<hhhh> =	-9999 bis 9999	: Höhe des Balkens, ausgehend vom Grafikkursor, <u>positiv nach unten</u>
<rrrr> =	0000 bis 9999	: Radius des Bogens (in dot), der an den Ecken eingefügt wird. Es sind positive und negative Werte möglich.

Beispiel: Waagrecht liegender Balken, 1200 Punkte breit, 200 Punkte hoch. Linker oberer Eckpunkt x = 700, y = 2700, Radius der Bögen: 40 pixels

```
&%%49 00&%%13 55
&%%08 700 2700
&%%37 1200 200 40
```



4.4.10. Box mit runden Ecken

35

Funktion: Mit diesem Befehl wird eine relativ positionierte Box (mit runden Ecken) gezeichnet. Die linke, obere Ecke befindet sich an der aktuellen Grafikkursor - Position.
Vorher vereinbarte Parameter wie Linienbreite, Farbe, Linienmode und Speichermode werden berücksichtigt.

Syntax: <ESC><ESC>35 <bbbb> <hhhh> <rrrr> **(27)(27)(51)(53)...**

Parameter:

<bbbb> =	-9999 bis 9999	: Breite der Box, ausgehend vom Grafikkursor, <u>positiv nach rechts</u>
<hhhh> =	-9999 bis 9999	: Höhe der Box, ausgehend vom Grafikkursor, <u>positiv nach unten</u>
<rrrr> =	0000 bis 9999	: Radius des Bogens (in dot), der an den Ecken eingefügt wird. Es sind positive und negative Werte möglich.

Beispiel: Box mit einer Breite von 800 Punkten, 200 Punkte hoch, Radius der Bögen: -80 Punkte.
&%&%35 800 200 -80



4.4.12. Ellipsenfläche

65

Funktion: Mit diesem Befehl wird eine Ellipsenfläche gezeichnet. Der Mittelpunkt der Ellipsenachsen befindet sich an der aktuellen Grafikkursor - Position. Die Ellipsenfläche wird mit dem aktuellen Raster ausgefüllt.

Syntax: <ESC><ESC>65 <rrrrr_x> <rrrrr_y> <www>
(27)(27)(54)(53)...

Parameter:

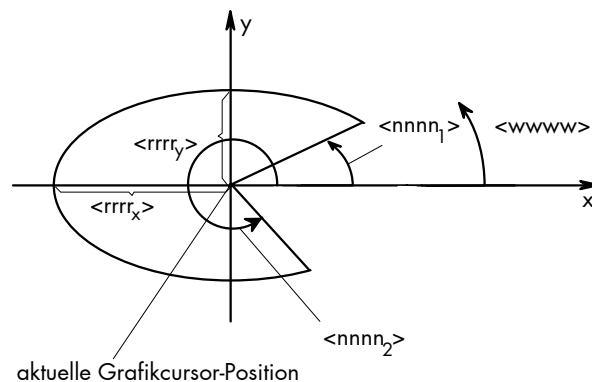
<rrrrr _x > =	0000 bis 3600	: Radius der Ellipse in x - Richtung, vom Grafikkursor ausgehend
<rrrrr _y > =	0000 bis 3600	: Radius der Ellipse in y - Richtung, vom Grafikkursor ausgehend
<www> =	0000 bis 3600	: Drehwinkel der Ellipsenachse in 1/10 Grad

Beispiel: Zeichne eine um 60 Grad gedrehte Ellipsenfläche ausgehend von der aktuellen Grafikkursor - Position.

```
&%%08 1250 2950  
&%%65 200 100 600
```



Das Ellipsenflächen-Segment wird durch die Parameter wie folgt beschrieben:

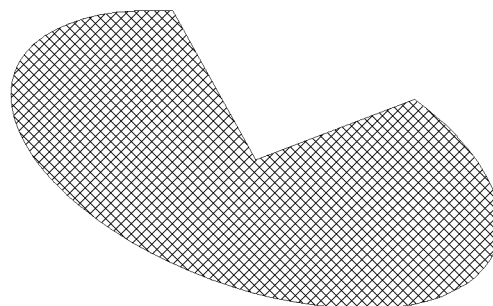


Beispiel:

Mit dem Raster 06 (vergl. Abschnitt: Raster) soll ein automatisch umrandetes Ellipsenflächen-Segment an der aktuellen Graficursor-Position gezeichnet werden.

&%%08 1250 2300

&%%4901&%%13 06&%%64 300 3300 200 400 700



4.4.14. Ellipsenflächen-Abschnitt

Funktion: Dieser Befehl bewirkt das Zeichnen eines Ellipsenflächen-Abschnitts an der aktuellen Grafikkursor-Position.
Die Fläche wird mit dem aktuell gültigen Raster gefüllt.

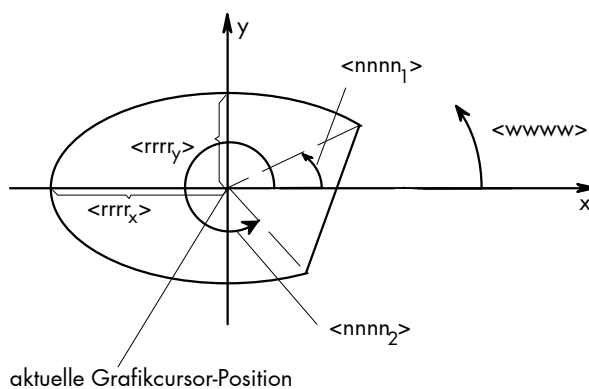
Syntax: <ESC><ESC>68 <nnnn1> <nnnn2> <rrrrx> <rrrry>
<www>
(27)(27)(54)(56)...

Parameter:

<nnnn1> =	0000 bis 3600	: Anfangswinkel des Flächenelements in 1/10 Grad
<nnnn2> =	0000 bis 3600	: Endwinkel des Flächenelements in 1/10 Grad
<rrrrx> =	0000 bis 9999	: Radius der Ellipse in x - Richtung
<rrrry> =	0000 bis 9999	: Radius der Ellipse in y - Richtung
<www> =	0000 bis 3600	: Drehwinkel der Ellipsenachsen in 1/10 Grad

Beachte: Die Fläche wird mit dem aktuellen Raster aufgefüllt.

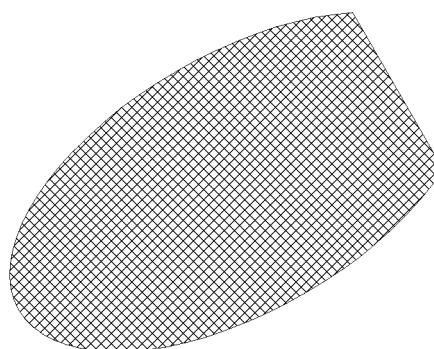
Der Ellipsenflächen-Abschnitt wird durch die Parameter wie folgt beschrieben:



Beispiel:

Mit dem Raster 6 (vergl. Abschnitt: Raster) soll ein automatisch umrandeter Ellipsenflächen-Abschnitt an der aktuellen Grafikcursor-Position gezeichnet werden.

```
&%%08 1250 2300
&%%4901&%%13 06
&%%68 450 3150 400 200 300
```



4.5. Grafik in Polarkoordinaten

Die Positionierung grafischer Elemente in Polarkoordinaten bezieht sich immer auf die aktuelle Grafikkursor - Position (= Startkoordinaten). Es handelt sich also um relative Positionierungen.

Eingestellte Gestaltungs - Parameter wie Linienbreite und -mode, Raster etc. werden berücksichtigt.

Der Grafikkursor steht auf dem zuletzt definierten Punkt.

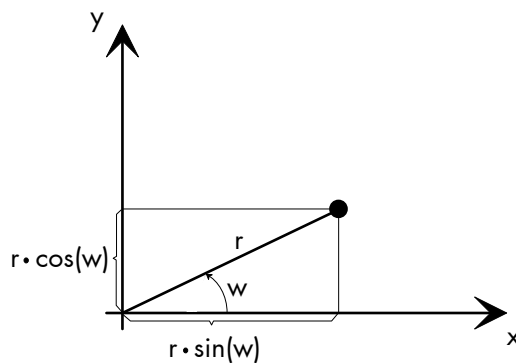
Beziehung zwischen Polar- und Relativ-Koordinaten:

$$X = r \cdot \sin(w) \quad Y = r \cdot \cos(w)$$

Beziehung zwischen Polar- und Absolut-Koordinaten:

$$X = x + r \cdot \sin(w) \quad Y = y + r \cdot \cos(w)$$

(x, y) = Koordinaten des Grafikkursors



4.5.1. Verschieben des Grafikcursors

38

Funktion: Mit diesem Befehl wird der Grafikcursor relativ zur aktuellen Position verschoben.

Syntax: <ESC><ESC>38 <rrrr> <www> (27)(27)(51)(56) ...

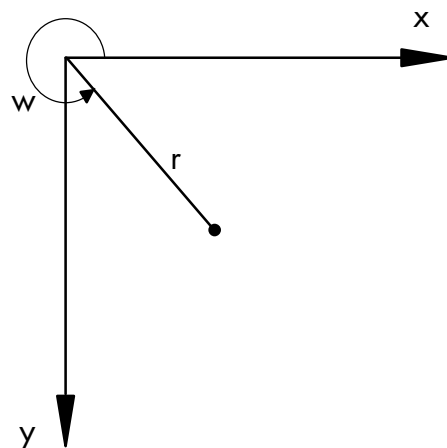
Parameter: <rrrr> = 0000 bis 9999 : Radius, vom aktuellen Grafikcursor ausgehend

<www> = 0000 bis 3600 : Winkel in Einheiten von 1/10 Grad

Beispiel: Setze den Grafikcursor nach einem Grafik - Reset auf die Position $x = y = 1000$

```
&%%43
```

```
&%%38 1414 3150
```



4.6. Übersicht: Grafikfunktionen und Darstellungselemente

In der nachfolgenden Tabelle wird zusammengefaßt, welche Zusammenhänge zwischen Grafikfunktionen und Darstellungselementen bestehen.

Symbolerklärung:

-	=	ohne Bedeutung für die Funktion	x	=	mit Bedeutung für die Funktion
R	=	Replace	O	=	Overstrike
C	=	Complement	E	=	Erase

Darstellungselement		Grafikfunktionen					Befehl				
		Linienmode	Linienbreite	Farbe	Raster	Hinterlegungsmodus				ESCESC...	
	<u>adressiert:</u>										
Grafikcursor	absolut	-	-	-	-	-	-	-	-	-	08
Grafikcursor (x-Richtung)	absolut	-	-	-	-	-	-	-	-	-	44
Grafikcursor (y-Richtung)	absolut	-	-	-	-	-	-	-	-	-	45
Fläche	absolut	-	-	X	X	R	O	C	E		01
Kreis	absolut	X	X	X	-		O		E		02
Linienzug	absolut	X	X	X	-		O		E		03
Kreisbogen	absolut	X	X	X	-		O		E		04
Kreisfläche	absolut	-	-	X	X	R	O	C	E		05
Kreisflächensegment	absolut	-	-	X	X	R	O	C	E		06
Balken	absolut	-	-	X	X	R	O	C	E		07
Balken mit runden Ecken	absolut	-	-	X	X	R	O	C	E		36
Box mit runden Ecken	absolut	-	-	X	X	R	O	C	E		34
Ellipse	absolut	X	X	X	-		O		E		56
Ellipsenfläche	absolut	-	-	X	X	R	O	C	E		55
Ellipsenflächen-Segment	absolut	-	-	X	X	R	O	C	E		63
Ellipsenflächen-Abschnitt	absolut	-	-	X	X	R	O	C	E		67
Grafikcursor	relativ	-	-	-	-	-	-	-	-	-	28
Fläche	relativ	-	-	X	X	R	O	C	E		21
Kreis	relativ	X	X	X	-		O		E		22
Linienzug	relativ	X	X	X	-		O		E		23
Kreisbogen	relativ	X	X	X	-		O		E		24
Kreisfläche	relativ	-	-	X	X	R	O	C	E		25
Kreisflächensegment	relativ	-	-	X	X	R	O	C	E		26
Balken	relativ	-	-	X	X	R	O	C	E		27
Balken mit runden Ecken	relativ	-	-	X	X	R	O	C	E		37
Box mit runden Ecken	relativ	-	-	X	X	R	O	C	E		35
Ellipse	relativ	X	X	X	-		O		E		66
Ellipsenfläche	relativ	-	-	X	X	R	O	C	E		65
Ellipsenflächen-Segment	relativ	-	-	X	X	R	O	C	E		64
Ellipsenflächen-Abschnitt	relativ	-	-	X	X	R	O	C	E		68
Grafikcursor	in Polark.	-	-	-	-	-	-	-	-	-	38
Linienzug	in Polark.	X	X	X	-		O		E		33
Umrandung, autom.		X	X	X	-		O		E		49

5. Funktionsgruppe DOWNLOAD

DOWNLOAD: Bei Bedarf lassen sich zusätzliche Schriften über die Schnittstelle in den MACRO / DOWNLOAD - Speicher (USER-RAM) laden.

Damit ergibt sich die Möglichkeit, Schriften zu ergänzen, zu modifizieren oder völlig neu einzurichten. Sie werden genau wie feste Zeichensätze benutzt. Allerdings werden Download-Schriften nicht über Zeichentabellen verwaltet, das heißt, die gewünschte Zeichencodierung für die Ländervariante muß beim Laden mitgeliefert werden.

Die Anzahl der Daten, die im USER-RAM gespeichert werden können, ist abhängig von der Konfiguration des Speichers und kann dem Statusblatt des jeweiligen Druckers entnommen werden.

Zusätzliche Schriften bzw. Zeichen können z.B. Logos, Gefahrenzeichen oder Unterschriften sein, die dann einfach durch eine Fontanwahl und einen Buchstaben gedruckt werden.

5.1. Einrichten und Ergänzen einer DOWNLOAD-Schrift

5.1.1. Laden eines einzelnen Zeichens



Funktion: Mit diesem Befehl wird ein einzelnes Zeichen in den MACRO/DOWNLOAD - Speicher des Druckers übertragen. Dieser Befehl wird verwendet, um Logos, Unterschriften oder Gefahrenzeichen flexibel zu nutzen.

Syntax:

```
<ESC><ESC>x <nnnn> <aa> <bbbb> <hhhh>
<HHHH> <llll> <rrrr>
<dd1> ... <ddn> <cc>
```

(27)(27)(120) ...

Parameter:

<code><nnnn></code>	= 0001 bis 9999	: Schriftartnummer
<code><aa></code>	= 00 bis FF	: Zeichennummer innerhalb d. Schriftart, Hex.-Zahl = Zeichencodierung
<code><bbbb></code>	= 0000 bis FFFF	: Breite des Zeichens in Bildpunkten (nur Zeichenbreite, keine Freifelder) vierstellige Hex.-Zahl
<code><hhhh></code>	= 0000 bis FFFF	: Höhe des Zeichens in Bildpunkten (incl. der Unterlänge), vierstellige Hex.-Zahl

<HHHH>	= 0000 bis FFFF	: Höhe des Zeichens in Bildpunkten ab der Schreiblinie, vierstellige Hex.-Zahl
<llll>	= 0000 bis FFFF	: linker Rand, vierstellige Hex.-Zahl
<rrrr>	= 0000 bis FFFF	: rechter Rand, vierstellige Hex.-Zahl
<dd _i >	= 00 bis FF	: Datenbyte, zweistellige Hex.-Zahl
<cc>	= 00 bis FF	: Checksumme, zweistellige Hex.-Zahl

Ist die angewählte Schriftart im RAM vorhanden, wird das zu ändernde Zeichen unter der entsprechenden Codierung geladen. Ist dieser Schrifttyp aber unbekannt, wird unter der eingegebenen Nummer ein neuer Zeichensatz mit maximal 255 Zeichen angelegt.

Die Anzahl der zu übertragenden Bytes n ergibt sich aus der Fläche des Zeichens (Höhe und Breite). Jede Zeile muß aus ganzzahligen Datenbytes bestehen. Überstehende Bits müssen als Pseudobits mit eingegeben werden.

Addiert man (8-bit-Addition ohne Übertrag) alle Bytes ab der Zeichennummer (Doppelbyte-Parameter werden als zwei einzelne Bytes betrachtet) inklusive der Checksumme zusammen, so muß sich die Summe 0 ergeben.

$$n(\text{high}) + n(\text{low}) + a + b(\text{high}) + b(\text{low}) + h(\text{high}) + h(\text{low}) + H(\text{high}) + H(\text{low}) + d_1 + d_2 + \dots + d_n + c = 0$$

Beachte: Für die einfache Erstellung von DOWNLOAD-Zeichen ist das PC-Softwarepaket "PCX-TOOL" erhältlich, mit dem gescannte und ggfs. nachbearbeitete PCX-Bilder direkt in das IDOL-DOWNLOAD-Format umgewandelt werden. Die so entstandenen

Dateien werden wie Druckdateien zum Drucker übertragen und stehen als DOWNLOAD-Font zur Verfügung.

Beispiel: Im Zeichensatz (Font) Nr. 0001 soll unter dem ASCII-Code 41 (Hex) nachfolgendes Zeichen abgelegt werden:

	Rasterbild										Bit - Codierung mit Pseudobits x	Hex. - Codierung
	1	2	3	4	5	6	7	8	9	10		
1	o	o	●	●	●	●	●	●	●	o	1111 111x	FE
2	o	o	●	●	●	●	●	●	●	o	1111 111x	FE
3	o	o	●	●	o	o	o	o	o	o	1100 000x	C0
4	o	o	●	●	o	o	o	o	o	o	1100 000x	C0
5	o	o	●	●	●	●	●	●	o	o	1111 110x	FC
6	o	o	●	●	●	●	●	●	o	o	1111 110x	FC
7	o	o	●	●	o	o	o	o	o	o	1100 000x	C0
8	o	o	●	●	o	o	o	o	o	o	1100 000x	C0
9	o	o	●	●	o	o	o	o	o	o	1100 000x	C0
10	o	o	●	●	o	o	o	o	o	o	1100 000x	C0

```
&%&%x 0001 41 0007 000A 000A 0002 0001 FE FE C0 C0 FC FC
C0 C0 C0 C0 2D
```

Der Befehl wird mit folgenden Parametern aufgerufen:

```
0001           = <nnnn>           = Zeichensatznummer
41             = <aa>             = ASCII-Code
007           = <bbbb>           = Zeichenbreite
000A          = <hhhh>           = Zeichenhöhe
000A          = <HHHH>           = Höhe ab Schreiblinie
0002          = <llll>           = linker Rand
0001          = <rrrr>           = rechter Rand
FE,FE,C0,C0,FC, = <dd1> ... <ddn> = Datenbytes
FC,C0,C0,C0,C0
2D            = c                 = Checksumme
```

Kriterium für die Checksumme:

$$41+00+07+00+0A+00+0A+00+02+00+01+FE+FE+ \\ C0+C0+FC+C0+C0+C0+C0+2D = 0$$

Um dieses Zeichen auszudrucken, wird der Zeichensatz angewählt und das Zeichen mit seiner Zeichennummer angesprochen.
(HEX '41' = ASCII 'A')

&%&%B 0001 AAAAAAAAAA

FFFFFFF

5.1.2. Parameter eintragen für eine **DOWNLOAD**-Schrift



Funktion: Mit diesem Befehl werden alle noch fehlenden Parameter für eine **DOWNLOAD**-Schrift gesetzt, um alle Funktionen wie bei residenten EPROM - Schriftarten ausführen zu können.

Syntax: <ESC><ESC>W <nnnn₁> <nnnn₂> <nnnn₃> <nn₄>
 <nn₅> <nnnn₆> <nnnn₇>

(27)(27)(87) ...

Parameter:	<nnnn ₁ >	= 0001 bis 9999	: Schriftartnummer für den Parametereintrag
	<nnnn ₂ >	= 0000 bis 9999	: Breite des Leerschrittes in Anzahl Bildpunkten
	<nnnn ₃ >	= 0000 bis 9999	: Höhe der Textzelle in Anzahl Bildpunkten
	<nn ₄ >	= 00 bis 04	: Versatz für Fettschrift in Anzahl Bildpunkten
	<nn ₅ >	= 00 bis 99	: Stärke des Unterstreichungs- striches in Anzahl Bildpunkten
	<nnnn ₆ >	= 0000 bis 9999	: Abstand zwischen Schreiblinie und Unterstreichungsstrich in Anzahl Bildpunkten
	<nnnn ₇ >	= 0000 bis 9999	: Abstand zwischen Schreiblinie und Unterkante der Textzelle in Anzahl Bildpunkten

Beispiel:

In der Schriftart Nr. 0001 soll unter dem ASCII-Code 41 (Hex) ein Zeichen abgelegt werden (vergl. Abschnitt 5.1.1 "Laden eines einzelnen Zeichens").

Es werden für die Schriftart Nr. 0001 folgende Parameter vereinbart:

0001 = <nnn1> : Zeichensatznummer

0010 = <nnn2> : Breite des Leerschrittes

0015 = <nnn3> : Höhe des Schriftfeldes

01 = <nn4> : Versatz für Fettschrift

02 = <nn5> : Stärke des Unterstreichungsstriches

0005 = <nnn6> : Abstand zwischen Schreiblinie
und Unterstreichungsstrich

0010 = <nnn7> : Abstand zwischen Schreiblinie
und Unterkante des Schreibfeldes

```
&%&%x 0001 4100 0700 0A 00 0A 0002 0001 FE FE C0 C0  
FC FC C0 C0 C0 C0 2D  
&%&%W 0001 0010 0015 01 02 0005 0010  
&%&%B 0001 AAAA
```


6. Funktionsgruppe MACRO (Elektronische Formulare)

MACRO: Wiederkehrende Befehlsfolgen lassen sich zu sogenannten MACROs zusammenfassen.

Diese MACROs werden an den Drucker geschickt und dort im MACRO / DOWNLOAD - Speicher (USER-RAM) abgelegt und verwaltet.

Sie bleiben dort bis zum Ausschalten des Druckers gespeichert, oder bis ein Löschbefehl geschickt wird.

In einem RFM (Flash-EPR0M) können die MACROs resident gespeichert werden, so daß sie nach dem Einschalten des Druckers sofort zur Verfügung stehen.

Die steckerkompatiblen MICROPLEX Drucksysteme bieten die Möglichkeit des elektronischen Formulardrucks.

Die Funktionsweise von elektronischen Formularen verdeutlicht sich anhand eines Beispiels:

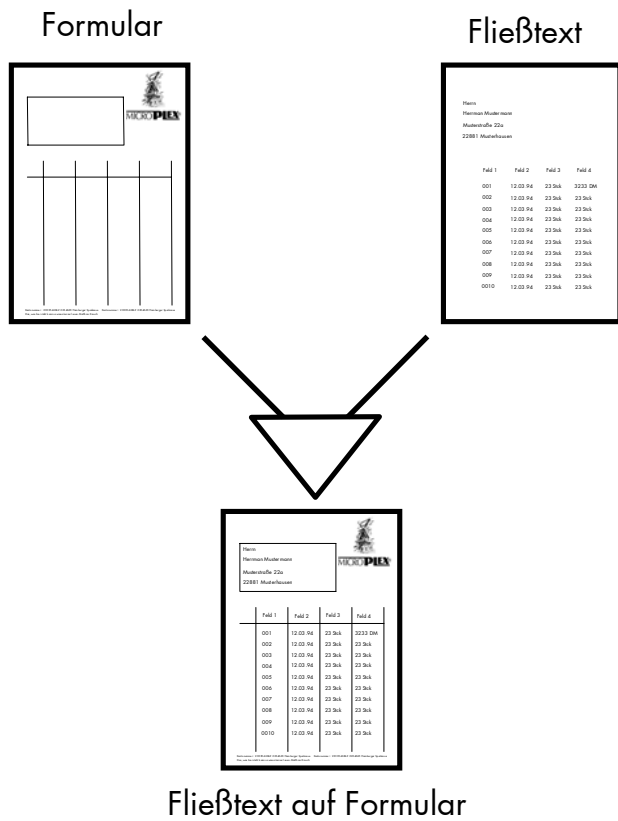
Herr Mustermann hat einen Drucker in seinem Büro. Er schreibt seine Korrespondenz auf vorgedrucktem Briefpapier mit dem Firmenlogo und den Kontoverbindungen. Außerdem nutzt Herr Mustermann seinen Drucker für Lieferscheine und Rechnungen. Auch hierfür hat er vorgedruckte Formulare in der Schublade. Jedesmal, wenn er seine Anwendung ändert, muß Herr Mustermann das Papier im Drucker wechseln.

Jetzt hat Herr Mustermann einen MICROPLEX-Drucker mit elektronischen Formularen gesehen. Dieser Drucker kann seinen vorgedruckten Briefbogentext zur selben Zeit wie seinen eigenen Brief drucken. Auch sein Rechnungsvordruck wird in einem Arbeitsgang mit seinem Rechnungstext gedruckt. Die verschiedenen Formulare sind permanent im Flash-Speicher des Druckers gespeichert.

Aber Herr Mustermann könnte seine Formulare auch auf dem MICROPLEX-Abteilungsdrucker in der Poststelle drucken, mit dem er über ein Netzwerk verbunden ist. Er überträgt das Formular einfach mitsamt dem Fließtext in die Poststelle. Er braucht jetzt keine verschiedenen Papiersorten in den Drucker zu legen und spart sich den Weg zur Poststelle.

Die Vorteile des elektronischen Formulareindrucks auf einen Blick:

- Die Fließdaten werden exakt in den "Vordruck" eingepaßt, da beides in einem Schritt gedruckt wird. Die Positioniergenauigkeit der Daten im Formular ist ca. 0,08 mm (bei einer Druckerauflösung von 300 dpi).



- Die maximale Druckgeschwindigkeit wird in der Regel nicht beeinflusst.
- Es können bis zu 99 verschiedene Formulare gleichzeitig im Drucker gespeichert werden.
- Jedes Formular kann in mehreren Druckern gleichzeitig gespeichert sein.
- Um ein Formular zu wechseln, reicht ein IDOL-Befehl aus. Es muß kein Papier gewechselt werden. Zum Drucken aller betriebsinternen Formulare reicht eine Papiersorte (z.B. Blankopapier DIN A4) aus.
- Elektronische Formulare lassen sich sehr schnell korrigieren, falls sich z.B. eine Telefonnummer oder die Adresse ändert.
- Die gesamten Lagerhaltungskosten verschiedener Formulare und die Entsorgung alter Formulare entfallen.
- Es kann nie zu Engpässen von Formularen am Arbeitsplatz kommen.
- Ein Formular kann an einen Blattauswurf gekoppelt werden (`<ESC><ESC>U <nn>`). Dann wird solange auf diesem "Vordruck" gedruckt, bis der Drucker wieder zurückgesetzt wird. Für die Kopplung des Formulars wird ein Befehl an den Drucker gesandt. Die bestehenden Druckdateien müssen nicht geändert werden.
- Es gibt die Funktionalität des automatischen "Durchschlags". Die Daten werden nur einmal gesendet und werden dann n-fach auf n Formularen gedruckt. Bei den Durchschlägen kann das Formular zusätzlich automatisch gewechselt werden. Jeder Durchschlag ist dann auf einem anderen Formular mit den gleichen Daten gedruckt (z.B. Rechnung und Rechnungskopie).
- Formulare können temporär im Drucker gespeichert werden, d.h. sie sind nur solange im Drucker, bis sie gelöscht werden oder der Drucker ausgeschaltet wird.
- Mit einem Flash-EPROM (Resident Forms Memory-Option) lassen sich die Formulare dauerhaft im SOLID-Drucker speichern. Sie stehen dann nach dem Einschalten des Druckers sofort zur Verfügung.

6.2. Definieren eines MACROs

Funktion: Mit diesen Befehlen wird ein MACRO im MACRO/DOWNLOAD - Speicher definiert. MACROs eröffnen dem Anwender die Möglichkeit, Texte sowie Befehlsfolgen (z.B. Grafik-Sequenzen, elektronische Formulare) unter einer Code - Nummer im Drucker abzuspeichern. MACROs können zu jeder Zeit an gewünschter Stelle mittels eines kurzen Befehls in einem laufenden Druckauftrag eingefügt werden.

Syntax:	<code><ESC><ESC>N <nn></code>	(27)(27)(78) ...
	<code><ESC><ESC>oN <aa> <nnnn></code>	(27)(27)(111)(78) ...

Parameter: `<nn>` = 01 bis 99 : MACRO-Nummer

Beim erweiterten Befehl (oN):

`<aa>` = 00 ... 99 : Attribut, dezimal

`<aa>` = 00 : zur Zeit default

`<aa>` = 03 : SIA-Macro (verknüpft eine Befehlsfolge mit einer Schnittstelle; siehe Beispiel nächste Seite)

- Verwenden Sie z.B. 9999 als MACRO -Nummer, diese wird nicht interpretiert (Anomalie von Attribut 03)
- Die Schnittstelle wird als Zeichenfolge angegeben (nach der Macro-Nummer). Die Länge der Zeichenfolge ist flexibel. Details zur Verwendung von sog. Terminierungszeichen finden Sie z.B. in Abschnitt 6.5 MACRO-Aufruf mit einem Schlüsselwort.

`<nnnn>` = 0001 ... 9999 : MACRO-Nummer

Beachte: Soll ein geladenes MACRO überschrieben werden, muß es vorher gelöscht werden.

Der Befehl wirkt als Anfangs- und Endkriterium für die Definition eines MACROs. Damit wird eine Verschachtelung von MACRO-Definitionen verhindert.

Beispiele: A) Speichern eines einfachen Textes als MACRO:
Die folgende IDOL-Sequenz definiert die Worte "Text – MACRO" als Macro Nr. 01.

```
&%%N 01Text – MACRO&%%N
```

B) SIA-Macro:
Die folgende IDOL-Sequenz "kombiniert" eine Emulation mit einer Schnittstelle. In diesem Beispiel soll der Drucker auf die gewünschte Emulation umschalten, sobald ein Datenstrom über die definierte Schnittstelle in den Drucker gelangt.

Hinweis: Bitte benutzen Sie die exakte Bezeichnung der Schnittstelle – diese finden Sie auf dem Statusblatt Ihres Druckers unter der Überschrift "Schnittstelle":

```
&%%dN 03 9999"Parallel" &%%em06&%%dN
```


6.3. Aufrufen eines MACROs



Funktion: Diese Befehle rufen das MACRO mit der angegebenen Kennnummer (Angabe zwei- oder vierstellig) auf, d.h. die unter dieser Kennnummer abgelegten Daten werden ausgeführt.

Syntax:	<code><ESC><ESC>O <nn></code>	(27)(27)(79) ...
	<code><ESC><ESC>oO <nnnn></code>	(27)(27)(111)(79) ...

Parameter:

<code><nn> = 00</code>	: Aufrufen des Flash-Speicher-Inhalts
<code><nn> = 01 bis 99</code>	: MACRO - Nummer

Beim erweiterten Befehl (oO):

<code><nnnn> = 0000</code>	: Aufrufen des Flash-Speicher-Inhalts
<code><nnnn> = 0001 bis 9999</code>	: MACRO - Nummer

Beachte: Im Gegensatz zur Definition von MACROs (vergleiche Kapitel 6.2) ist beim Aufruf von MACROs eine Verschachtelung bis zu einer Tiefe von 10 möglich.

Beispiel: Es wird ein MACRO mit der Nummer 01 aufgerufen.

```
&%&%O 01
```

Jetzt wird der Text "Text - MACRO" an der aktuellen Textcursor-Position gedruckt, der im Abschnitt 6.1 als MACRO 01 gespeichert wurde.

6.4. Automatischer Aufruf eines MACROs

Funktion: Das unter der MACRO - Kennnummer (Angabe zwei- oder vierstellig) abgelegte MACRO wird automatisch bei jedem Form Feed ausgeführt.

Bei jedem automatischen Aufruf wird der Textcursor an die obere linke Ecke gesetzt.

Mit dem gleichen Befehl und der MACRO - Kennnummer `<nn> = 00` (oder vierstellig) oder einem CLEAR - Befehl (`<ESC><ESC>j`) wird der automatische Aufruf wieder abgeschaltet.

Syntax:	<code><ESC><ESC>U <nn></code>	(27)(27)(85) ...
	<code><ESC><ESC>oU <nnnn></code>	(27)(27)(111)(85) ...

Parameter: `<nn> = 00` : Automatischen Aufruf ausschalten
`<nn> = 01 bis 99` : MACRO-Nummer

Beim erweiterten Befehl (oU):

`<nnnn> = 0000` : Automatischen Aufruf ausschalten
`<nnnn> = 0001 bis 9999` : MACRO - Nummer

Beispiel: Es wird das MACRO mit der Kennziffer 01 automatisch bei jedem Form Feed ausgeführt.

`&%%&%U 01`

Zusammen mit dem Beispiel aus Abschnitt 6.1 würde jetzt auf jeder Seite zusätzlich der Text "Text - MACRO" gedruckt.

6.5. MACRO-Aufruf mit einem Schlüsselwort (Keyword)



Funktion: Mit diesem Befehl können Schlüsselwörter für MACROs definiert werden.
Ein Macro-Aufruf oder eine Macro-Kopplung kann dann durch das zugehörige Schlüsselwort aktiviert werden.

Syntax: <ESC><ESC>ky <a> "String" <k> <n timer> **(27)(27)(107)(121) ...**

Parameter:

<a>	: Befehlstyp (ASCII – Zeichen, einstellig)
<a> = +	: Schlüsselwort (Keyword) definieren
<a> = -	: Ein Schlüsselwort löschen
<a> = *	: Alle Schlüsselwörter löschen
"String"	: Schlüsselwort, Textstring Die Länge des Keyword ist flexibel. Deshalb wird das Keyword mit Terminierungszeichen "geklammert". Das erste Zeichen nach dem Parameter <a> definiert das Terminierungszeichen. Dieses Terminierungszeichen ist frei wählbar. Es muß lediglich sichergestellt werden, daß dieses Terminierungszeichen nicht in dem Keyword vorkommt.
<k>	: Macro-Kennbuchstabe (ASCII – Zeichen)
<k> = O	: Overlay Macro
<k> = U	: Form Feed Macro
<n timer> = 0001 bis 9999:	Nummer des MACROs, welches durch das Schlüsselwort gestartet werden soll.

Beispiele:

&%&%ky + "Rechnung"○1000	Schlüsselwort Rechnung mit Macro 1000 koppeln
&%&%ky - "Kopie"	Schlüsselwort Kopie löschen
&%&%ky *	Alle Schlüsselwörter löschen

6.6. Löschen eines MACROs

P
oP

Funktion: Der Befehl löscht das MACRO mit der Kennnummer <nn> (oder bei vierstelliger Angabe: <nnnn>).

Syntax:	<ESC><ESC>P <nn>	(27)(27)(80)...
	<ESC><ESC>oP <aa> <nnnn>	(27)(27)(111)(80)...

Parameter:

<nn> = 00	: alle IDOL MACROs löschen
<nn> = 01 bis 99	: MACRO - Nummer

Beim erweiterten Befehl (oP):

<aa> = 00 ... 99	: (Attribut, 2-stellig, dezimal) Bislang nicht realisiert, verwenden Sie 00
------------------	---

<nnnn> = 0000	: alle IDOL MACROs löschen
<nnnn> = 0001 bis 9999	: MACRO - Nummer

Beispiel: MACRO 01 wird gelöscht. Ein vorhandener automatischer Aufruf wird beendet.

```
&%&%U 00 &%&%P 01
```

6.7. FLASH-EPROM (RFM) programmieren

Resident Forms Memory stellt eine Funktionalität des MICROPLEX Controllers (MPC) dar. Es dient dazu, einen beliebigen Datenstrom in einem Bereich des FLASH-Speichers oder in einer Datei (rfm.dat) dauerhaft zu speichern. Bei jedem Einschalten des Druckers werden (falls vorhanden) automatisch zuerst diese Daten verarbeitet. Üblicherweise bestehen die Daten aus Macros und/oder ladbaren Schriften, die dann zur weiteren Benutzung im User-RAM eingelagert werden.

Funktion: Mit diesem Befehl wird das FLASH-EPROM (RFM) mit einem verbesserten Algorithmus programmiert. (Das FLASH-EPROM muß vor dem Löschen komplett auf Null programmiert werden). Der Programmiervorgang wird damit verlängert (ca. 1,5 min).

Syntax: <ESC><ESC>*bf* <nn> ... <ESC><ESC>*bf*

(27)(27)(98)(102)...

Parameter: <nn> : Funktionskennung (z.Zt. ohne Bedeutung, sollte 00 sein)

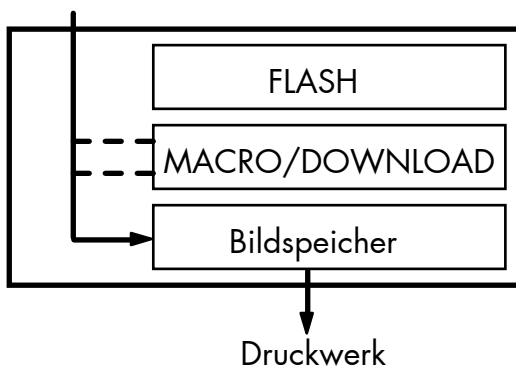
Beachte: Dieser Befehl ist ab der Firmware-Version 2.9 einsetzbar. Es wurde eine Prüfziffer eingeführt. Die Daten im FLASH-EPROM werden nur ausgeführt, wenn die Überprüfung fehlerfrei verläuft. Fehlerhafte Daten im FLASH-EPROM werden auf dem Statusblatt unter "Macros" angezeigt:

FLSH IDL * CRC-ERROR *****

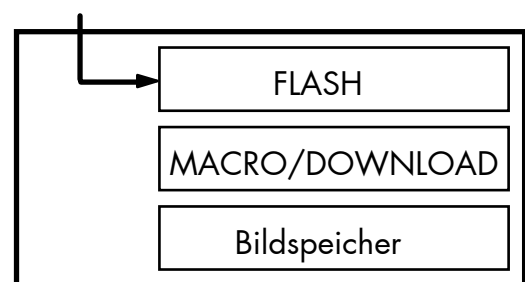
Der Inhalt des FLASH-EPROMs läßt sich jetzt mit dem IDOL-Befehl "Aufrufen eines MACROs" mit dem Parameter 00 zur Ausführung bringen. Dabei werden vorher alle MACROs gelöscht !

Beim Einschalten des Druckers werden die Daten aus dem FLASH-EPROM in den MACRO-DOWNLOAD-Speicher geladen. Die Daten werden genauso behandelt, als kämen sie von der Drucker-Schnittstelle.

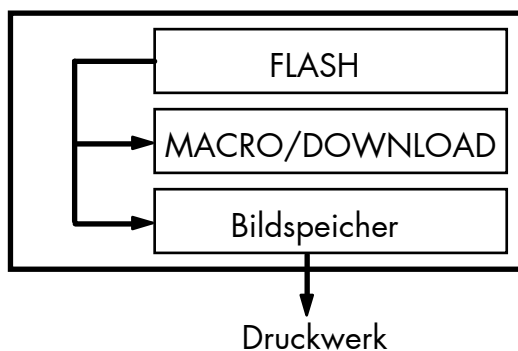
1. Normalbetrieb:



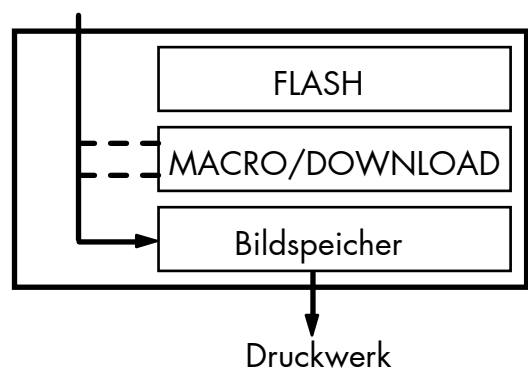
2. FLASH-Eprom programmieren:



3. Einschalten:



4. Normalbetrieb:



Beispiel:

&%%&%bf00Dieser Text wird in das FLASH programmiert.
Nach jedem Einschalten wird der Inhalt des FLASH-Speichers in
den MACRO-DOWNLOAD-Speicher übertragen.
Für dieses Beispiel bedeutet das, daß dieser Text nach jedem
Einschalten gedruckt wird. &%%&%bf

Beachte: Mit den Bitmap-Befehlen ist ein direkter Zugriff zum Bildspeicher des Druckers möglich.

Die Daten werden als ASCII - kodierte Zeichen übertragen. Jedes gesetzte Bit im Bildspeicher entspricht einem Punkt.

Leerschritte zwischen den Datenbytes sind verboten.

Wird mit der Bildinformation die Größe des Eingangspuffers überschritten, so wird auf eine Stausicherung verzichtet.

Beispiel: Es wird ein Raster mit Vergrößerungsfaktor gedruckt.
Vergrößerungsfaktor (Breite 4, Höhe 4) gedruckt.

```
&%%m441250250000640011 FOFOFOFOFOFOFOFOFOFOFOFOF  
FOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOF  
OFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOF  
FOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOFOF
```



7.3. Neuer BITMAP-Befehl (Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen)



Funktion: Dies ist ein neuer Befehl zum Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen. Dieser Befehl bietet Vorteile gegenüber den Befehlen <ESC><ESC>M und <ESC><ESC>m . So erlaubt der neue Befehl z.B. die Verwendung des Kompressionsmodes PCL.

Syntax: <ESC><ESC>ti <k₁nn> ... <k₉₉Bilddaten>
(27)(27)(116)(105) ...

Parameter: Jeder Parameter (wie Grafikformat, Kompressionsmode, Auflösung...) enthält eine 2-stellige Typkennung gefolgt von einem entsprechenden n-stelligen Parameter.

<k_xnn> : k_x = Parametertyp (2-stellig)

Zulässige Werte für x:

x = 01	: Grafikformat	nn	= 2-stellig
x = 90	: PCL Start der Rasterdaten		
x = 91	: PCL Kompressionsmode	n	= 1-stellig
x = 92	: Bitmap Auflösung	nnnn	= 4-stellig
x = 98	: PCL Ende der Rasterdaten		
x = 99	: Ende der Parameter-Kette (letzte Typkennung), unmittelbar danach <u>müssen Bilddaten folgen</u> (siehe Beispiele auf den folgenden Seiten).		

Hinweise: Weitere Details zu den Parametern finden Sie in den folgenden Abschnitten. Alle 90er -Typkennungen beenden den ti Befehl.

Hinweise: Wenn die **Raster Grafik** (das Bild) eine Angabe zur **Auflösung** enthält (z.B. TIFF-Bilder enthalten solche Auflösungsinformationen, PCX-Bilder nicht), dann hängt die Größe des Bildes beim Ausdrucken nicht von der Auflösung des Druckers ab.

Der **Textcursor** ist die Startposition für die Platzierung und das Ausfüllen eines Druckfensters mit Pixel-Informationen: Der Textcursor zeigt auf die **linke, obere Ecke der Raster Grafik**. Die Ausrichtung der Grafik ist abhängig von der Schreibrichtung (siehe Befehl <ESC><ESC>C).

Nach der Fertigstellung eines TIF oder PCX –Bildes zeigt der Textcursor unverändert auf die Startposition. Bei PCL –Bildern dagegen wandert der Cursor zeilenweise nach unten (wobei er links bleibt).

Grafikformat (k₁):

02	=	PCX
03	=	HP Modes (PCL)
04	=	TIFF CCITT Group 4

PCL Kompressionsmode (k₉₁):

0	=	unkodiert (default)
1	=	Laufängerkodierung (Run-Length Encoding)
2	=	Tagged Image File Format (TIFF Fax 4)
3	=	Delta Row
5	=	Adaptive Kompression

Bitmap Auflösung (k₉₂):

0075	=	75 dots per inch
0100	=	100 dots per inch
0150	=	150 dots per inch
0200	=	200 dots per inch
0300	=	300 dots per inch
0600	=	600 dots per inch

Beispiele:

1. Drucken eines **Tiff -Bildes**:

```
&%&%ti 01 04 99Bilddaten
```

Beachte: Parameter k92 wird nicht verwendet, weil in TIFF-Bilddaten Informationen zur Auflösung enthalten sind.

2. Drucken eines **Bildes via PCL**:

```
&%&%ti 90                &%&%co2 Start der Rasterdaten
&%&%ti 91 02            &%&%co2 Kompressionsmode 2
&%&%ti 92 0600         &%&%co2 Auflösung 600 dpi
&%&%ti 01 03 0135 99Daten der ersten Bildzeile
&%&%ti 01 03 0132 99Daten der zweiten Bildzeile
                    {
                    Anzahl Bytes (4 stellig)
```

. . .

```
&%&%ti 01 03 0141 99Daten der letzten Bildzeile
&%&%ti 98                &%&%co2 Ende der Rasterdaten
```


8. Funktionsgruppe BARCODE

Barcodes oder Strichcodes werden heute überall zum Automatisieren von Abläufen verwandt. Die Breite der schwarzen Balken und der Lücken kodiert die jeweilige Information, welche dann von Barcodescannern gelesen werden kann.

Entsprechend ihrem Aufbau unterscheidet man eindimensionale und zweidimensionale Barcodes.

8.1. Übersicht: Generieren eines BARCODES

bc

Funktion: Durch den Befehl lassen sich Barcodes der verschiedensten Typen flexibel parametrisieren.

Syntax: `<ESC><ESC>bc <k1nn> ... <k9"cc">`
(27)(27)(98)(99) ...

Parameter: Jeder der Parameter k_1 bis k_9 (wie Typ, Höhe, Breite, ...) besitzt eine einstellige Parametertyp-Kennung, gefolgt von dem dazugehörigen mehrstelligen Parameter. Dadurch ist es möglich, den Barcode einmal zu initialisieren, um später nur noch die Barcodedaten zu definieren.

`<kxnn>` : k_x = Parametertyp (1 - stellig)

Zulässige Werte für x:

x = 1	: Barcodetyp	nn	= 2-stellig
x = 2	: Barcodeflag	hh	= 2-stellig (Hex)
x = 3	: Barcodehöhe	n ₄	= 4-stellig
x = 4	: Breite, schmaler Balken	n ₄	= 4-stellig
x = 5	: Breite, breiter Balken	n ₄	= 4-stellig
x = 6	: Barcodehöhe, Trennzeichen	n ₄	= 4-stellig
x = 7	: Schmale Barcodelücke	n ₄	= 4-stellig
x = 8	: Breite Barcodelücke	n ₄	= 4-stellig
x = 9	: Barcodedaten (terminiert)	"cc"	= String

Hinweis: Die Parameter haben bei den verschiedenen Barcodetypen zum Teil etwas andere Bedeutungen. Aus diesem Grund sind in den folgenden Abschnitten weitere Einzelheiten zu diesen Parametern beschrieben.

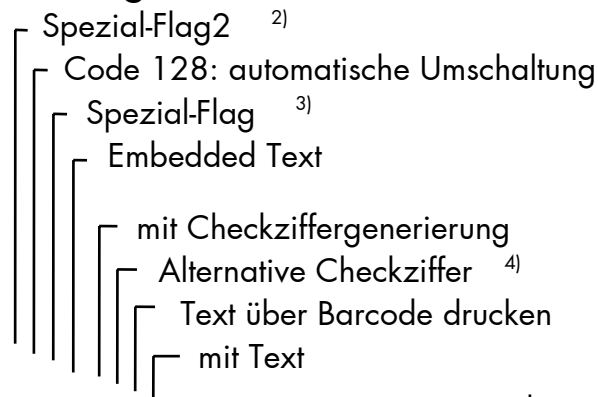
Barcodetyp (k₁):	1	=	2/5 Interleaved (USD-1)
	2	=	2/5 5 - Strich Industrie
	3	=	2/5 3 - Strich Datalogic
	4	=	2/5 3 - Strich Matrix
	5	=	Code 39 (3/9) (USD-3)
	6	=	Codabar (USD-4)
	7	=	EAN 13 (IAN 13, JAN 13)
	8	=	EAN 8 (IAN 8, JAN 8)
	9	=	MSI
	10	=	Code 128 (USD-6, EAN 128) [A,B,C], UCC 128
	11	=	UPC A
	12	=	UPC E
	13	=	EAN 13 + Add On (2 bzw. 5)
	14	=	EAN 8 + Add On (2 bzw. 5)
	15	=	PDF 417
	16	=	USD-5
	17	=	Postnet
	18	=	DataMatrix *1)
	19	=	KIX
	20	=	MaxiCode (UPS) *1)
	21	=	Code 11
	22	=	Code 93
	23	=	UPC-A+ add on
	24	=	UPC-E + add on
	25	=	FIM
	26	=	Planet
	27	=	UPS One Code
	28	=	Aztec *1)
	29	=	QR-Code *1)

*1) Option

Barcodeflag (k₂):

Das Barcodeflag ist als zweistellige ASCII-kodierte Hexadezimalziffer anzugeben.

Bit-Kodierung:



Hexadezimal:

Beispiele:

0000 0000	00 = ohne Text, ohne Checkziffer
0000 0001	01 = mit Textunterdruck
0000 0011	03 = mit Textüberdruck
0000 1000	08 = mit Checkziffergenerierung
0000 1001	09 = mit Textunterdruck und Checkziffer
0000 1011	0B = mit Textüberdruck und Checkziffer

²⁾ Details zum Spezial-Flag2:

Code 128: Textunterdruck für DPD

³⁾ Details zum Special-Flag:

Code 128: FNC1 Steuerzeichen einfügen
 Code 39: bei Textdruck Start/Stoppszeichen mitdrucken
 EAN: mittleren Trennbalken nicht verlängern
 Alle 2-dim. Barcodes: Barcode oberhalb der Schreiblinie

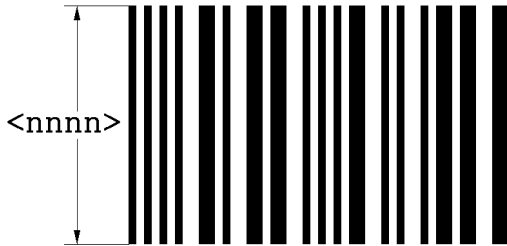
⁴⁾ Details zur Alternativen Checkziffer:

2/5 Interleaved: wird als Deutscher Postcode formatiert
 MSI: erzeugt 3 verschiedene Checkzifferkombinationen
 Code 128: fügt zusätzlich noch Modulo-10 Checkziffer ein
 Code 11: zweite Modulo-11 Checkziffer

Barcodehöhe (k₃):

<nnnn> : 0000 ... 9999 = Höhe des
Barcodes in 1/300 inch ⁵⁾

Barcode ohne Textunterdruck



Barcode mit Textunterdruck



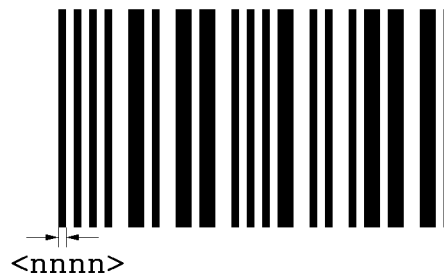
Wenn ein Textunterdruck erzeugt wird, ist die Höhe des Barcodes um 25 dots reduziert.

EAN und UPC Barcodes mit Textunterdruck



Breite schmaler Balken (k₄):

<nnnn> : 0000 ... 9999 = Breite in 1/300 inch ⁵⁾



⁵⁾ Beispiel: Bei Verwendung eines 300 dpi Druckers (passend zur IDOL Standardauflösung) führt ein Wert von 900 zu einer Barcodehöhe von 3 Zoll (76.2mm).

Breite breiter Balken (k_5): <nnnn> : 0000 ... 9999 = Breite in 1/300 inch ⁵⁾



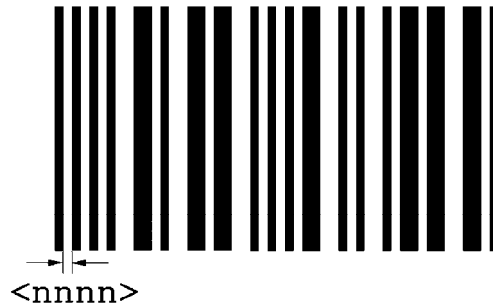
Hinweis: Bei den Mehrbreiten-Barcodes wird die Definition des breiten Balkens für den 2-fach breiten Balken herangezogen. Alle anderen Balkenbreiten werden in einem festgelegten Verhältnis vom schmalen Balken abgeleitet.

Trennzeichenhöhe (k_6): <nnnn> : 0000 ... 9999 = Höhe in 1/300 inch ⁵⁾
(EAN und UPC)



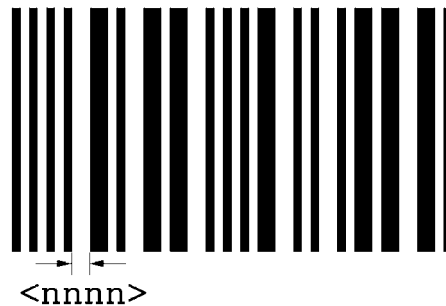
Hinweis: Wird die Trennzeichenhöhe nicht definiert, so ist sie identisch mit der Barcodehöhe.

Breite der schmalen Lücke (k7): <nnnn> : 0000 ... 9999 = Lückenbreite in 1/300 inch ⁵⁾



Hinweis: Werden die Angaben für die Lückenbreiten nicht vorgegeben, so entsprechen sie den Werten für die Balken.

Breite der breiten Lücke (kg): <nnnn> : 0000 ... 9999 = Lückenbreite in 1/300 inch ⁵⁾



Hinweis: Werden die Angaben für die Lückenbreiten nicht vorgegeben, so entsprechen sie den Werten für die Balken.

Barcode-Daten (kg): Die Daten, die in einen Barcode umgesetzt werden sollen, sind als Textstring anzugeben. Die Länge der Barcodedaten ist flexibel. Deshalb werden die Barcodedaten mit Terminierungszeichen "geklammert". **Das erste Zeichen des Parameters <kg> definiert das Terminierungszeichen. Dieses Terminierungszeichen ist frei wählbar.** Es muß lediglich sichergestellt werden, daß dieses Terminierungszeichen nicht in den Barcodedaten vorkommt.

Beispiel:

In diesem Beispiel wird das Zeichen " als Terminierungszeichen verwendet:

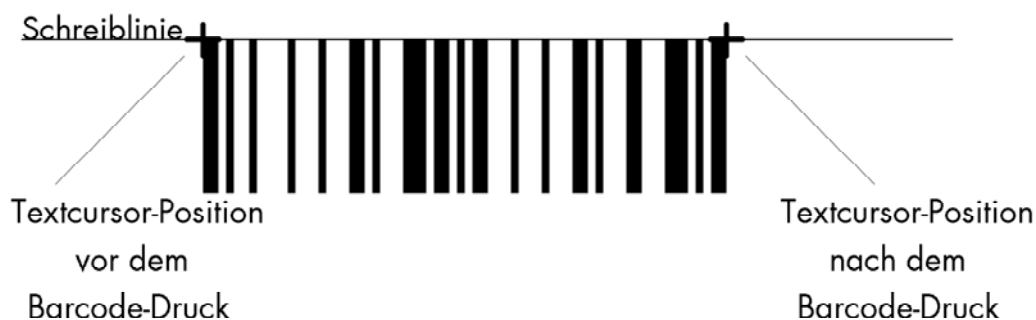
`<ESC><ESC>bc ...9"8834573456"`

The diagram shows the string `<ESC><ESC>bc ...9"8834573456"` with brackets and labels:

- A bracket from the first double quote to the last double quote is labeled "Daten".
- A bracket from the first double quote to the end of the string is labeled "Terminierungszeichen (Anfang Daten)".
- A bracket from the last double quote to the end of the string is labeled "Terminierungszeichen (Ende Daten)".

Hinweise zur Textcursor-Position und zum Textunterdruck:

Der Barcode wird - genau wie Text - abhängig von der Schreibrichtung generiert. Der Textcursor befindet sich in der linken oberen Ecke (d.h. der Barcode wird unterhalb der Schreiblinie nach rechts erzeugt). Anschließend befindet sich der Textcursor am Ende des Barcodes.



Der Unterdruck wird automatisch zum Barcode zentriert.

Der Unterdruck wird mit der aktuellen Schrift und allen eingestellten Schriftparametern erzeugt.

Beispiel:

Um einen Interleaved 2/5 Barcode zu erzeugen, werden beim ersten Mal alle Parameter mit folgender Sequenz angegeben:

```
&%&%bc 101 200 30380 40008 50016 9*4911119844*
```

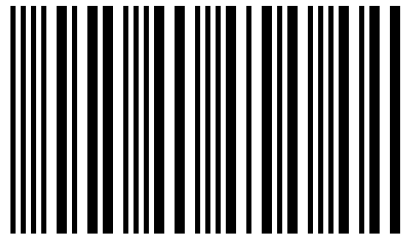
Später reicht dann eine Sequenz, in der nur die veränderten Parameter (hier die Barcode-Daten) angegeben werden:

```
&%&%bc 9*4911119845*
```

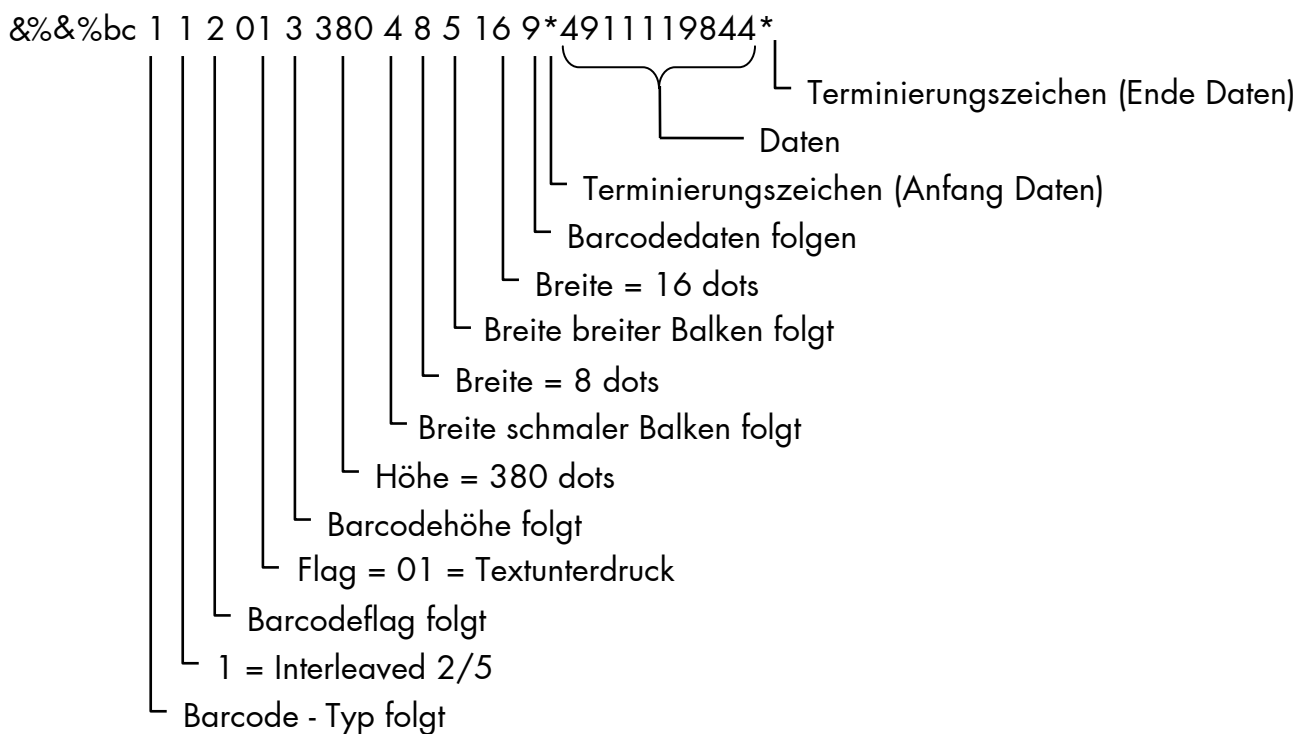

Die Eingabe von führenden Nullen zur Darstellung der vollen Stellenzahl der Parameter kann durch die Verwendung einer Leertaste vermieden werden (z.B. k₄: 4 8 statt 40008).

Beispiel: Barcode Interleaved 2/5 mit dem Wert 4911119844:

&%&%bc 1 1 2 01 3 380 4 8 5 16 9*4911119844*



4911119844



8.2. Eindimensionale Barcodes

Innerhalb der Gruppe der eindimensionalen Barcodes unterscheidet man zwischen Zweibreiten- und Mehrbreiten-Codes.

Bei **Zweibreiten-Barcodes** können innerhalb einer Strichcodierung maximal zwei verschiedene Breiten für Striche und Lücken auftreten. Die Breiten für Striche und Lücken stehen im Verhältnis 2:1 bis 3:1.

Strichcodierungen mit mindestens 3 Breiten (Striche und Lücken) werden **Mehrbreiten-Barcodes** genannt. Bei diesen Codearten sind höhere Anforderungen an die Qualität von Drucker und Lesegerät zu stellen.

Bei **Höhenmodulierten-Barcodes** sind Striche und Lücken gleich breit, dafür wird mit verschiedenen hohen Strichen gearbeitet.

In den folgenden Abschnitten werden nur die für den jeweiligen Barcodetyp charakteristischen Parameter näher erläutert. Ansonsten gelten die Parameterfestlegungen aus Kapitel 8.1 (Übersicht: Generierung eines BARCODES).

8.2.1. 2/5 Interleaved (USD-1) (Zweibreiten-Barcode)

Diese Barcodes werden verwendet zur Erzeugung eines sogenannten Ident- und Leitcodes (Deutsche Post).

Barcodetyp (k₁):

<nn> : 01 : Barcodetyp 2/5 Interleaved (USD-1)

Barcodeflag (k₂):

Bit-Kodierung (vgl. Kapitel 8.1):

	┌	Bit für alternative Checkziffer gesetzt		
0000 1100	<nn>	: 0C	:	erzeugt Checkziffer für deutschen Postbarcode
0000 1101	<nn>	: 0D	:	mit Checkziffer und Textunterdruck, formatiert für deutschen Postbarcode

Beispiele:

2/5 Interleaved Barcodes mit Checkziffer und Textunterdruck:

Zeichenfolge „56310243031“

&%&%bc 101 20D 30380 40008 50016 9*56310243031*



56.310 243.031 3

Zeichenfolge „2134807501640“



21348.075.016.40 1

Beispiele für verschiedene 2/5 Barcodes:

Darstellung der Zeichenfolge „0123456789“ mit Unterdruck:

1. 2/5 5 - Strich Industrie

&%%bc 102 201 30120 40005 50012 9#0123456789#



0123456789

2. 2/5 3 - Strich Datalogic

&%%bc 103 201 30120 40005 50015 9*0123456789*



0123456789

3. 2/5 3 - Strich Matrix

&%%bc 104 201 30120 40005 50010 9"0123456789"



0123456789


8.2.2. Code 39 (3/9) (USD-3) (Zweibreiten-Barcode)

Barcodetyp (k₁):

<nn> : 05 : Barcodetyp 39 (3/9) (USD-3)

Barcodeflag (k₂):

Bit-Kodierung (vgl. Kapitel 8.1):



 0010 0001 <nn> : 21 : Textunterdruck mit Darstellung Start/Stop-Zeichen *

Hinweis: Bei diesem Barcodetyp sind generell am Anfang und am Ende Start/Stop-Zeichen (Sternchen) zu verwenden. Über die Bit-Kodierung wird festgelegt, ob diese Zeichen (*) im Barcodedruck mit dargestellt werden.

Beispiele:

Darstellung der Zeichenfolge „0123456789“ mit Unterdruck und Checkziffer:

&%&%bc 105 209 30120 40005 50010 9*0123456789*



01234567892

Darstellung der Zeichenfolge „0123456789“ mit Unterdruck, Checkziffer und Start/Stop-Zeichen:

&%&%bc 105 229 30120 40005 50010 9*0123456789*



01234567892

8.2.4. Code 128 (Mehrbreiten-Barcode)

Barcodetyp (k₁):

<nn> : 10 : Barcodetyp Code 128

Hinweis:

Der Code 128 kennt drei verschiedene Zeichensätze (A, B und C). Der Zeichensatz A dient zur Kodierung der ASCII Zeichen 00 bis 5F (hexadezimal), der Zeichensatz B für die ASCII Zeichen 20 bis 7F und der Zeichensatz C für die ASCII Zeichen 30 bis 39 (die Ziffern 0 bis 9, vergleiche Anhang I: Code – Übersetzungstabelle). Zur Bestimmung des Zeichensatzes muß das erste Zeichen in den Barcode-Daten den Startcode enthalten!

Hex. Entsprechendes ASCII Zeichen:

67	g	= Startcode Zeichensatz A
68	h	= Startcode Zeichensatz B
69	i	= Startcode Zeichensatz C

Beispiel:

&%&%bc ... 9" g7365945"

Die Verwendung von 'g' als erstem Zeichen der Barcodedaten aktiviert den Zeichensatz A.

Barcodeflag (k₂):

Bit-Kodierung (vgl. Kapitel 8.1):

			Bit EAN 128
			Bit Checkziffergenerierung
			Bit alternative Checkziffer
0010 0000	<nn>	: 20	: am Anfang wird automatisch FNC1 eingefügt (FNC1 kennzeichnet einen EAN 128 Barcode)
0100 0000	<nn>	: 40	: automatischer Zeichensatzwechsel
0000 1100	<nn>	: 0C	: eine Modulo 10 Checkziffer wird vor der regulären Modulo 103 Checkziffer eingefügt

Einfügen von Kontrollcodes:

Bei der Interpretation der Barcodedaten gibt es zwei Interpreter-Ebenen:

1. Zeichen-Interpretation.
Auf dieser Ebene werden die Zeichen in Barcode umgesetzt.
2. Kontrollzeichen-Ebene.
Auf dieser Ebene kann die Interpretation der Barcodezeichen beeinflusst werden (z.B. Wechseln des Zeichensatzes innerhalb des Barcodes).

Der Zeichen-Interpreter befindet sich standardmäßig auf der Zeichen-Interpretationsebene. Um die Interpreter-Ebene innerhalb der Barcodedaten zu wechseln, sind folgende nicht druckbare Zeichen in die Barcodedaten „einzubauen“:
(vgl. Anhang I: Code – Übersetzungstabelle)

Hex.	ASCII	
OE	SO	: Schaltet auf die Kontrollzeichen-Ebene.
OF	SI	: Schaltet zurück auf die Zeichen-Interpretationsebene.

Auf der Kontrollzeichen-Ebene stehen folgende Kontrollzeichen zur Verfügung:

!	= FNC1	%	= Zeichensatz A
"	= FNC2	&	= Zeichensatz B
#	= FNC3	'	= Zeichensatz C
\$	= FNC4	(= SHIFT

Beispiel: Innerhalb der Barcodedaten wird der Zeichensatz gewechselt

```
&%&%bc .....9"i223488<SO>&<SI>abcdefg"
```



hier sind die Codes der nicht druckbaren Zeichen zu verwenden

Automatischer Zeichensatzwechsel:

Beim Code 128 kann ein automatischer Wechsel zwischen Zeichensatz B und C aktiviert werden. Dies bewirkt einen Wechsel in den Zeichensatz C, wenn mehr als 4 aufeinander folgende Ziffern (numerische Daten) entdeckt werden, ansonsten wird im Zeichensatz B kodiert. Vorteil der Kodierung im Zeichensatz C ist, daß je 2 Ziffern zu einem Daten-Zeichen codiert werden (Reduzierung der Barcodelänge).

Einschalten des automatischen Zeichensatzwechsels:

Barcodeflag (k₂) : 40 = automatischer Zeichensatzwechsel.

Achtung: Ist der automatische Zeichensatzwechsel aktiviert, kann nicht mehr in die Kontrollzeichen-Ebene gewechselt werden. Die Befehle dazu haben keine Wirkung.

Beispiele:

Darstellung von Zeichenfolgen mit Unterdruck:

- Code 128 (Zeichensatz A)

&%&%bc 110 209 30120 40010 50020 60150 9"gCODE"



CODE

- Code 128 (Zeichensatz B)

&%&%bc 110 209 30120 40010 50020 60150 9"hcode123456"



code123456

3. Code 128 (Zeichensatz C)

Zeichensatz C wird verwendet zur Kodierung von Ziffernpaaren.

&%&%bc 110 209 30120 40010 50020 60150 9"i123456"



123456

4. Code 128 (Zeichensatz B mit autom. Wechsel zum Zeichensatz C)

Zeichensatz C wird verwendet zur Kodierung von Ziffernpaaren.

&%&%bc 110 249 30120 40010 50020 60150 9"icode123456"



code123456

8.2.5. EAN 8, EAN 13 und UPC A, UPC E (Mehrbreiten-Barcode)

Die UPC A und UPC E Barcodes sind die amerikanischen Varianten der EAN 8 und EAN 13 Barcodes.

(EAN European Article Numbering)

(UPC Universal Product Code)

Barcodetyp (k_1):

<nn>	: 7	: Barcodetyp EAN 13 (IAN 13, JAN 13)
<nn>	: 8	: Barcodetyp EAN 8 (IAN 8, JAN 8)
<nn>	: 11	: Barcodetyp UPC A
<nn>	: 12	: Barcodetyp UPC E

Barcodedaten (k_9):

Die Länge der Barcodedaten ist bei diesen Barcodetypen fest

EAN 8 = 8 Stellen,

EAN 13 = 13 Stellen

UPC E = 8 Stellen,

UPC A = 12 Stellen.

Wird die Checkziffergenerierung gewählt (vgl. Kapitel 8.1 Kodierung Barcodeflag k_2), reduziert sich die Daten-Stellenzahl um 1 Stelle.

Beispiele:

Darstellung von Zeichenfolgen mit Textunterdruck und Checkziffer:

1. EAN 8

&%&%bc 108 209 30120 40005 60145 9"1234123"



2. EAN 13

&%&%bc 107 209 30120 40005 60145 9"412345123456"

**Beachte:**

Bei diesem Barcode darf der Barcode-Befehl nicht direkt am linken Rand positioniert werden, da sonst der Textunterdruck nicht richtig positioniert wird.

3. UPC E

&%&%bc 112 209 30220 40008 60250 9*0123456*

**Beachte:**

Bei diesem Barcode darf der Barcode-Befehl nicht direkt am linken Rand positioniert werden, da sonst der Textunterdruck nicht richtig positioniert wird.

4. UPC A

&%&%bc 111 208 30120 40005 60120 9"37937831782"



8.2.6. EAN 8 Add On und EAN 13 Add On (Mehrbreiten-Barcode)

Barcodetyp (k₁): (EAN European Article Numbering)

<nn> : 13 : Barcodetyp EAN 13 + Add On (2 bzw. 5)
 <nn> : 14 : Barcodetyp EAN 8 + Add On (2 bzw. 5)

Barcodedaten (k₉):

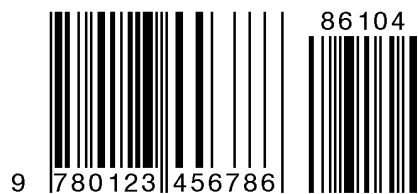
Überschreitet die Länge der Barcodedaten bei den Barcodetypen EAN 8 und EAN 13 die festgelegte Stellenzahl um 2 oder 5 Stellen, so wird ein entsprechender Barcode mit Add On generiert. Werden z.B. bei einem EAN 13 Barcode 18stellige Barcodedaten geschickt, so wird ein 13stelliger EAN mit 5stelligem Add On generiert.

Wird die Checkziffergenerierung gewählt (vgl. Kapitel 8.1 Kodierung Barcodeflag k₂) reduziert sich die Daten-Stellenzahl um 1 Stelle.

Beispiele:

EAN 13 Add On ohne Checkziffer

&%&%bc 113 201 30300 40004 60340 9"978012345678686104"



Beachte:

Bei diesem Barcode darf der Barcode-Befehl nicht direkt am linken Rand positioniert werden, da sonst der Textunterdruck nicht richtig positioniert wird.

EAN 8 Add On ohne Checkziffer

&%&%bc 114 201 30120 40005 60145 9"1234123847"



8.2.7. Postnet, Planet, Kixbar und OneCode (Höhenmodulierte Barcodes)

Hauptanwendungsgebiet dieser Barcodes sind Postbarcodes für englischsprachige Länder.

Barcodetyp (k_1):

<nn>	: 17	: Barcodetyp Postnet
<nn>	: 19	: Barcodetyp KIX
<nn>	: 26	: Barcodetyp Planet
<nn>	: 27	: Barcodetyp OneCode (UPS)

Hinweis:

Bei fehlenden Angaben zu Breite und Höhe der Balken wird automatisch die Standardgröße erzeugt.
Für abweichende Größenfestlegungen werden die Parameter k_3 und k_6 verwendet:

Höhe kleiner Balken (k_3):

<nnnn> : 0000 ... 9999 : Höhe kleiner Balken in dot.

Höhe großer Balken (k_6) (Trennzeichenhöhe):

<nnnn> : 0000 ... 9999 : Höhe großer (oder mittlerer) Balken in dot.

Barcodedaten (k_9):

Die Angabe der Barcodedaten erfolgt als Zeichenkette.

Zeichenvorrat:	Postnet:	0 bis 9
	KIX:	0 bis 9 und A bis Z
	OneCode:	0 bis 9


Typische Datenlänge:	Postnet und Planet:	5, 6 oder 9 Zeichen
	OneCode:	20, 25, 29 oder 31 Ziffern

Beispiele:

Darstellung von Zeichenfolgen mit Textunterdruck:

1. Postnet mit Checkziffer

&%&%bc 117 209 9"433062144"



4330621443

2. Planet

&%&%bc 126 209 9"433062144"



4330621443

3. KIX ohne Checkziffer

&%&%bc 119 201 9"0123456789ABCDEFGHIJKLMNZ"



0123456789ABCDEFGHIJKLMNZ

4. OneCode

Beispiel mit 20 Ziffern Datenlänge:


&%&% bc 127 201 9"01234567094987654321"



01234567094987654321

Beispiel mit 31 Ziffern Datenlänge:

&%&% bc 127 201 9"0123456709498765432101234567891"



0123456709498765432101234567891

8.3. Zweidimensionale Barcodes

In den folgenden Abschnitten werden nur die für den jeweiligen Barcodetyp charakteristischen Parameter näher erläutert. Ansonsten gelten die Parameterfestlegungen aus Kapitel 8.1 (Übersicht: Generierung eines BARCODES).

8.3.1. PDF 417

Barcodetyp (k_1):

<nn> : 15 : Barcodetyp PDF 417

Sicherheitsstufe (k_2):

<nn> : 00 ... 08 : Sicherheitsstufe

Der PDF 417 Barcode kann in verschiedenen Sicherheitsstufen erzeugt werden. Mit steigender Sicherheitsstufe werden im Barcode mehr Daten zur Fehlerkorrektur integriert. Dies bedeutet, daß der Barcode mit steigender Sicherheitsstufe größer wird. Gleichzeitig können jedoch Fehler, die durch äußere Einflüsse auftreten (z.B. zerkratzen oder beschriften des Barcodes), sicherer korrigiert werden.

Höhe einer Zeile (k_3):

<nnnn> : 0000 ... 9999 : Höhe des Balkens einer Zeile in dot.

Der Standardwert für die Höhe des Balkens einer Zeile ist 9 dot. Wird der Parameter mit '0000' angegeben, wird der Standardwert verwendet. Andere Werte überschreiben diesen Standardwert.

Breite schmaler Balken (k₄):

<nnnn> : 0000 ... 9999 : Breite eines schmalen Balkens in dot.

Der Standardwert für einen schmalen Balken ist 3 dot.
Wird der Parameter mit '0000' angegeben, wird der Standardwert verwendet. Andere Werte überschreiben diesen Standardwert.

Anzahl Druckspalten (k₅):

<nnnn> : 0000 ... 0030 : Anzahl der gewünschten Druckspalten.

Wird der Parameter mit '0000' angegeben, wird die Mindestanzahl der benötigten Druckspalten automatisch berechnet. Ist die gewählte Spaltenzahl nicht realisierbar, wird die Spaltenzahl auf den nächstmöglichen Wert abgeglichen.

Länge der Barcodedaten (k₆):

<nnnn> : 0000 ... 9999 : Länge der Barcodedaten in Bytes.

Mit diesem Parameter k₆ kann die Länge der Barcodedaten in Bytes angegeben werden. Seine Verwendung ist immer dann ratsam, wenn kein Zeichen als Terminierungszeichen verwendet werden kann. Dies ist der Fall, wenn alle verfügbaren Zeichen in den Barcodedaten vorkommen oder der Anwender nicht sicher sein kann, daß sein Terminierungszeichen nicht in den Barcodedaten auftaucht.

Ist dagegen ein Zeichen verfügbar, so können die Daten des PDF-Barcodes (wie bei den anderen Barcodes) durch Terminierungszeichen geklammert werden. Der Parameter k₆ wird dann weggelassen; das erste Zeichen der Barcodedaten definiert das Terminierungszeichen, welches dann auch am Ende der Barcodedaten verwendet wird. Dieses Terminierungszeichen darf also in den Barcodedaten nicht vorkommen.

Barcodedaten (k9):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Beispiele:

Der folgende Barcode wird hier nacheinander mit denselben Nutzdaten aber mit unterschiedlichen Sicherheitsstufen ($k_2 = 0, 2$ und 8) gedruckt. Anhand dieses Beispiels ist gut zu erkennen, wie sich der Barcode mit steigender Sicherheitsstufe vergrößert.

k₁: Barcode : PDF 417
k₃: Höhe : Standard
k₄: Breite : Standard
k₅: Anzahl Druckspalten : 3
k₆: Datenlänge : 7 Bytes
k₉: Daten : BARCODE

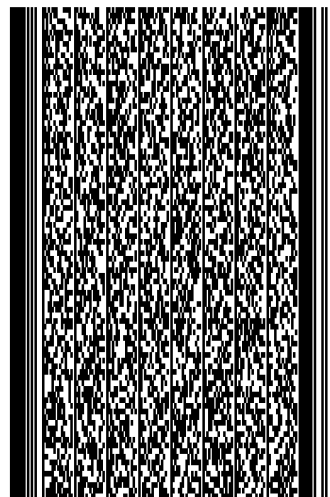
&%%bc 1 15 2 00 3 0 4 0 5 3 6 7 9BARCODE



&%%bc 1 15 2 02 3 0 4 0 5 3 6 7 9BARCODE



&%%bc 1 15 2 08 3 0 4 0 5 3 6 7 9BARCODE



8.3.2. USD-5

Barcodetyp (k₁):

<nn> : 16 : Barcodetyp USD-5

Parameter:

Barcodedaten (k₉):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Zeichenvorrat: 0 bis 9

Datenlänge (Anzahl Ziffern pro Barcode): 5, 7, 9, 11, 14, 17, 20

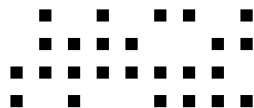
Beachte:

Alle anderen Parameter haben bei diesem Befehl keine Wirkung. So ist die Größe der Quadrate vorgegeben mit 1,7 mm mal 1,7 mm, der Abstand der Quadrate beträgt 4,0 mm (in beiden Richtungen).

Beispiel:

USD-5 Barcode mit einer Datenlänge von 5 Ziffern:

&%&%bc 1 16 9*12345*



8.3.3. DataMatrix

Barcodetyp (k₁):

<nn> : 18 : Barcodetyp DataMatrix

Variante (k₂): z.Zt. ist nur der Standardwert ('00') zulässig.
In diesem Fall wird der ECC 200 - Barcode erzeugt.

Höhe und Breite eines Symbolelements (Modul, „Pad“) (k₃):

<nnnn> : 0000 ... 9999 : Höhe und Breite des Symbolelements.

Der Standardwert für die Höhe und die Breite eines Symbolelements ist 3 dot. Wird '0000' als Wert für den Parameter k₃ angegeben, wird der Standardwert verwendet. Andere Werte überschreiben diesen Standardwert.

Anzahl Druckreihen (k₄):

<nnnn> : 0000 ... 9999 : Anzahl der gewünschten Druckreihen.

Wird '0000' als Wert für den Parameter k₄ angegeben, so optimiert die Software automatisch die Abmessungen des Barcodes. Wird dagegen ein Wert vorgegeben, der nicht umgesetzt werden kann, so wird kein Barcode erzeugt (daher besser weglassen).

Anzahl Druckspalten (k₅):

<nnnn> : 0000 ... 0030 : Anzahl der gewünschten Druckspalten.

Wird '0000' als Wert für den Parameter k₅ angegeben, so optimiert die Software automatisch die Abmessungen des Barcodes. Wird dagegen ein Wert vorgegeben, der nicht umgesetzt werden kann, so wird kein Barcode erzeugt (daher besser weglassen).

Länge der Barcodedaten (k_6):

<nnnn> : 0000 ... 9999 : Länge der Barcodedaten
in Bytes.

Mit diesem Parameter k_6 kann die Länge der Barcodedaten in Bytes angegeben werden. Seine Verwendung ist immer dann ratsam, wenn kein Zeichen als Terminierungszeichen verwendet werden kann. Dies ist der Fall, wenn alle verfügbaren Zeichen in den Barcodedaten vorkommen oder der Anwender nicht sicher sein kann, daß sein Terminierungszeichen nicht in den Barcodedaten auftaucht.

Ist dagegen ein Zeichen verfügbar, so können die Daten des Data Matrix-Barcodes (wie bei den anderen Barcodes) durch Terminierungszeichen geklammert werden. Der Parameter k_6 wird dann weggelassen; das erste Zeichen der Barcodedaten definiert das Terminierungszeichen, welches dann auch am Ende der Barcodedaten verwendet wird. Dieses Terminierungszeichen darf also in den Barcodedaten nicht vorkommen.

Barcodedaten (k_9):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Beispiel:

&%%bc 118 200 30016 40016 50016 6000 9*60113122000098403*



8.3.4. MaxiCode (UPS)

Barcodetyp (k_1):

<nn> : 20 : Barcodetyp MaxiCode (UPS)

Barcodemode (k_2):

<nn> : 02, 03, 04, 06 : verfügbare Modi

Länge der Barcodedaten (k_6):

<nnnn> : 0000 ... 0200 : Länge der Barcodedaten
in Bytes.

Mit diesem Parameter k_6 kann die Länge der Barcodedaten in Bytes angegeben werden. Seine Verwendung ist immer dann ratsam, wenn kein Zeichen als Terminierungszeichen verwendet werden kann. Dies ist der Fall, wenn alle verfügbaren Zeichen in den Barcodedaten vorkommen oder der Anwender nicht sicher sein kann, daß sein Terminierungszeichen nicht in den Barcodedaten auftaucht.

Ist dagegen ein Zeichen verfügbar, so können die Daten des MaxiCode -Barcodes (wie bei den anderen Barcodes) durch Terminierungszeichen geklammert werden. Der Parameter k_6 wird dann weggelassen; das erste Zeichen der Barcodedaten definiert das Terminierungszeichen, welches dann auch am Ende der Barcodedaten verwendet wird. Dieses Terminierungszeichen darf also in den Barcodedaten nicht vorkommen.

Barcodedaten (k_9):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Aufbau des UPS MaxiCode:

```
"PostalCode CountryCode Service SecondaryMessages<EOT>"
```

Die verschiedenen Codes können durch die von UPS vorgeschriebenen nicht druckbaren Zeichen GS oder RS getrennt werden. Am Ende der Barcodedaten muß ein EOT stehen:

(vgl. Anhang I: Code – Übersetzungstabelle)

Hex. ASCII

1D GS : Trennzeichen

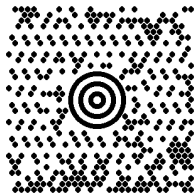
1E RS : Trennzeichen

04 EOT : End of Text

Weitere Informationen finden Sie in den internationalen Spezifikationen zum MaxiCode.

Beispiel:

&%&%bc 120 204 6000 9*068107317 840 001 THIS IS A test_message<EOT>*



8.3.5. Aztec

Bei diesem zweidimensionalen Barcode befinden sich im Zentrum mehrere konzentrische Quadrate (zentrales Erkennungsmuster, „Bull-Eye“), darum herum sind die weiteren quadratischen Symbolelemente angeordnet.

Barcodetyp (k_1):

<nn> : 28 : Barcode Typ Aztec

Barcodemode (k_2):

<nn> : 00 ... 03 : verfügbare Modi

- | | | |
|---|---|--|
| 0 | = | automatisch |
| 1 | = | kompakt (falls möglich, dann hat das zentrale Erkennungsmuster 2 quadratische Ringe) |
| 2 | = | „full range“ (zentrales Erkennungsmuster hat 3 quadratische Ringe) |
| 3 | = | Runen (nicht realisiert) |

Höhe und Breite der Symbolelemente (Module, „Pads“) (k_3):

<nnnn> : 0000 ... 9999 : Höhe und Breite der Symbolelemente

Der Standardwert für die Höhe und Breite der Symbolelemente ist 3 Punkte (Dots). Wenn '0000' als Wert für den Parameter k_3 gewählt wird, dann wird der Standardwert verwendet. Vom Standardwert abweichende Vorgabewerte überschreiben den Standardwert.

Sicherheitsstufe (k₄):

<nnnn> : 0005 ... 0095 : Sicherheitsstufe in Prozent

Der Standardwert für die Sicherheitsstufe ist 23 %.

Wenn '0000' als Wert für den Parameter k₄ gewählt wird, dann wird der Standardwert verwendet.

Der Aztec Barcode kann in verschiedenen Sicherheitsstufen erzeugt werden. Mit steigender Sicherheitsstufe werden im Barcode mehr Daten zur Fehlerkorrektur integriert. Dies bedeutet, daß der Barcode mit steigender Sicherheitsstufe größer wird.

Gleichzeitig können jedoch Fehler, die durch äußere Einflüsse auftreten (z.B. zerkratzen oder beschriften des Barcodes), sicherer korrigiert werden.

Anzahl der Datenschichten (Data Layers) (k₅):

<nnnn> : 0001 ... 0032 : Anzahl der Datenschichten (Minimum)

Wenn '0000' als Wert für den Parameter k₅ gewählt wird, dann optimiert die Software die Größe des Barcodes automatisch.

Aztec Barcodes können in verschiedenen Größen erzeugt werden.

Der Parameter k₅ ermöglicht die Wahl der Größe in Stufen:

der Wert '0001' führt zu einem Symbol mit einer Datenschicht (Layer), der Wert '0032' führt zu einem Symbol mit 32 ringförmigen Datenschichten um das zentrale Erkennungsmuster.

Wenn nicht genug Barcodedaten zur Verfügung stehen, um die Schichten (Layer) zu füllen, füllt die Software den Barcode automatisch auf.

Allerdings führt die Vorgabe von z.B. 3 Schichten nicht auf jeden Fall zur Generierung eines Barcodes mit 3 Schichten.

Ursache: Falls die Datengrenzen überschritten werden, optimiert die Software die Größe des Barcodes automatisch auf die nächstgrößere Anzahl von Datenschichten.

Länge der Barcodedaten (k₆):

<nnnn> : 0000 ... 9999 : Länge der Barcodedaten
in Bytes.

Mit diesem Parameter k₆ kann die Länge der Barcodedaten in Bytes angegeben werden. Seine Verwendung ist immer dann ratsam, wenn kein Zeichen als Terminierungszeichen verwendet werden kann. Dies ist der Fall, wenn alle verfügbaren Zeichen in den Barcodedaten vorkommen oder der Anwender nicht sicher sein kann, daß sein Terminierungszeichen nicht in den Barcodedaten auftaucht.

Ist dagegen ein Zeichen verfügbar, so können die Daten des Aztec-Barcodes (wie bei den anderen Barcodes) durch Terminierungszeichen geklammert werden. Der Parameter k₆ wird dann weggelassen; das erste Zeichen der Barcodedaten definiert das Terminierungszeichen, welches dann auch am Ende der Barcodedaten verwendet wird. Dieses Terminierungszeichen darf also in den Barcodedaten nicht vorkommen.

Barcodedaten (k₉):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Beispiel:

&%&%bc 128 200 30007 40000 50019 60000 9*Code 2D!
Microplex*

Hinweis: In diesem Beispiel enthalten die Barcodedaten einen
"Wagenrücklauf" (Carriage Return) mit Zeilenvorschub (Line Feed).



8.3.6. QR-Code

Der QR-Code (Quick Response) erzeugt eine zweidimensionale Matrix (Symbologie), die aus quadratischen Modulen besteht. Solche Barcodes werden für die Artikelnummerierung und Verfolgung verwendet, neuere Anwendungen unterstützen die Nutzer von Mobiltelefonen (Mobile Tagging).

Barcodetyp (k_1):

<nn> : 29 : Barcodetyp QR-Code

Barcodemode (k_2):

Zurzeit ist nur der Standardwert ('00') zulässig.

Höhe und Breite der Symbolelemente („Pads“) (k_3):

<nnnn> : 0000 ... 9999 : Höhe und Breite der Symbolelemente

Sicherheitsstufe (k_4):

<nnnn> : 0001 ... 0004 : Sicherheitsstufe (Level)

0001	=	Stufe L
0002	=	Stufe M
0003	=	Stufe Q
0004	=	Stufe H

Der QR Barcode kann in verschiedenen Sicherheitsstufen erzeugt werden. Mit steigender Sicherheitsstufe werden im Barcode mehr Daten zur Fehlerkorrektur integriert. Dies bedeutet, daß Fehler, die durch äußere Einflüsse auftreten (z.B. zerkratzen oder beschriften des Barcodes), sicherer korrigiert werden.

Größe, Anzahl der Module (k₅):

<nnnn>	: 0000 ... 0040	: Anzahl der Module (Größe)
	0000	= automatisch
	0001	= 21 * 21 Module
		bis zu (in 4er Schritten)
	0040	= 177 * 177 Module

Länge der Barcodedaten (k₆):

<nnnn> : 0000 ... 9999 : Länge der Barcodedaten in Bytes.

Mit diesem Parameter k₆ kann die Länge der Barcodedaten in Bytes angegeben werden. Seine Verwendung ist immer dann ratsam, wenn kein Zeichen als Terminierungszeichen verwendet werden kann. Dies ist der Fall, wenn alle verfügbaren Zeichen in den Barcodedaten vorkommen oder der Anwender nicht sicher sein kann, daß sein Terminierungszeichen nicht in den Barcodedaten auftaucht.

Ist dagegen ein Zeichen verfügbar, so können die Daten des QR-Code -Barcodes (wie bei den anderen Barcodes) durch Terminierungszeichen geklammert werden. Der Parameter k₆ wird dann weggelassen; das erste Zeichen der Barcodedaten definiert das Terminierungszeichen, welches dann auch am Ende der Barcodedaten verwendet wird. Dieses Terminierungszeichen darf also in den Barcodedaten nicht vorkommen.

Barcodedaten (k₉):

Mit diesem Parameter werden die Barcodedaten als Zeichenkette angegeben.

Beispiel:

Der folgende Barcode kann z.B. mit einem Smartphone gescannt werden; es erfolgt eine Weiterleitung zur MICROPLEX – Website.

`&%%bc 129 30007 4 4 5 0 9*www.microplex.de*`



9. RFID Funktionsgruppe

Eine Reihe von MICROPLEX Druckern sind optional mit einer integrierten Multiprotokoll RF Schreib/Leseinheit ausgestattet (Werksoption). Diese Drucker funktionieren als komplettes System für das schreiben, lesen und bedrucken von RFID Etiketten (**R**adio**F**requenz**ID**entifikation).

Die Befehle für die RFID Funktionen werden wie die Druckdaten über den aktiven Druckdatenkanal übertragen (Centronics, USB, Ethernet,...).

Die RFID Funktionen des Druckers ermöglichen es, z.B. die ID (Transponder-Identifikationsnummer) der RFID Etiketten zu lesen und über den Statuskanal des Druckers zum Host zu übertragen.

Außerdem kann der Nutzer dabei eigene Identifikationsdaten mitschicken (z.B. eine Seiten-Identifikationsnummer).

Die Transponder-Identifikationsnummer des Etiketts kann so eindeutig den auf diesem Etikett gedruckten Informationen zugeordnet werden.

Hinweis: Um die RFID Funktionen des Druckers nutzen zu können, benötigen Sie das entsprechende RFID Material. Lesen Sie bitte hierzu das Bedienhandbuch Ihres MICROPLEX Druckers.

Note: Wenn das RFID-Tag nicht gelesen/programmiert werden kann, wird das Etikett mit einem Überschreibe-Muster gedruckt, um anzuzeigen, dass dieses Etikett nicht zu verwenden ist. Weitere Informationen finden Sie im Bedienhandbuch Ihres Druckers.

9.1. Befehle für RFID

9.1.1. Übersicht

Funktion: Der rf Befehl kann durch die Verwendung einer Reihe von Parametern für die verschiedenen RFID -Aufgaben flexibel parametrisiert werden.

Syntax: <ESC><ESC>rf <k_{xx}nn> ... <k₉₉"cc">
(27)(27)(114)(102)...

Parameter: Jeder rf Befehl enthält eine oder mehrere Typ-Kennungen. Auf die meisten dieser Typ-Kennungen folgt ein dazugehöriger Parameter.

Zulässige Werte für die Parameter <k_{xx}nn> des rf Befehls:

Typ-Kennung k _{xx}	Parameter	Beschreibung
00	kein Parameter erlaubt	Beendet die RFID-Bearbeitung des aktuellen Etiketts (Etikett, welches sich unter der RF Schreib/Leseinheit des Druckers befindet).
01	kein Parameter erlaubt	Lese ID Der Transponder wird gelesen (Etiketten ID Nummer), die Antwort wird zum Host gesendet (über den SOLID Statuskanal; ASCII formatiert).

Typ-Kennung k_{xx}	Parameter	Beschreibung
11	nn = 2-stellig dezimal	Datenblocknummer Beim schreiben: 00 = EPC* = default 01 = Benutzer Beim lesen: nn = Blocknummer
12	n = 1-stellig	Datenformat 0 = binär = default (ASCII dargestellt wie sie sind) 1 = Hex-ASCII
13	n = 1-stellig	Schreibschutz 0 = kein Schutz = default 1 = Schreibschutz
99	"cc" = String	Schreibe Etikett (Write Tag) Letzte Typkennung, unmittelbar danach müssen die Daten folgen (Siehe Beispiel in Abschnitt 9.1.3)

*EPC: Electronic Product Code

9.1.2. Etiketten-Identifikationsnummer lesen (Vor dem Druckvorgang)

Der im folgenden Beispiel verwendete IDOL - Befehl ermöglicht es, die Etiketten-Identifikationsnummer (ID) über den Statuskanal zu erhalten, bevor die Druckdaten für dieses Etikett gesendet wurden.

Beispiel: Die Etiketten-Identifikationsnummer (ID) soll vor dem Drucken gelesen werden.

Senden Sie den folgenden Befehl über den Druckdatenkanal zum Drucker:

&%&%rf 01

Dieser Befehl führt dazu, dass der Drucker eine "Statusmeldung" sendet.

Die Statusmeldung enthält die Etiketten-Identifikationsnummer:

```
1020<S><Label ID><S><T>
```

Falls die RFID - Hardware die Etiketten-Identifikationsnummer nicht lesen und senden kann, wird der Drucker eine andere 1020 Meldung über den Statuskanal senden.

Diese Meldung enthält ein "X" anstatt der Etiketten ID:

```
1020<S><X><S><T>
```

Hinweise: Der oben beschriebene Befehl kann die Druckgeschwindigkeit verringern. Wird ein höheres Ausgabevolumen benötigt, sollte ein anderer Befehl bevorzugt werden (siehe Abschnitt 9.2).

Die folgenden Abschnitte enthalten weitere Details!

9.1.3. Daten in einen Datenblock des RFID – Etiketts schreiben

Der im folgenden Beispiel verwendete IDOL - Befehl ermöglicht es, Daten in Datenblöcke des RFID – Etiketts zu schreiben.

Beispiel: Der Text „MICROPLEX“ soll in den Benutzerblock des RFID – Etiketts geschrieben werden.

Senden Sie den folgenden Befehl über den Druckdatenkanal zum Drucker:

&%&%rf 11 01 99*MICROPLEX*

Dieser Befehl führt dazu, dass die Zeichenfolge „MICROPLEX“ in den RFID Chip geschrieben wird (ohne Schreibschutz).

Beachte: Die Daten sind als Textstring anzugeben, das Datenformat kann mit dem Parameter k_{12} festgelegt werden (binär = default, siehe Abschnitt 9.1.1). Die Länge der Daten ist flexibel. Deshalb werden die Daten mit Terminierungszeichen "geklammert". Das erste Zeichen des Parameters k_{99} definiert das Terminierungszeichen. Dieses Terminierungszeichen ist frei wählbar. Es muss lediglich sichergestellt werden, dass dieses Terminierungszeichen nicht in den Daten vorkommt. Im obigen Beispiel wurde das Zeichen * als Terminierungszeichen verwendet.

Beachte: Der oben beschriebene Befehl kann die Druckgeschwindigkeit verringern.

Wenn das RFID-Tag aus irgendeinem Grund nicht programmiert werden kann, wird das Etikett mit einem Überschreibe-Muster gedruckt, um anzuzeigen, dass dieses Etikett nicht zu verwenden ist.

9.2. Generierung Seiten-ID-Nummer, Abfrage RFID Daten nach dem Drucken

Für diese Aufgaben werden Statusabfragen (Status Request) verwendet, mehr Details zu diesen IDOL - Befehlen sind in Kapitel 12 SPEZIAL - Befehle und Statusabfragen.

Der folgende IDOL-Befehl wird für die Abfrage von RFID Daten und außerdem für die Generierung einer Seiten ID Nummer verwendet:

<ESC><ESC>: 0019<ddd><Page ID>

Dieser Befehl ist eine Statusabfrage an den Drucker, die Antwort (Statusmeldung) wird über den SOLID Statuskanal ausgegeben. Diese Meldung beinhaltet die RFID Daten und die Seiten ID Nummer.

Parameter: <ddd> = Stringlänge (3 stellig, dezimal).
000 ist zu verwenden, wenn kein Textstring mitgegeben werden soll.
Die maximale Stringlänge beträgt 31 Zeichen.

<Page ID> = Textstring (ASCII - Zeichen).
Hier kann z.B. eine Seiten-Identifikations-nummer (Page ID) eingegeben werden.
(Der Textstring wird als Seiten-Quittungsstring eingelagert).

Beachte: Statusabfragen (Befehle vom Typ <ESC><ESC>: <nnnn>) können nur dann verwendet werden, wenn der Drucker ON LINE ist. Ist der Drucker dagegen im OFF LINE Zustand, können keine Befehle zum Drucker übertragen werden.

Der Drucker kann eine Meldung senden, um Ihnen zu zeigen, wann er in den OFF LINE Mode wechselt:

Wählen Sie dazu den automatischen Meldungsmodus (EEPROM Wort 23 Bit 13 = 0). Dann überträgt der Drucker seine Statusmeldungen (OFF LINE, Papierstau ...) automatisch zum Host.

Hinweise: Weitere Einzelheiten zur Abfrage von Statusmeldungen sind im Kapitel 12 dieses Handbuchs beschrieben.

Beispiel: Jedem RFID Etikett soll eine Kunden-Seiten-Identifikation zugeordnet werden. Diese Page ID soll den Text "SEITE:" und eine 4stellige fortlaufende Nummer enthalten (diese wird vom Nutzer generiert).

Für das erste Etikett sieht der Befehl für die RFID Statusabfrage dann wie folgt aus:

```
&%&%: 0019 009 SEITE:0001
```

Der obige Befehl wird zusammen mit den Druckdaten für dieses Etikett zum Drucker übertragen.

Er bewirkt, dass der Drucker nach der Bedruckung dieses Etiketts eine "Fertigmeldung" (Statusmeldung) sendet.

Die Statusmeldung enthält die Etiketten-Identifikationsnummer (Label ID):

```
1000<S><Page ID><S><Label ID><S><T>
```

Anhand der ebenfalls enthaltenen Seiten-Identifikationsnummer (Page ID) erkennt der Nutzer, welche Seite gerade erfolgreich gedruckt wurde.

Falls die RFID - Hardware die Etiketten-Identifikationsnummer nicht lesen und senden kann, wird der Drucker eine andere 1000 Meldung über den Statuskanal senden.

Diese Meldung enthält ein "X" anstatt der Etiketten ID:

```
1000<S><Page ID><S><X><S><T>
```

Die folgenden Abschnitte enthalten weitere Details!

9.3. Syntax der RFID Daten (Statusmeldungen)

Die RFID Funktionen des Druckers ermöglichen es, die ID (Transponder-Identifikationsnummer) der RFID Etiketten zu lesen und zusammen mit weiteren für den Nutzer relevanten Daten über den Statuskanal des Druckers zum Host zu übertragen.

Die RFID Daten (Statusmeldungen des Druckers) werden mit folgender Syntax übertragen:

<Key-No.><S><Page ID><S><Label ID><S><T>

- Key-No. : 4...6 stellige dezimale Statusnummer (ASCII Zeichen).
S : Separator (z.B. SPACE).
Page ID : Textstring, z.B. Seiten-Identifikationsnummer.
(Wie im Statusabfragebefehl vorgegeben, vergleiche Abschnitt 9.2. Ist dort nichts angegeben worden, wird nur der zugehörige Separator ausgegeben).
Beachte: Falls der Befehl **Etiketten-Identifikationsnummer** vor dem Druckvorgang **lesen** verwendet wurde (siehe Abschnitt 9.1.2), ist **keine Page ID** und kein Separator in den RFID Daten enthalten.
Label ID : Etiketten-Identifikationsnummer (wird von der integrierten RF Schreib/Leseinheit des Druckers gelesen).
T : Terminator (z.B. ASCII NUL (HEX 00))

Hinweise: Es werden nur ASCII Zeichen verwendet.

Separator und Terminator können konfiguriert werden (siehe Anhang).

Weitere Einzelheiten zu den Statusmeldungen sind im Kapitel 12 dieses Handbuchs beschrieben.

Beispiele: In der folgenden Statusmeldung wird das Zeichen / als Separator verwendet, als Terminator wird das Zeichen \ verwendet.

a) Etiketten-Identifikationsnummer lesen (vor dem Druckvorgang):

Befehl: &%&%rf 01

Statusmeldung: 1020/058000098B7C1F6B/\

Diese Statusmeldung des Druckers enthält folgende Informationen:

Key-No. = 1020 : Dies ist eine Antwort auf einen RFID Befehl.

Label ID = 058000098B7C1F6B : Identifikationsnummer des Etiketts
(in diesem Beispiel beträgt die Länge 16 Zeichen)

b) Generierung einer Seiten ID Nummer, Abfrage der RFID Daten nach dem Drucken:

Befehl: &%&%: 0019 009 SEITE:0001

Statusmeldung: 1000/SEITE:0001/058000098B7C1F6B/\

Diese Statusmeldung des Druckers enthält folgende Informationen:

Key-No. = 1000 : Das Etikett (eine Seite) wurde korrekt gedruckt.

Page ID = SEITE:0001 : Dies ist die vom Nutzer verwendete Seiten-Identifikation (Page ID). Siehe vorherige Abschnitte: Parameter des Befehls für die Statusabfrage.

Label ID = 058000098B7C1F6B : Identifikationsnummer des Etiketts (in diesem Beispiel beträgt die Länge 16 Zeichen)

10. Funktionsgruppe NETZWERK

Mit den Befehlen der Funktionsgruppe NETZWERK werden Druckaufträge verwaltet, die einen Drucker über ein Netzwerk erreichen. Die Befehle kennzeichnen einen Druckauftrag eindeutig. Anhand dieser Kennzeichnungen werden Druckaufträge von Programmen identifiziert und verwaltet, ohne die Druckdaten zu interpretieren. Diese Funktionalität ist unabhängig vom verwendeten Netzbetriebssystem.

10.1. Jobanfang**na**

Funktion: Mit diesem Befehl wird der Anfang eines Druckauftrages markiert. Dieser Befehl setzt den druckerinternen Seitenzähler auf eins.
Soll ein Druckauftrag mit einer Anfangs- und Endekennung versehen werden, wird der Druckauftrag mit Hilfe der Befehle "Jobanfang" und "Jobende" begrenzt.

Syntax:	<code><ESC><ESC>na <n> <aaaa></code>	(27)(27)(110)(97) ...
----------------	--	------------------------------

Parameter:	<code><n> =</code>	0...9	: Befehlstyp
	<code><n> =</code>	0	: reserviert
	<code><n> =</code>	1	: Standard
	<code><n> =</code>	2	: automatische Inkrementierung der Jobnummer
	<code><n> =</code>	3...9	: reserviert
	<code><aaaa> =</code>	0001...9999	: Jobnummer, dezimal (Angabe nicht erforderlich, wenn <code><n>=2</code> gewählt wurde)

Beachte: Die Jobnummer "0000" sollte nicht verwendet werden. Diese Nummer wird automatisch vergeben, wenn vom Anwender keine Jobnummer festgelegt wird.
Der Befehl "Jobanfang" löst zunächst ein implizites FORM FEED aus. Das bedeutet, falls sich Daten im Eingangspuffer befinden, werden diese erst ausgegeben, bevor der neue Job begonnen wird.

Beispiel: Ein Druckauftrag soll die Jobnummer 1 122 erhalten.

```
&%&%na 1 1122
```

10.2. Jobende

ne

Funktion: Dieser Befehl markiert das Ende eines Druckauftrages. Der Parameter "Jobnummer" muß mit der des Jobanfanges übereinstimmen. Mit dem Befehl "Jobende" wird in Kombination mit dem Befehl "Jobanfang" ein Druckauftrag mit einer Jobnummer versehen.

Syntax: <ESC><ESC>ne <n> <eeee> **(27)(27)(110)(101) ...**

Parameter:	<n> =	0...9	: Befehlstyp
	<n> =	0	: reserviert
	<n> =	1	: Standard
	<n> =	2	: Ende eines Jobs mit autom. Inkrementierung (vgl. 9.1)
	<n> =	3...9	: reserviert
	<eeee> =	0001...9999	: Jobnummer, dezimal (Angabe nicht erforderlich, wenn <n>=2 gewählt wurde)

Beachte: Der Befehl "Jobende" löst ein unbedingtes FORM FEED aus. Das bedeutet, die Seite wird in jedem Fall ausgegeben, auch wenn sich keine Daten im Eingangspuffer befinden.

Beispiel: Der Druckauftrag 1122 wird beendet.

```
&%&%ne 1 1122
```

10.3. Jobindex**ni**

Funktion: Mit diesem Befehl werden Zeichen festgelegt, die in den Datenstrom als Indexinformationen integriert werden. Diese Indexinformationen können z.B. genutzt werden, um die Daten in einem Archivsystem wiederzufinden. Die Indexdaten werden nicht gedruckt. Die mit diesem Befehl festgelegten Indexdaten bleiben aktiv, bis mit diesem Befehl neue Indexdaten festgelegt werden.

Syntax: `<ESC><ESC>ni <n> <xxx> <z1> ... <zn>`

(27)(27)(110)(105) ...

Parameter:

<code><n> =</code>	0...9	: Befehlstyp Informationen zu den verschiedenen Befehlstypen: siehe folgende Seite
<code><n> =</code>	3	: Standard
<code><xxx> =</code>	000...999	: Anzahl der Indexzeichen
<code><z₁ ... z_n></code>		: Indexdaten

Beispiel: Den folgenden Seiten soll für die Archivierung der Index "Rechnungen 2003" vorangestellt werden.

`&%%ni 3 15Rechnungen 2003`

Einzelheiten zu den Varianten des Jobindex-Befehls:

Mit dem Jobindex-Befehl können verschiedene Indexvarianten erzeugt werden (Parameter <n> : Befehlstyp).

Es kann der Gültigkeitsbereich des Jobindex-Befehls festgelegt werden. Entweder nur für die Seite, auf der er gesetzt wurde, oder für mehrere Seiten bis zur nächsten Änderung.

Die Indexdaten können auf verschiedene Weise übergeben werden:

1. Indexdaten mit bekannter Länge. Die Anzahl der Indexbytes wird mit dem Indexbefehl zusammen angegeben.
2. Indexdaten unbekannter Länge. Hier wird das erste Zeichen der Indexdaten als Terminator definiert. Danach werden die folgenden Daten als Indexdaten verwendet. Dies geschieht solange, bis das zu Beginn definierte Terminatorzeichen wieder erscheint. Dies bedeutet: das definierte Terminatorzeichen darf in den Indexdaten **nicht** vorkommen !

Außerdem können Indexdaten an bereits bestehende Indexinformationen angehängt werden, ohne daß die bereits definierten Indexdaten gelöscht werden.

Die verschiedenen Anwendungsmöglichkeiten sind in folgender Tabelle dargestellt:

Befehlstyp <n>	Index definieren	Index anhängen	Gültig für aktuelle Seite	Gültig bis zur nächsten Änderung	Terminator
1	●			●	●
2		●			●
3	●			●	
4		●			
5	●		●		●
6	●		●		

Die Gültigkeit der angehängten Indexdaten hängt von der Gültigkeit der bereits bestehenden Indexdaten ab.

Die Syntax der verschiedenen Indexbefehle mit den einzelnen Parametern ist wie folgt:

Syntax: <ESC><ESC>*ni* <1> <T> <z₁ ... z₂₅₆> <T>

(27)(27)(110)(105) ...

Parameter: <n> = 1 Index definieren. Löscht alle bisher gesetzten Indexinformationen. Der mit diesem Befehl gesetzte Index bleibt über Seitengrenzen hinweg erhalten bis zur nächsten Änderung.

<T> = ASCII : Terminator-Definition für Indexdaten

<z₁ ... z₂₅₆> : Indexdaten (ASCII)

<T> = ASCII : Terminator für Indexdaten

Syntax: <ESC><ESC>*ni* <2> <T> <z₁ ... z₂₅₆> <T>

(27)(27)(110)(105) ...

Parameter: <n> = 2 Indexinformationen anhängen. Hängt Informationen an einen bestehenden Index an, ohne die bereits definierten Indexdaten zu löschen.

<T> = ASCII : Terminator-Definition für Indexdaten

<z₁ ... z₂₅₆> : Indexdaten (ASCII)

<T> = ASCII : Terminator für Indexdaten

Syntax: <ESC><ESC>*ni* <3> <xxx> <z₁ ... z₂₅₆>

(27)(27)(110)(105) ...

Parameter: <n> = 3 Index definieren. Löscht alle bisher gesetzten Indexinformationen. Der mit diesem Befehl gesetzte Index bleibt über Seitengrenzen hinweg erhalten bis zur nächsten Änderung.

<xxx> = 000...999 : Anzahl der Indexzeichen

<z₁ ... z_{xxx}> : Indexdaten (ASCII)

Syntax: <ESC><ESC>*ni* <4> <xxx> <z₁ ... z₂₅₆>

(27)(27)(110)(105) ...

Parameter: <n> = 4 Indexinformationen anhängen. Hängt Informationen an einen bestehenden Index an, ohne die bereits definierten Indexdaten zu löschen.

<xxx> = 000...999 : Anzahl der Indexzeichen

<z₁ ... z_{xxx}> : Indexdaten (ASCII)

Syntax: <ESC><ESC>ni <5> <T> <z₁ ... z₂₅₆> <T>

(27)(27)(110)(105) ...

Parameter: <n> = 5 Index definieren. Löscht alle bisher gesetzten Indexinformationen. Der mit diesem Befehl gesetzte Index ist nur für die Seite gültig, auf der er definiert wurde. Dies gilt auch für Indexinformationen, die an diesen Index angehängt wurden.

<T> = ASCII : Terminator-Definition für Indexdaten

<z₁ ... z₂₅₆> : Indexdaten (ASCII)

<T> = ASCII : Terminator für Indexdaten

Syntax: <ESC><ESC>ni <6> <xxx> <z₁ ... z₂₅₆>

(27)(27)(110)(105) ...

Parameter: <n> = 6 Index definieren. Löscht alle bisher gesetzten Indexinformationen. Der mit diesem Befehl gesetzte Index ist nur für die Seite gültig, auf der er definiert wurde. Dies gilt auch für Indexinformationen, die an diesen Index angehängt wurden.

<xxx> = 000...999 : Anzahl der Indexzeichen

<z₁ ... z_{xxx}> : Indexdaten (ASCII)

11. Funktionsgruppe CONTROLLER

Die Konfiguration von MICROPLEX – Druckern ist in der EEPROM-Sequenz gespeichert. Die EEPROM-Sequenz setzt sich aus bis zu 4 Zeilen zusammen, diese werden oben auf dem Statusblatt des Druckers angezeigt.

Mit den Befehlen der Funktionsgruppe CONTROLLER kann die Konfiguration des Druckers geändert werden (z.B. Einschaltwerte des Druckers ändern, neue Funktionen freischalten).

Beachte: Es sollte ein Statusblatt gedruckt werden, bevor die Konfiguration geändert wird. So läßt sich der alte Zustand des Druckers bei Bedarf leichter wieder herzustellen.

11.1. Freigabeschlüssel (Softkey)

+m

Funktion: Mit diesem Befehl können neue Funktionen freigeschaltet werden. Nachdem Sie die Seriennummer des Gerätes (oder ein Statusblatt) an MICROPLEX gesendet haben, erhalten Sie nach dem Erwerb der neuen Funktionalität einen Freigabeschlüssel (Softkey).
Die neuen Funktionen sind verfügbar, nachdem der Softkey mit dem folgenden IDOL – Befehl zum Drucker geschickt wurde.

Syntax: <ESC><ESC> +m <hhhhhhh> (27)(27)(43)(109) ...

Parameter: <hhhhhhh> = 00000000 bis FFFFFFFF : Schlüsselwort
(hexadezimal, 8 stellig)

Schalten Sie den Drucker aus und wieder ein und drucken Sie ein neues Statusblatt, auf dem alle verfügbaren Optionen gelistet sind.

11.3. Verwendung von vordefinierten Konfigurationssätzen (User Configs)

cf

Funktion:

Diese Befehle ermöglichen es dem Anwender, eine Anzahl von "vordefinierten" Druckerkonfigurationen zu verwenden (bis zu 50 verschiedene "Sätze" von EEPROM-Sequenzen sind möglich).

cg

Normalerweise werden Sie Einstellungen an Ihrem Drucker vornehmen (Ränder etc.), um Ihre speziellen Anforderungen zu erfüllen. Danach können Sie diese aktuelle Konfiguration über das Bedienfeld speichern

(Menüebene 1 \Konfiguration \User Configuration \Define).

Ab jetzt können Sie diese Konfiguration wieder herstellen.

Jeder Konfigurationssatz beinhaltet unabhängige Werte für die Ränder, Druckrichtung, Fonts, den Ländercode, etc.

Alle vordefinierten Konfigurationssätze (User Configs) werden im Controller gespeichert und sind über das Bedienfeld des Druckers sowie über die folgenden IDOL-Befehle auswählbar.

Syntax:	<code><ESC><ESC>cf <n></code>	(27)(27)(99)(102)...
	<code><ESC><ESC>cg <dd></code>	(27)(27)(99)(103)...

Parameter:

`<n>` = 0 bis 9 : Nummer des Konfigurationssatzes, einstellige Dezimalzahl
 0 = aktiver Konfigurationssatz im Auslieferungszustand des Druckers

Erweiterter Befehl (cg):

`<dd>` = 0 bis 49 : Nummer des Konfigurationssatzes, zweistellige Dezimalzahl
 0 = aktiver Konfigurationssatz im Auslieferungszustand des Druckers

Beachte:

Dieser Befehl aktiviert den ausgewählten Konfigurationssatz und ein Reset wird initialisiert (<ESC><ESC>re). Falls es den gewählten Konfigurationssatz nicht gibt, wird ein neuer Satz erzeugt (enthält dann die aktuellen Einstellwerte).

Ein Befehl zum "Zurücksetzen auf Fabrikwerte (Factory Default)" wirkt sich auf den aktuell gewählten (aktiven) Konfigurationssatz aus.

Beispiele:

- A) Der vordefinierte **Konfigurationssatz** (User Config) Nummer 5 soll in "Rechnung" **umbenannt** werden:

```
&%&%+a 01 005 99"Rechnung"
```

Von nun an ist Rechnung (der von Ihnen gewünschte neue Name) mit diesem Konfigurationssatz verknüpft (zugewiesen).

Der Drucker verwendet diesen Namen z.B. für die betreffenden Bedienfeldmeldungen.

Hinweise: In Abhängigkeit vom Drucker kann die Anzahl der Zeichen eingeschränkt sein. Außerdem unterstützen nicht alle Druckerdisplays Sonderzeichen oder auch Groß- und Kleinbuchstaben.

- *¹ Hinweis: Auf dem Statusblatt Ihres Druckers wird weiterhin die Nummer des (aktiven) Konfigurationssatzes angezeigt (gleich hinter der Bezeichnung „SERVICE INFORMATION“). Diese Nummer benötigen Sie für den +a IDOL-Befehl, wenn Sie diesem Konfigurationssatz bereits einen Namen gegeben haben, diesen aber abändern möchten.

- B) Ein Netzwerkadministrator will dem Drucker eine **neue MAC - Adresse** geben. (Diese Umkonfiguration sollte nur durchgeführt werden, wenn die bei der Fabrikation an das MICROPLEX Gerät vergebene MAC – Adresse nicht weiter verwendet werden soll).

```
&%&%+a 02 0 99"CC:3C:3F:00:C0:00"
```

Die MAC – Adresse des Druckers wurde in CC:3C:3F:00:C0:00 geändert.

11.4.1. Interne Variablen drucken (Seriennummer, Uhr)



Funktion: Dieser Befehl ermöglicht es dem Anwender, die Ausdrücke z.B. mit einer Zeitangabe zu versehen (Nutzung der Echtzeituhr, vergleiche vorherigen Abschnitt).

Syntax: <ESC><ESC>dv <a><#####>
(27)(27)(100)(118) ...

Parameter: <a> = s **Seriennummer** (keine weiteren Parameter)

 <a> = d **Datum**

 <#####> = *dd.mm.yy* Formatierungsmaske (Beispiel) *¹ *²

 dd : Tag, zweistellige Dezimalzahl *³

 mm : Monat, zweistellige Dezimalzahl *³

 yy : Jahr, zweistellige Dezimalzahl

Alternative Formatierungen:

 ddd : Wochentag (Montag bis Sonntag)

 k oder kk: Kalenderwoche *³

 mmm : Monat (Kurzform Text: 3 Zeichen)

 yyyy: Jahr, vierstellige Dezimalzahl

 <a> = t **Zeit**

 <#####> = *hh.mm.ss* Formatierungsmaske (Beispiel) *¹ *²

 hh : Stunden, zweistellige Dezimalzahl *³

 mm : Minuten, zweistellige Dezimalzahl *³

 ss : Sekunden, zweistellige Dezimalzahl *³

*¹ **Beachte:** Zwischen dem Parameter <a> und einer Formatierungsmaske ist zurzeit kein Leerzeichen (und auch kein anderes Zeichen) erlaubt.

- *² Die Formatierungsmaske ist mit Terminierungszeichen "geklammert". Mit dem ersten Zeichen der Formatierungsmaske definieren Sie das Terminierungszeichen (in den obigen Beispielen wurde * als Terminierungszeichen gewählt). Weitere Details zur Verwendung von Terminierungszeichen finden Sie im Abschnitt Funktionsgruppe Barcode.
- *³ Bei Werten von 1 bis 9 wird eine führende Null hinzugefügt.

Beispiele:

&%%&dv d*dd.mm.jj*	führt zu	22.01.14
&%%&dv d*m-dd-yyyy*	führt zu	1-22-2014
&%%&dv d*ww/yy*	führt zu	04/14
&%%&dv d*ddd*	führt zu	Mittwoch
&%%&dv t*hh.mm*	führt zu	09.55
&%%&dv t*h:mm:ss*	führt zu	9:55:11

x = 05 : Dateiname	"ccc" = String schreibt die Datei als "Druckdatei" in das Verzeichnis "idol/printfiles"
x = 06 : Dateiname	"ccc" = String schreibt die Datei in das Fontverzeichnis
x = 99 : Befehlsende-Kennung,	unmittelbar danach muss der <u>Dateiinhalte</u> <u>folgen</u>

Hinweis: Verwenden Sie bitte die Bedienfeldfunktion ' Inhaltsverzeichnis ' des Druckers, um eine Übersicht der Verzeichnisstruktur zu erhalten:
Bedienebene 1 \Datei-Verwaltung \Inhaltsverzeichnis

Beispiel:

Es soll eine Druckdatei gespeichert werden, der Name der Datei ist
Testpage1.prn:

`&%%&%+f 05"Testpage1.prn"99` hier folgen die Daten der Druckdatei &%%&%v&%%&%+f

Beachte: Wenn Sie den Parameter 02 (Dateilänge) nicht verwenden, muss das Ende der Datei mit dem Befehl `&%%&%+f` gekennzeichnet werden.

12. SPEZIAL - Befehle und Statusabfragen

12.1. Spezial - Befehle

Funktion: Der Befehl <ESC><ESC>: kann für einfache Aufgaben wie die Erzeugung eines Statusblatts, einer Fontliste oder einer Menüseite verwendet werden. Weitere Anwendungen werden in den folgenden Abschnitten beschrieben.

Syntax:	<ESC><ESC>: <nnnn>	(27)(27)(58)...
----------------	--------------------	------------------------

Parameter: <nnnn> = 0001 bis 9999 = vierstellige Dezimalzahl

<nnnn> = 0001 : dieser Befehl generiert ein **Statusblatt**

<nnnn> = 0002 : dieser Befehl generiert eine **Liste aller Fonts**, die auf dem Drucker installiert sind

<nnnn> = 0003 : dieser Befehl generiert ebenfalls eine Liste aller Fonts

<nnnn> = 0004 : dieser Befehl generiert eine **Menüseite** (Übersicht der verfügbaren Bedienfeldfunktionen des Druckers)

<nnnn> = 1234 : mit diesem Befehl wird die aktuelle Konfiguration **auf die Fabrikwerte** zurückgesetzt.

12.2. Statusabfragen

12.2.1. Konfiguration des Status Kanals

Die Funktionalität Status-Out wird über die Option "STAT.OUT" freigegeben mit einem Freigabeschlüssel (Sofkey).

Die Status-Meldungen sind unterteilt in:

- Meldungen, die nur automatisch gesendet werden (automatische Meldungen)
- Meldungen, die nur auf Anfrage gesendet werden (Anfrage - Meldungen)
- Meldungen, die sowohl automatisch wie auch auf Anfrage gesendet werden.

Anfrage-Meldungen sind mit der Options-Freigabe immer möglich.

Damit automatische Statusmeldungen zum Host übertragen werden können, ist ein Bit in der EEPROM Sequenz zu setzen:

Word 23 Bit 13 (Bedienfeld: Config 23)

Bit 13 = 0 : Mode 1, Meldungen werden automatisch und nach Abfrage übertragen.

Bit 13 = 1 : Mode 2, Meldungen werden nur nach einer Abfrage übertragen.

Diese Konfiguration des Status Kanals kann über das Bedienfeld des Druckers erfolgen (Bedienfeldfunktion Konfig. Wort).

Auch der IDOL Befehl <ESC><ESC>ee kann verwendet werden, um das Bit 13 zu setzen.

Syntax des Befehls <ESC><ESC>ee:

<ESC><ESC>ee <ddd> <hhhh>

Parameter <ddd> : Das Wort, welches geändert werden soll (hier: 023).

Parameter <hhhh> : Der neue Wert für das spezifizierte Wort.

Die Wörter der EEPROM Sequenz sind hexadezimal kodiert. Um ein Bit zu verändern, muß das ganze Wort vollständig neu beschrieben werden.

12.2.2. Befehle für die Statusabfrage

Funktion: Die Befehle für die Statusabfrage bieten die Möglichkeit, Informationen über den Drucker Status (Papierstau, Offline, Papier Ende,...) und den Druck Job Status (Idle, Busy, Seite gedruckt,...) zu erhalten.

<p>Syntax: <ESC><ESC>: <nnnn> (27)(27)(58)...</p>

Parameter: <nnnn> = Parameter (vierstellige Dezimalzahl)

<nnnn> = 0000 : keine Meldung (default)

<nnnn> = 0001 : Statusblatt drucken.
Ein Statusblatt wird gedruckt und über den Statuskanal zum Host übertragen.

<nnnn> = 0010 : Generiere eine Drucker Statusmeldung und sende sie über den Statuskanal.

<nnnn> = 0011 : Generiere eine Drucker Statusmeldung und sende sie über den Statuskanal.

<nnnn> = 0018 : Seiten-Quittungsstring einlagern, wird aber bereits mit der vorletzten Seite ausgegeben. Er gilt nur für eine Seite.

Format: <ESC><ESC>:0018xxxSTRING

xxx = Stringlänge 3 stellig

<nnnn> = 0019 : Seiten-Quittungsstring einlagern. Dieser String ist ein Seitenparameter, er wird der zugehörigen Meldung 1000 als Parameter angehängt. Er gilt nur für eine Seite.

Format: <ESC><ESC>:0019xxxSTRING

xxx = Stringlänge 3 stellig

<nnnn> = 0040 : Generiere alle MPC-Hardware Konfiguration Statusmeldungen und sende sie über den Statuskanal.

<nnnn> = 0041 : Generiere alle Drucker Konfiguration Statusmeldungen und sende sie über den Statuskanal.

<nnnn> = 0042 : Generiere alle EEPROM Konfiguration Statusmeldungen und sende sie über den Statuskanal.

<nnnn> = 0043 : Generiere alle MPC-Firmware Konfiguration Statusmeldungen und sende sie über den Statuskanal.

Beachte:

Der Befehl <ESC><ESC>: <nnnn> kann nur dann verwendet werden, wenn der Drucker ON LINE ist. Ist der Drucker im OFF LINE Zustand, können keine Befehle zum Drucker übertragen werden.

Eine OFF LINE Meldung wird nur dann zum Host übertragen, wenn sich der Drucker im automatischen Meldungsmodus befindet (Wort 23 Bit 13 = 0).

Beachte:

Nähere Einzelheiten erfahren Sie durch die Nutzung unserer Angebote an MICROPLEX Schulungen und die zugehörigen Dokumentationen: "The EEPROM-Sequence" und "Firmware Funktionalität Status Out".

13. DRUCKWERK – Befehle (PRINT ENGINE)

13.1. Druckgeschwindigkeit einstellen

pe

Funktion: Mit dieser Funktion kann die Druckgeschwindigkeit verändert werden (nicht bei allen Druckwerken verfügbar, Details finden Sie im Bedienhandbuch Ihres Druckers) ¹⁾.

Syntax: <ESC><ESC>pe <m> <ssss> **(27)(27)(112)(101)...**

Parameter:

<m>	= 0 bis 1	Maßeinheit (einstellig)
<m>	= 0	: in 1/10 Zoll pro Sekunde
<m>	= 1	: in mm pro Sekunde
<ssss>	= 0001 bis 9999	: Geschwindigkeitswert (vierstellige Dezimalzahl; der Einstellbereich hängt vom Druckwerk ab)

13.2. Druckdichte (Schwärzung, Kontrast) einstellen

pd

Funktion: Dieser Befehl wählt die Druckdichte (ist nicht für alle Druckwerke verfügbar, Details finden Sie im Bedienhandbuch Ihres Druckers) ¹⁾.

Syntax: <ESC><ESC>pd <nnn> **(27)(27)(112)(100)...**

Parameter: <nnn> = 000 bis 100 : Druckdichte in Prozent (dreistellige Dezimalzahl; der Einstellbereich hängt vom Druckwerk ab)

Beachte: Bitte beachten Sie, dass die Verwendung von hohen Kontrastwerten (größer als 100%) zu einer Verringerung der Lebensdauer der Verbrauchsmaterialien/Teile des Druckers führen kann. Lesen Sie bitte das Bedienhandbuch Ihres Druckers.

13.3. Material und Synchronisation einstellen (Thermo Mode)

tm

Funktion: Mit Hilfe dieser Funktion erfolgt die Einstellung des Druckers auf das aktuell verwendete Material (Medium). (Nicht verfügbar für alle Druckwerke, Details finden Sie im Bedienhandbuch Ihres Druckers) ¹⁾.

Syntax:	<ESC><ESC>tm <n>	(27)(27)(116)(109)...
----------------	------------------	------------------------------

Parameter: <n> = 0 bis 1 : Mode (einstellig)

<n> = 0 : Etiketten = synchronisierter Druckbetrieb (Etiketten bedeutet formatiertes Material. Der Drucker nutzt die Lücken (Gaps) oder Blackmarks auf dem Druckmaterial zur Synchronisation).

<n> = 1 : Endlos = Druckbetrieb ohne Synchronisation

¹⁾ Der Befehl wird ignoriert, wenn der Drucker diese Funktion nicht unterstützt.

14. Funktionsgruppe EIN- und AUSGABESTEUERUNG

14.1. SPS – Control (Option)



Funktion: Mit diesem Befehl werden die Funktionen der Option SPS – Control (**S**peicher**P**rogrammierbare **S**teuerung) unterstützt.
 SPS – Control Eingänge werden abgefragt.
 SPS – Control Ausgänge werden gesetzt.
 Diese optionale Hardwareschnittstelle ermöglicht es, Druckvorgänge in MICROPLEX-Druckern mit Abläufen in nachgeschalteten Prozesseinheiten (z.B. Kuvertierungseinrichtungen oder Transportbänder) zu synchronisieren. Die elektrischen Eingänge können abgefragt und die Ausgänge der SPS – Control gesetzt werden. So können andere Geräte angesteuert und die Verarbeitung der Druckdaten mit externen Signalen und maschinenabhängigen Laufzeiten synchronisiert werden. Die Statusmeldungen zeigen an, ob der Drucker ONLINE ist oder ein Fehler aufgetreten ist (z.B. Papierende, Tonerende, Papierstau).

Syntax: <ESC><ESC>*cio* <nn> <dddd> <p_g>...<p₁>
(27)(27)(99)(105)(79)...

Parameter:

<nn> = 01 bis 04	: I/O – Mode (2 stellige Dezimalzahl)
<nn> = 01	: Daten an Ausgabe-Port senden. („Setzen der Ausgänge“)
<nn> = 02	: Warten, bis Druckjob gedruckt ist.
<nn> = 03	: Warten auf externe Signale. („Abfrage der Eingänge“)
<nn> = 04	: Ausgabe-Port für kurze Zeit ändern.
<dddd> = 0000 bis 9999	: Zeitvorgabe (4 stellige Dezimalzahl). Dieser Parameter wird nur für I/O – Mode 03 und 04 verwendet.

$\langle p_8 \rangle \dots \langle p_1 \rangle$: Port (8-stellige Bitdarstellung)
 $\langle p_i \rangle = 0, 1, x$: Zustände der Ports.
 Bedeutung abhängig vom I/O – Mode.

Hinweis: Die auf den folgenden Seiten beschriebenen Modi können auch kombiniert verwendet werden, um z.B. den Ablauf innerhalb eines Produktionsprozesses zu synchronisieren.

14.1.1. Daten an Ausgabe-Port senden (I/O-Mode 01)

In diesem Mode können die **Ausgänge der SPS-Control verändert** werden. Ausgänge können gesetzt, gelöscht oder in ihrem augenblicklichen Zustand belassen werden.

Parameter: $\langle nn \rangle = 01$ I/O – Mode

Hinweis: Ein Parameter $\langle dddd \rangle$ ist in diesem Mode 01 nicht notwendig und auch nicht erlaubt.

$\langle p_8 \rangle \dots \langle p_1 \rangle$ Die Parameter $\langle p_i \rangle$ repräsentieren die Ausgänge der SPS-Control.

Die Werte haben folgende Bedeutung:

$\langle p_i \rangle = 0$: Ausgang i löschen (Reset auf „0“).
 $\langle p_i \rangle = 1$: Ausgang i setzen (auf „1“).
 $\langle p_i \rangle = x, X$: Ausgang i unverändert lassen.

Beispiel: Die Ausgänge 3, 4 und 6 der SPS-Control sollen gesetzt, der Ausgang 5 gelöscht werden und alle anderen unverändert bleiben.

&%%&%cio01 XX1011XX

A8	A7	A6	A5	A4	A3	A2	A1
----	----	----	----	----	----	----	----

Ausgänge

14.1.2. Warten, bis Druckjob vollständig gedruckt ist (I/O-Mode 02)

Enthält eine Druckseite einen solchen SPS-Befehl, so wird die Generierung des **Druckdatenstromes** nach der Übermittlung dieser Seite **gestoppt**. Die Bearbeitung des Druckdatenstromes wird so lange angehalten, **bis** diese Seite (sowie **alle** vorherigen **Seiten**) **gedruckt** und korrekt ausgegeben wurden.

Parameter: <nn> = 02 I/O – Mode

<d> : Dieser Parameter ist im Mode 02 einstellig.
Als Wert ist 0 (Null) anzugeben !

Hinweis: Parameter <p_i> sind in diesem Mode 02 nicht notwendig und auch nicht erlaubt.

Beispiel: Es soll eine Seite gedruckt werden. Bevor die Daten der nächsten Seite bearbeitet werden, soll gewartet werden, bis die erste Seite den Drucker korrekt verlassen hat.

Dies ist Text für die erste Seite.

&%%&%v

&%%&%cio02 0

Dies ist Text für die zweite Seite.

&%%&%v

Beachte: Stellen Sie sicher, daß jede Druckseite, die diesen SPS-Befehl enthält, korrekt mit einem Form Feed abgeschlossen wird. Der Drucker kommt sonst zum völligen Stillstand. Ohne einen korrekt gesetzten Form Feed - Befehl würde die betreffende Seite nie gedruckt werden.

14.1.3. Warten auf externe Signale (I/O-Mode 03)

In diesem Mode können die **Eingänge der SPS-Control abgefragt** werden. Der nachfolgende Datenstrom wird erst weiter bearbeitet, wenn die angegebenen Eingänge gleichzeitig die gewünschten Zustände (gesetzt oder gelöscht) haben. Mit diesem Mode kann auf externe Ereignisse synchronisiert werden.

Parameter:	<nn> = 03	I/O – Mode
	<dddd>	: Zeitvorgabe in 10 ms Schritten (dezimal). Mit diesem Parameter wird angegeben, wie lange ein Eingangssignal mindestens anliegen muß, damit es vom Drucker akzeptiert wird. (Vermeidung von Fehlern durch Spannungsspitzen).
	<p8> ... <p1>	Die Parameter < p _i > repräsentieren in diesem Mode die Eingänge der SPS-Control. Die Werte der Parameter haben folgende Bedeutung:
	<p _i > = 0	: Warte, bis Eingang i gelöscht ist ("0").
	<p _i > = 1	: Warte, bis Eingang i gesetzt ist ("1").
	<p _i > = x,X	: Eingang i bleibt unbewertet.

Beispiel: Der Drucker soll warten, bis die Eingänge 3 und 4 der SPS-Control mindestens 50 ms gesetzt und die Eingänge 5 und 6 gelöscht sind. Alle anderen Eingänge sollen unbewertet bleiben.

```
&%%&%cio03 0005 XX0011XX
```

E8	E7	E6	E5	E4	E3	E2	E1
----	----	----	----	----	----	----	----

Eingänge

14.1.4. Ausgabe-Port für kurze Zeit ändern (I/O-Mode 04)

In diesem Mode können die **Ausgänge** der SPS - Control nach dem Druck einer Seite **für eine** vorgegebene **Zeit** (temporär) auf einen definierten **Pegel gesetzt** werden.

Nach der gewählten Zeitdauer werden die Ausgänge der SPS-Control wieder zurückgesetzt.

Ausgänge können gesetzt, gelöscht oder in ihrem augenblicklichen Zustand belassen werden.

Parameter:

<nn> = 04 I/O – Mode

<dddd> : Zeitvorgabe in 10 ms Schritten (dezimal).
Mit diesem Parameter wird angegeben, wie lange das neue Signalmuster am Ausgabe-Port gesetzt bleibt. Nach der angegebenen Zeitdauer werden die Ausgänge der SPS-Control wieder zurückgesetzt.

<p8> ... <p1> Die Parameter <p_i> repräsentieren die Ausgänge der SPS-Control:

Die Werte der Parameter haben folgende Bedeutung:

<p_i> = 0 : Ausgang i löschen (Reset auf „0“).

<p_i> = 1 : Ausgang i setzen (auf „1“).

<p_i> = x,X : Ausgang i unverändert lassen.

Beachte:

Der Befehl wird ausgeführt, sobald die Seite (die diesen Befehl enthält) gedruckt wurde. Die maximale Zeitdauer für das Ausgangssignal wird bestimmt durch die Zeit bis zur Fertigstellung der nächsten Seite.

Beispiel:

Der Ausgang 1 der SPS-Control soll nach dem Druck der Seite für die Zeitdauer von 1 Sekunde auf 1 gesetzt werden, alle anderen Ausgänge sollen unverändert bleiben. Danach wird der vorherige Zustand wieder hergestellt (hier Ausgang 1 = 0).

```
&%&%cio01 00000000
```

- Daten für eine Seite

```
&%&%cio04 0100 XXXXXXX1
```

```
&%&%v
```

A8	A7	A6	A5	A4	A3	A2	A1
----	----	----	----	----	----	----	----

Ausgänge

Hinweis:

Die verschiedenen Modi können auch kombiniert verwendet werden, um z.B. den Ablauf innerhalb eines Produktionsprozesses zu synchronisieren.

15. ANHANG I Code - Übersetzungstabelle

Dezimal	Hex.	ASCII	Binär
0	00	NUL	0000 0000
1	01	SOH	0000 0001
2	02	STX	0000 0010
3	03	ETX	0000 0011
4	04	EOT	0000 0100
5	05	ENQ	0000 0101
6	06	ACK	0000 0110
7	07	BEL	0000 0111
8	08	BS	0000 1000
9	09	HT	0000 1001
10	0A	LF	0000 1010
11	0B	VT	0000 1011
12	0C	FF	0000 1100
13	0D	CR	0000 1101
14	0E	SO	0000 1110
15	0F	SI	0000 1111
16	10	DLE	0001 0000
17	11	DC1	0001 0001
18	12	DC2	0001 0010
19	13	DC3	0001 0011
20	14	DC4	0001 0100
21	15	NAK	0001 0101
22	16	SYN	0001 0110
23	17	ETB	0001 0111
24	18	CAN	0001 1000
25	19	EM	0001 1001
26	1A	SUB	0001 1010
27	1B	ESC	0001 1011
28	1C	FS	0001 1100
29	1D	GS	0001 1101
30	1E	RS	0001 1110
31	1F	US	0001 1111

Dezimal	Hex.	ASCII	Binär
32	20	SP	0010 0000
33	21	!	0010 0001
34	22	"	0010 0010
35	23	#	0010 0011
36	24	\$	0010 0100
37	25	%	0010 0101
38	26	&	0010 0110
39	27	'	0010 0111
40	28	(0010 1000
41	29)	0010 1001
42	2A	*	0010 1010
43	2B	+	0010 1011
44	2C	,	0010 1100
45	2D	-	0010 1101
46	2E	.	0010 1110
47	2F	/	0010 1111
48	30	0	0011 0000
49	31	1	0011 0001
50	32	2	0011 0010
51	33	3	0011 0011
52	34	4	0011 0100
53	35	5	0011 0101
54	36	6	0011 0110
55	37	7	0011 0111
56	38	8	0011 1000
57	39	9	0011 1001
58	3A	:	0011 1010
59	3B	;	0011 1011
60	3C	<	0011 1100
61	3D	=	0011 1101
62	3E	>	0011 1110
63	3F	?	0011 1111

Dezimal	Hex.	ASCII	Binär
64	40	@	0100 0000
65	41	A	0100 0001
66	42	B	0100 0010
67	43	C	0100 0011
68	44	D	0100 0100
69	45	E	0100 0101
70	46	F	0110 0110
71	47	G	0100 0111
72	48	H	0100 1000
73	49	I	0100 1001
74	4A	J	0100 1010
75	4B	K	0100 1011
76	4C	L	0100 1100
77	4D	M	0100 1101
78	4E	N	0100 1110
79	4F	O	0100 1111
80	50	P	0101 0000
81	51	Q	0101 0001
82	52	R	0101 0010
83	53	S	0101 0011
84	54	T	0101 0100
85	55	U	0101 0101
86	56	V	0101 0110
87	57	W	0101 0111
88	58	X	0101 1000
89	59	Y	0101 1001
90	5A	Z	0101 1010
91	5B	[0101 1011
92	5C	\	0101 1100
93	5D]	0101 1101
94	5E	^	0101 1110
95	5F	_	0101 1111

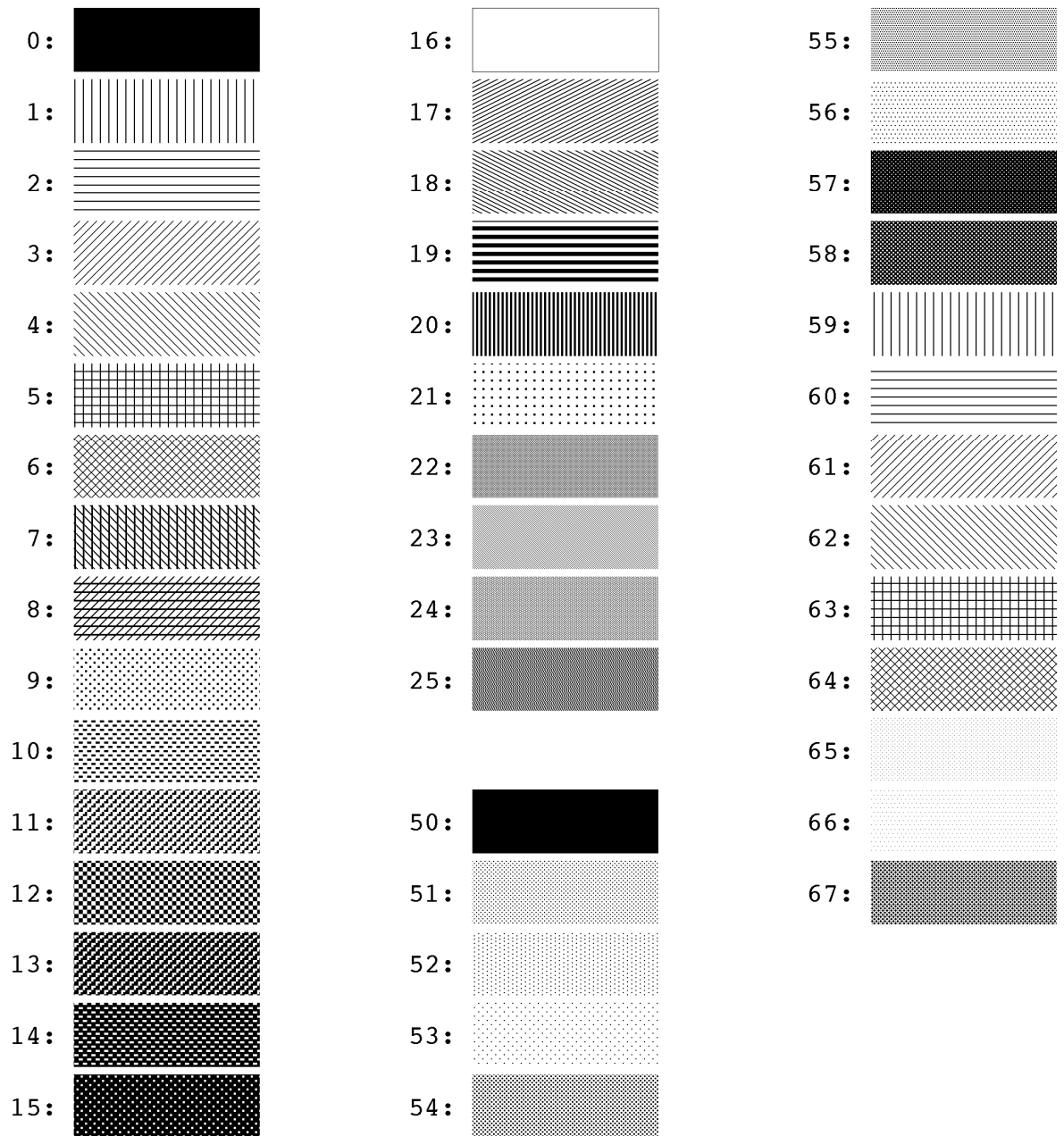
Dezimal	Hex.	ASCII	Binär
96	60	`	0110 0000
97	61	a	0110 0001
98	62	b	0110 0010
99	63	c	0110 0011
100	64	d	0110 0100
101	65	e	0110 0101
102	66	f	0110 0110
103	67	g	0110 0111
104	68	h	0110 1000
105	69	i	0110 1001
106	6A	j	0110 1010
107	6B	k	0110 1011
108	6C	l	0110 1100
109	6D	m	0110 1101
110	6E	n	0110 1110
111	6F	o	0110 1111
112	70	p	0111 0000
113	71	q	0111 0001
114	72	r	0111 0010
115	73	s	0111 0011
116	74	t	0111 0100
117	75	u	0111 0101
118	76	v	0111 0110
119	77	w	0111 0111
120	78	x	0111 1000
121	79	y	0111 1001
122	7A	z	0111 1010
123	7B	{	0111 1011
124	7C		0111 1100
125	7D	}	0111 1101
126	7E	~	0111 1110
127	7F		0111 1111

Dezimal	Hex.	ASCII	Binär
128	80		1000 0000
129	81		1000 0001
130	82		1000 0010
131	83		1000 0011
132	84		1000 0100
133	85		1000 0101
134	86		1000 0110
135	87		1000 0111
136	88		1000 1000
137	89		1000 1001
138	8A		1000 1010
139	8B		1000 1011
140	8C		1000 1100
141	8D		1000 1101
142	8E		1000 1110
143	8F		1000 1111
144	90		1001 0000
145	91		1001 0001
146	92		1001 0010
147	93		1001 0011
148	94		1001 0100
149	95		1001 0101
150	96		1001 0110
151	97		1001 0111
152	98		1001 1000
153	99		1001 1001
154	9A		1001 1010
155	9B		1001 1011
156	9C		1001 1100
157	9D		1001 1101
158	9E		1001 1110
159	9F		1001 1111

Dezimal	Hex.	ASCII	Binär
160	A0		1010 0000
161	A1		1010 0001
162	A2		1010 0010
163	A3		1010 0011
164	A4		1010 0100
165	A5		1010 0101
166	A6		1010 0110
167	A7		1010 0111
168	A8		1010 1000
169	A9		1010 1001
170	AA		1010 1010
171	AB		1010 1011
172	AC		1010 1100
173	AD		1010 1101
174	AE		1010 1110
175	AF		1010 1111
176	B0		1011 0000
177	B1		1011 0001
178	B2		1011 0010
179	B3		1011 0011
180	B4		1011 0100
181	B5		1011 0101
182	B6		1011 0110
183	B7		1011 0111
184	B8		1011 1000
185	B9		1011 1001
186	BA		1011 1010
187	BB		1011 1011
188	BC		1011 1100
189	BD		1011 1101
190	BE		1011 1110
191	BF		1011 1111

Dezimal	Hex.	ASCII	Binär
192	C0		1100 0000
193	C1		1100 0001
194	C2		1100 0010
195	C3		1100 0011
196	C4		1100 0100
197	C5		1100 0101
198	C6		1100 0110
199	C7		1100 0111
200	C8		1100 1000
201	C9		1100 1001
202	CA		1100 1010
203	CB		1100 1011
204	CC		1100 1100
205	CD		1100 1101
206	CE		1100 1110
207	CF		1100 1111
208	D0		1101 0000
209	D1		1101 0001
210	D2		1101 0010
211	D3		1101 0011
212	D4		1101 0100
213	D5		1101 0101
214	D6		1101 0110
215	D7		1101 0111
216	D8		1101 1000
217	D9		1101 1001
218	DA		1101 1010
219	DB		1101 1011
220	DC		1101 1100
221	DD		1101 1101
222	DE		1101 1110
223	DF		1101 1111

Dezimal	Hex.	ASCII	Binär
224	E0		1110 0000
225	E1		1110 0001
226	E2		1110 0010
227	E3		1110 0011
228	E4		1110 0100
229	E5		1110 0101
230	E6		1110 0110
231	E7		1110 0111
232	E8		1110 1000
233	E9		1110 1001
234	EA		1110 1010
235	EB		1110 1011
236	EC		1110 1100
237	ED		1110 1101
238	EE		1110 1110
239	EF		1110 1111
240	F0		1111 0000
241	F1		1111 0001
242	F2		1111 0010
243	F3		1111 0011
244	F4		1111 0100
245	F5		1111 0101
246	F6		1111 0110
247	F7		1111 0111
248	F8		1111 1000
249	F9		1111 1001
250	FA		1111 1010
251	FB		1111 1011
252	FC		1111 1100
253	FD		1111 1101
254	FE		1111 1110
255	FF		1111 1111

16. ANHANG II Rasterarten

17. ANHANG III Befehlsübersicht

Befehl	Funktion	Seite
ESCESC		
+a	Alphanumerische Zuweisungen	292
+c	Echtzeituhr (Real-Time Clock)	294
+f	Datei-Management Befehl	297
+m	Freigabeschlüssel (Softkey)	289
=	Einschaltwerte (Power-On Werte) festlegen	290
01	Fläche, absolut positioniert	163
02	Kreis, absolut positioniert	165
03	Linienzug, absolut positioniert	166
04	Kreisbogen, absolut positioniert	167
05	Kreisfläche, absolut positioniert	168
06	Kreisflächensegment, absolut positioniert	1691 88
07	Balken, absolut positioniert	170
08	Grafik-Cursor, absolut positioniert	160
10	Linienbreite	144
11	Linienmode	145
12	Farbe von Grafikelementen	148
13	Raster	149
14	Hinterlegungsmode	153
16	Raster laden	151
18	Druckfarbe wählen	64
19	Hinterlegungsmode für Bitmaps	155
21	Fläche, relativ zum aktuellen Grafik-Cursor	181
22	Kreis, relativ zum aktuellen Grafik-Cursor	183
23	Linienzug, relativ zum aktuellen Grafik-Cursor	184
24	Kreisbogen, relativ zum aktuellen Grafik-Cursor	186
25	Kreisfläche, relativ zum aktuellen Grafik-Cursor	187
26	Kreisflächensegment, relativ zum aktuellen Grafik-Cursor	188
27	Balken, relativ zum aktuellen Grafik-Cursor	189
28	Verschieben des Grafik-Cursors, relativ zum aktuellen Grafik-Cursor	180
33	Linienzug, relativ in Polarkoordinaten	200
34	Box mit runden Ecken, absolut positioniert	172
35	Box mit runden Ecken, relativ zum aktuellen Grafik-Cursor	191
36	Balken mit runden Ecken, absolut positioniert	171

37	Balken mit runden Ecken, relativ zum aktuellen Grafik-Cursor	190
38	Verschieben des Grafik-Cursors, relativ in Polarkoordinaten	199
40	Abbildungsfunktionen für Text- und Grafik-Cursor	138
41	Definieren der internen Grafikauflösung	140
42	Setzen der Grafik - Ränder	141
43	Grafik - Reset	143
44	Grafik-Cursor in x-Richtung, absolut positioniert	161
45	Grafik-Cursor in y-Richtung, absolut positioniert	162
49	Automatisches Umranden von Grafik-Flächen	156
50	Vektorfont mit Breiten- und Höhenfaktor	128
51	Schreibrichtung mit Winkelangabe (Grafik-Fonts)	133
52	Zeichenneigung mit Winkelangabe (Grafik-Fonts)	134
55	Ellipsenfläche, absolut positioniert	174
56	Ellipse, absolut positioniert	173
57	Maßeinheit zum Vergrößern von Outlinefonts und Stickfonts	130
59	Graustufen-Raster	150
60	Skalieren von Outlinefonts und Stickfonts	129
61	Skalieren der Höhe von Outlinefonts und Stickfonts	131
62	Skalieren der Breite von Outlinefonts und Stickfonts	132
63	Ellipsenflächen-Segment, absolut positioniert	175
64	Ellipsenflächen-Segment, relativ zum aktuellen Grafik-Cursor	194
65	Ellipsenfläche, relativ zum aktuellen Grafik-Cursor	193
66	Ellipse, relativ zum aktuellen Grafik-Cursor	192
67	Ellipsenflächen-Abschnitt, absolut positioniert	177
68	Ellipsenflächen-Abschnitt, relativ zum aktuellen Grafik-Cursor	196
:	SPEZIAL - Befehle und Statusabfragen	299
A	Vergrößerung von Zeichen eines Pixelfonts	127
a	Anwählen des Papierablagefachs und Stacker Mapping	40
B	Auswahl der Schriftart	108
bc	Generieren eines BARCODES	233
bf	FLASH-EPROM programmieren	222
bo	Fettdruck (Befehl 2)	119
C	Bestimmung der Schreibrichtung	109
cio	SPS - Control (Option)	305
cf	Verwendung von vordefinierten Konfigurationssätzen (User Configs)	291
cg	Verwendung von vordefinierten Konfigurationssätzen (User Configs)	291
co	Kommentar	74
cu	Schneidebefehl (inkl. Tear off und Peel off)	72
cx	Absolute Positionierung des Textcursors in x-Richtung	87
cy	Absolute Positionierung des Textcursors in y-Richtung	88

D	Absolutes Positionieren des Textcursors	82
dg	Durchschlag - Funktion	47
ds	Mehrfachdruck einer Seite ohne Papierfach-Anwahl	44
du	Duplex - Druck	57
dx	Vervielfältigen von Informationsbereichen in x-Richtung	49
dy	Vervielfältigen von Informationsbereichen in y-Richtung	49
E	Speichern der aktuellen Textcursor - Position	83
em	Auswahl der Emulation	29
F	Anfahren der gespeicherten Textcursor - Position	84
f	Fettdruck (Befehl 1)	118
G	Anwählen des Papiereinzugs und Feeder Mapping	37
g	Umschalten des Papierformats	34
h	Mehrfachdruck eines kompletten Druckjobs	45
I	Inverse Zeichendarstellung (Befehl 1)	114
i	Rasterhinterlegung (Befehl 2)	116
J	Der eingeschränkte CLEAR - Befehl	51
j	Der CLEAR - Befehl	53
ky	MACRO-Aufruf mit einem Schlüsselwort (Keyword)	219
L	Auswahl der Zeichentabelle	106
l	Auswahl der Emulation	29
M	Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen (BITMAP) ohne Vergrößerungsfaktor	225
m	Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen (BITMAP) mit Vergrößerungsfaktor	227
N	Definieren eines MACROs	215
na	Jobanfang, Netzwerk	282
ne	Jobende, Netzwerk	283
ni	Jobindex, Netzwerk	284
O	Aufrufen eines MACROs	217
oN	Definieren eines MACROs (Erweiterter Befehl)	215
oO	Aufrufen eines MACROs (Erweiterter Befehl)	217
oP	Löschen eines MACROs (Erweiterter Befehl)	221
oU	Automatischer Aufruf eines MACROs (Erweiterter Befehl)	218
P	Löschen eines MACROs	221
pb	Rasterhinterlegung (Befehl 2)	117
pc	Inverse Zeichendarstellung (Befehl 2)	115
pd	Druckdichte (Schwärzung, Kontrast) einstellen	303
pe	Druckgeschwindigkeit einstellen	303
pf	Textgestaltungsfunktionen	96

ph	Horizontalen Tabulator ausführen	103
pr	Speichern und Wiederherstellen von aktuellen Textparametern	214
ps	Sperrschrift (Befehl 2)	123
pu	Unterstreichen (Befehl 2)	121
pv	Vertikalen Tabulator ausführen	104
re	Reset - Befehl	54
rf	Befehl für RFID	272
ri	Textränder auf Einschaltländer zurücksetzen	94
rl	Linken Textrand auf die aktuelle Textcursor - Position setzen	90
rm	Textränder auf die Papierränder setzen	95
ro	Oberen Textrand auf die aktuelle Textcursor - Position setzen	91
rr	Rechten Textrand auf die aktuelle Textcursor - Position setzen	93
rs	Druckauflösung wählen	33
rt	Druckrichtung wählen	70
ru	Unteren Textrand auf die aktuelle Textcursor - Position setzen	92
S	Vertikale Relativpositionierung des Textcursors	85
s	Sperrschrift (Befehl 1)	122
T	Horizontale Relativpositionierung des Textcursors	86
th	Horizontalen Tabulator an aktueller Cursorposition setzen	101
ti	Neuer BITMAP-Befehl (Positionieren und Ausfüllen eines Druckfensters für Pixel-Informationen)	229
tk	Heftbefehl (Staple)	73
tl	Alle horizontalen und vertikalen Tabulatoren löschen	105
tm	Material und Synchronisation einstellen (Thermo Mode)	304
tt	Transparentcode ändern (temporär)	28
tu	Two-Up Mode (Anordnung von mehreren Druckseiten je Seitenformat)	59
tv	Vertikalen Tabulator an aktueller Cursorposition setzen	102
tx	Festlegen der horizontalen Zeichenschrittweite	110
ty	Festlegen der vertikalen Zeichenschrittweite	112
U	Automatischer Aufruf eines MACROs	218
u	Unterstreichen (Befehl 1)	120
v	Der FORM FEED - Befehl	56
W	Parameter eintragen für eine DOWNLOAD-Schrift	208
x	Laden eines einzelnen Zeichens einer DOWNLOAD-Schrift	204
Y	Löschen einer DOWNLOAD-Schrift	210
Z	Seitennummerierung	61
Z+	Seitenzähler erhöhen	62
Z-	Seitenzähler verringern	63
Z=	Seitenzähler setzen	61
zl	Null mit Schrägstrich	99

zn	Hochstellen (negative Halbzeilenschaltung)	124
ZP	Drucken der Seitennummer	62
zp	Tiefstellen (positive Halbzeilenschaltung)	125

18. Index

	Seite
2	
2/5 3 - Strich Datalogic Barcode	234
2/5 3 - Strich Matrix Barcode	234
2/5 5 – Strich Barcode	234
2/5 Interleaved Barcode	234, 243
A	
A3/A4-Format	20
Abbildungsfunktionen, Cursor	138
Ablagefach, Papier	40
Absolutes Positionieren	
Grafik.....	159
Grafik-Cursor.....	160
Grafik-Cursor x-Richtung.....	161
Grafik-Cursor y-Richtung.....	162
Textcursor	82
All - Point - Addressibility (APA)	17
Alphanumerische Zuweisungen	292
Archivierung, On-Line	22
Auflösung	
interne Grafikauflösung.....	140
Auflösung des Druckers	33
Aufruf eines MACROs	217
automatisch	218
Aztec Barcode	264

B

Balken	
absolut positioniert.....	170
relativ positioniert	189
Balken mit runden Ecken	
absolut positioniert.....	171
relativ positioniert	190
Barcode	
Generieren	233
Typen.....	234
BARCODE (FG)	233
Barcodes, eindimensional	242
Barcodes, zweidimensional	256
Befehlsübersicht	317
BITMAP	
mit Vergrößerungsfaktor	227
ohne Vergrößerungsfaktor.....	225
BITMAP (FG)	225
BITMAP(neu)	
Positionieren und Ausfüllen... ..	229
Blattgröße	20
Blocksatz	96
Box mit runden Ecken, absolut positioniert.....	172
Box mit runden Ecken, relativ positioniert	191

C

Carriage Return	25
Checksumme	207
Checkziffer, Barcode	247
Clear-Befehl	
eingeschränkt	51
uneingeschränkt	53
Clipping - Ränder	141
Codabar Barcode	234
Code - Übersetzungstabelle	311
Code 128 Barcode	247
Code 39 (3/9) Barcode	234
Code 39 Barcode	245
CODE V	67
complement	153, 155
CONTROLLER (FG)	289
Cursor	81
absolut positionieren (in x-Richtung)	87
absolut positionieren (in y-Richtung)	88
aktuelle Position speichern	83
gespeicherte Position anfahren	84
relativ positionieren (horizontal)	86
relativ positionieren (vertikal)	85

D

DataMatrix Barcode	260
Datei- Management	297
DOWNLOAD	203
DOWNLOAD-Schrift	204
Laden eines Zeichens	204
Löschen	210
Parameter eintragen	208
Druckauflösung	33
Druckdichte	303
Druckfarbe	64
Druckgeschwindigkeit	303, 319
Druckrichtung	70
Druckseiten, je Seitenformat	59
Druckwerk - Befehle	303
Duplex - Druck	57
Durchschlag	47

E

EAN 128 Barcode	247
EAN 13 Add On Barcode.....	253
EAN 13 Barcode	234, 251
EAN 8 Add-On Barcode.....	253
EAN 8 Barcode	234, 251
Echtzeituhr	294
EIN- und AUSGABESTEUERUNG (FG)	305
Einschaltwerte	290
Einzug, Auswahl.....	37
Ellipse	
absolut positioniert.....	173
relativ positioniert	192
Ellipsenfläche	
absolut positioniert.....	174
relativ positioniert	193
Ellipsenflächen-Abschnitt	
absolut positioniert.....	177
relativ positioniert	196
Ellipsenflächen-Segment	
absolut positioniert.....	175
relativ positioniert	194
Emulation, Auswahl.....	29
Emulationen, verfügbare	15
Epson	69
erase.....	153, 155
Etiketten	271

F

Fabrikwerte	299
Farbe von Grafikelementen.....	148
Feeder Mapping	37
Fettdruck	
Befehl 1	118
Befehl 2	119
Fläche	
absolut positioniert	163
relativ positioniert.....	181
FLASH-Eprom programmieren	222
FONT (FG)	75
Fonts, skalierbare	126
FORMFEED.....	56
Freigabeschlüssel	289
Funktionalität, IDOL	16

G

GKS (FG)	137
Grafik - Ränder, setzen der.....	141
Grafik-Cursor	
absolut positionieren.....	160
positionieren mit Polarkoordinaten.....	199
relativ positionieren	180
Grafik-Cursor x-Richtung	
absolut positionieren.....	161
Grafik-Cursor y-Richtung	
absolut positionieren.....	162
Grafikfonts	126
Schreibrichtung mit Winkelangabe	133
Zeichenneigung mit Winkelangabe	134
Grafikformat.....	230
Grafikfunktion und Darstellungselement	201
Grafikfunktionen	
allgemeine	138
Grafik-Reset.....	143
Grafischer Elemente, Eigenschaften	144
Grafisches - Kern - System	137
Graustufen-Raster	150

H

Heften	73
Hinterlegungsarten	153
Hinterlegungsmode, Bitmaps	155
Hochstellen	124

I

IDOL	15, 28
IDOL - Befehle	24
IDOL - Befehle, Aufbau	23
IGP	65
Informationsbereiche vervielfältigen	49
interne Variablen drucken	295
Inverse Zeichendarstellung	114, 115

J

JOB (FG).....	27
Jobanfang	282
Jobende	283
Jobindex	284

K

Key-No.	278
Keyword MACRO	219
KIX Barcode	254
Kommentar	74
Kompressionsmode	230
Kontrast	303
Koordinatensystem	19
Nullpunkt.....	20
Kreis	
absolut positioniert.....	165
relativ positioniert	183
Kreisbogen	
absolut positioniert.....	167
relativ positioniert	186
Kreisfläche	
absolut positioniert.....	168
relativ positioniert	187

Kreisflächensegment	
absolut positioniert	169
relativ positioniert.....	188
L	
Label ID	278
Linienbreite.....	144
Linienmode.....	145
Linienzug	
absolut positioniert	166
in Polarkoordinaten	200
relativ positioniert.....	184
M	
MACRO	
aufrufen	217
automatisch aufrufen	218
definieren	211
Keyword.....	219
löschen.....	221
MACRO (FG).....	211
MACRO / DOWNLOAD (FG)	203
MACRO definieren.....	215
Maßeinheit zum Vergrößern, Outlinefonts und Stickfonts	130
Material	304
Material Synchronisation.....	304
Mehrfachdruck	
Seite, ohne Papierfach-Anwahl	44
vollständiger Druckjob	45
Metrik.....	19
A3/A4-Format.....	20
MP-CONFIG.....	21
MSI Barcode.....	234, 246
N	
negative Halbzeilenschaltung.....	124
NETZWERK (FG)	281
Null mit Schrägstrich).....	99
N-Up Mode.....	59

O

OneCode Barcode	254
Outlinefonts	126
Maßeinheit zum Vergrößern	130
Skalieren	129
overstrike	153, 155

P

Page ID	278
Papier einlegen	36
Papierablagefach, Auswahl	40
Papieranforderung	36
Papiereinzug, Auswahl	37
Papierformat, Umschaltung	34
Parameter, Symbolik der	23
PDF 417 Barcode	256
Peel-Off	72
Philosophie	15
Pixelfonts	
Vergrößern von Zeichen	127
Polarkoordinaten	198
positive Halbzeilenschaltung	125
Postnet Barcode	254
Power On Werte	290
Prescribe	68
Print Engine – Befehle	303

Q

QR-Code Barcode	268
-----------------------	-----

R

Ränder.....	89
auf Einschaltblätter setzen.....	94
auf Papierränder setzen.....	95
linken Texttrand setzen.....	90
oberen Texttrand setzen.....	91
rechten Texttrand setzen.....	93
unteren Texttrand setzen.....	92
Raster.....	149
Raster laden.....	151
Raster, Graustufen.....	150
Rasterarten, Tabelle.....	315
Rasterhinterlegung.....	116, 117
Read ID.....	274
Real-Time Clock.....	294
Rechtsbündig.....	96
Relatives Positionieren	
Grafik.....	179
replace.....	153, 155
Reset - Befehl.....	54, 320
RFID Daten.....	278
RFID Daten schreiben.....	275
RFID Datenabfrage.....	274, 276
RFID Etiketten.....	271
RFID FG.....	271

S

Schlüsselwort.....	219
Schneidebefehl.....	72
Schreiben, RFID.....	275
Schreibrichtung mit Winkelangabe, Grafikfonts.....	133
Schreibrichtung, Auswahl.....	109
Schriftart, Auswahl.....	108
Schwärzung.....	303
Seitenbeschreibungssprache IDOL.....	28
Seiten-Identifikationsnummer.....	276
Seitennummer	
Drucken.....	62
Seitennummerierung.....	61
Zähler erhöhen.....	62, 63
Zähler setzen.....	61

Zähler verringern	63
Separator	278
Skalierfont	
Breite	132
Höhe.....	131
Slashed Zero	99
Softkey	289
Speichermode	153
Speichern und Wiederherstellen Textparameter	
Definition.....	214
Sperrschrift	
Befehl 1	122
Befehl 2.....	123
Spezial - Befehle	299
SPS – Control	305
Stacker Mapping	40
Staple.....	73
Status Kanal	300
Statusabfrage	301
Statusmeldungen.....	278
Stickfonts	
Maßeinheit zum Vergrößern.....	130
Skalieren	129
T	
Tabulator	100
horizontal	101
löschen.....	105
vertikal	102
Tabulator (horizontal) ausführen	103
Tabulator (vertikal) ausführen	104
Tear off.....	72
temporäre Konfigurationsänderungen	28
Terminator	278
Textgestaltungsfunktion	96
Textstring	278
Thermo Mode.....	304
Tiefstellen.....	125
Tools, Software.....	21
Transparentcode	24
Transparentcode, temporär	28

Two-Up Mode 59

U

Übersicht Befehle..... 317
Übersicht Grafikfunktionen und Darstellungselemente 201
Umranden von Grafik-Flächen 156
Unterstreichen
 Befehl 1 120
 Befehl 2 121
UPC A 234
UPC E..... 234
UPC-A Barcode 251
UPC-E Barcode 251
UPS MaxiCode Barcode 262
USD-5 Barcode 259
User Configs..... 291

V

Vektorfonts
 mit Breiten und Höhenfaktor 128
Vektorgrafik..... 16
Vergrößerung von Zeichen, Pixelfonts 127
Vervielfältigen von Informationsbereichen 49
vordefinierte Konfigurationssätze 291

Z

Zeichenneigung mit Winkelangabe, Grafikfonts 134
Zeichensätze 16
Zeichenschrittweite
 horizontal 110
 vertikal 112
Zeichentabelle, Auswahl 106
Zeilenabstand 112
Zentrieren..... 96