

**Honrad Albers**

**HP-41 Barcodes mit  
dem HP-IL-System**

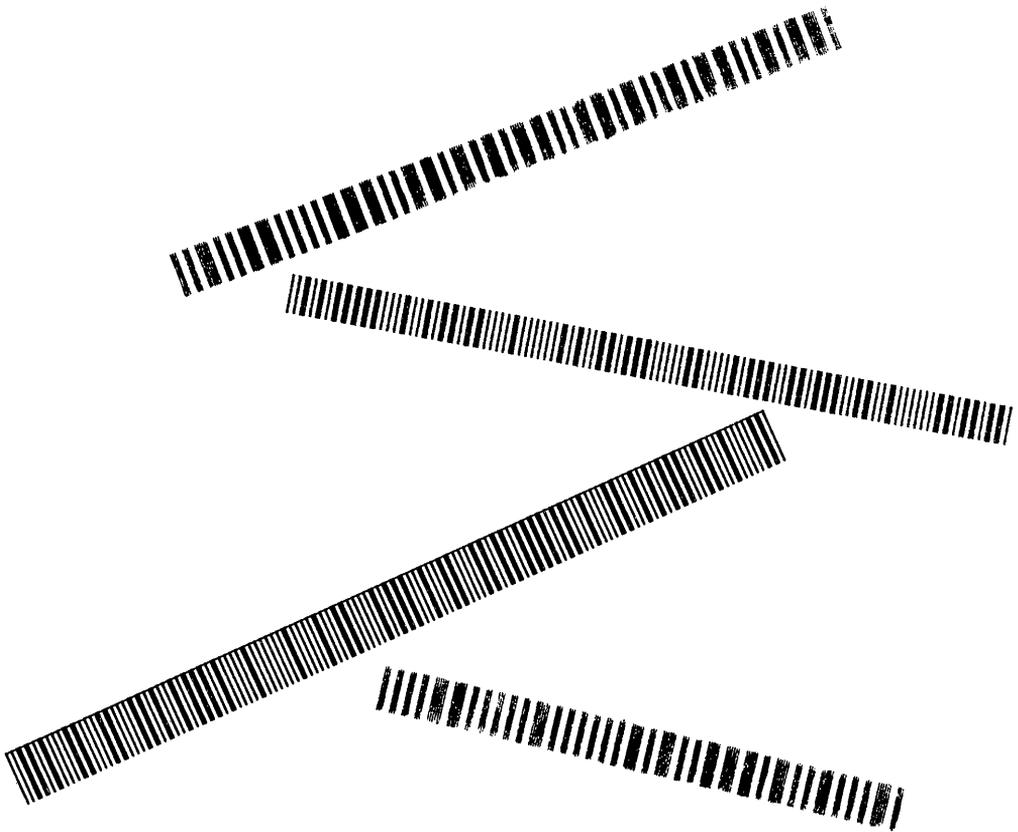


**Helder mann Verlag Berlin**



Honrad Albers

# HP-41 Barcodes mit dem HP-IL-System



**Heldermann Verlag Berlin**

Konrad Albers  
Postfach 6023  
Südring 42  
2300 Klausdorf

**CIP - Kurztitelaufnahme der Deutschen Bibliothek**

Albers, Konrad: HP-41 Barcodes mit dem HP-IL-System / Konrad Albers.-  
Berlin: Heldermann, 1986.  
ISBN 3-88538-804-9



**UNIVERSITÄTS-  
BIBLIOTHEK**

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

C Copyright 1986, Heldermann Verlag, Nassauische Str. 26, D-1000 Berlin 31.

ISBN 3-88538-804-9

## INHALTSVERZEICHNIS

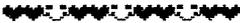
Vor- und Dankeswort des Autors .....	xiii
--------------------------------------	------

### Kapitel 1 HP 41 BARCODES UND DIE GERÄTE

I. Was sind Barcodes? .....	1
II. Vor- und Nachteile von Barcodes und Lesestift für HP 41 .....	1
a. Vorteile .....	1
b. Nachteile .....	2
III. Die Barcodetypen zum HP 41 - eine Übersicht .....	2
a. Barcodes von Alpha- und synthetischen Textzeilen, Druckersonderzeichen .....	2
b. Barcodes von Zahlen und für sequentiell einzulesende Daten .....	3
c. Barcodes von Anweisungen, Befehlen, Funktionen und für die Synthetik .....	3
d. Programmbarcodes .....	3
IV. Grundsätzliches .....	3
a. Zum Aufbau .....	3
1. Die Byte-Tabelle und das Codierungsschema mit Binärzahlen .....	4
2. Der Typindikator und weitere Informationen .....	9
3. Die Prüfsummen 4-Bit, 8-Bit und die Spiegelprüfsumme .....	9
b. Zur Herstellung .....	10
1. Drucker 82162A / Plotter 7470A - ThinkJet-Drucker 2225B - Ein Vergleich .....	11
2. Rechner HP 41/71/75 und Drucker HP-IL 82162A .....	12
Der ESCAPE-Modus .....	12
Der 8-Bit-Modus .....	13
Noch ein Wort zum 8-Bit Spaltenmodus .....	14
2.1 Das Druckerpapier, Ausnutzung, Archivierung .....	16
2.2 Der Akkuzustand .....	16
2.3 Druckerumbauten .....	17
3. Rechner HP 41/71/75 und HP-IL Plotter 7470A .....	17
3.1 Plotterpapier, Stifte und Tusche .....	17
4. Wann Plotter - wann Drucker? .....	18
5. Die Programme zur Barcodeherstellung für HP 41 .....	18
V. Rechner und notwendige oder wünschenswerte Geräte .....	19
a. Barcode-Leser HP 82153A - notwendig für das Einlesen .....	19
b. HP-IL-Modul 82160A und HP-IL-Drucker 82162A - notwendig zur Herstellung .....	19
c. XF - Erweitertes Funktions/Speicher-Modul HP 82180A - nötig für PRGM-Barcodes .....	19
d. Plotter-Modul HP 82184A - wünschenswert/notwendig .....	19
e. HP-IL-Plotter 7470A - wünschenswert .....	19
f. Port-Extender - evtl. notwendig .....	20

### Kapitel 2 ALPHA-EINZELZEICHEN, ALPHA- UND SYNTHETISCHE TEXTZEILEN DRUCKER-SONDERZEICHEN

I. Alpha-Einzelzeichen .....	21
a. Zweibyte Funktions- und Druckerzeichen 0-127 .....	21
1. Aufbau .....	21
1.1 Der Typindikator .....	22
1.2 Spezieller typbezogener Dezimalwert .....	22
1.3 Die 4-Bit Prüfsumme .....	22
2. Herstellen von Zweibyte Funktions- und Druckersonderzeichen .....	23

2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "A127BC" ...	65/23
2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "A127Bc" .....	66/23
2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "A127bc" .....	66/24
3.	Fertigen von "Zweibyte Funktions- und Druckerzeichen 0-127" auf dem Plotter .....	24
3.1	Rechner HP 41 CV - Plotter, XF- und PM PRGM "127Pbc" .....	67/25
4.	Arbeiten mit 'Zweibyte Funktions- und Druckerzeichen 0-127' ....	26
5.	Tabellen der 'ZWEIBYTE FUNKTIONS- UND DRUCKERZEICHEN 0-127' ....	27
b.	Dreibyte Alpha-Einzelzeichen 0-255 :replace und rappend .....	29
1.	Aufbau .....	29
1.1	Der Typindikator .....	29
1.2	Die 8-Bit Prüfsumme .....	29
2.	Herstellen von Dreibyte Alpha-Einzelzeichen 0-255 .....	29
2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "A255BC" ...	70/29
2.2	Rechner HP 41 C - Drucker und XF-Modul PRGM "A255Bc" .....	71/31
2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "A255bc" .....	72/31
3.	Herstellen von Dreibyte Alpha-Einzelzeichen 0-255 auf dem Plotter .....	31
3.1	Rechner HP 41 CV - Plotter, XF- und PM PRGM "255Pbc" .....	73/31
4.	Arbeiten mit Dreibyte Alpha-Einzelzeichen 0-255 .....	33
5.	Tabellen der 'ALPHA-EINZELZEICHEN 0-255 :REPLACE- rAPPEND' ...ab	34
<b>II.</b>	<b>Alpha- und synthetische Textzeilen</b> .....	42
a.	Alpha-Textzeilen .....	42
1.	Aufbau .....	42
1.1	Der Typindikator .....	42
1.2	Die 8-Bit Prüfsumme und der Text .....	43
2.	Herstellen von Text-BC auf dem Thermodrucker 82162A .....	43
2.1	Nur Rechner HP 41 CV - Drucker, keine Module PRGM "TEXTBC" ..	75/43
2.2	Rechner HP 41 CV - Drucker, XF-Modul PRGM "TEXTBc" .....	77/44
2.3	Rechner HP 41 CV - Drucker, XF- und PM PRGM "TEXTbc" .....	78/44
3.	Herstellen von Text-BC mit dem ThinkJet-Drucker HP-IL 2225B ....	46
3.1	Rechner HP 41 CV - ThinkJet-Drucker, XF-M PRGM "TXTJbc" .....	79/46
4.	Arbeiten mit Alpha-Textzeilen .....	48
b.	Synthetische Programmierung: was ist das? .....	48
c.	Synthetische Textzeilen .....	48
1.	Aufbau, der Typindikator und die 8-Bit Prüfsumme .....	48
2.	Herstellen von "SYNTEXTZEILEN" auf dem Thermodrucker .....	49
2.1	Nur Rechner HP 41 CV - Drucker, keine Module PRGM "SYNTBC" ..	84/49
2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "SYNTBc" .....	85/50
2.3	Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "SYNTbc" ..	87/51
3.	Arbeiten mit synthetischen Textzeilen .....	51
3.1	Massensteuerung aller Flags .....	51
	Flag-Bedeutungen und Flag-Zustand .....	52
3.2	Flagsteuerung mit 'STOFLAG' .....	54
<b>III.</b>	<b>Druckersonderzeichen</b>  .....	55
1.	Aufbau, Typindikator und Prüfsumme .....	55
2.	Herstellen von Druckersonderzeichen .....	55
2.1	Rechner HP 41 CV - Drucker, keine Module PRGM "BLDSBC" .....	88/55
2.2	Rechner HP 41 CV - Drucker, XF-Modul PRGM "BLDSBc" .....	89/57
2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "BLDSbc" .....	91/57
3.	Arbeiten mit Druckersonderzeichen-Barcodes .....	58
3.1	Umrechnungsprogramm: "BLDSDZ" .....	92/59
3.2	Umrechnungsprogramm: "DZBLDS" .....	93/60
3.3	Längere Druckersonderzeichen  .....	60
	- Der 8-Bit-Modus .....	60
	- ESCAPE-Grafik-Modus .....	63
	- 8-Bit-ESCAPE-Grafik-Modus-Mix .....	64
	Programmbarcodes für Kapitel 2 .....	ab 65

## Kapitel 3 ZAHLEN UND SEQUENTIELLE DATEN

<b>I. Zahlen</b>	96
1. Aufbau	96
1.1 Der Typindikator	97
1.2 Die 8-Bit Prüfsumme	97
2. Herstellen von Zahlenbarcodes aus dem X-Register	98
2.1 Nur Rechner HP 41 CV - Drucker, keine Module PRGM "ZALNBC"	114/98
2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "ZALNBc"	117/100
2.3 Rechner HP 41 C - Drucker, nur Plottermodul PRGM "ZALNbc"	119/101
3. Vielziffrige Zahlen in Programmen	102
3.1 Umrechnungsprogramm "ZXRADB"	119/102
- Noch eine kurze Betrachtung zur Programmierlösung	103
4. Kurzformexponenten	103
4.1 Kurzformexponenten-Programm "KFEXBC"	120/104
5. Arbeiten mit Zahlenbarcodes	105
<b>II. Sequentielle Daten</b>	105
1. Aufbau	105
1.1 Der Typindikator	106
1.2 Die Folgenummer	106
1.3 Die 8-Bit Prüfsumme	107
2. Herstellen von sequentiellen Daten-Barcodes	107
2.1 Nur Rechner HP 41 CV - Drucker, keine Module PRGM "ZFDABC"	121/107
2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "ZFDAbc"	125/110
2.3 Rechner HP 41 C - Drucker, XF- und PM PRGM "ZFDAbc"	129/111
- Hinweis zu "ZALNBC" im Vergleich zu "ZFDABC"	112
3. Arbeiten mit Barcodes für sequentielle Daten	112
- Einschränkung für Alphadaten	112
- Möglichkeiten mit der Plottermodul-Funktion 'BCREGX'	112
- Programmbarcodes für Kapitel 3	ab 114

## Kapitel 4 ANWEISUNGEN, BEFEHLE, FUNKTIONEN UND SYNTHETISCHES ZUM PROGRAMMIEREN

<b>I. 1-Byte-Tastenfeldfunktionen-Barcodes</b>	130
1. Aufbau, Typindikator und Spiegelprüfsumme	130
2. Herstellen von 1-Byte-Tastenfeldfunktionen auf dem Thermodrucker	130
2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "1BYTBC"	211/130
2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "1BYTBc"	212/130
2.3 Rechner HP 41 C - Drucker, Plottermodul PRGM "1BYTbc"	213/132
- Noch ein Wort zu 'R/S-STOP'	132
<b>II. 2-Byte - Tastenfeldfunktionen ohne Argument</b>	132
a. Programmierbare Rechnerfunktionen	132
1. Aufbau	133
1.1 Der Typindikator	133
1.2 Der Dezimalwert des Funktionscode	133
1.3 Die 4-Bit Prüfsumme	133
2. Herstellen von 2-Byte-Barcodes der programmierbaren Rechnerfunktionen	133
2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "2BYTBC"	214/133
2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "2BYTBc"	215/135
- Kurze Betrachtung zu einer anderen PRGM-Lösung mit XF-M	135
2.3 Rechner HP 41 C - Drucker, XF- und PM PRGM "2BYTbc"	216/135
b. Die Funktion 'INDIREKT' als Barcode	135
1. Aufbau, Typindikator, 4-Bit Prüfsumme und Funktionscode	136
2. Herstellen von Barcodes für die 'IND'-Funktion	136
2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "INDBC"	136

2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "INDBc" .....	218/137
2.3	Rechner HP 41 C - Drucker, nur Plottermodul PRGM "INDBc" ..	218/137
c.	Nicht programmierbare Rechnerfunktionen ohne Argument .....	137
1.	Aufbau .....	137
1.1	Der Typindikator .....	138
1.2	Der Dezimalcode der nichtprogrammierbaren Funktion .....	138
1.3	Die 4-Bit Prüfsumme .....	138
2.	Herstellen von Barcodes nicht programmierbarer Rechnerfunktionen .....	138
2.1	Nur Rechner HP 41 C Drucker, keine Module PRGM "NPFBC" ....	219/138
2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "NPFbc" .....	220/138
2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "NPFbc" .....	221/139
d.	Besondere Funktionen und deren Zweibytebarcodes .....	140
1.	↑-- Module Mathematik, Statistik und Vermessung .....	140
2.	μ-- Module Finanzen, Netzwerkanalyse und Baustatik .....	141
	- Weitere bisher 'unbekannte Größen' .....	141
3.	eGØBEEP-- - HP-IL-Funktionen .....	141
	- 'NEWM' und 'LIST' programmierbar - 'PRP' "NONEXISTENT" .....	142
4.	-- 'GTO IND ___' und 'XEQ IND ___' .....	143
5.	<del>XT+H</del> IÄ___ 'XEQ 00' bis 'XEQ e' = 000 - 255 .....	143
6.	Der 'Q-Bote' .....	143
7.	Byteschnapper 'Snap 2' bis 'Snap 9' .....	145
e.	Synthetische Befehle aus Textzeilen .....	147
1.	Aufbau, Typindikator und 8-Bit Prüfsumme .....	148
2.	Herstellen von BC für 'SNAP 2'- Programmierung auf dem Thermodrucker .....	148
2.1	Nur Rechner HP 41 C- Drucker, keine Module PRGM "SNP2BC" ..	221/148
3.	Herstellen der Tabellen für 'SNAP 2'- Programmierung auf dem Plotter .....	149
3.1	Rechner HP 41 CV - Plotter, Plottermodul PRGM "S2Pbc" ....	222/149
	Tabellen "BARCODES FÜR SYNTHETISCHES PROGRAMMIEREN MIT SNAP 2"	150
f.	XROM-Funktionen .....	153
1.	Aufbau .....	153
1.1	Der Typindikator und die übrigen Kenndaten .....	153
1.2	Die XROM-Identifikationsnummer und der Funktionscode .....	153
1.3	Die 4-Bit Prüfsumme .....	154
2.	Herstellen von Barcodes für XROM-Funktionen auf dem Thermodrucker .....	154
2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "XROMBC" .	225/154
2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "XROMbc" .....	226/155
2.3	Rechner HP 41 C - Drucker, Plottermodul PRGM "XROMbc" ....	227/155
3.	Herstellen von XROM-Tabellen mit dem Plotter .....	156
3.1	Rechner HP 41 CV - Plotter, Plottermodul PRGM "XRMPbc" ....	228/156
	Funktionen mit XROM-Nr. des HP-IL-Moduls .....	158
	Funktionen mit XROM-Nr. des X/Funktions-Moduls .....	159
	Funktionen mit XROM-Nr. des TIME-Moduls .....	160
	Funktionen mit XROM-Nr. des Magnetkartenlesers .....	161
	Funktionen mit XROM-Nr. des Barcode-Lesers .....	161
	Funktionen mit XROM-Nr. der Plotter-Moduls .....	162
	Funktionen mit XROM-Nr. des Extend. I/O-Moduls .....	163
	Funktionen mit XROM-Nr. des Development-Moduls .....	164
	Funktionen mit XROM-Nr. des -W&W CCD A-Moduls .....	165
	Funktionen mit XROM-Nr. des CCD-I/O FNS-Moduls .....	166
g.	XROM-Nummern von synthetischen Befehlen gibt es nicht .....	167
h.	Arbeiten mit Tastenfeld- und XROM-Barcodes .....	169
	- Nicht programmierbare Funktionen durch die Hintertür .....	169
III.	<b>Mehrbytebarcodes - Tastenfeldfunktionen mit und ohne Argument</b> .....	170
a.	Prgmierbare sowie nicht prgmierbare Rechner- und XROM-Funktionen .	170

1.	Aufbau .....	170
1.1	Der Typindikator .....	171
1.2	Die Dezimalwerte des Funktionscodes .....	171
1.3	Die 8-Bit Prüfsumme .....	171
2.	XROM-XEQ Funktionsbarcodes: der Unterschied .....	171
3.	Das Umrechnungsprogramm "XR-XEQ" .....	230/172
4.	Herstellen von XEQ-Funktionsbarcodes .....	172
4.1	Nur Rechner HP 41 CV - Drucker, keine Module PRGM "XEQBC" ..	231/172
4.2	Rechner HP 41 CV - Drucker, XF-Modul PRGM "XEQbc" .....	234/175
4.3	Rechner HP 41 CV - Drucker, XF- und PM PRGM "XEQbc" .....	236/176
b.	Nicht programmierbare Rechner- und XROM-Funktionen mit Argument ..	177
1.	Die CATALOG-Funktion 'CAT' .....	177
1.1	Aufbau .....	177
1.2	Der Typindikator, Identifikations- und CAtalog-Code .....	177
1.3	Die 8-Bit Prüfsumme .....	177
1.4	Herstellen von 'CAT' - 0 bis 6 Barcodes .....	177
1.4.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "CATBC" ..	239/177
1.4.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "CATbc" .....	239/178
1.4.3	Rechner HP 41 C - Drucker, Plottermodul PRGM "CATbc" .....	240/178
2.	Die Funktionen 'GTO.' und 'GTO..' .....	178
2.1	Aufbau .....	179
2.2	Der Typindikator und die Zeilennummer .....	179
2.3	Die 8-Bit Prüfsumme .....	179
2.4	Herstellen von 'GTO.'- und 'GTO..' -Barcodes .....	179
2.4.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "GTOZBC" ..	240/179
2.4.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "GTOZbc" .....	242/179
2.4.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "GTOZbc" .....	243/180
3.	Die Funktion 'DELETE' - Löschen einer Anzahl PRGM-Zeilen .....	181
3.1	Aufbau, Typindikator, Zeilenanzahl und 8-Bit Prüfsumme .....	181
3.2	Herstellen von 'DEL___' Barcodes .....	181
3.2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "DELBC" ..	243/181
3.2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "DELbc" .....	244/181
3.2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "DELbc" .....	245/181
4.	Die Funktion 'COPY'ren von XROM-Programmen in den Rechner .....	183
4.1	Aufbau .....	183
4.2	Der Typindikator, Identifikations-Code und Programm-Name .....	183
4.3	Die 8-Bit Prüfsumme .....	183
4.4	Herstellen von 'COPY' - Barcodes .....	183
4.4.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "COPYBC" ..	246/183
4.4.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "COPYbc" .....	247/183
4.4.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "COPYbc" .....	249/185
5.	Die Funktion 'CLP' - Löschen von Programmen .....	185
5.1	Aufbau, Typindikator, Identifikationscode, P-Name und Prf-Summe	185
5.2	Herstellen von 'CLP_' - Barcodes .....	185
5.2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "CLPBC" ..	249/185
5.2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "CLPbc" .....	251/186
5.2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "CLPbc" .....	252/186
6.	Die Funktion 'SIZE___' - Einstellen einer Registerzuweisung ...	186
6.1	Aufbau, Typindikator, Idi-Code, SIZE-Anzahl und 8-Bit Prf-Summe	187
6.2	Herstellen von 'SIZE___' - Barcodes .....	187
6.2.1	Nur Rechner HP 41 C Drucker, keine Module PRGM "SIZEBC" ...	253/187
6.2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "SIZEbc" .....	254/188
6.2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "SIZEbc" .....	254/188
7.	Die Funktion 'ASN ___' - Tastenzuweisungen .....	189
7.1	Aufbau .....	189
7.2	Der Typindikator, Identifikations- und Tastencode .....	189
7.3	Definition und Berechnung des Tastencode .....	189
7.4	Die 8-Bit Prüfsumme .....	189

7.5	Herstellen von ASN-Barcodes .....	190
7.5.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "ASNBC" ..	255/191
7.5.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "ASNbc" .....	257/192
7.5.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "ASNbc" .....	259/192
7.6	Tastenbelegung im Vergleich zu Barcodebenutzung .....	192
8.	Die Funktion 'LIST___' - Ausdruck von Programmzeilen .....	193
8.1	Aufbau .....	193
8.2	Der Typindikator und die XEQ-Identifikation .....	193
8.3	Die 'LIST___' - Anzahl .....	193
8.4	Die 8-Bit Prüfsumme .....	193
8.5	Herstellen von 'LIST___' - Barcodes .....	193
8.5.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "LISTBC" ..	260/193
8.5.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "LISTbc" .....	261/194
8.5.3	Rechner HP 41 C - Drucker, XF und PM PRGM "LISTbc" .....	262/195
9.	Die Funktion 'PRP' - Ausdruck von Programm listings .....	196
9.1	Aufbau, Typindikator, Identifikations-Code und Prüfsumme .....	196
9.2	Herstellen von 'PRP'-Barcodes .....	196
9.2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "PRPBC" ..	262/196
9.2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "PRPbc" .....	264/197
9.2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "PRPbc" .....	265/197
10.	Die Funktion 'NEWM' - Initialisierung einer Kassette/Diskette ..	197
10.1	Aufbau, Typindikator, Identifikation und Anzahl der Einträge ..	197
10.2	Herstellen von 'NEWM'-Barcodes .....	197
10.2.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "NEWMBC" ..	266/197
10.2.2	Rechner HP 41 C - Drucker, XF-Modul PRGM "NEWMbc" .....	267/197
10.2.3	Rechner HP 41 C - Drucker, XF- und PM PRGM "NEWMbc" .....	268/199
11.	Die Funktion 'PLNG' - Feststellen einer PRGM-Länge (CCD-Modul) ..	199
c.	Barcodes von 'GTO/XEQ/LBL "ALPHA"' - zur Direktausführung .....	199
1.	Aufbau .....	199
2.	Der Typindikator mit Identifikations-Code .....	199
3.	Die 8-Bit Prüfsumme .....	200
4.	Herstellen von 'GTO/XEQ/LBL "ALPHA"' - Barcodes .....	200
4.1	Nur Rechner HP 41CV - Drucker, keine Module PRGM "GXLABC" ..	268/200
4.2	Rechner HP 41 CV - Drucker, XF-Modul PRGM "GXLABc" .....	270/201
4.3	Rechner HP 41 CV - Drucker, XF- und PM PRGM "GXLABc" .....	272/202
d.	Arbeiten mit Mehrbytebarcodes .....	202
e.	Das Rätselprogramm "Wunsch" .....	203
1.	Nur Rechner HP 41 C - Drucker, keine Module PRGM "Wunsch" ..	271/203
f.	Simuliertes synthetisches Tastenfeld mit PRGM "WDL" .....	203
	und SYNFKTION-Barcodes .....	203
1.	Rechner HP 41 C - Drucker, keine Module PRGM "WDL" .....	274/204
2.	Aufbau .....	205
3.	Der Typindikator, die SYNFKTION und das 'END' .....	205
4.	Die 8-Bit Prüfsumme .....	205
5.	Herstellen von SYNFKTION-Barcodes .....	205
5.1	Nur Rechner HP 41 C - Drucker, keine Module PRGM "SSTFBC" ..	274/205
5.2	Rechner HP 41 CV - Plotter, Plottermodul PRGM "SSTFbc" ....	275/206
6.	'BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD' .....	ab 208
	Programmbarcodes für Kapitel 4 .....	ab 211

## Kapitel 5 PROGRAMM-BARCODES UND BARCODEANALYSE

I.	Barcodes von Programmen .....	278
1.	Globale Marken, Tastenzuweisungen, Autostart und das 'END' ....	278
2.	Aufbau .....	279
2.1	Der Typindikator und die übrigen Leitinformationen .....	279
2.2	'Abgebrochene' Funktionen und Länge einer Reihe .....	279
2.3	Die Zeilennummer und die 8-Bit Prüfsumme .....	281

3.	Herstellen von Programmbarcodes auf dem Thermodrucker	281
3.1	Rechner HP 41 CV - Drucker, XF-Modul PRGM "PRGMbc"	293/281
3.2	Rechner HP 41 CV - Drucker, XF- und PM PRGM "PRGMbc"	295/283
4.	Herstellen von Programm-Barcodes auf dem ThinkJet-Drucker	284
4.1	Rechner HP 41 CV - ThinkJet-Drucker, XF- + PM PRGM "PRTJbc"	297/285
5.	Herstellen von Programmbarcodes auf dem Plotter 7470A	286
5.1	Rechner HP 41 CV - Plotter, XF- und PM PRGM "PGMPbc"	302/286
	Einige Bemerkungen zum PRGM-Längentest und zu weiteren Programmlösungen	288
<b>II.</b>	<b>Barcodeanalyse</b>	289
1.	Nur Rechner HP 41 C - Drucker, PRGM "BCANLY"	306/289
	Eine kurze Betrachtung zum Nutzungsgrad der Typindikator-Möglichkeiten	291
	Programmbarcodes für Kapitel 5	ab 292

## Kapitel 6 DER 'EAN' UND ANDERE BARCODES

<b>I.</b>	<b>Barcodes mit vier Strichen</b>	308
a.	Der Europäische Artikel-Nummern-Barcode 'EAN'	308
b.	Der 'UPC' Universal Product Code	308
<b>II.</b>	<b>Barcodes mit zwei Strichbreiten</b>	309
a.	Andere Barcodes	309
<b>III.</b>	<b>Zukünftige Weiterentwicklungen</b>	311
<b>II.</b>	<b>Schlußwort</b>	311
	<b>Anhang:</b> 'STO b'-PRGM "PTbc" für HP 41CX, PM und ThinkJet-Drucker	312
	<b>Stichwortverzeichnis</b>	318
	<b>Literaturhinweise</b>	320

## VERZEICHNIS DER PROGRAMME

Name	Inhalt	Bytes	Reg	Seiten
Kapitel 2				
"A127BC" - Zweibyte Funktions- und Druckerzeichen 0-127		209	30	65/23
"A127Bc"		170	25	66/23
"A127bc"		118	17	66/24
"127Pbc" - Anfertigung einer Tabelle - Plotter		530	76	67/25
"A255BC" - Dreibyte Alpha-Einzelzeichen 0-255		250	36	70/29
"A255Bc"		208	30	71/31
"A255bc"		154	22	72/31
"255Pbc" - Anfertigung einer Tabelle - Plotter		552	79	73/31
"TEXTBC" - Textzeilen aus dem Alpha-Register		382	55	75/43
"TEXTBc"		302	44	77/44
"TEXTbc"		201	29	78/44
"TXTJbc" - desgleichen mit ThinkJet-Drucker		1053	151	79/46
"SYNTBC" - Synthetische Textzeilen nach Einzelbyte-Eingabe		308	44	84/49
"SYNTBc"		251	36	85/50
"SYNTbc"		191	28	87/51
"BLDSBC" - Druckersonderzeichen mit 'BLDSPEC'		380	55	88/55
"BLDSBc"		307	44	89/57
"BLDSbc"		203	29	91/57
"BLDSdz" - Umrechnung 'BLDSPEC'-SD-Zahl in Dezimalwert		306	44	92/59
"DZBLDS" - Umrechnung Dezimalwert in 'BLDSPEC'-SD-Zahl		308	44	93/60
Kapitel 3				
"ZALNBC" - Zahlen aus dem X-Register		674	97	114/98
"ZALNBc"		527	76	117/100
"ZALNbc"		39	6	119/101
"ZXRADB" - Umrechnung Zahlen aus X in Alpha-Dezimalbytes		181	26	119/102
"KFEXBC" - Kurzformexponenten		258	37	120/104
"ZFDABC" - Zahlen und Folgedaten für sequentielle Eingabe		889	127	121/107
"ZFDABc"		739	106	125/110
"ZFDAbc"		125	18	129/111
Kapitel 4				
"1BYTBC" - 1-Byte Tastenfeldfunktionen		285	41	211/130
"1BYTBc"		256	37	212/130
"1BYTbc"		219	32	213/132
"2BYTBC" - Zweibyte Tastenfeldfunktionen ohne Argument		245	35	214/133
"2BYTBc"		215	31	215/135
"2BYTbc"		169	25	216/135
"INDBC" - Die Funktion 'INDIREKT'		106	16	217/136
"INDBc"		79	12	218/137
"INDbc"		45	7	218/137
"NPFBC" - Nicht prgmierbare Rechnerfunktionen ohne Argument		209	30	219/138
"NPFBc"		179	26	220/138
"NPFbc"		118	17	221/139

Die Seitenzahlen vor '/' bezeichnen die Barcode- und nach '/' die Textseiten

Name	Inhalt	Bytes	Req	Seiten
"SNP2BC"	- 'SNAP 2'-BC für synthetische Programmierung	184	27	221/148
"S2Pbc"	- Tabellen von Barcodes für 'SNAP 2' - Plotter	598	86	222/149
"XROMBC"	- XROM-Funktionen	247	36	225/154
"XROMBc"		217	31	226/155
"XROMbc"		167	24	227/155
"XRMPbc"	- Tabellen von XROM-Funktionen	508	73	228/156
"XR-XEQ"	- Umrechnung von XROM-Zahlen in Bytedezimalwerte	186	27	230/172
"XEQBC"	- 'XEQ'-Rechner- und XROM-Funktionen	590	85	231/172
"XEQBc"		560	80	234/175
"XEQbc"		494	71	236/176
"CATBC"	- Die CATALOG-Funktion 'CAT'	135	20	239/177
"CATBc"		131	19	239/178
"CATbc"		74	11	240/178
"GTOZBC"	- Die Funktionen 'GTO.____' und 'GTO..'	227	33	240/179
"GTOZBc"		197	29	242/179
"GTOZbc"		142	21	243/180
"DELBC"	- Die Funktion 'DELETE' - Löschen von PRGM-Zeilen	213	31	243/181
"DELBc"		183	27	244/181
"DELbc"		128	19	245/181
"COPYBC"	- 'COPY'ren von XROM-Programmen in den Rechner	325	47	246/183
"COPYBc"		254	37	247/183
"COPYbc"		131	19	249/185
"CLPBC"	- Die Funktion 'CLP __' Löschen von Programmen	309	45	249/185
"CLPBc"		241	35	251/186
"CLPbc"		113	17	252/186
"SIZEBC"	- Die Funktion 'SIZE____' - Registerzuweisung	206	30	253/187
"SIZEBc"		176	26	254/188
"SIZEbc"		120	18	254/188
"ASNBC"	- Die Funktion 'ASN __' - Tastenzuweisung	415	60	255/191
"ASNBc"		342	49	257/192
"ASNbc"		213	31	259/192
"LISTBC"	- Die Funktion 'LIST____' - Ausdruck von PRGM-Zeilen	223	32	260/193
"LISTBc"		193	28	261/194
"LISTbc"		129	19	262/195
"PRPBC"	- Die Funktion 'PRP __' - Druck von PRGM-Listings	335	48	262/196
"PRPBc"		262	38	264/197
"PRPbc"		121	18	265/197
"NEWMBC"	- Die Funktion 'NEWM____' - Mediuminitialisierung	216	31	266/197
"NEWMBc"		186	27	267/197
"NEWMbc"		126	18	268/199
"GXLABC"	- 'GTO/XEQ/LBL "ALPHA"' - Direktausführung	405	58	268/200
"GXLABc"		334	48	270/201
"GXLABc"		206	30	272/202
"Wunsch"	- Ein Rätselprogramm	142	21	273/203
"WDL"	- Eingabeprogramm simuliertes synthetisches T-Feld	34	5	274/204

Die Programme für die Funktionen 'CAT' bis 'NEWM' stellen Barcodes her, welche das Argument beinhalten. Nicht nur 'COPY' - sondern 'COPY "PRPLOT" als BCODE.

Name	Inhalt	Bytes	Reg	Seiten
"SSTFBC"	- Simulierte Synthetische Tastenfeld-BC	204	30	274/205
"SSTPbc"	- Tabellen für SYNFUNKTIONEN-BC - Plotter	517	74	275/206
Kapitel 5				
"PRGMbc"	- Programmbarcodes	820	118	292/281
"PRGMbc"		292	42	295/283
"PRTJbc"	- Programmbarcodes - ThinkJet-Drucker	1040	149	297/285
"PGMPbc"	- Programmbarcodes - Plotter	772	111	302/286
"BCANLY"	- Barcode-Analyse	233	34	306/289

## Vor- und Dankeswort

Bei uns in Norddeutschland kennen wir ein plattdeutsches Sprichwort: "Den een sien Uhl is den annern sien Nachtigal". Wäre die Dokumentation von HP über Barcodes nicht so 'een Uhl', hätte ich vermutlich keine Idee entwickelt dies Buch zu schreiben. Ich stellte mir aber vor, viele andere HP-41-Benutzer würden diese Lücke ebenfalls bedauern; so begann ich mit der Arbeit, um das Thema Barcodes zum HP 41 so erschöpfend wie möglich darzustellen. Nun lege ich Ihnen, der großen Anwendergemeinde dieses Rechners, ein Fachbuch vor, von dem ich wünsche, daß es allen eine Arbeitshilfe wird und darüberhinaus dem Interessierten die Zusammenhänge darlegt.

Auf besondere Höflichkeitsfloskeln bei den Erläuterungen zur Programmbedienung habe ich im Interesse kurzer, dennoch verständlicher Darstellung verzichtet. Deswegen mögen die 'Anweisungen' manchmal etwas schroff wirken. Bitte seien Sie dieserhalb nachsichtig; danke. Ins Detail gehende Programmklärungen sind nicht vorgesehen. Dies Buch ist mehr ein Dienstleistungsbuch, welches vorwiegend das Arbeiten mit Barcodes zum Ziel hat. Gleichwohl besteht die Möglichkeit, es regelrecht durcharbeiten, um sich mit der Materie Barcodes zum HP 41 in allen Einzelheiten vertraut zu machen. Hierbei muß über die Programmlösungen dann Schritt für Schritt nachgedacht werden. Insgesamt habe ich mich bemüht, einerseits dem Neuling nicht zuviel abzuverlangen, andererseits den erfahrenen Anwender nicht zu langweilen.

Bleiben Fragen offen, bin ich gern bereit im Rahmen der Möglichkeiten Antworten zu geben. Doch bitte ich darum, zuvor die jeweiligen Zusammenhänge noch einmal aufmerksam und eventuell unter Zuhilfenahme der aufgeführten weiteren Literatur durcharbeiten; vielleicht fällt 'der Groschen' dann auch ohne Rückfragen. Die Experten unter meinen Lesern werden möglicherweise sowohl an meinen Programmen, als auch am Inhalt diesen oder jenen Abschnitts Anmerkungen anbringen wollen. Sie werden etwa fragen, ob es noch nötig sei, den 'Q-Boten' oder die nicht existierenden 'XROM'-Zahlen von synthetischen Befehlen zu behandeln. Bitte bedenken Sie zweierlei:

- Anders kann man vieles machen. Gefragt ist die bessere, kürzere, schnellere Lösung ohne Komfortverlust. Hierfür bin ich stets, vor allem im Interesse der Weiterentwicklung, aufgeschlossen.
- Die kürzere Programmlösung ist oftmals schwieriger zu verstehen. Ein solches Buch muß aber für den Anfänger, der grundlegendes wissen möchte, verständlich bleiben, dem Kenner etwas bieten, der dadurch vielleicht ein 'Aha'-Erlebnis hat und auch den Experten ansprechen, dessen Reaktion hie und da 'ja aber' sein mag.

Bevor Sie zu Verbesserungsvorschlägen ansetzen - nicht Veränderungen - bitte, prüfen Sie Ihre Anregungen unter diesen Gesichtspunkten. Halten sie nach Ihrer Meinung stand, sind Ihre besseren Lösungen sehr willkommen.

Ich möchte hier auch meinen Helfern und Beratern noch einmal danken, die meine Arbeit unterstützt und gefördert haben. Ohne eine Rangfolge danke ich namentlich: den Mitarbeitern der Firma Carl Reese, Kiel, besonders Frau Siegmund, der 'mars plot'-Beraterin und Herrn Lorenz, HP-Abteilung, für die geduldige Bereitstellung immer neuer Muster, Herrn Volker Radek und Herrn Dieter Kapp, von HP Bad Homburg, Herrn Hans-Peter Frahm, von Triumph-Adler, Kiel, das Buch habe ich auf einer TA SE 310 mit Blocksatz geschrieben, Herrn Heinz Dalkowski und Herrn Dr. Norbert Helder mann für ihre Ratschläge und guten Anregungen.

Regelrechte Barcode-Arbeitssysteme, so wie für den HP 41 in diesem Buch behandelt sind mir von anderer Seite nicht bekanntgeworden. Auch in dieser Hinsicht ist unser Rechner offenbar 'einsame Spitze'. Ihnen allen wünsche ich viel Erfolg und Spaß bei der Arbeit.



## Kapitel 1

### HP 41 BARCODES UND DIE GERÄTE

#### I. Was sind Barcodes?

Das Wort BAR kommt aus der englischen Sprache und bedeutet in unserem Zusammenhang Strich. Barcodes sind mit Strichen unterschiedlicher Stärke und nach bestimmter Anordnung verschlüsselte Informationen. Mit einem optischen Lesestift können sie gelesen und bei Einsatz entsprechender Geräte in Klarinformation dargestellt werden.

Umgekehrt kann man mit einem geeigneten Drucker oder Plotter Klarinformationen in Strichcodes verschlüsseln.



Beispiel für HP 41 Barcode

Hewlett-Packard hat für das Rechner-System HP 41 einen speziellen Barcode entwickelt, der ein in sich geschlossenes System darstellt und mit dem das Arbeiten erheblich erleichtert wird. Eine Reihe von Eingaben sind ohne weiteren Aufwand überhaupt nur mit Barcodes und Lesestift möglich. Das vorliegende Buch erschließt Ihnen die problemlose Nutzenanwendung aller Vorteile.

Andere Barcodes sind in verschiedenen Bereichen im Einsatz; sie dienen überwiegend der Datenerfassung. Jedermann kennt inzwischen den auf vielen Verpackungen vorhandenen EAN. Es ist dies der 'Europäische Artikel-Nummern'-Barcode, den der HP-Lesestift natürlich nicht 'versteh'.



Beispiele für den EAN

Außerdem sind eine Reihe weiterer Typen entwickelt worden, die jedoch entweder in Europa keine Bedeutung haben oder die hier speziell für interne Zwecke eingesetzt werden. Beispiele: industrielle Dateneingabe, Lagerkontrolle, Bibliotheken, Personenkennanlagen, Testlabors in Krankenhäusern, Fotografische Industrie, Einzelhandel oder Dokumentiersysteme. Diese Barcodes werden in Kapitel 6 vorgestellt. Der EAN wird genauer erläutert. Weiterhin wird ein kurzer Ausblick auf die zukünftige Entwicklung gegeben.

#### II. Vor- und Nachteile von Barcodes und Lesestift für HP 41

Wo Licht ist, gibt es natürlich auch Schatten. Die Vorteile überwiegen indes bei weitem. Um jedoch keine unerfüllbaren Erwartungen zu wecken, werden Vor- und Nachteile aufgeführt.

##### a. Vorteile

- Der Lesestift ist ein recht preiswertes, enorm leistungsfähiges Zusatzgerät.

- Barcodes lassen sich mit entsprechenden Programmen auf Druckern oder dem Plotter auch für Druckvorlagen einfach, schnell und sehr preisgünstig herstellen.
- Informationsträger ist Papier - ebenfalls sehr preiswert.
- Dies ist somit die einzige Möglichkeit kostensparender Massenvervielfältigung von Daten und Programmen jedweder Art.
- Das Einlesen von Strichcodes mit dem Lesestift geschieht wesentlich schneller als die Eingabe über das Tastenfeld.
- Auch ungeübte Anwender beherrschen die Technik sofort.
- Tipfehler sind ausgeschlossen.
- Tastenbelegungen oft gebrauchter Funktionen entfallen - dies spart Programmspeicher ein.
- Viele nicht tastbare Daten (z.B. Kleinbuchstaben) werden von Barcodes problemlos eingegeben.
- Synthetische Befehle und Anweisungen oder Texte sind sehr schnell und einfach einzulesen.
- Nur von Barcodes - neuerlich auch mit dem CCD-Modul - ist es möglich, ohne angeschlossene Peripheriegeräte XROM-Befehle zu programmieren.
- Programme können auch als Barcodes 'PV' geschützt werden.
- Tastenbelegungen von Programmen können in Programm-BC mit aufgezeichnet werden.

#### b. Nachteile

- Die Informationsdichte ist geringer als bei anderen HP 41 Datenträgern: Magnetkarten, Kassetten oder Disketten.
- Längere Programme sind deswegen nicht so schnell einzulesen wie von anderen Datenträgern.
- Der Lesestift testet nicht, ob ausreichend Speicherplatz vorhanden ist.
- Das Rechnerflag 11 für automatischen Programmstart kann in Barcodes nicht verschlüsselt werden.
- Genau wie beim Magnetkartenleser ist programmgesteuertes Einlesen nicht möglich. Eine Aufforderung zur Programm- oder Dateneingabe mit anschließendem Autostart kann jedoch programmiert werden.
- Gewisse ASCII-Zeichen können nicht direkt in Register eingelesen werden. Man muß den 'Umweg' über den Programmspeicher gehen.
- Benutzt man keine Schutzfolie zum Einlesen, können Barcodes infolge Papierabrieb unbrauchbar werden.

### III. Die Barcodetypen zum HP 41 - eine Übersicht

Sämtliche Text-, Zahlen-, Programm- oder sonstige Funktionen können als Barcodes dargestellt werden. Gemäß seinem Aufbau gehört der HP 41 BC zur Grundform des Schwarzcode, einer Strichcodierung mit zwei verschiedenen 'Balkenbreiten' und gleich großen Zwischenräumen. Im Gegensatz dazu stehen Strichcodierungen anderer BC mit bis zu 4 verschiedenen Balkenbreiten und unterschiedlichen Zwischenräumen, die ebenfalls Informationsträger sind. Den HP 41 Barcode kann man in vier Kategorien einteilen:

#### a. Barcodes von Alpha- und synthetischen Textzeilen, Druckersonderzeichen

Diese Barcodes werden in Kapitel 2 behandelt. Jede Art von Alpha-Einzelzeichen oder Zeichenfolgen, seien es nun Texte oder irgendwelche Kombinationen von ASCII-Zeichen für synthetische Programmierung, zur Flagsteuerung oder für Druckersonderzeichen, die mit der Funktion 'BLDSPEC' oder synthetisch aufgebaut wurden, können als Barcodes dargestellt werden. Innerhalb dieses Typs gibt es weitere Unterscheidungen; z.B. ob der Inhalt des Alpha-Registers überschrieben werden soll oder ob der Text an den Inhalt dieses Registers anzuhängen ist. Diese Art Barcodes ist besonders für die Synthetik von großer Bedeutung.

### b. Barcodes von Zahlen und für sequentiell einzulesende Daten

BC dieses Typs werden in Kapitel 3 besprochen. Zahlen aus dem X-Register werden bis zur höchstmöglichen Stellenzahl positiv oder negativ dargestellt. Hier ist ganz besonders der Barcodetyp für das sequentielle Einlesen von Zahlen oder Alpha-Daten in Rechnerregister mit der Funktion 'WNDDTA' interessant. Hervorragend ist dabei, daß solche Daten zwangsläufig der Reihe nach und auch in Register > 99 direkt, ohne den Umweg über indirekte Adressierung eingelesen werden können.

### c. Barcodes von Anweisungen, Befehlen, Funktionen und für die Synthetik

Solche BC werden in Kapitel 4 behandelt. Dieser Typ BC ist wohl der vielfältigste überhaupt. Wir werden eine größere Zahl von recht unterschiedlichen Programmen zur BC-Herstellung kennenlernen. Bei diesen Barcodes ist herausragend, es müssen für die Programmierung von XROM-Befehlen keine Peripheriegeräte angeschlossen sein. Der Unterschied zu XEQ-BC wird klar herausgestellt. Synthetische Befehle können verblüffend einfach eingegeben werden. Ein synthetisches Tastenfeld wird simuliert. Ein Rätselprogramm lädt zum Tüfteln ein, und es werden Barcodes vorgestell, die 'offiziell' nicht existieren.

### d. Programmbarcodes

BC von Programmen sind ihrer Natur nach in erster Linie Speicherbarcodes. Deswegen kommt in Kapitel 5 der Plotter auch nicht zu kurz. Außerdem ist das 'XF-Modul' für die Anfertigung von PRGM-BC mit dem HP 41 und dem IL-Drucker unbedingt notwendig. Warum Programmbarcodes nur mit Rechner und Drucker, ohne Module schlecht möglich sind, wird genau erläutert. Wichtig ist, Programmbarcodes können Tastenbelegungen und PV-Schutz aufzeichnen.

## IV. Grundsätzliches

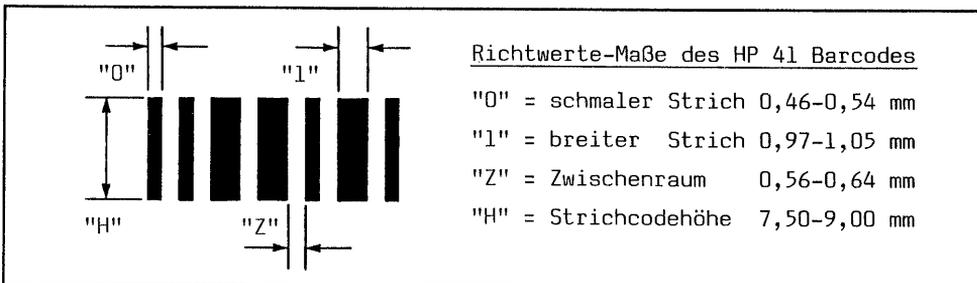
Zum besseren Detailverständnis ist es vorteilhaft, die für alle BC-Typen zum HP 41 geltenden Grundlagen zu kennen. Wer bereits umfangreiches Wissen hierrüber hat, kann dies Kapitel getrost überspringen und zur Sache ab Kapitel 2 kommen.

Alle HP 41 Barcodes und nur davon wird bis Kapitel 5 gesprochen, haben die folgenden Merkmale gemeinsam:

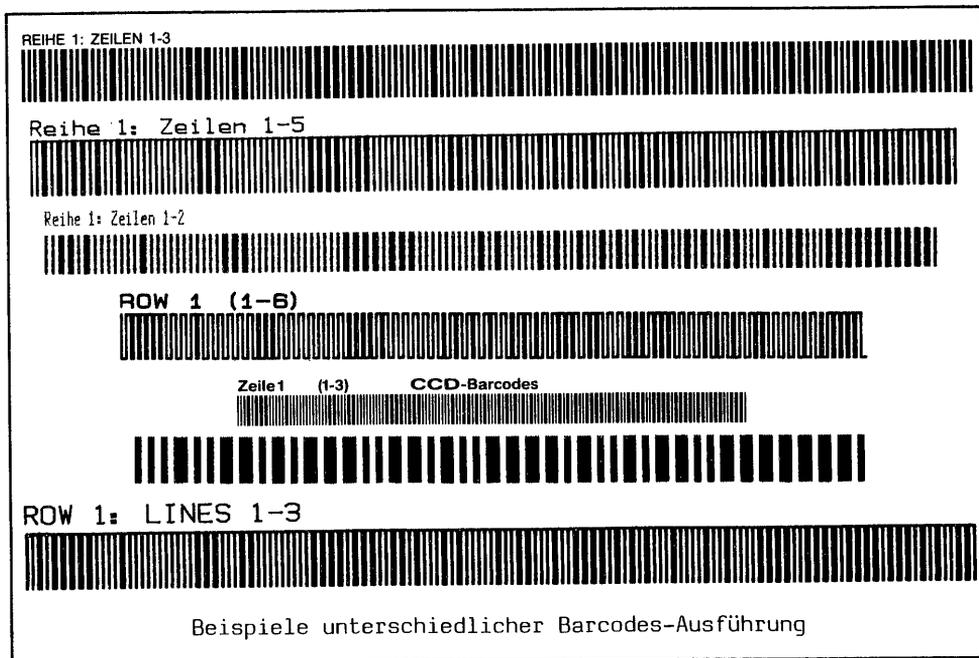
- Es ist ein 'Zweibalken-Barcode' mit gleich großen Zwischenräumen, die als Informationsträger ungenutzt sind.
- Kontrastreiche Darstellung nur schwarz auf weißem Grund - keine Farbdarstellung möglich.
- Maximale Informationsdichte einschließlich Prüfsumme und Indikatoren ist 16 Bytes pro Zeile. Kürzere, bis ein Byte-BC sind möglich - längere nicht.
- Die Zusatzinformation für die Kennung der Abtastrichtung (Start/Stop-Signal) ist immer gleich.

### a. Zum Aufbau

Der äußere Aufbau muß gewissen Anforderungen an die Abmessungen der breiten und schmalen Striche sowie der Zwischenräume genügen:



Werden diese Anforderungen erfüllt, so kann die Ausführung der Barcodes innerhalb recht weiter Grenzen variieren. Eines ist dabei allerdings sehr wichtig: die Zwischenräume müssen gleich groß sein. Die schmalen und breiten Striche können noch so präzise hergestellt sein, unterschiedliche Zwischenräume machen den Barcode unleserlich. Nachfolgend einige Beispiele unterschiedlich hergestellter Barcodes, die alle vom Lesestift dekodiert werden.



### 1. Die Byte-Tabelle und das Codierungsschema mit Binärzahlen

Die Byte-Tabelle zum HP 41 auf den Seiten 6 und 7 zeigt die dezimalen und hexadezimalen Werte der Bytes 0-255, gesamt also 256 Bytes mit ihren Druckerzeichen. Für unseren Zweck ist allgemein nur der Dezimalwert von Interesse. Jedes Byte kann je nach Stellung im Speicher eine ganze oder teilweise Funktion oder ein Alphazeichen darstellen. So stellt z.B. das Byte 68 allein die Funktion 'X<Y?' dar. Geht ein Textbyte (ab Nr. 240) voraus, so ist das Byte 68 der Buchstabe "D". Steht ein Vorsilbenbyte voran, z.B. Nr. 145 = 'STO\_', so wird dies Byte zur Nachsilbe für das Speicherregister 68. Der Wert des Byte wird hierbei nie verändert, sondern lediglich die Stellung im Speicher, bzw. innerhalb einer Barcodereihe. Somit kann für jedes Byte immer mit gleichem Wert gearbeitet werden. Nähere Einzelheiten hierzu findet man im Buch "Synthetische Programmierung auf dem HP-41C/CV" von W.C. Wickes, Helder mann Verlag Berlin.

Damit der Lesestift die Information einer BC-Reihe aufnehmen und für den Rechner dekodieren kann, muß ein Codierungsschema zugrunde liegen, welches mit optischen Mitteln erfassbar ist. Die Grundlage hierzu ist das Binär- oder Dualzahlensystem. Dieses Stellenwertsystem benötigt für die Darstellung jeder Zahl nur zwei Ziffern: 0 und 1. Null ist die Leerstelle. Die Basis für jede Stelle mit einer Eins ist die Zahl 2. (Die 1 ist ein Symbolwert. In manchen Darstellungen von Dualzahlen wird für eine besetzte Stelle ein 'L' geschrieben). Die Stellen werden von rechts nach links mit 0,1,2,3 usw. gezählt. Der Stellenwert ist die Potenz aus Basis und Stellenzahl als Exponent. Diese Stelle wird als Bit (englisch binary digit) bezeichnet. Unser System besteht aus 8 Bit. Es sind von

rechts nach links die Stellen 0 bis 7. Diese 8 Bit bilden zusammen ein Byte, (Byte ist ein künstliches Wort) welches seinerseits die kleinste Informationseinheit ist. Je nachdem, ob das Bit besetzt ist = 1 oder nicht = 0, repräsentiert es einen Wert, der sich aus 2 oder 0 (Basis), hoch Stellenzahl (Exponent) errechnet. Demgemäß sieht das Byte 68 so aus:

0	64	0	0	0	4	0	0	← Bit Dezimalwert
0	1	0	0	0	1	0	0	← Basis 0 oder 2
7	6	5	4	3	2	1	0	← Stellenziffer als Exponent

Nur die Stellen (Bit) 2 und 6 sind mit einer 1 besetzt und haben demzufolge die Basis 2; die übrigen Bit haben die Basis 0. Die Queraddition ergibt  $2^2 = 4 + 2^6 = 64$  zusammen = 68. Es können nun auch gleich die Grenzen des 8-Bit-Systems erkannt werden: Die kleinste Zahl als Bytewert ist 00000000 = Null. Sind alle Bit mit einer 1 besetzt, ergibt sich

128	64	32	16	8	4	2	1	← Bit Dezimalwert
1	1	1	1	1	1	1	1	← Alle Bit = 1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	← Basis überall 2
7	6	5	4	3	2	1	0	← Stellenziffer = Exponent

← 8 Bit = 1 Byte →

Die Queraddition aller Dezimalwerte der 8 Bit ergibt 255. Dies ist somit der höchstmögliche Bytewert in einem 8-Bit-System. Insgesamt stehen 0-255 = 256 unterschiedliche Bytes zur Verfügung.

In verschiedenen Rechengängen, besonders bei Umformungen und bei der Darstellung von Indikatoren, wird mit dem Halbbyte gerechnet. Ein Halbbyte heißt Nybble (eigentlich nibble = abknabbern). Das Nybble besteht entweder aus den Bit 0-3 = 2. Nybble oder 4-7 = 1. Nybble. Ein Halbbyte aus anderen Bitfolgen, z.B. 2-5, gibt es nicht. Die größte mit einem Nybble darstellbare Zahl ist 1111 = 15. Die Darstellungsbreite reicht hier von 0-15 = insgesamt 16 Ziffern. Die auf den Seiten 6 und 7 stehende Byte-Tabelle spiegelt diese Systematik wieder. Es gibt 0-15 = 16 Spalten mit 0-15 = 16 Reihen an Bytes; gesamt  $16 \times 16 = 256$ . Wird nämlich zur 00001111 = 15 die Zahl 1 hinzugezählt, so tritt ein 'Überlauf' vom 2. ins 1. Nybble ein. Alle Bit des 2. Nybble werden Null, das 1. Bit des 1. Nybble wird 1 = dezimal 16. Die Zählung beginnt im 2. Nybble erneut bis wieder 15 überschritten ist, und es tritt ein weiterer Überlauf auf 32 ein, bis schließlich 255 erreicht ist. Die 'Trennstelle' zwischen 1. und 2. Nybble ist der Überlauf nach 16: '0001 0000'. Diese Zahl begegnet uns tatsächlich auch häufig beim Zerlegen von Bytewerten in Nybble. Nehmen wir unser Byte 68 und zerlegen es:  $68:4=4$  Rest 4. Die erste 4 ist die Bitstelle im 1. Nybble und die zweite 4 diejenige im 2. Halbbyte: '0100 0100'. Die Zusammenfügung ergibt wieder 68 nämlich:  $4 \times 16 = 64$  (1. Nybble) + 4 (2. Nybble) = 68.

Wir erkennen nun auch, wie ein Byte für den Lesestift erfassbar gemacht werden kann. Es wird einfach die 1 mit einem dicken Strich und die Null mit einem dünnen Strich dargestellt. Die Zwischenräume trennen lediglich die beiden Striche für den Zweck besserer Lesbarkeit.

Die folgenden praktischen Beispiele mit Barcodes, die auf dem IL-Drucker hergestellt sind, wiederholen die Zusammenhänge noch einmal.

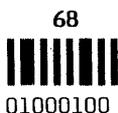
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8 $\blacklozenge$ LBL 00 01 $\overline{\text{K}}$	2 $\overline{\text{X}}$ LBL 01 02 $\overline{\text{K}}$	3 $\leftarrow$ LBL 02 03 $\overline{\text{K}}$	4 $\alpha$ LBL 03 04 $\overline{\text{K}}$	5 $\beta$ LBL 04 05 $\overline{\text{K}}$	6 $\Gamma$ LBL 05 06 $\overline{\text{K}}$	7 $\downarrow$ LBL 06 07 $\overline{\text{K}}$	8 $\Delta$ LBL 07 08 $\overline{\text{K}}$	9 $\sigma$ LBL 08 09 $\overline{\text{K}}$	10 $\blacklozenge$ LBL 09 10 $\overline{\text{K}}$	11 $\blacktriangleright$ LBL 10 11 $\overline{\text{K}}$	12 $\mu$ LBL 11 12 $\overline{\text{K}}$	13 $\blacktriangleleft$ LBL 12 13 $\overline{\text{K}}$	14 $\blacktriangleright$ LBL 13 14 $\overline{\text{K}}$	15 $\blackstar$ LBL 140 15 $\overline{\text{K}}$
1	16 $\theta$ 1 17 $\Omega$	18 $\delta$ 2 18	19 $\acute{\alpha}$ 3 19	20 $\grave{\alpha}$ 4 20	21 $\check{\alpha}$ 5 21	22 $\grave{\alpha}$ 6 22	23 $\theta$ 7 23	24 $\ddot{\alpha}$ 8 24	25 $\theta$ 9 25	26 $\ddot{\alpha}$ 26	27 $\text{EEK}$ 27	28 $\text{NEG}$ 28	29 $\alpha$ 29	30 $\text{XEQ}$ 30	31 $\text{SPARE}$ 31
2	32 RCL 00 32 (space)	34 $\cdot$ RCL 01 34	35 $\#$ RCL 02 35	36 $\$$ RCL 03 36	37 $\%$ RCL 04 37	38 $\&$ RCL 05 38	39 $\cdot$ RCL 06 39	40 $\langle$ RCL 07 40	41 $\rangle$ RCL 08 41	42 $*$ RCL 09 42	43 $+$ RCL 10 43	44 $,$ RCL 11 44	45 $-$ RCL 12 45	46 $\cdot$ RCL 13 46	47 $\backslash$ RCL 14 47
3	48 $\theta$ STO 00 48	50 2 STO 01 50	51 3 STO 02 51	52 4 STO 03 52	53 5 STO 04 53	54 6 STO 05 54	55 7 STO 06 55	56 8 STO 07 56	57 9 STO 08 57	58 $:$ STO 09 58	59 $;$ STO 10 59	60 $<$ STO 11 60	61 $=$ STO 12 61	62 $>$ STO 13 62	63 $?$ STO 14 63
4	64 $e$ + 64	66 $B$ * 66	67 $C$ / 67	68 $D$ X<Y? 68	69 $E$ X>Y? 69	70 $F$ X $\leq$ Y? 70	71 $G$ $\Sigma$ + 71	72 $H$ $\Sigma$ - 72	73 $I$ HMS+ 73	74 $J$ HMS- 74	75 $K$ MOD 75	76 $L$ % 76	77 $M$ %CH 77	78 $N$ P-R 78	79 $O$ R-P 79
5	80 LN 80	82 R SQRT 82	83 S Y $\times$ 83	84 T CHS 84	85 U e $\times$ 85	86 V LOG 86	87 W 10 $\times$ 87	88 X e $\times$ -1 88	89 Y SIN 89	90 Z COS 90	91 $\Gamma$ TAN 91	92 $\backslash$ ASIN 92	93 J ACOS 93	94 $\uparrow$ ATAN 94	95 - DEC 95
6	96 1/X 96	98 b FACT 98	99 c X $\neq$ 0? 99	100 d X>0? 100	101 e LNI+X 101	102 f X<0? 102	103 g X=0? 103	104 h INT 104	105 i FRC 105	106 j D-R 106	107 k R-D 107	108 l HMS 108	109 m HR 109	110 n RND 110	111 o OCT 111
7	112 p CL $\Sigma$ T	114 r PI Y	115 s CLST X	116 t R $\uparrow$ L	117 u RDN M $\Gamma$	118 v LASTX N $\Sigma$	119 w CLX O $\Gamma$	120 x X=Y? P $\uparrow$	121 y X $\neq$ Y? Q $\equiv$	122 z SIGN T $\uparrow$	123 $\pi$ X $\leq$ 0? a	124 l MEAN b	125 $\rightarrow$ SDEV c	126 $\Sigma$ AVIEW d	127 t CLD e
0		2	3	4	5	6	7	8	9	A	B	C	D	E	F

Die Byte-Tabelle zum HP 41

8	128	◆	129	*	130	⊗	131	←	132	α	133	β	134	Γ	135	↓	136	Δ	137	σ	138	◆	139	↘	140	μ	141	↙	142	↖	143	⊗	
	DEG		RAD		GRAD		ENTER		STOP		RTN		BEEP		CLA		ASHF		PSE		CLRG		AOFF		AON		OFF		PROMPT		ADV		
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
9	144	⊖	145	Ω	146	⊗	147	⊖	148	α	149	⊖	150	α	151	⊖	152	⊖	153	⊖	154	⊖	155	⊖	156	⊖	157	⊖	158	⊖	159	⊖	
	RCL		STO		STO+		STO-		STO*		STO/		ISG		DSE		VIEW		ΣREG		ASTO		ARCL		FIX		SCI		ENG		STONE		
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
A	160	XROM	161	!	162	"	163	#	164	\$	165	%	166	&	167	.	168	<	169	>	170	*	171	+	172	,	173	-	174	.	175	/	
	XROM		XROM		XROM		XROM		XROM		XROM		XROM		XROM		SF		CF		FS?C		FC?C		FS?		FC?		GTO IND		XEQ IND		SPARE
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
B	176	⊖	177	1	178	2	179	3	180	4	181	5	182	6	183	7	184	8	185	9	186	:	187	:	188	<	189	=	190	>	191	?	
	SPARE		GTO 00		GTO 01		GTO 02		GTO 03		GTO 04		GTO 05		GTO 06		GTO 07		GTO 08		GTO 09		GTO 10		GTO 11		GTO 12		GTO 13		GTO 14		
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
C	192	⊖	193	⊖	194	⊖	195	⊖	196	⊖	197	⊖	198	⊖	199	⊖	200	⊖	201	⊖	202	⊖	203	⊖	204	⊖	205	⊖	206	⊖	207	⊖	
	GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		GLOBAL		X <		LBL
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
D	208	P	209	⊖	210	R	211	S	212	T	213	U	214	V	215	W	216	X	217	Y	218	Z	219	[	220	\	221	]	222	↑	223	-	
	GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		GTO		
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12
E	224	⊖	225	⊖	226	b	227	c	228	d	229	e	230	f	231	g	232	h	233	i	234	j	235	k	236	l	237	m	238	n	239	o	
	XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		XEQ		
	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
F	240	P	241	⊖	242	r	243	s	244	t	245	u	246	v	247	w	248	x	249	y	250	z	251	⊖	252	⊖	253	⊖	254	⊖	255	⊖	
	TEXT 0		TEXT 1		TEXT 2		TEXT 3		TEXT 4		TEXT 5		TEXT 6		TEXT 7		TEXT 8		TEXT 9		TEXT 10		TEXT 11		TEXT 12		TEXT 13		TEXT 14		TEXT 15		
	T	Z	Y	X	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖	⊖
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																	

Die Byte-Tabelle zum HP 41

Der Lesestift liest die schmalen und breiten Striche Bit für Bit und fügt jeweils 8 Bit zu einem Byte zusammen, welches je nach Stellung im Speicher oder in einer Barcodereihe interpretiert wird:



Steht diese Bit-Folge als Byte allein, ist es gemäß Byte-Tabelle die Rechner-Funktion 'X<Y?'.



Geht ein Text-Byte voraus, ist es der Buchstabe "D".



Steht ein Vorsilben-Byte voran, ist es dazu die Nachsilbe; hier z.B. 'St0 68'. Das Byte 68 ist dann die Registeradresse für das Daten-Speicherregister 68. Wäre eine andere Vorsilbe vorhanden, z.B. 'LBL', würde das Byte die Marke 68 darstellen. Es wird hierbei gleichzeitig deutlich, wie platzaufwendig die HP 41 Barcodes besonders mit dem IL-Drucker sind. Die gleich großen Trennräume zwischen 0 und 1 sind ungenutzt. In anderen Systemen, die dann allerdings nicht so problemlos mit einem relativ preisgünstigen Drucker hergestellt werden können, ist die Informationsdichte größer. Teils werden hier auch die Trennräume für Informationen genutzt.

Nun denn, hier ist dies nicht der Fall, dadurch um so einfacher. So allein geht es allerdings nicht; es sind für BC-Informationen noch eine ganze Anzahl 'Verwaltungskennzeichnungen' nötig. Zunächst muß der Lesestift die Abtastrichtung erkennen können, und er muß 'wissen', eine neue Informationsreihe beginnt: dazu dienen Start/Stop-Signale. Ohne solche Abtastidentifikation könnte nämlich unser Byte von links nach rechts 01000100 heißen, von rechts nach links hingegen 00100010 und dies wäre dann das Byte 34. Beim HP 41 Barcode wird Start/Stop und Abtastrichtung mit zwei zusätzlichen Strichen 00 links und zwei zusätzlichen Strichen 10 rechts definiert. Die Abtastrichtung ist hierdurch egal. Die Information wird stets korrekt gelesen, weil der Lesestift beim Einlesen 'verkehrt herum' die Datenfolge richtigstellt.

Nun ist es nicht möglich, eine Bytefolge einfach durch Hinzufügen der Start/Stop-Striche zu erstellen, weil eine Reihe wie:



nicht verstanden wird. Einlese-Ergebnis ist 'CKSUM ERR'. Die Verwaltung, die Leitinformation fehlt, um welche Art von Barcode es sich handelt, was damit geschehen soll und ob die Reihe logisch ist, d.h. ob die ausgewiesene Prüfsumme sich aus der vorhandenen Anzahl Bytes richtig ermittelt. Außerdem sind in der Leitinformation weitere typspezifische Daten verschlüsselt, welche dem Rechner mitteilen, wie die Barcodereihe zu verarbeiten ist. Z.B. Text-BC 'append' oder 'replace', Zahlen-BC einzeln oder Folgedaten für Register, PRGM-BC 'PV'-Schutz oder nicht, vollständig ausgeführte oder am Ende der BC-Reihe unterbrochene Funktion, XROM- oder XEQ-Befehl, u.s.w.; es gibt viele Unterscheidungen.

Ganz allgemein hat eine Barcodereihe die Form:

00	Prüfsumme	Indikator	weitere Informationen	Inhalt BC-Reihe	10
			←----- Leitinformation -----→		

Die Prüfsumme steht immer voran. Sie, und die übrigen Leitinformationen belegen dabei je nach Barcodetyp 1/2 bis 3 Bytes. Nach dieser 'Verwaltung' einer BC-Reihe folgt der eigentliche Inhalt, welcher vom Rechner verarbeitet wird.

## 2. Der Typindikator und weitere Informationen

In Abschnitt III. wurde eine Übersicht über die verschiedenen Barcodetypen gegeben. Der Typindikator teilt dem Rechner mit, um welchen Typ BC es sich handelt. Ferner enthält der Typindikator oft auch Informationen über Alternativen innerhalb des Typs. Nochmal Text als Beispiel: 'replace' oder 'append' und weiterhin ist die Anzahl Textzeichen der Reihe enthalten. Bei der Beschreibung der einzelnen Barcodetypen wird jeweils die Zusammensetzung der unterschiedlichen Typindikatoren genau besprochen.

Weitere Informationen sind z.B. bei Programmbarcodes nötig, um innerhalb einer Reihe nicht vollständig abgeschlossene Befehle zu erfassen. Es könnte sein, die Vorsilbe 'STO' ist das letzte Byte einer Reihe (max. 16 Bytes pro Reihe), und die folgende Reihe beginnt mit Byte 68. In diesem Fall muß innerhalb der Reihe enthalten sein, daß eine Zweibytefunktion abgebrochen wurde, damit 'STO' auf seine Nachsilbe 68 gewissermaßen warten kann. Werden längere Textzeilen als Barcodes verschlüsselt, muß auch hier im Indikator enthalten sein, die in der nächsten Reihe folgenden Bytes sind Buchstaben bis zu der vom Textbyte bezifferten Anzahl. Die jeweiligen Codezahlen der verschiedenen Typindikatoren sind von HP für diesen Barcode festgelegt und können nicht verändert werden.

## 3. Die Prüfsummen 4-Bit, 8-Bit und die Spiegelprüfsumme

Bis auf die Einbytefunktionen, die eine sogenannte Spiegelprüfsumme besitzen, wird jede Barcodereihe von einem Prüfsummen-Nybble mit 4-Bit Prüfsumme oder einem Prüfsummenbyte mit 8-Bit Prüfsumme angeführt.

4-Bit Prüfsummen kommen nur bei Zweibytefunktionen vor. Alle übrigen Mehrbyte-Barcodereihen haben eine 8-Bit Prüfsumme. Die Prüfsummen werden errechnet, indem die Dezimalwerte aller Nybble von Zweibytefunktionen, bzw. aller Bytes der Reihe addiert werden. Ist der Wert  $\leq 15$  bzw.  $\leq 255$ , so ist dies die Prüfsumme der Reihe. Ist die Summe der Addition  $> 15$  bzw.  $> 255$ , tritt ein Überlauf ein, weil es ja nicht möglich ist mit 4 Bit  $> 15$  oder mit 8 Bit  $> 255$  darzustellen. In diesem Fall ist der Wert des letztlich verbleibenden Überlaufs die Prüfsumme. Beispiel: Byteaddition = 315;  $315-255=$  Prüfsumme 60. Byteaddition = 1327; 1327 minus sooft 255 bis der Überlauf getilgt und die verbleibende Summe  $\leq 255$  ist. Programmtechnisch ist ein solches Subtraktionsverfahren natürlich nicht zu gebrauchen. Einfacher ist  $1327:256=5$  Rest 47; Prüfsumme  $5+47=52$ . Genauso berechnet sich die 4-Bit Prüfsumme. Es müssen dabei die Werte jedoch nicht unbedingt pro Nybble addiert werden, sondern es kann der Wert des 2. Byte und der Wert des 2. Nybble von Byte 1 zusammengezählt werden. Beispiel: 255 (Byte 2) + 13 (Nybble 2 Byte 1) =  $268:16=16$  Rest 12 Prüfsumme 28 - oh! 28 ist  $> 15$ , was nun? In diesem Fall wiederholt man die Rechnung einfach:  $28:16=1$  Rest 12 Prüfsumme 13. Bei Addition nur der Nybble ist das Ergebnis gleich. Die Zahl 255 heißt binär 11111111 - aufgeteilt in Nybble 1111 = 15 und 1111 = 15. Addition:  $15+15+13=43:16=2$  Rest 11 Prüfsumme 13. Auch bei den 8-Bit Prüfsummen wird die Rechnung nach gleichem Verfahren wiederholt, bis die Prüfsumme  $\leq 255$  ist.

Es gibt einige Besonderheiten, die in den entsprechenden Kapiteln genau erläutert werden. Bei Programmbarcodes beispielsweise, wird die Prüfsumme der

vorhergehenden Reihe mit in die Berechnung der Prüfsumme der nachfolgenden Reihe einbezogen.

Es stellt sich noch die Frage, ob eine Prüfsumme Null sein kann. Die Berechnungsroutine für die Prüfsumme ist die Addition aller Bytes (bzw. Nybble) :  $256 \text{ (bzw. } 16) = x \text{ Rest } y \text{ Prüfsumme } x+y$ . Bei einer Byteaddition 255 wäre die Prüfsumme 255. Aber auch bei  $255:256$ , um bei der Routine zu bleiben, ist das Ergebnis 0 Rest 255. Ist die Byteaddition 256, beträgt der Überlauf 1, und die Prüfsumme ist 1. Ebenso ergibt die Routine  $256:256=1 \text{ Rest } 0$ . Eine Prüfsumme 0 ist demzufolge ausgeschlossen.

Entwickeln wir nun noch eine allgemeine Routine für die Berechnung der Prüfsumme und untersuchen wie oft die Routine durchlaufen werden muß, um kein Ergebnis  $> 255$  zu erhalten. Angenommen, eine Byteaddition einer Reihe ergibt 511.  $511:256=1 \text{ Rest } 255$ , Addition = 256. Wie wir festgestellt haben, gibt es diese Prüfsumme in einem 8-Bit-System nicht. Die Rechnung muß wiederholt werden, und wir erhalten (siehe oben) Prüfsumme 1. Eine Solche Situation einer höheren Prüfsummenberechnung als 255 aus dem Ergebnis der Division durch 256 + Rest (Modulo) kann sich bei max. 16 Bytes in einer Reihe 14,xyz mal ergeben, weil Byte 16 die Prüfsumme ist. Angenommen 15 Bytes ergeben die Addition 3583, so ergibt sich  $:256 = 13 \text{ Rest } 255$  rechnerische Prüfsumme 268. Dies ist die höchstmögliche Byteaddition, welche einen Überlauf  $> 255$  verursacht. Schlußfolgerung: es genügt in jedem Fall die Prüfsummenroutine 2x zu durchlaufen. Dies gilt auch bei Prüfsummen, die aus Nybble berechnet werden.

Bleibt noch, eine geeignete Programmroutine zu entwickeln. Es handelt sich um eine 2x auszuführende Division mit Einbeziehung des Restes = Modulo. Im X-Register steht die Byteaddition. Die nachfolgende Rechneroutine erfüllt die Anforderungen, und sie wird bei vielen Herstellungsprogrammen angewendet. Eine nämliche Routine hat Andreas Marktscheffel in der CCD-Zeitschrift 'PRISMA' veröffentlicht. Die kürzeste Routine ist:  $255 \text{ (15) MOD } X=0? \text{ LASTX}$  von G. Kruse.

Die sogenannte Spiegelprüfsumme gibt es nur bei 1-Byte-Befehlen. Hier wird einfach das Informationsnybble 2 spiegelbildlich als Nybble 1 des betreffenden Byte dargestellt. Die Prüfsumme ist zugleich der Indikator. Das nachstehende Beispiel verdeutlicht dies:

```

01 SF xy
02 XEQ zz
03 LBL zz
04 RCL X
05 256 (16)
06 ST/Z
07 MOD
08 +
09 INT
10 FS?C xy
11 RTN

```



00 Start/Stop |1000|0001| 10 Start/Stop  
 Nybble für Spiegelprüfsumme Nybble für Information

### b. Zur Herstellung

Grundsätzlich lassen sich Barcodes auf allen grafikfähigen Druckern herstellen. Problemlos ist die Programmierung natürlich mit speziell dafür vorgesehenen Geräten. Alle Programme zur Barcodeherstellung, egal auf welchem Drucker oder Plotter sie angefertigt werden, zergliedern sich stets in zwei grobe Teile:

1. Das Zusammenfügen der Bytes einer Barcodereihe des entsprechenden Typs mit Prüfsumme, Indikator und evtl. weiteren Informationen sowie schließlich dem Inhalt der Reihe
2. Der Aufbereitung dieser Bytefolgen in Steuerbefehle zum eigentlichen Drucken der Barcodes.

Für einen bestimmten Drucker oder Plotter sind hierbei immer gerätespezifische Befehle zu verwenden. Im vorliegenden Buch sind einige Programme in Kapitel 2 und 5 für den ThinkJet-Drucker HP-IL 2225B geschrieben. Viele 'kleine' Programmbeispiele und alle Programm-XEQ's sind ebenfalls mit dem ThinkJet hergestellt. Alle übrigen sind für den IL-Drucker oder den Plotter 7470A vorgesehen.

### 1. Drucker 82162A / Plotter 7470A - ThinkJet-Drucker 2225B - Ein Vergleich

Der HP-IL-Drucker 82162A und der HP-IL-Plotter mit Plottermodul im Rechner sind speziell für die Herstellung von HP 41 Barcodes ausgerüstet. Für den Drucker ist kein weiteres Zubehör erforderlich.

Der ThinkJet-Drucker ist nicht besonders für die BC-Herstellung eingerichtet. Trotzdem lassen sich selbstverständlich Barcodes drucken, nur ist der Aufwand viel größer. Der Thermodrucker und auch der Plotter mit Plottermodul drucken direkt aus dem Alpharegister des Rechners. Die schmalen Striche, Zwischenräume und breite Striche werden automatisch gedruckt. Im Gegensatz hierzu müssen die Bytefolgen für den ThinkJet-Drucker erst in barcodedruckfähige 'Baustein-Bytefolgen' umbereitet werden. Dies ist grundsätzlich für alle anderen grafikfähigen Drucker ohne BC-Modus ähnlich.

Replace-Text: OK?



Diese Barcodereihe stellt die Bytefolgen 77 = Prüfsumme, 115 = Indikator: replace und die Information 79 = 0, 75 = K und 63 = ? dar. Mit Anfangs- und Endstrichen sieht diese Reihe binär so aus:

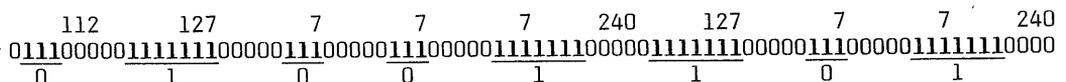
00		Anfangsstriche
01001101	=	77 Prüfsumme
01110011	=	115 Indikator
01001111	=	79 0
01001011	=	75 K
00111111	=	63 ?
10		Endstriche

Es muß nun eine Umformung der Bits eines jeden dieser Bytes in grafikdruckfähige Bausteine vorgenommen werden. Diese Bausteine sind ihrerseits Bytefolgen, welche im Grafikmodus eines Druckers schmale und breite Striche mit Zwischenräumen drucken.

Der ThinkJet-Drucker verarbeitet in diesem Modus nur ganze Bytes. Ist ein Bit '0' bleibt die Druckfläche auf dem Papier 'blank', ist es '1' wird ein Punkt gedruckt. Die Kapazität beträgt pro Zeile maximal 1280 Punkte; dies sind 160 Bytes x 8 Bit. Innerhalb dieser Grenze muß der Aufbau einer BC-Reihe bleiben. Es sind maximal 640 Punktzeilen möglich. Die Bausteine müssen so optimiert werden, daß die Informationsdichte wirtschaftlich bleibt, die Lesbarkeit nicht beeinträchtigt wird und die Programmlaufzeit mit dem HP 41 erträglich ist. In den erwähnten ThinkJet-Programmen Kapitel 2 und 5 werden als Bausteine die Bytes

- 112 = 01110000 = 3 Punkte aneinander - schmaler Strich
- 127 = 01111111 = 7 Punkte aneinander - breiter Strich
- 7 = 00000111 = Anschluß an 127 für schmalen Strich
- 240 = 11110000 = Anschluß an 7 für breiten Strich

verwendet. Die 5 Nullen sind der Trennraum zwischen schmalen und breiten Strichen. Die Umformung des Byte 77 in grafikdruckfähige Bausteine sieht dann so aus:



Es sind also 10 Bausteinbytes notwendig für den Barcodedruck des Byte 77. Je nach Anteil der Nullen und Einsen erfordert jedes Informationsbyte 8 - 12 Bausteine. 160 Bausteine stehen gesamt pro Zeile zur Verfügung, abzüglich 5 Bytes

für Anfangs- und Endstriche verbleiben 155 Bausteinbytes. Hieraus können unter Berücksichtigung der längstmöglichen Text- oder Programmzeile maximal 14 Bytes Gesamtinformation in einer BC-Reihe werden. Anhand der Programme wird der Aufbau näher erläutert.

Man erkennt jedoch jetzt bereits, welche Informationsmengen verarbeitet werden müssen. Einerseits ist der ThinkJet-Drucker für die Barcodeherstellung nicht so elegant zu programmieren, andererseits stößt unser HP 41 hier klar an seine Leistungsgrenze im Hinblick auf Übertragungsgeschwindigkeit.

## 2. Rechner HP 41/71/75 und Drucker HP-IL 82162A

Für die Steuerung der Druckerfunktionen wird ein Rechner benötigt, der die entsprechenden Befehle senden kann. Der Drucker besitzt die Möglichkeit Barcodes auf zwei verschiedene Arten zu erstellen; das Ergebnis ist gleich:

### Der ESCAPE-Modus

Stets nach dem Einschalten befindet sich der Drucker im ESCAPE-Modus. (Erklärung ESCAPE- und 8-Bit-Modus siehe Bedienungshandbuch IL-Drucker.) Mit dem Befehl 'OUTA' können in diesem Modus ASCII-codierte Zeichenketten aus dem Alpharegister gesendet werden, deren Bytefolgen der Drucker als Instruktion versteht. So ist es möglich, das Gerät in einen speziellen Druckermodus, den sogenannten 'Spaltenmodus' zu setzen, in welchem alle auf die Instruktionsbytes folgenden Zeichen als Barcodes ausgedruckt werden. Die z.B. für 2 Zeichen Barcodes nötigen Instruktionsbytes sind folgende:

Dezimalwerte: 27 42 122 | 48 50 66 | 161 129 - ALPHA-Inhalt: **E\*z02B!\***

- Die Bytes 27 42 122 versetzen den Drucker in den Spaltenmodus
- Die Bytes 48 50 66 teilen dem Drucker mit, es sind 02 folgende Zeichen als Barcodes zu drucken
- Die Bytes 161 129 sind die beiden als Barcodes auszudruckenden Bytes.

In diesem Beispiel stellen sie das Alphazeichen "A" aus der Tabelle der Funktions- und Alpha-Einzelzeichen 0-127 dar. In einer solchen Sequenz können bis zu 16B einschließlich Prüfsummen- und Leitinformationsbytes auf das B folgen.

Replace-Text: **E\*z02B!\***



Textzeile mit dem Lesestift einlesen, XEQ 'OUTA' ausführen, und es wird die folgende Barcodezeile gedruckt:



```

01*LBL *ESCAPE*
02 *E*z02B *
03 OUTA
04 END

```

Sofern mehr Bytes im Alpharegister vorhanden sind als durch die vor B definierte Anzahl (01-16), so werden diese nicht als Barcodes, sondern als Alphazeichen im Anschluß an die Barcodereihe linksbündig mit ausgedruckt, weil der Drucker dann nicht mehr im Spaltenmodus ist. Alphazeichen, die vor der Sequenz stehen, werden ignoriert.

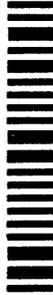
Weil der Empfang der entsprechenden Instruktionen den Drucker aktiviert, ist es prinzipiell gleichgültig mit welchem Rechner diese Befehle gesendet werden. Die gleichen Instruktionen können nämlich ebensogut mit dem Basic-Rechner HP 71 B oder dem HP 75 C/D gesendet werden. Damit ist es grundsätzlich möglich, HP 41 Barcodes auch mit Basic-Programmen auf dem Drucker zu erstellen. Auf der nächsten Seite finden Sie die obenbeschriebene Sequenz als Programmbeispiel in Basic für den HP 75. Außerdem ein Beispiel für überzählige Bytes.

```

10 PRINTER IS 'P1'
20 A$=CHR$(27)&"*z02B"
30 B$=CHR$(161)&CHR$(129)
)
40 PRINT A$&B$

```

Beispiele für überzählige Zeichen  
links 8-Bit- rechts ESCAPE-Modus



HP41

HP 41

Sollte übrigens der Drucker beim Testen dieses Beispiels auf dem HP 41 den Alpharegisterinhalt normal und nicht als Barcodes ausgeben, so ist der Drucker nach dem Einschalten inzwischen durch Betätigen irgendeiner Druckerfunktion automatisch in den 8-Bit-Modus versetzt. Schalten Sie den Drucker aus und wieder ein oder senden Sie die folgende Sequenz:

```

01*LBL "8BT/ESC"
02 SF 17
03 **
04 OUTA
05 END

```

—BARCODETEXTZEILE—

Replace-Text: !



Nur für Programmeinlesung

Flag 17 muß gesetzt sein, damit der Zeilenvorschub unterdrückt wird. Zeile 03 ist das Byte 252, welches den Drucker vom 8-Bit-Modus wieder in den ESCAPE-Modus versetzt. Dies Byte 252 kann nicht direkt in das Alpharegister eingelesen werden. Der Versuch friert die Anzeige harmlos ein. Durch Entfernen und Wiedereinsetzen des Lesestiftes wird das Display wieder aktiv. Als Programmzeile funktioniert alles bestens. Sofern Sie ein XF-Modul besitzen kann 252 'XT0A' eingegeben werden. Bytes mit Dezimalwerten > 239, also ab 240 können nicht direkt in Register eingelesen werden.

Der 8-Bit-Modus

Sobald irgendeine Druckerfunktion nach dem Einschalten ausgeführt wurde, befindet sich der Drucker automatisch im 8-Bit-Modus. Auch in diesem Modus können ASCII-codierte Zeichenketten aus dem Alpharegister als Barcodes gedruckt werden. Die hierfür nötigen Informationen für das im ESCAPE-Modus gedruckte "A" sehen folgendermaßen aus:

Dezimalwerte: 27 124 | 214 129 | 161 129 - ALPHA-Inhalt:

- Die Bytes 27 124 versetzen den Drucker vom ESCAPE-Modus ohne vorausgehende Ausführung einer Druckerfunktion in den 8-Bit-Modus.
- Die Bytes 214 129 setzen den Spaltenmodus, teilen dem Drucker mit, 2 Bytes sind als Barcodes zu drucken und drucken die Start/Stop-Striche.
- Die Bytes 161 129 sind die als Barcodes zu druckenden Bytes = "A".

Das Byte 128 teilt mit, es ist ein Byte als Barcode zu drucken, das Byte 143 sagt dem Rechner 16 Bytes sind Barcodes. Genau sind es 127 + n Bytes. Im Gegensatz zur Byteanzahl im ESCAPE-Modus, wo 2 Ziffern 01-16 vor dem B benötigt werden, braucht der 8-Bit-Modus nur jeweils 1 Zeichen zur Definition der Anzahl

Bytes in der Barcodezeile.

Sind mehr Zeichen im Alpharegister vorhanden als das Byte 127+n vorgibt, werden diese rechtsbündig im Anschluß an die Barcodereihe mit ausgedruckt. Vor der Sequenz stehende Bytes werden ignoriert. Gesetzte Drucker-Flags 12 oder 13 haben keinen Einfluß auf den Barcodedruck. Überzählige Bytes im 8-Bit-Modus werden doppeltbreit gedruckt.

Auch im 8-Bit-Modus könnten HP 41 Barcodes prinzipiell mit Basic-Programmen und den Rechnern HP 71B oder 75 C/D erstellt werden. Eine entsprechende Programmsequenz für unser Beispiel sieht so aus:

```
10 PRINTER IS 'p1'
20 A$=CHR$(27)&CHR$(124)
&CHR$(214)&CHR$(129)
30 B$=CHR$(161)&CHR$(129)
)
40 PRINT A$&B$
```



Praktisch ist dies angesichts der komfortablen Programmierung mit dem HP 41 und hier ganz besonders mit dem Plottermodul natürlich nicht von Bedeutung.

Noch ein Wort zum 8-Bit Spaltenmodus

Im ESCAPE-Modus wird der Drucker durch die Bytes 27 42 122 und im 8-Bit-Modus durch das Byte 214 in den Spaltenmodus gesetzt. Befindet sich der Drucker im ESCAPE-Modus, so ist die vollständige Sequenz 27 124 (Umschalten in den 8-Bit-Modus) und 214 einzugeben. Es gibt nun noch eine wesentlich elegantere Möglichkeit den Drucker gleichzeitig in den 8-Bit- und den Spaltenmodus zu versetzen: nämlich durch die Befehlsfolge 'CLX' 'ACCOL' (es kann auch 0 oder '.' in das X-Register geschrieben werden). Im Alpharegister braucht dann nur noch das Byte 127+n für die Byteanzahl Barcodes und die entsprechende Anzahl Informationsbytes zu stehen. Der Aufbau einer längeren Informationssequenz im Alpharegister entfällt. Außerdem ist es nur mit 'ACCOL' möglich Null-Bytes (Dezimal 0 nicht 48)

zu senden. Weder im ESCAPE- noch im 8-Bit-Modus werden Nullbytes mit den beschriebenen Sequenzen gedruckt. Die ESCAPE-Sequenz für das Drucken eines Null-Bytes wäre 'SF 17' - "253 27 42 122 48 49 66 0" - 'OUTA'. Die 8-Bit-Sequenz lautet 'SF 17' - "27 124 214 128 0" - 'OUTA'. In beiden Fällen werden nur die Start/Stop-Striche, nicht aber die Null, bestehend aus 8 schmalen Strichen gedruckt. Mit der Funktion 'ACCOL' heißt die Befehlsfolge 'CLX','SF 17','ACCOL','128','OUTA','CF 17','ACCOL'. Diese Sequenz druckt die nebenstehenden Barcodes Flag 17 muß vor 'OUTA' zur Unterdrückung des Zeilenvorschubes gesetzt sein, weil der Befehl 'OUTA' andernfalls den Barcodeausdruck eines nicht im Alpharegister vorhandenen Ersatzbyte auslöst. Bei gesetztem Flag 17 ist das Byte 128 die Instruktion "drucke ein Byte Barcode". Danach muß Flag 17 zur Freigabe des Zeilenvorschubs wieder gelöscht werden. 'ACCOL' bewirkt dann den Barcodedruck eines Null-Bytes. Sofern Flag 17 für nachfolgende Operationen gesetzt bleiben soll, kann anstelle 'CF 17' auch 'SF 25' (Fehlerignorflag) eingegeben werden. Geschieht dies nicht, erfolgt "TRANSMIT ERR".



Null-Byte

Wenn Sie diese Beispiele durchprobieren möchten, können Sie die nachstehenden Textzeilen entsprechend einlesen:

Für ESCAPE-Sequenz

Replace-Text: } 253 27 42 122 48 49 66 0



Erst als Programmzeile eingeben, dann 'SF 17','SST','OUTA'.

Für 8-Bit-Sequenz

Replace-Text: v 27 124 214 128 0



Auch diese Barcodes müssen zuerst als Programmzeile eingegeben werden

Für 'ACCOL'

Replace-Text: 128



Dies Byte kann direkt ins Alpharegister eingelesen werden

Bei Einsatz des Plottermoduls gibt es keine Einschränkung. Hiermit kann ein Null-Byte mit der Folge 'CLA','-1','BCD' als Barcode gedruckt werden. Zur Demonstration der verschiedenen Druckermodi nachfolgend drei kleine Programme, die Sie nacheinander mit dem Lesestift einlesen, packen (GTO..), ausführen und löschen können.

PROGRAMM : "ESCAPE" Benötigte Register: 5 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-3



Reihe 3: Zeilen 4-7



GTO ..



XEQ "ESCAPE"



CLP "ESCAPE"



01\*LBL "ESCAPE"  
02 SF 17  
03 "!"  
04 OUTA  
05 ADV  
06 ADV  
07 END

Zeile 03: 253 27 42 122 48 50 66 161 129 - Ohne 'LBL' und 'END' 16 Bytes

PROGRAMM : "8-BIT-" Benötigte Register: 4 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-5



Reihe 3: Zeilen 6-7



GTO ..



XEQ "8-BIT-"



CLP "8-BIT-"



01\*LBL "8-BIT-"  
02 SF 17  
03 "E" ~~"E"~~  
04 OUTA  
05 ADV  
06 ADV  
07 END

Zeile 03: 27 124 214 129 161 129 - Ohne 'LBL' und 'END' 13 Bytes

Man sieht, der 8-Bit-Modus ist 3 Bytes 'sparsamer' als der ESCAPE-Modus. Die Programmierung mit 'ACCOL' verbraucht auch weniger Bytes.

Die Ausführung mit 'ACCOL' verbraucht zwar genausoviel Bytes wie der 8-Bit-Modus, dennoch ist die Programmierung einfacher und es können Null-Bytes gesendet werden.

PROGRAMM: "ACCOLL" Benötigte Register: 4 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-7



Reihe 3: Zeilen 8-9



GTO ..



01\*LBL "ACCOLL"

02 SF 17

03 .

04 ACCOL

05 " "

06 OUTA

07 ADV

08 ADV

09 END

XEQ "ACCOLL"



CLP "ACCOLL"



Zeile 05: 129 161 129 - Ohne 'LBL' und 'END' 13 Bytes

Diese vorstehenden Barcodes sind mit dem ThinkJet-Drucker HP 2225B hergestellt. In allen Programmen der nachfolgenden Kapitel wird die 'ACCOL'-Funktion und der 8-Bit-Modus für den Barcodeausdruck mit dem HP-IL-Drucker 82162A benutzt.

### 2.1 Das Druckerpapier, Ausnutzung, Archivierung

Wegen des erforderlichen Kontrastes muß mit dem Thermodrucker weißes, schwarzdruckendes Papier verwendet werden. Beste Ergebnisse werden mit dem HP-Papier 82175A erzielt. Blaudruckendes Papier kann nicht verwendet werden. Schwarzdruckendes Papier mit vergrauter Oberfläche, für sonstige Drucke völlig in Ordnung, wird mit Barcodes bedruckt meist "abgelehnt". Es mag sein, gute Fotokopien bewirken eine Verbesserung. Leider sind alle Thermopapiere nicht lichtbeständig. Gute Fotokopien und hier ohne weiteres auch Verkleinerungen sind für eine Archivierung deswegen immer anzuraten.

Weil das Papier mit Barcodes nur am rechten Rand bedruckt wird, beschriften viele Programme in den folgenden Kapiteln das Papier rechtsbündig. Es kann so für einen zweiten Durchlauf umgedreht und noch einmal verwendet werden. Später wird der Streifen dann mittig auseinandergeschnitten.

Eine 16-Bytes Barcodereihe vom Thermodrucker ist mit Beschriftung fast 35 cm lang. Für das Einlesen hat es sich bestens bewährt den Streifen unter dem Lese Stift hindurchzuziehen und nicht mit dem Lesestift darüber hinwegzugehen wie sonst üblich. Leider ist ein solcher Streifen länger als ein DIN A 4 Papier Format. Für eine dauerhafte Archivierung ist hier die Verkleinerung mit einem Bürokopierer die einzige Lösung. Wegen der Querbeschriftung werden die meisten Barcodestreifen auf DIN A 4 sicherlich von oben nach unten aufgeklebt.

### 2.2 Der Akkuzustand

Die Herstellung von Barcodes auf dem Thermodrucker und auch auf dem ThinkJet ist energieaufwendiger denn sonstiges Drucken. Besonders während längerem Arbeiten oder bei der Herstellung von Programmbarcodes könnte es zum Erlahmen der

Akkuspannung kommen. Es ist deswegen ratsam, den Drucker stets mit angeschlossenem Netzgerät zu betreiben und vorsorglich einen geladenen Akku in Reserve zu halten.

## 2.3 Druckerumbauten

 **PACK** Es besteht die 'Experten'-Möglichkeit den Thermodrucker umzubauen, so daß Barcodes mit weniger als der Hälfte an Platzbedarf der originalen HP-Spaltenmodus BC entstehen. Es ist dies sicherlich ein berechtigter Wunsch angesichts 35 cm Länge bei 16 Bytes. Diese Möglichkeit ist jedoch dem Anwender normalerweise nicht zugänglich, und HP dokumentiert dies nicht. Es wird deswegen hier darauf verzichtet. Wer näheres darüber zu erfahren wünscht, kann sich an den CCD wenden. Nebenstehend ein kleines Beispiel.

## 3. Rechner HP 41/71/75 und HP-IL Plotter 7470A

Für die Steuerung des Plotters hat HP eine spezielle 'Graphics Language' entwickelt, kurz HP-GL genannt, deren Syntax aus nur 2 Bytes pro Befehl besteht. Genau wie für den Drucker könnte deswegen auch für die Arbeit mit dem Plotter jeder Rechner eingesetzt werden, der entsprechende Befehlsfolgen senden kann. Außer der Aufbereitung der Bytefolgen in Steuerbefehle zum Plotten der Barcodes kommt hierbei noch die Steuerung der Stiftpositionen, die Auswahl der geeigneten Beschriftungstypen, Austausch von Stiften unterschiedlicher Stärke und gegebenenfalls Wechsel der Tuschefarbe hinzu. Obwohl es grundsätzlich möglich ist entsprechende Barcode-Programme nur für Rechner und Plotter zu schreiben, ist dies für den HP 41 angesichts der problemlosen Programmierung mit dem Plottermodul nicht interessant. Die ähnlich wie beim ThinkJet-Drucker zu bewältigenden großen Datenmengen, der häufige Wechsel von Stiftpositionen für das Plotten einer Barcodereihe und die für solche Aufgaben wesentlich zu geringe Arbeitsgeschwindigkeit des HP 41, führen zu unerträglichen Programmlaufzeiten. Einen schnelleren Rechner einzusetzen, ist wegen des geringen Preises eines Plottermoduls keine Alternative für HP 41 Barcodes. Das Plottermodul ist zudem außer für HP 41 BC, für die sonstige Plottersteuerung, für die Herstellung von anderen Barcodes und für die Arbeit mit dem Thermodrucker einzusetzen. Im Gegensatz zu den PRGM für Thermo- und ThinkJet-Drucker gibt es in diesem Buch kein Plotterprogramm ohne das Plottermodul, weil dies nicht mehr wirtschaftlich ist. Im PM sind ganze Befehlsfolgen der HP-GL zu einem Befehl für den HP 41 zusammengefaßt. Die Arbeitsgeschwindigkeit für die Barcodeherstellung überschreitet fast die Grenze der möglichen Zeichengeschwindigkeit für Faser- oder Tuschestifte.

### 3.1 Plotterpapier, Stifte und Tusche

HP bietet zum Plotter spezielles Plotterpapier und Faserstifte an. Für gelegentliches Arbeiten ist dies auch völlig ausreichend, man kann sogar sagen: optimal. Der bei längeren Plots jedoch recht schnell stiftverbreiternde Abrieb der Faserzeichner, besonders beim Barcodeplotten, ist nicht zufriedenstellend. Die Striche werden 'schleichend' ständig breiter, die Leerräume nehmen ab.

Ein tuschefestes Plotterpapier und spezielle Zeichenstifte für Tusche ergeben wesentlich präzisere und völlig gleichbleibende Resultate. Alle Plotter-Barcodes für dieses Buch sind mit dem Plotter 7470A, einem solchen speziellen Plotterpapier und Tuschestiften sowie Zeichentusche der Firma STAEDTLER hergestellt. Für die Beschriftung wurden Spitzen 'marsplot' in Normalausführung Typ 750 PL CF für Zeichengeschwindigkeiten bis 30cm/s mit Stärken 0,3 - 0,5 mm und für das Barcodeplotten Hartmetallausführung Typ 757 PL CF für Zeichengeschwindigkeiten bis 50 cm/s Stärke 0,3 mm verwendet. Beide Spitzentypen haben eine gelbe Sicherungsschraube. Als Tusche hat sich nur die STAEDTLER 'marsplot' Typ 748 PLH schwarz bewährt. Bei anderen schnelltrocknenden Tuschen führen die vielen dicht nebeneinander, bzw. teils übereinander liegenden Striche der Barcodes nach gewisser Zeit zur Verstopfung des Tuschefflusses. Andere Spitzentypen

zeitigen ebenfalls nicht den gewünschten Erfolg. Es mag sein, andere Hersteller führen in ihrem Programm ebenfalls geeignete Geräte, die hier nicht beschrieben werden können, weil das Testen und Herausfinden der für Barcodes brauchbaren Papier-, Tusche-, Spitzenkombination aus dem Angebot eines Herstellers bereits sehr zeitaufwendig war.

#### 4. Wann Plotter - wann Drucker?

Der programmtechnische Aufwand und das Gebot einer rationellen Papiernutzung rechtfertigen den Plottereinsatz eigentlich nur für die Barcodeherstellung längerer Texte, Programme oder für Tabellen. In allen anderen Fällen wie Herstellung von Einbyte-, Zweibyte- oder Mehrbytebefehlen, kurze auch synthetische Texte oder andere Bytefolgen, Zahlenbarcodes usw., sind Drucker besser geeignet. In diesem Buch wird so verfahren: Kurzbarcodes sind auf dem Thermo- oder ThinkJet-Drucker und Programmbarcodes oder Tabellen mit dem Plotter gefertigt.

#### 5. Die Programme zur Barcodeherstellung für HP 41

Alle Programme dienen der Herstellung mit dem Thermodrucker, dem ThinkJet-Drucker oder dem Plotter. ThinkJet-Drucker und Plotterprogramme gibt es nur für Texte, Programm-BC und Tabellen. Die übrigen Programme für den Thermodrucker betreffen jeweils nur einen Barcodetyp. Im Interesse der Kürze und geringen Speicherplatzbelegung sowie möglichst schnellem Einlesen sind Themen nicht zusammengefaßt, auch dann nicht wenn gleiche Routinen immer wieder vorkommen und als Unterprogramm verwendet werden könnten, weil dies immer eine Verlängerung des Gesamtprogramms bedeutet, indem die Hauptprogrammrountinen wesentlich mehr Speicherplatz belegen. Dafür sind alle Programme so komfortabel wie möglich, in Dialogeingabe und mit Bedienungs- oder Eingabe-fehlermeldung gestaltet. Soweit möglich, werden für einen 2. Papierstreifendurchlauf alle Ausdrücke rechtsbündig ausgeführt.

Viele Programme für den Thermodrucker sind 3-fach vorhanden. Die Programm-Namen für den Druck gleicher Barcodes auch auf verschiedenen Geräten folgen dabei einer bestimmten Ordnung: Programmnamen für Programme

- nur für Rechner und Thermodrucker ohne Module enden mit "...BC"
- für Rechner mit XF-Modul und Thermodrucker enden auf "...Bc"
- für Rechner, XF- /Plotter-Modul und Drucker enden auf "...bc"
- für Rechner, XF- /Plotter-Modul und Plotter enden auf "...Pbc"
- für Rechner, XF-Modul und ThinkJet-Drucker enden ohne Plotter-Modul auf "..TJbc" und mit Plotter-Modul auf "..TJbc"

Die ersten Buchstaben gleicher Programmtypen sind identisch, z.B. "SYNTBC" und "SYNTbc" oder "SYNTbc". Einige Programmnamen folgen dieser Systematik nicht. Dies sind dann keine Programme zur Barcodeherstellung, sondern meist für Umrechnungen wie "XR-XEQ". Ein spezielles Programm heißt "Wunsch". Es gibt Ihnen ein kleines Rätsel auf.

Die auf "...BC" endenden Programme bieten auch dem Anwender recht umfangreiche Arbeitsmöglichkeiten, der nicht über Zusatzmodule verfügt. Die Bedienung aller Programme wird so ausführlich wie nötig beschrieben. Selbstverständlichkeiten wie: Drucker = primäre Einheit oder Programm mit Lesestift einlesen usw. sind weggelassen. Wenn nicht ausdrücklich anders erwähnt, steht der Drucker-Modus-Schalter immer auf "MAN". Auf die detaillierte Beschreibung der Programmschritte wird verzichtet. Für ein tiefergehendes Ergründen der Programme wird Einzelschrittausführung empfohlen. Die Kenntnis des Rechnersystems wird dabei vorausgesetzt. Sofern einzelne Schritte bei synthetischen Befehlen zu 'MEMORY LOST' führen würden, wird ausdrücklich gewarnt.

## V. Rechner und notwendige oder wünschenswerte Geräte

Viele Programme ohne Module erfordern 'SIZE 18'. Wer mit einem Rechner HP 41 C ohne Speicher-Erweiterungsmodule BC herstellen möchte, hätte für solche Programme nur noch 41 Register zur Verfügung. Dies ist für die Nutzung aller Möglichkeiten nicht ausreichend. Es sollte dann mindestens 1 Speicher-Erweiterungs-Modul, besser jedoch das Quard-Modul oder ein HP 41 CV vorhanden sein. Mit dieser Ausstattung können dann bis auf PRGM-BC alle anderen Typen hergestellt werden. Optimal ist natürlich der HP 41 CX mit integriertem XF-Modul.

### a. Barcode-Leser HP 82153A - notwendig für das Einlesen

Nur für das Einlesen vorhandener Barcodes ist der Lesestift das einzig nötige Zusatzgerät zum Rechner, auch für den HP 41 C ohne Module. Die in diesem Buch vorhandenen Tabellen gewährleisten damit bereits eine wesentliche Vereinfachung vieler Eingaben, nicht tastbarer Zeichen und Bytefolgen sowie für synthetische Programmierung. Auch wer keinen Drucker besitzt, kann so bytesparend programmieren und seine eigenen Programme kürzer und schneller machen. Der Lesestift ist zudem eines der preiswertesten Zusatzgeräte.

### b. HP-IL-Modul 82160A und HP-IL-Drucker 82162A - notwendig zur Herstellung

Für die Nutzung vieler Programme zur Barcode-Herstellung sind ein IL-Modul und ein Thermodrucker die mindestnotwendigen Geräte. Hierbei sollte der Rechner ein HP 41 C mit Quard-Modul oder ein HP 41 CV sein. Bis auf Programm-BC können mit dieser Ausrüstung bereits alle übrigen Typen erzeugt werden.

### c. XF - Erweitertes Funktions/Speicher-Modul HP 82180A - nötig für PRGM-BC

Sonst ist dies Modul wünschenswert. Zur Herstellung von Programm-Barcodes auf dem Thermodrucker ist das XF-Modul allerdings notwendig; warum, wird in Kapitel 5 erläutert. Aber auch für die Nutzung kürzerer Programmlaufzeiten und geringerer Speicherplatzbelegung ist das XF-Modul zu empfehlen. Dies Modul beinhaltet eine Reihe Befehle, deren Umsetzung sonst nur mit synthetischer Programmierung realisiert werden kann. Hierbei muß dann allerdings ein erheblich höherer Byteverbrauch in Kauf genommen werden. Beispielsweise bringt die XF-Funktion 'ATOX' den Dezimalwert eines ASCII-Zeichens vom Alpha- ins X-Register und löscht zugleich das Alphazeichen. Ohne XF-Modul, synthetisch programmiert, sind für den gleichen Vorgang 51 Bytes erforderlich. Außerdem bietet dies Modul einen externen Massenspeicher von 124 Registern, in den Programme, Daten oder Texte abgelegt werden können. Dieser Massenspeicher kann noch um 2 X-Memory-Module, kurz XM, für umfangreichere Arbeiten auf gesamt 600 Register erweitert werden. Der HP 41 CX beinhaltet alle Funktionen und Erweiterungen des XF-Moduls und - dies ist für BC zwar nicht wichtig, auch des TIME-Moduls. XM-Module sind auch für den HP 41 CX erforderlich.

### d. Plotter-Modul HP 82184A - wünschenswert/notwendig

Dies Modul ist zur Barcodesherstellung auch dann sehr wünschenswert, wenn kein Plotter vorhanden ist. Die enorme Leistungsfähigkeit dieses Moduls wird deutlich anhand der Programme "ZFDABC" = 125 Register und "ZFDabc" = 18 Register für Zahlen- und Folgedaten aus Kapitel 3. Eine Reihe sehr nützlicher Funktionen für die völlig problemlose BC-Herstellung sind in diesem Modul zusammengefaßt. Für die Arbeit mit dem Plotter HP 7470A ist dies Modul notwendig.

### e. HP-IL-Plotter 7470A - wünschenswert

Für die Herstellung von Programm-Barcodes oder die Anfertigung der Tabellen, wie sie in diesem Buch vom Original auf 80% verkleinert gedruckt sind, ist der Plotter sicherlich nicht nur wünschenswert, sondern notwendig, weil solche Dokumentation weder mit dem Thermodrucker noch mit dem ThinkJet-Drucker in er-

träglichen Arbeitszeiten hergestellt werden können. Zur Erstellung von anderen nicht HP 41 Barcodes ist das Plottermodul allerdings notwendig.

f. Port-Extender - evtl. notwendig

Finden nicht alle Einsteck-Module und Geräte in den 4 Ports des Rechners Platz, so ist der Einsatz eines Port-Extenders die einzige Lösung. Dies Zusatzgerät belegt selbst einen Port des Rechners und bietet weitere 7 Einsteck-Ports, die den Nummern der Rechnerports entsprechen. Damit stehen 10 Ports bereit. Bestens bewährt hat sich der PE 4100 mit eigener Stromversorgung bei Entfernung vom Rechner von CT CORVALLIS TEAM GMBH, Max-Planck-Straße 22, 6382 Friedrichsdorf .

## Kapitel 2

### ALPHA-EINZELZEICHEN, ALPHA- UND SYNTHETISCHE TEXTZEILEN DRUCKER-SONDERZEICHEN

#### I. Alpha-Einzelzeichen

Ein wichtiges Anwendungsgebiet für Barcodes beim Programmieren mit Text ist die Herstellung von Textzeilen. Gerade der Dialog zwischen Benutzer und Rechner, Abfrage und Eingabe sowie Fehlermeldungen bei Fehlbedienung erleichtern die Arbeit sehr und gestalten sie recht sicher. Überschriften und Ergebniskommentare oder Anmerkungen innerhalb der Programmlistings erhöhen die Verständlichkeit. Mit den tastbaren Funktionen ist jedoch oft eine wesentlich größere Byteanzahl nötig, sofern man nicht tastbare Zeichen oder Kleinbuchstaben in Programmen verwenden möchte. Außer mit den manchmal umständlichen Methoden der synthetischen Programmierung ist dies außer mit dem neuen CCD-Modul nur mit Barcodes und hier sogar sehr einfach möglich. Ein Beispiel: die Zeile "Text zu lang" in den Programmen für Textzeilenherstellung "TEXTBC" usw., würde folgende Schritte erfordern:

<u>Ohne Module</u>		
01 "T"		
02 ACA		
03 SF 13		
04 "EXT ZU LANG"		
05 0		
06 33		
07 BLDSPEC		
08 ARCL X		
09 ACA		
10 PRBUF		
11 CF 13		
12 .END.		
Text zu lang! 32 Bytes		

<u>Mit XF-Modul</u>		
01 "T"		
02 ACA		
03 SF 13		
04 "EXT ZU LANG"		
05 33		
06 XTOA		
07 ACA		
08 PRBUF		
09 CF 13		
10 .END.		
Text zu lang! 28 Bytes		

<u>Mit Barcodeeingabe</u>		
01 "Text zu lang!"		
02 PRA		
03 .END.		
Text zu lang! 16 Bytes		

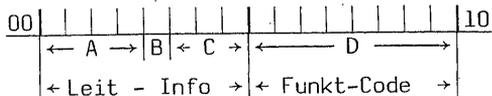
Die Eingabe erfolgt mit Hilfe der Alpha-Einzelzeichen aus der Tabelle "Zweibyte Funktions- und Druckerzeichen 0-127". An Alpha-Einzelzeichen sind zwei Arten zu unterscheiden: Zweibyte- und Dreibyte-Zeichen. Da einige der Zweibyte-Zeichen im RUN- und ALPHA-Modus unterschiedlich arbeiten, sind diese mit "Zweibyte Funktions- und Druckerzeichen 0-127" bezeichnet. Die Dreibyte-Zeichen nennen wir "Alpha-Einzelzeichen 0-255 :REPLACE |— APPEND". Über beide Typen gibt es nachstehend Tabellen.

#### a. Zweibyte Funktions- und Druckerzeichen 0-127

Diese Barcodes sind, wie Sie sicherlich am Beispiel "Text zu lang!" erkannt haben, außerordentlich leistungsfähig.

##### 1. Aufbau

Der Aufbau wird jeweils am deutlichsten mit einer kleinen Skizze und einer Bezifferung der Prüfsumme, des Indikators, der sonstigen Daten und des danach folgenden Inhalts der Reihe dargestellt. Prinzipiell ist der Aufbau immer ähnlich. Dennoch, es sind die einzelnen Bit eines Aufbaus, die eine einwandfreie Funktion gewährleisten. Toleranzen sind nicht möglich.



A = 4-Bit Prüfsumme

B = 1-Bit auf '0' gesetzt = Indikator für 2 Byte-Nicht-XROM-Funktion

C = 3-Bit auf '1' gesetzt = Indikator für diesen Zweibyte-Typ

D = Spezieller typbezogener Dezimalwert für den Funktions-Code

### 1.1 Der Typindikator

Der Typindikator ist im zweiten Nybble des ersten Byte verschlüsselt. B teilt dem Rechner mit, es handelt sich um einen Nicht-XROM-Zweibyte-BC (B = Null). C gibt Auskunft über den Zweibyte Funktions- und Drucker-Typ 0-127 (E = Eins).

### 1.2 Spezieller typbezogener Dezimalwert

Bei diesem von HP so kreiertem Barcode-Typ ist das 1. Nybble von D verdoppelt. Das Byte dezimal Null ist D = 16; das Byte dezimal 15 ist D = 31. Dabei sind die Druckerzeichen für D = 0-15 und D = 16-31 identisch. Die nachfolgenden Barcodes zeigen dies. Beide lesen das ASCII-Zeichen 15 =  $\text{␣}$  ins Alpha-Register. 'XEQ PRA' druckt das Zeichen bei gesetztem Flag 12 doppeltbreit.



Ab Byte 16 ist bei diesem Typ das 1. Nybble von D = 32, d.h. Nybble 1 muß verdoppelt werden. Für das ASCII-Zeichen 16 =  $\text{␣}$  - binär 0001000 ist D demgemäß = 32 - binär 00100000. Für 54 - binär 00110110 ist D = 102, indem aus Nybble 1 = 48 (0011)x2 = 96 wird + 6 (0110) aus Nybble 2 = dezimal 102.

Es läßt sich nun auch sofort ermitteln, welcher höchste Dezimalwert mit verdoppeltem 1. Nybble dargestellt werden kann. Es ist dies 127 - binär 01111111. Das erste Nybble ist 7 (0111)x16=112+15 (1111 - Nybble 2)=127. Für Darstellung als Funktionscode ist Nybble 1 von 127 zu verdoppeln = 14 - binär 11100000 = 224. Addiert man die 15 (1111) von Nybble 2 hinzu, erhält man 239 - 11101111. Eine weitere Verdoppelung von Nybble 1 ist nicht mehr möglich. Tatsächlich lassen sich problemlos auch bei den noch zu besprechenden Dreibyte-Einzelzeichen nur solche zwischen 16 und 239 in beliebige Register, außer in Programmzeilen, einlesen. Auch Textzeilen, die Bytes zwischen 0-15 oder ab 240 enthalten, werden nicht korrekt in Register eingelesen.

### 1.3 Die 4-Bit Prüfsumme

Die 4-Bit Prüfsumme wird im 1. Nybble A des ersten Byte dargestellt. Sie errechnet sich ganz einfach aus der Addition des Wertes (D+C):16 INT + (D+C) 16 MOD. Ein Beispiel für die Prüfsumme Buchstabe "Z" = ASCII-# 90 - 01011010. Zuerst Nybble 1 verdoppeln = 80x2=160. Sodann + Nybble 2 = 10 ergibt D = 170. D+1 (1=Indikator C, B ist ja = 0) = 171:16 (weil 4-Bit Prüfsumme) = 10 Rest 11 = 21 geteilt durch 16 ergibt 1 Rest 5; Prüfsumme also 6. Weil diese im 1. Nybble A vom Leit-Info-Byte dargestellt wird, ist 6x16 zu nehmen = 96+1 (Indikator B+C) ergibt 97 als Wert für Byte 1, die Leitinformation. In diesem Beispiel lauten also die Bytes für das ASCII-Zeichen "Z" als Barcodes 97,170 - 0110000110101010 A = 0110 (6) - B = 0 - C = 001 - D = 10101010. Dies ganze scheint komplizierter als es tatsächlich ist. Man muß sich nur ein wenig mit der Systematik des 8-Bit Systems vertraut gemacht haben, dann ist alles transparent.

2. Herstellen von Zweibyte Funktions- und Druckerzeichen 0-127 auf dem Drucker

Für die Herstellung dieser Barcodes auf dem Thermodrucker gibt es drei Programme. Die Bedienung ist gleich.

2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "A127BC"

Programmbedienung:

- SIZE: es werden keine Datenregister benötigt

XEQ "A127BC"



- "ASCII-Code?" - einen Code zwischen 0-127 eingeben; z.B. 65 für "A".
- Falls ASCII > 127 - Fehlermeldung: "ASCII <= 127".

01*LBL "A127BC"	24 RDN	47 ST/ Z	70 ADV	93 1
02 CF 22	25 X=0?	48 MOD	71 RTN	94 +
03 SF 28	26 "++"	49 +	72 GTO 00	95 .
04 CF 29	27 ACA	50 INT	73*LBL 01	96 1
05 FIX 0	28 ADV	51 16	74 SF 25	97 BLDSPEC
06*LBL 00	29 CF 12	52 *	75 X=0?	98 X<>Y
07 127	30 ADV	53 E	76 ACCOL	99 BLDSPEC
08 "ASCII-CODE?"	31 RCL X	54 +	77 X=0?	100 RTN
09 TONE 7	32 16	55 SF 17	78 GTO 03	101*LBL 05
10 PROMPT	33 ST/ Z	56 .	79*LBL 02	102 .
11 FC?C 22	34 MOD	57 ACCOL	80 XEQ 04	103 X<>Y
12 GTO 00	35 STO Z	58 RDN	81 ARCL X	104 BLDSPEC
13 ABS	36 X<>Y	59 --	82 RDN	105 RTN
14 X<>Y?	37 INT	60 XEQ 03	83*LBL 03	106*LBL 06
15 GTO 06	38 ST+ X	61 XEQ 02	84 OUTA	107 ACX
16 RCL X	39 ST+ Y	62 RDN	85 CLA	108 ADV
17 SF 12	40 16	63 XEQ 01	86 RTN	109 "ASCII <= 127"
18 ACX	41 *	64 ADV	87*LBL 04	110 ACA
19 .	42 ST+ Z	65 ADV	88 127	111 ADV
20 X<>Y	43 SIGN	66 CF 17	89 X<>Y	112 TONE 0
21 BLDSPEC	44 +	67 CLST	90 X<=Y?	113 GTO 00
22 - -	45 RCL X	68 BEEP	91 GTO 05	114 END
23 ARCL X	46 16	69 ADV	92 -	209 Bytes

Die Routine Zeilen 30-54 zur Verdoppelung Nybble 1 Byte D und Prüfsummenberechnung stammt aus einem Programm von Andreas Marktscheffel aus der CCD-Zeitschrift 'PRISMA' 3/4 1984 Seite 21. Selbstverständlich gibt es andere Lösungen, doch diese zeigt besonders gut die Speicherarithmetik.

2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "A127Bc"

65 A

Programmbedienung:

- SIZE: es werden keine Datenregister benötigt

XEQ "A127Bc"



- "ASCII-Code?" - einen Code zwischen 0-127 eingeben.
- Falls ASCII > 127 - Hinweis: "ASCII <= 127".

Das Programmlisting finden Sie auf der folgenden Seite. Das ASCII-Zeichen "A" wird als Barcode wie nebenstehend mit Wiederholung der Codeziffer und des Druckerzeichens ausgeführt.



01*LBL "A127bc"	19 " " "	37 SIGN	55 XEQ 02	73*LBL 02
02 CF 22	20 XTOA	38 +	56 RDN	74 XTOA
03 SF 28	21 ACA	39 RCL X	57 XEQ 01	75*LBL 03
04 CF 29	22 ADV	40 16	58 ADV	76 OUTA
05 FIX 0	23 CF 12	41 ST/ Z	59 ADV	77 CLA
06*LBL 00	24 ADV	42 MOD	60 CF 17	78 RTN
07 127	25 RCL X	43 +	61 CLST	79*LBL 04
08 "ASCII-CODE?"	26 16	44 INT	62 BEEP	80 ACX
09 TONE 7	27 ST/ Z	45 16	63 ADV	81 ADV
10 PROMPT	28 MOD	46 *	64 ADV	82 "ASCII <= 127"
11 FC?C 22	29 STO Z	47 E	65 RTN	83 ACA
12 GTO 00	30 X<>Y	48 +	66 GTO 00	84 ADV
13 ABS	31 INT	49 SF 17	67*LBL 01	85 TONE 0
14 X>Y?	32 ST+ X	50 .	68 SF 25	86 GTO 00
15 GTO 04	33 ST+ Y	51 ACCOL	69 X=0?	87 END
16 SF 12	34 16	52 RDN	70 ACCOL	170 Bytes
17 CLA	35 *	53 "	71 X=0?	
18 ARCL X	36 ST+ Z	54 XEQ 03	72 GTO 03	

Das XF-Modul erspart 39 Bytes nur mit der Funktion 'XTOA'.

### 2.3 Rechner HP 41 C - Drucker XF- und Plottermodul PRGM "A127bc"

#### Programmbedienung:

- SIZE: man kommt ohne Speicherregister aus

XEQ "A127bc"



- "ASCII-Code?" - einen Code zwischen 0-127 eingeben.
- Falls ASCII > 127 - Fehlermeldung: "ASCII <= 127".

01*LBL "A127bc"	12 GTO 00	23 16	34 -2	45 ACX
02 CF 22	13 ABS	24 ST/ Z	35 BCCSM	46 ADV
03 SF 28	14 X>Y?	25 MOD	36 BCO	47 "ASCII <= 127"
04 CF 29	15 GTO 01	26 X<>Y	37 ADV	48 ACA
05 FIX 0	16 SF 12	27 INT	38 ADV	49 ADV
06*LBL 00	17 ACX	28 ST+ X	39 CF 17	50 TONE 0
07 127	18 " "	29 16	40 CLST	51 GTO 00
08 "ASCII-CODE?"	19 XTOA	30 *	41 BEEP	52 END
09 TONE 7	20 ACA	31 +	42 RTN	118 Bytes
10 PROMPT	21 CF 12	32 " " "	43 GTO 00	
11 FC?C 22	22 RCL X	33 XTOA	44*LBL 01	

Hier zeigt sich die enorme Leistungsfähigkeit des Plotter-Moduls. Die Berechnung der Prüfsumme und der Ausdruck der Barcodes benötigen nur die drei Programmschritte 34 bis 36.

### 3. Fertigen von "Zweibyte Funktions- und Druckerzeichen 0-127" auf dem Plotter

Wer einen Plotter hat, kann sich mit dem nachstehend beschriebenen Programm die DIN A 4 - Tabelle der "Zweibyte Funktions- und Druckerzeichen 0-127" erstellen, so wie sie verkleinert in diesem Buch wiedergegeben ist.

Für die exakte Darstellung wird die Benutzung von Tuschestiften, wie beschrieben empfohlen; aber auch Faserstifte werden hierfür gute Ergebnisse erbringen.

### 3.1 Rechner HP 41 CV - Plotter, XF- und Plottermodul PRGM "127Pbc"

Die Kenntnis der Befehle des PM und der HP-GL wird vorausgesetzt. Das Plotter-Modul beinhaltet eine hohe Zahl der möglichen HP-GL-Befehle. So kann die Gestaltung der Schrift mit den Funktionen 'CSIZE' oder 'CSIZED' erfolgen. Einfacher ist dies jedoch für unseren Zweck mit dem Befehl 'SI', weil hier die Schriftgröße absolut in cm und nicht als Grafikeinheit gewählt wird. Eine Verlangsamung der Zeichengeschwindigkeit ist wie sie in Zeilen 57 und 58 mit dem HP-GL Befehlssatz "VS19" und 'OUTA' programmiert ist, im PM nicht vorgesehen. Ebenso können Fehlerabfrage-Instruktionen (Zeilen 147,148) mit dem PM nicht ausgeführt werden. Andererseits sind bis auf den Befehl 'BCO' alle übrigen Barcodefunktionen (Zeilen 140-153) so bequem nur mit dem PM zu verwirklichen.

Nachstehend zunächst die Programmliste; die Programmbedienung folgt auf Seite 26. Die PRGM-Zeile 06 lautet dezimal: 247 31 240 0 0 67 144 8. Mit dieser und den folgenden 07 und 08 werden auf einmal alle Flags 00-43 gesteuert. Im Abschnitt 'Anfertigen eines Flagstatus' wird dies besprochen.

01*LBL "127Pbc"	42 PEN	83 .	124 SF 00	165 ST+ 02
02 "HP7470A"	43 167	84 PEN	125 FS? 00	166 XEQ 15
03 FINDID	44 SCALE	85 "PEN 2= .5 SWZ"	126 GTO 03	167 ISG 01
04 STO 04	45 78	86 TONE 7	127 RCL X	168 GTO 01
05 SELECT	46 "SCHRIFT? J/N"	87 PROMPT	128 16	169 XEQ 09
06 "X++Ca"	47 AON	88 2	129 ST/ Z	170 GTO a
07 RCL I	48 TONE 7	89 PEN	130 MOD	171*LBL 06
08 STOF LAG	49 STOP	90 7	131 X<>Y	172 CLA
09 SIZE?	50 AOFF	91 9	132 INT	173 PEN
10 5	51 ATOX	92 MOVE	133 ST+ X	174 RCL 00
11 X>Y?	52 X*Y?	93 SF 17	134 16	175 INT
12 PSIZE	53 SF 10	94 "ZWEIBYTE FUNKT"	135 *	176 ARCL X
13*LBL 00	54*LBL a	95 "IONS- UND"	136 +	177 "t "
14 2	55 RCL 04	96 LABEL	137 "-"	178 XTOA
15 "SEITE? 1-2"	56 LISTEN	97 " DRUCKERZEICHEN"	138 XTOA	179 RCL 03
16 TONE 7	57 "VS19"	98 "t 0-127"	139 2	180 RCL 02
17 PROMPT	58 OUTA	99 LABEL	140 BCCKSM	181 1.3
18 FC? 22	59 XEQ 15	100 CF 17	141*LBL 03	182 -
19 GTO 00	60 20.2	101 .	142 FS?C 00	183 MOVE
20 INT	61 STO 02	102 PEN	143 "0"	184 LABEL
21 ABS	62 .015	103 "PEN 2= .3 SWZ"	144 ASTO L	185 RTH
22 X>Y?	63 STO 01	104 TONE 7	145 RCL 04	186*LBL 00
23 GTO 00	64 "PAPIER?"	105 PROMPT	146 LISTEN	187 "PLOT ENDE"
24 1	65 TONE 7	106 RCL 04	147 "0E"	188 AVIEW
25 -	66 PROMPT	107 LISTEN	148 OUTA	189*LBL 09
26 00	67 "PEN 2= .4 SWZ"	108 "SI.225..25"	149 INA	190 BEEP
27 *	68 TONE 7	109 OUTA	150 CLA	191 .
28 .127	69 PROMPT	110 LISTEN	151 ARCL L	192 .
29 +	70 2	111 "VS"	152 SF 25	193 PEN
30 STO 00	71 PEN	112 OUTA	153 BC	194 MOVE
31 7.13532	72 FRAME	113*LBL 01	154 FS? 10	195 RTH
32 STO 03	73 .	114 RCL 03	155 XEQ 06	196*LBL 15
33 PINIT	74 13	115 INT	156 ISG 00	197 .
34 11	75 MOVE	116 RCL 02	157 GTO 02	198 PEN
35 263	76 167	117 MOVE	158 GTO 08	199 PDIR
36 15	77 13	118 1	159*LBL 02	200 LORG
37 182	78 DRAW	119 PEN	160 ISG 03	201 ACOS
38 LIMIT	79 RCL 04	120 CLA	161 GTO 01	202 LDIR
39 .	80 LISTEN	121 RCL 00	162 7.13532	203 END
40 252	81 "SI.225..35"	122 INT	163 STO 03	530 Bytes
41 .	82 OUTA	123 X=0?	164 14.5	

Programmbedienung:

- 'PEN' 1: Falls Tusche - 0,3 mm schwarz, sonst Faserstift P .3 einsetzen
- SIZE: wenn der vorhandene 'SIZE' < 5 ist, stellt das PRGM 'SIZE' 005 ein

XEQ "127Pbc"



- "Seite? 1-2": Sofern die Arbeit unterbrochen werden mußte können Sie wählen, ob Seite 1 oder Seite 2 geplottet werden soll. Nach Eingabe 1 oder 2 'R/S' drücken.
  - "Schrift? J/N": Durch Eingabe von "J" = ja können Sie bestimmen, ob die Barcodes mit Dezimalziffern 0-127 und den dazugehörigen ASCII-Zeichen (dies erst ab ASCII-# 33) vom Plotter beschriftet werden sollen oder "N" = nicht. Die Tabellen in diesem Buch sind mit Zeichen des Druckers beschriftet. Dann 'R/S'
  - "Papier?": Kontrolle ob Papier eingelegt ist. - 'R/S'
  - "PEN 2= .4 SWZ": Bei der Arbeit mit Tuschestiften Aufforderung einen Stift mit 0,4 mm Strichstärke und schwarzer Tusche in Stifthalter 2 einzusetzen.
  - "PEN 2= .5 SWZ": Stiftwechsel auf Stärke 0,5 mm.
  - "PEN 2= .3 SWZ": Bei Plotterbeschriftung nötiger Stiftwechsel. Falls mit Faserstiften P .3 gearbeitet wird kann hier gleich 'R/S' gedrückt werden.
- Nach Fertigstellung Seite 1 ertönt 'BEEP' und die Frage "Papier?". Nach Papierwechsel dann 'R/S' drücken, und die Seite 2 wird gefertigt. Ist auch diese geplottet, steht im Display "PLOT ENDE", und Sie besitzen eine DIN A 4 Tabelle dieser nützlichen Barcodes.

4. Arbeiten mit 'Zweibyte Funktions- und Druckerzeichen 0-127'

Die Barcodes 0 und 127 sind keine ASCII-Zeichen, sondern Funktions-Barcodes. Befindet sich der Rechner im RUN-Modus, schaltet 0 in den ALPHA-Modus um und löscht das Register. Sonst wirkt 0 wie die Korrekturtaste 'CLX/A'. '127' bewirkt die Funktion 'APPEND'. Im ALPHA-Modus überschreibt das erste eingelesene Textzeichen (1-126) das Register, sofern nicht vorher 127 eingegeben wird. Im PRGM-Modus eröffnen die Zeichen 1-127 eine neue Textzeile; Nachfolgezeichen werden solange angehängt, bis 15 Zeichen 'voll' sind oder vorher durch Drücken der Taste 'ALPHA' abgeschlossen wird. Im Gegensatz zum ALPHA-Register kann eine abgeschlossene Programmtext-Zeile mit 127 nicht wieder eröffnet werden.

Im RUN-Modus schalten die Zeichen 37 = % - 42 = \* - 43 = + - 45 = - und 47 = / nicht in den ALPHA-Modus um, sondern führen die entsprechende mathematische Funktion aus. Alle übrigen BC schalten in den ALPHA-Modus und überschreiben (außer 127) das Register.

Im übrigen können in Funktions-Namen nach 'XEQ ALPHA' auch mit diesem Barcode-Typ nur die tastbaren Zeichen eingelesen werden.

Die Seiten 27 und 28 sind die Tabellen dieser Zweibyte Funktions- und Druckerzeichen 0-127 welche mit dem Plotter 7470A angefertigt wurden.

# ZWEIBYTE FUNKTIONS- UND DRUCKERZEICHEN 0-127

0 \*



1 x



2 z



3 e



4 a



5 b



6 r



7 d



8 A



9 o



10



11 >



12 u



13 L



14 r



15 f



16 0



17 Q



18 s



19 A



20 a



21 A



22 a



23 0



24 o



25 0



26 u



27 E



28 e



29 \*



30 f



31 z



32



33 !



34 "



35 #



36 f



37 %



38 &



39 .



40 (



41 )



42 \*



43 +



44 ,



45 -



46 .



47 /



48 0



49 1



50 2



51 3



52 4



53 5



54 6



55 7



56 8



57 9



58 :



ZWEIBYTE FUNKTIONS- UND DRUCKERZEICHEN 0-127

80 P



81 Q



82 R



83 S



84 T



85 U



86 V



87 W



88 X



89 Y



90 Z



91 [



92 \



93 ]



94 ^



95 \_



96 `



97 a



98 b



99 c



100 d



101 e



102 f



103 g



104 h



105 i



106 j



107 k



108 l



109 m



110 n



111 o



112 p



113 q



114 r



115 s



116 t



117 u



118 v



119 w



120 x



121 y



122 z



123 {



124 }



125 +



126 =



127 ~

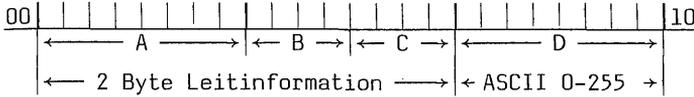


b. Dreibyte Alpha-Einzelzeichen 0-255 :replace und ⌈append

Im Gegensatz zu den Zweibyte Funktions- und Druckerzeichen handelt es sich bei diesen Einzelzeichen um Textzeilen mit nur einem ASCII-Zeichen.

1. Aufbau

Alle Textbarcodes sind gleich aufgebaut:



A = 1 Byte 8-Bit Prüfsumme

B = 1. Nybble 4-Bit Typindikator

C = 2. Nybble 4-Bit Anzahl Zeichen der Zeile

D = Ein ASCII-Zeichen zwischen 0 und 255

1.1 Der Typindikator

Es wird dem Rechner mitgeteilt, daß es sich um eine Textzeile handelt, ob der Text das Alpharegister überschreiben soll = :replace oder ob der Text an den Inhalt des Alpharegisters anzuhängen ist = ⌈append und wieviele Zeichen die Zeile enthält. Die Alternativen sind:

B = 7 - Textzeile Alpharegister überschreiben

B = 8 - Textzeile an Alpha-Inhalt anhängen

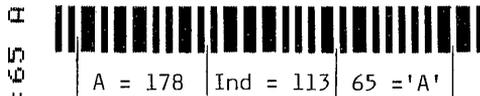
C = 1-14 - Anzahl Zeichen der Zeile

D = 1-14 - ASCII-Zeichen zwischen 0 bis 255

Dieser Typindikator ist für alle Textbarcodes gleich. Weil die Dreibyte Einzelzeichen stets nur ein Byte D haben, ist der Typindikator in diesem Fall für die Alternativen immer gleich: für replace - 01110001 = dezimal 113, für append - 10000001 = dezimal 129. Dieser Indikator-Bytewert errechnet sich aus Nybble 1 = 7x16=112+1 (Nybble 2) gesamt 113 oder 8x16=128+1 ergibt 129.

1.2 Die 8-Bit Prüfsumme

Diese Prüfsumme wird durch Addition des Indikatorbyte plus D ermittelt. Der höchste Additionswert kann hier nur 129+255=384 sein. Es ist deswegen nicht nötig, die Prüfsummenroutine 2x zu durchlaufen. Der Rechengang ist einfach: 384:256=1 Rest 128 - Prüfsumme 129. Für D = 0 und Indikator 113 ist die Prüfwert 128. Die BC für den Replacebuchstaben "A" als Beispiel lauten:



2. Herstellen von Dreibyte Alpha-Einzelzeichen 0-255 :replace - ⌈append

Für die Einzelherstellung dieser Zeichen gibt es drei Programme mit gleicher Bedienung.

2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "A255BC"

Programmbedienung:

- SIZE: es werden keine Datenregister benötigt

Zur Erhöhung der Startbequemlichkeit folgen hierauf bei allen PRGM XEQbarcodes.

XEQ "A255BC"



- "ASCII 0-255?": Dezimalwert des entsprechenden Zeichens eingeben - 'R/S'
- "APPEND? J/N": Für append = "J" - für replace = "N" eingeben - 'R/S'
- Falls die ASCII-Eingabe > 255 ist, erfolgt Fehlerhinweis: "ASCII <= 255".

01*LBL "A255BC"	26 STOP	51 129	76 ADV	101 X=Y?
02 CF 13	27 AOFF	52 FC? 00	77 CF 17	102 GTO 06
03 CF 22	28 ASTO X	53 113	78 CLST	103 -
04 SF 28	29 X=Y?	54 +	79 BEEP	104 1
05 CF 29	30 SF 00	55 RCL X	80 ADV	105 +
06 FIX 0	31 RCL Z	56 256	81 ADV	106 .
07*LBL 00	32 RCL X	57 ST/ Z	82 RTN	107 1
08 CF 00	33 SF 12	58 MOD	83 GTO 00	108 BLDSPEC
09 255	34 FS? 00	59 +	84*LBL 01	109 X<Y
10 "ASCII 0-255?"	35 "+"	60 INT	85 SF 25	110 BLDSPEC
11 TONE 7	36 FC? 00	61 SF 17	86 X=0?	111 RTN
12 PROMPT	37 ":	62 .	87 ACCOL	112*LBL 05
13 FC?C 22	38 ARCL X	63 ACCOL	88 X=0?	113 ACX
14 GTO 00	39 "+ "	64 "	89 GTO 03	114 ADV
15 ABS	40 X=0?	65 XEQ 03	90*LBL 02	115 "ASCII <= 255"
16 INT	41 "+"	66 RDN	91 XEQ 04	116 ACA
17 X<Y?	42 XEQ 04	67 XEQ 02	92 ARCL X	117 ADV
18 GTO 05	43 ARCL X	68 FS? 00	93 RDN	118 TONE 0
19 ENTER↑	44 RDN	69 "	94*LBL 03	119 GTO 00
20 "J"	45 RDN	70 FC? 00	95 OUTA	120*LBL 06
21 ASTO X	46 ACA	71 "a"	96 CLA	121 .
22 ENTER↑	47 ADV	72 XEQ 03	97 RTN	122 X<Y
23 "APPEND? J/N"	48 CF 12	73 RDN	98*LBL 04	123 BLDSPEC
24 AON	49 ADV	74 XEQ 01	99 127	124 END
25 TONE 7	50 FS? 00	75 ADV	100 X<Y	250 Bytes

Bedienung und Erläuterungen zu PRGM "A255Bc" finden Sie auf Seite 31.

01*LBL "A255Bc"	21 AON	41 ADV	61 "	81 GTO 03
02 CF 13	22 TONE 7	42 FS? 00	62 FC? 00	82*LBL 02
03 CF 22	23 STOP	43 129	63 "a"	83 XTOA
04 SF 28	24 AOFF	44 FC? 00	64 XEQ 03	84*LBL 03
05 CF 29	25 ATOX	45 113	65 RDN	85 OUTA
06 FIX 0	26 X=Y?	46 +	66 XEQ 01	86 CLA
07*LBL 00	27 SF 00	47 RCL X	67 ADV	87 RTN
08 CF 00	28 RCL Z	48 256	68 ADV	88*LBL 04
09 255	29 RCL X	49 ST/ Z	69 CF 17	89 ACX
10 "ASCII 0-255?"	30 SF 12	50 MOD	70 CLST	90 ADV
11 TONE 7	31 FS? 00	51 +	71 BEEP	91 "ASCII <= 255"
12 PROMPT	32 "+"	52 INT	72 ADV	92 ACA
13 FC?C 22	33 FC? 00	53 SF 17	73 ADV	93 ADV
14 GTO 00	34 ":	54 .	74 RTN	94 TONE 0
15 ABS	35 ARCL X	55 ACCOL	75 GTO 00	95 GTO 00
16 INT	36 "+ "	56 "	76*LBL 01	96 END
17 X<Y?	37 XTOA	57 XEQ 03	77 SF 25	208 Bytes
18 GTO 04	38 ACA	58 "	78 X=0?	
19 74	39 ADV	59 XEQ 02	79 ACCOL	
20 "APPEND? J/N"	40 CF 12	60 FS? 00	80 X=0?	

Zu PRGM "A255BC" die Dezimalwerte zu Zeile 35: 242 0 127; Zeile 39: 242 127 32; Zeile 41: 242 127 0; Zeile 64: 241 130 (Drucke 3 Barcodes); Zeile 69: 241 129; Zeile 71: 241 113.

:79 0 F75 K F63 ?

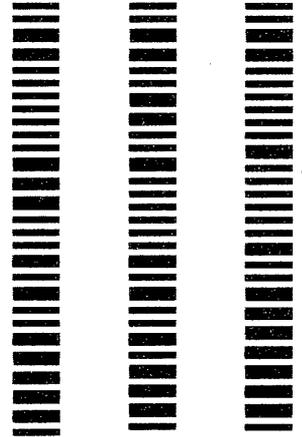
2.2 Rechner HP 41 C - Drucker und XF-Modul PRGM "A255Bc"

Programmbedienung: wie "A255BC"

XEQ "A255Bc"



Das Programmlisting finden Sie auf Seite 30 unten. Die Zeilen 'TONE 7' und 'TONE 0' sind in allen Programmen die synthetisch eingegebenen 'TONE 57' und 'TONE 10'. Mit Barcodes ist diese Eingabe sehr einfach; das Verfahren wird in Kapitel 4 besprochen.



2.3 Rechner HP 41 C - Drucker, XF- und PM PRGM "A255bc"

Programmbedienung: wie "A255BC"

XEQ "A255bc"



Rechts nebenstehend finden Sie ein Beispiel dieser Barcodes, welche in das Alpha-Register oder in eine Programmzeile eingelesen werden können.

01*LBL "A255bc"	14 GTO 00	27 SF 00	40 CF 12	53 BEEP
02 CF 13	15 ABS	28 RCL Z	41 FS? 00	54 RTN
03 CF 22	16 INT	29 RCL X	42 "	55 GTO 00
04 SF 28	17 X>Y?	30 SF 12	43 FC? 00	56*LBL 01
05 CF 29	18 GTO 01	31 FS? 00	44 "a"	57 ACX
06 FIX 0	19 74	32 "+"	45 XTOA	58 ADV
07*LBL 00	20 *APPEND? J/N*	33 FC? 00	46 -3	59 *ASCII (<= 255"
08 CF 00	21 AON	34 " "	47 BCCKSM	60 ACA
09 255	22 TONE 7	35 ARCL X	48 BCO	61 ADV
10 *ASCII 0-255?"	23 STOP	36 "+ "	49 ADV	62 TONE 0
11 TONE 7	24 AOFF	37 XTOA	50 ADV	63 GTO 00
12 PROMPT	25 ATOX	38 ACA	51 CF 17	64 END
13 FC?C 22	26 X=Y?	39 ADV	52 CLST	154 Bytes

3. Herstellen von Dreibyte Alpha-Einzelzeichen 0-255 auf dem Plotter 7470A

Analog zum Programm "127Pbc" im vorhergehenden Abschnitt kann mit dem nachstehenden Programm auf dem Plotter eine DIN A 4 Tabelle der Alpha-Einzelzeichen 0-255 genauso erstellt werden, wie sie anschließend abgedruckt ist.

3.1 Rechner HP 41 CV - Plotter, XF- und PM PRGM "255Pbc"

Der Aufbau dieses Plotterprogramms entspricht dem PRGM für die Zeichen 0-127. Der Ablauf ist nahezu identisch.

Programmbedienung:

- 'PEN' 1: Falls Tusche - 0,3 mm schwarz, sonst Faserstift P .3 einsetzen.
- SIZE: sofern der eingestellte 'SIZE' < 5 ist, stellt das PRGM 'SIZE' 005 ein.

XEQ "255Pbc"



- "Seite? 1-8": Die gesamte Tabelle hat 8 Seiten. Kann nicht ohne Unterbrechung gearbeitet werden, so wird das Plotten mit jeder Seite zwischen 1-8 neu begonnen wenn 1...8 - 'R/S' gedrückt wird.
  - "Schrift? J/N": Genau wie bei "0-127". Außerdem werden die Zeichen 128 - 156 nicht geplottet. Die Tabellen dieses Buches sind bis auf die Zeichen 0-31 und 128-159 mit dem Plotter beschriftet. Die anderen Zeichen sind mit dem Thermodrucker hergestellt. - 'R/S'
  - "PAPIER?": Kontrolle, ob Papier eingelegt ist. - 'R/S'
  - "PEN 2= ...SWZ" - wie "127Pbc" = 0.4; 0.5; 0.3 mm; sonst Faserstifte.
- Nachdem je eine Seite fertiggestellt ist, ertönt 'BEEP', Auswurf der Seite und die Frage "PAPIER?". Nach Papierwechsel dann 'R/S', nach Seite 8 "PLOT ENDE".

01*LBL "255Pbc"	44 SCALE	87 PROMPT	130 GTO 08	173 RTN
02 "HP7470A"	45 78	88 2	131*LBL 02	174*LBL 05
03 FINDID	46 "SCHRIFT? J/N"	89 PEN	132 ISG 03	175 2
04 STO 04	47 AON	90 7	133 GTO 01	176 PEN
05 SELECT	48 TONE 7	91 9	134 7.1274	177 RDN
06 "I++Ca"	49 STOP	92 MOVE	135 STO 03	178 RDN
07 RCL I	50 AOFF	93 SF 17	136 14.5	179 RCL 00
08 STOFAG	51 ATOX	94 "ALPHA-EINZELZEI"	137 ST+ 02	180 INT
09 SIZE?	52 X*Y?	95 "FCHEN 0--"	138 XEQ 15	181 ARCL X
10 5	53 SF 10	96 LABEL	139 ISG 01	182 "+ "
11 X*Y?	54*LBL a	97 "255 :REPLACE --"	140 GTO 01	183 128
12 PSIZE	55 RCL 04	98 "+ APPEND"	141 XEQ 09	184 X*Y?
13*LBL 00	56 LISTEN	99 LABEL	142 GTO a	185 GTO 06
14 8	57 "VS19"	100 CF 17	143*LBL 03	186 -
15 "SEITE? 1-8"	58 OUTA	101 .	144 RCL 03	187 GTO 07
16 TONE 7	59 XEQ 15	102 PEN	145 INT	188*LBL 06
17 PROMPT	60 20.2	103 "PEN 2= .3 SWZ"	146 RCL 02	189 RDN
18 FC? 22	61 STO 02	104 TONE 7	147 MOVE	190*LBL 07
19 GTO 00	62 .015	105 PROMPT	148 1	191 XTOA
20 INT	63 STO 01	106 RCL 04	149 PEN	192 RDN
21 ABS	64 "PAPIER?"	107 LISTEN	150 CLA	193 1.3
22 X*Y?	65 TONE 7	108 "SI.225,..25"	151 RTN	194 -
23 GTO 00	66 PROMPT	109 OUTA	152*LBL 04	195 MOVE
24 1	67 "PEN 2= .4 SWZ"	110 LISTEN	153 XTOA	196 LABEL
25 -	68 TONE 7	111 "VS"	154 RDN	197 RTN
26 32	69 PROMPT	112 OUTA	155 RCL 00	198*LBL 08
27 *	70 2	113*LBL 01	156 INT	199 "PLOT ENDE"
28 .255	71 PEN	114 XEQ 03	157 XTOA	200 AVIEW
29 +	72 FRAME	115 113	158 RDN	201*LBL 09
30 STO 00	73 .	116 XEQ 04	159 RDN	202 BEEP
31 7.1274	74 13	117 "-"	160 3	203 .
32 STO 03	75 MOVE	118 FS? 10	161 BCKSM	204 .
33 PINIT	76 167	119 XEQ 05	162 ASTO L	205 PEN
34 11	77 13	120 ISG 03	163 RCL 04	206 MOVE
35 263	78 DRAW	121 "*"	164 LISTEN	207 RTN
36 15	79 RCL 04	122 XEQ 03	165 RDN	208*LBL 15
37 182	80 LISTEN	123 129	166 "OE"	209 .
38 LIMIT	81 "SI.225,..35"	124 XEQ 04	167 OUTA	210 PEN
39 .	82 OUTA	125 "-"	168 INA	211 PDIR
40 252	83 .	126 FS? 10	169 CLA	212 LORG
41 .	84 PEN	127 XEQ 05	170 ARCL L	213 ACOS
42 PEN	85 "PEN 2= .5 SWZ"	128 ISG 00	171 SF 25	214 LDIR
43 167	86 TONE 7	129 GTO 02	172 BC	215 END 552 Bytes

Auch in diesem PRGM wird synthetisch, mit HP-GL und Befehlen des Plottermoduls gearbeitet. Die Zeile 121 ist ein "NOP". Zeile 125 dezimal = 240 255.

#### 4. Arbeiten mit Dreibyte Alpha-Einzelzeichen 0-255

Weil diese Barcodes Textzeilen mit einem ASCII-Zeichen sind, funktionieren sie wie Programmtextzeilen. Im RUN-Modus eingelesen, schaltet die Eingabe replace oder append zusätzlich auf ALPHA, überschreibt das Register oder hängt die Zeichen an den Inhalt an. Der Text ist danach abgeschlossen, der Rechner bleibt im ALPHA-Modus. Jede weitere Eingabe überschreibt das Register oder füllt es bis 24 Zeichen auf. Die Zeichen 0-15 und 240-255 können nicht direkt mit dem Lese Stift ins Alpha-Register eingegeben werden. Geschieht dies trotzdem, wird der Rechner harmlos blockiert. Lesestift herausnehmen, wieder einstecken, und die Sache ist behoben. Für die Barcodes 1-15 stehen ersatzweise die Zeichen aus der Tabelle 0-127, die übrigens alle beliebig mit denen 0-255 kombiniert werden können, zur Verfügung. Auf die direkte Eingabe der Barcodes 0-15 und 240-255 muß verzichtet werden. Bei Kombination von Zeichen aus beiden Tabellen ist zu beachten, daß 0-255 jede Eingabe abschließt, während das erste Zeichen "0-127" das Alpharegister überschreibt und Folgezeichen angehängt werden. Bei geschlossener Zeile vorher also 127-append einlesen. Ein :REPLACE-Zeichen aus 0-255 überschreibt in jedem Fall das Alpharegister, ein +APPEND-Zeichen schließt die Zeile immer ab.

Im PRGM-Modus gibt es keine Einschränkungen. Jede Einzelzeichen-Textzeile wird eingelesen und im laufenden Programm entsprechend replace oder append ausgeführt. Auch hier können die ASCII-Codes aus beiden Tabellen kombiniert werden.

Eine Sonderstellung nehmen die Zeichen 0 und 127 ein. Im Alpha-Modus ist 127 in bei Fällen :R oder +A das ASCII-Zeichen "␣". Als Programmzeile bewirkt :127 nichts und +127 hängt das ASCII-Zeichen 127 an den Alpha-Inhalt an. Soll nun im laufenden Programm das Zeichen 127 replace ins Alpharegister gebracht werden, so ist dies nicht möglich. Es muß 'CLA', +APPEND 127 programmiert werden. Das ASCII-Zeichen Null bleibt allein nicht im Alpharegister stehen, es 'verflüchtigt' sich. Replace wirkt '0' in einem laufenden Programm wie 'CLA'. Append wird '0' an den Inhalt des Alpharegisters angehängt; bei leerem Alpharegister geschieht nichts.

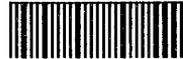
Für den Aufbau einer synthetischen Textzeile sind die Einzelzeichen 0-127 bzw. 0-255 in Programmen bytesparend gegenüber 'BLDCPES' und 'XTOA'. Ein Beispiel: der Aufbau der synthetischen Textzeile 06 im PRGM "255Pbc" erfordert folgende Bytes: 31 240 0 0 67 144 8.

Die Ausführung mit Barcodes erspart 1 Register. 'BLDSPEC' ist gegenüber 'XTOA' noch viel länger. In der Praxis ersparen die Barcodes dann noch mehr Bytes, wenn Daten im Stack erhalten bleiben sollen. Diese Tabellen sind insbesondere auch für Anwender von Interesse, die keinen Drucker besitzen. Die Herstellung geschlossener, synthetischer Textzeilen wird im Abschnitt II. besprochen.

Noch einige Besonderheiten der ASCII-Zeichen > 127: das Zeichen 138 wird aus dem Alpha-Register nicht gedruckt, 141 ebenso nicht, bewirkt aber einen Zeilenvorschub. Im PRGM-Modus werden alle Zeichen ab 128 in Listings nicht gedruckt. Die Abstände zwischen den Anführungszeichen werden ab Byte 161 bis 183 ständig größer. Bei bestimmten Voraussetzungen drucken die Bytes 184-199 die 'Chinesischen Zeichen' '⠄' '⠅' '⠆' '⠇' und die Bytes 198 199 können den Ausdruck stoppen. Byte 232 wird im MAN-Modus rechts gedruckt.

<u>Mit BC 0-255</u>	<u>Mit 'XTOA'</u>
20 Bytes	27 Bytes
01*LBL "Z 06"	01*LBL "Z 06"
02 "⠄"	02 CLA
03 "␣"	03 31
04 "+␣"	04 XTOA
05 "+␣"	05 240
06 "+C"	06 XTOA
07 "␣"	07 0
08 "+␣"	08 XTOA
09 END	09 XTOA
	10 67
⠄⠅⠆⠇	11 XTOA
	12 144
	13 XTOA
	14 8
	15 XTOA
	16 END
	⠄⠅⠆⠇

ALPHA-EINZELZEICHEN 0-255 :REPLACE - FAPPEND

: 0 *	F0 *	: 1 *	F1 *
			
: 2 x	F2 x	: 3 <	F3 <
			
: 4 a	F4 a	: 5 ß	F5 ß
			
: 6 r	F6 r	: 7 d	F7 d
			
: 8 A	F8 A	: 9 a	F9 a
			
: 10	F10	: 11 >	F11 >
			
: 12 u	F12 u	: 13 <	F13 <
			
: 14 r	F14 r	: 15 #	F15 #
			
: 16 e	F16 e	: 17 Q	F17 Q
			
: 18 s	F18 s	: 19 A	F19 A
			
: 20 a	F20 a	: 21 A	F21 A
			
: 22 a	F22 a	: 23 o	F23 o
			
: 24 o	F24 o	: 25 0	F25 0
			
: 26 u	F26 u	: 27 E	F27 E
			
: 28 e	F28 e	: 29 #	F29 #
			
: 30 f	F30 f	: 31 #	F31 #
			

ALPHA-EINZELZEICHEN 0-255 :REPLACE - FAPPEND

: 32



F32



: 33 !



F33 !



: 34 "



F34 "



: 35 #



F35 #



: 36 \$



F36 \$



: 37 %



F37 %



: 38 &



F38 &



: 39 '



F39 '



: 40 (



F40 (



: 41 )



F41 )



: 42 \*



F42 \*



: 43 +



F43 +



: 44 ,



F44 ,



: 45 -



F45 -



: 46 .



F46 .



: 47 /



F47 /



: 48 0



F48 0



: 49 1



F49 1



: 50 2



F50 2



: 51 3



F51 3



: 52 4



F52 4



: 53 5



F53 5



: 54 6



F54 6



: 55 7



F55 7



: 56 8



F56 8



: 57 9



F57 9



: 58 :



F58 :



: 59 ;



F59 ;



: 60 <



F60 <



: 61 =



F61 =



: 62 >



F62 >



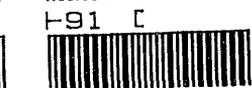
: 63 ?



F63 ?



ALPHA-EINZELZEICHEN 0-255 : REPLACE - F-APPEND



ALPHA-EINZELZEICHEN 0-255 :REPLACE - FAPPEND

: 96 `	F96 `	: 97 a	F97 a
: 98 b	F98 b	: 99 c	F99 c
: 100 d	F100 d	: 101 e	F101 e
: 102 f	F102 f	: 103 g	F103 g
: 104 h	F104 h	: 105 i	F105 i
: 106 j	F106 j	: 107 k	F107 k
: 108 l	F108 l	: 109 m	F109 m
: 110 n	F110 n	: 111 o	F111 o
: 112 p	F112 p	: 113 q	F113 q
: 114 r	F114 r	: 115 s	F115 s
: 116 t	F116 t	: 117 u	F117 u
: 118 v	F118 v	: 119 w	F119 w
: 120 x	F120 x	: 121 y	F121 y
: 122 z	F122 z	: 123 {	F123 {
: 124	F124	: 125 }	F125 }
: 126 ~	F126 ~	: 127 ʃ	F127 ʃ

# ALPHA-EINZELZEICHEN 0-255 : REPLACE - TAPPEND

: 128 *	T128 *	: 129 *	T129 *
: 130 x̄	T130 x̄	: 131 †	T131 †
: 132 α	T132 α	: 133 β	T133 β
: 134 Γ	T134 Γ	: 135 ↓	T135 ↓
: 136 Δ	T136 Δ	: 137 σ	T137 σ
: 138	T138	: 139 λ	T139 λ
: 140 μ	T140 μ	: 141	T141
: 142 τ	T142 τ	: 143 ϖ	T143 ϖ
: 144 θ	T144 θ	: 145 Ω	T145 Ω
: 146 δ	T146 δ	: 147 á	T147 á
: 148 ã	T148 ã	: 149 ä	T149 ä
: 150 ä	T150 ä	: 151 ö	T151 ö
: 152 õ	T152 õ	: 153 0	T153 0
: 154 ũ	T154 ũ	: 155 ð	T155 ð
: 156 œ	T156 œ	: 157 ≠	T157 ≠
: 158 £	T158 £	: 159 ⌘	T159 ⌘

ALPHA-EINZELZEICHEN 0-255 :REPLACE - FAPPEND

: 160



F160



: 161 !



F161 !



: 162 "



F162 "



: 163 #



F163 #



: 164 \$



F164 \$



: 165 %



F165 %



: 166 &



F166 &



: 167 '



F167 '



: 168 <



F168 <



: 169 >



F169 >



: 170 \*



F170 \*



: 171 +



F171 +



: 172 ,



F172 ,



: 173 -



F173 -



: 174 .



F174 .



: 175 /



F175 /



: 176 0



F176 0



: 177 1



F177 1



: 178 2



F178 2



: 179 3



F179 3



: 180 4



F180 4



: 181 5



F181 5



: 182 6



F182 6



: 183 7



F183 7



: 184 8



F184 8



: 185 9



F185 9



: 186 :



F186 :



: 187 ;



F187 ;



: 188 <



F188 <



: 189 =



F189 =



: 190 >



F190 >



: 191 ?



F191 ?



ALPHA-EINSELZEICHEN 0-255 : REPLACE - FAPPEND

: 192 @



F192 @



: 193 A



F193 A



: 194 B



F194 B



: 195 C



F195 C



: 196 D



F196 D



: 197 E



F197 E



: 198 F



F198 F



: 199 G



F199 G



: 200 H



F200 H



: 201 I



F201 I



: 202 J



F202 J



: 203 K



F203 K



: 204 L



F204 L



: 205 M



F205 M



: 206 N



F206 N



: 207 O



F207 O



: 208 P



F208 P



: 209 Q



F209 Q



: 210 R



F210 R



: 211 S



F211 S



: 212 T



F212 T



: 213 U



F213 U



: 214 V



F214 V



: 215 W



F215 W



: 216 X



F216 X



: 217 Y



F217 Y



: 218 Z



F218 Z



: 219 [



F219 [



: 220 \



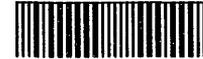
F220 \



: 221 ]



F221 ]



: 222 ^



F222 ^



: 223 \_



F223 \_



ALPHA-EINZELZEICHEN 0-255 : REPLACE - F-APPEND

: 224 `



F224 `



: 225 a



F225 a



: 226 b



F226 b



: 227 c



F227 c



: 228 d



F228 d



: 229 e



F229 e



: 230 f



F230 f



: 231 g



F231 g



: 232 h



F232 h



: 233 i



F233 i



: 234 |



F234 |



: 235 k



F235 k



: 236 l



F236 l



: 237 m



F237 m



: 238 n



F238 n



: 239 o



F239 o



: 240 p



F240 p



: 241 q



F241 q



: 242 r



F242 r



: 243 s



F243 s



: 244 t



F244 t



: 245 u



F245 u



: 246 v



F246 v



: 247 w



F247 w



: 248 x



F248 x



: 249 y



F249 y



: 250 z



F250 z



: 251 {



F251 {



: 252 |



F252 |



: 253 }



F253 }



: 254 ~



F254 ~



: 255 ƒ



F255 ƒ



## II. Alpha- und synthetische Textzeilen

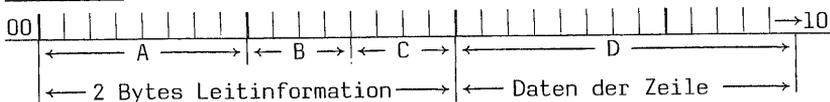
Angeichts der Einzelzeichen aus denen ja die Textzeilen aufgebaut sind, scheinen Barcodes für ganze Textzeilen überflüssig zu sein. Die schnellste Möglichkeit einen Text insbesondere mit nicht tastbaren Zeichen einzugeben, ist indes das Einlesen einer BC-Reihe. Eine Replace- und eine Appendzeile füllen das Alpharegister fehlerfrei. Programmgesteuert kann dies mit der Lesestiftfunktion 'WNNDDTA' geschehen. Sinnvolle Anwendungen ergeben sich immer da, wo gleiche Texte öfter verwendet werden, z.B. Teilüberschriften oder Monatsnamen bei Aufstellungen. Zur Speicherung längerer Texte sind Barcodes nicht geeignet.

Für synthetische Programmierung sind bisher teils längere Hilfsprogramme oder das CCD-Modul nötig, um eine beliebige Kombination von Bytes in den Programmspeicher zu bringen. 'Urvater' aller dieser Byteladeprogramme ist das sehr komfortable Programm "BC" mit 441 Bytes Speicherbedarf aus dem bereits zitierten Wickes. Mit den Programmen für die Anfertigung einer synthetischen Textzeile wird für das praktische Programmieren überhaupt kein Speicherplatz benötigt, weil die SYNTEXTZEILEN-Barcodes vorher angefertigt werden. Nach Herstellung einer solchen BC-Zeile wird das Programm "SYNT.." gelöscht. Im Gegensatz dazu muß ein Byteladeprogramm während des Programmierens im Rechner sein. Ebenso ist die Eingabe von synthetischen Befehlen mit einer speziellen Art von Text-BC zum Kinderspiel geworden; ja es können sogar größere Zahlen als Texte bytesparend verwendet werden.

### a. Alpha-Textzeilen

Barcodes von Alpha-Textzeilen ermöglichen, den Text mit einem Lesestiftstrich ins Alpharegister oder eine Programmzeile einzulesen. Es können alle ASCII-Zeichen zwischen 1 und 127 verwendet werden. Ein Null-Byte darf im Text nicht vorkommen.

#### 1. Aufbau



A = 1 Byte 8-Bit Prüfsumme

B = 1. Nybble 4-Bit Typindikator

C = 2. Nybble 4-Bit Anzahl Zeichen der Zeile

D = maximal 14 Bytes Textzeichen

#### 1.1 Der Typindikator

Der Typindikator ist identisch mit dem der Dreibyte-Einzelzeichen aus Abschnitt I.b.1.1. Außer B = 7 replace oder 8 append, sind vor allem C und D in Textzeilen variabel. Hierzu zwei Beispiele: eine Textzeile mit 7 Zeichen soll das Alpharegister überschreiben. B = 7 - C = 7 - binär 01110111 - Dezimalwert des Byte = 119. Eine andere Textzeile soll 14 Zeichen an den Alphainhalt anhängen: B = 8 - C = 14 - binär 10001110 - Dezimalwert des Byte = 142. Zur Vereinfachung also  $112+n$  ( $n$  = Anzahl Textzeichen) oder  $128+n$  für das Indikatorbyte aus Nybble B+C. Hier begegnet uns bereits die Zerlegung eines Bytes in Nybble. Das erste Nybble B ergibt  $7 \times 16 = 112$  oder  $8 \times 16 = 128$ , die obigen Dezimalwerte. Warum nun in D nur 14 Zeichen? Das Alpharegister 'faßt' doch 24 Zeichen. Nun, die BC von Textzeilen können sowohl ins Alpharegister als auch in PRGM-Zeilen eingelesen werden. In Programmzeilen ist aber die Zeichenanzahl auf maximal 15 festgelegt, dabei zählt das Appendzeichen mit. Weiterhin können in einer HP 41 BC-Zeile höchstens 16 Bytes dargestellt werden: Prüfsumme, Indikator = 2, dazu

14 Textbytes. Das Appendzeichen bei B = 8 wird automatisch hinzugefügt; ferner ist die Anzahl Textzeichen der Reihe in einem Nybble (C) verschlüsselt, in welchem maximal die Zahl 15 möglich ist.

1.2 Die 8-Bit Prüfsumme und der Text

Die Prüfsumme errechnet sich aus der Dezimalwert-Addition aller Textzeichen und des Indikatorbyte. Beispiel für eine Replacezeile:

```

Feld D:   T e x t           z u           l a n g
Dezimal:  84 101 120 116   32 122 117   32 108  97 110 103
Indikator = 112 + 12 Zeichen - Wert des Indikatorbyte = dezimal 124
Gesamtaddition = (B/C) + D = 1266 : 256 = 4 Rest 242 - Prüfsumme 246

```

Prüfsumme und Indikator stehen der Barcode-Textzeile voran. Die BC-Zeile lautet

Replace-Text: Text zu lang



Sofern der Text an den Inhalt des Alpharegisters angehängt werden soll ist sie

Append-Text: Text zu lang



Prüfsumme und Indikator werden selbstverständlich beim Einlesen nicht ins Alpharegister oder eine Programmzeile gebracht. Probieren Sie dies ruhig einmal aus.

2. Herstellen von Text-BC auf dem Thermodrucker 82162A

Für die Herstellung von Textzeilenbarcodes stehen drei Programme zur Verfügung, die nachstehend beschrieben werden. Die Bedienung aller PRGM ist gleich.

2.1 Nur Rechner HP 41 CV - Drucker, keine Module "TEXTBC"

Programmbedienung:

- SIZE: mindestens 016

XEQ "TEXTBC"

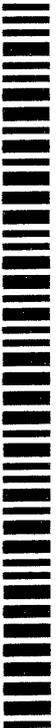


- "APPEND? J/N" - "J" = append - "N" = replace - 'R/S'
- "TEXT?" - Sie können nun maximal 14 Zeichen und mit dem Lesestift auch alle nicht tastbaren Zeichen 1-126 aus der Tabelle 'Zweibyte Funktions- und Druckerzeichen 0-127' eingeben. - 'R/S'

Bei mehr als 14 Zeichen erfolgt Hinweis: "Text zu lang!"  
Ist das Alpharegister leer - Hinweis: "Kein Text!" - Für eine zweite Textzeile dann einfach 'R/S' drücken.

Das Programmlisting finden Sie auf Seite 44. Die Routine LBL 03 ist dem Buch J.S. Dearing "Tricks, Tips und Routinen für Taschenrechner der Serie HP-41", Deutsche Ausgabe von H. Dalkowski, Helder mann Verlag, Berlin - Seite 35 entnommen. Diese Routine wird noch häufig verwendet. Die Schritte 96-106 dieses Programms berechnen die Prüfsumme. Die Befehle Zeilen 108 und 109 versetzen den Drucker in den 8-Bit Spaltenmodus. 'TONE 7' und 'TONE 0' sind auch in diesem wie in allen anderen Programmen die synthetischen 'TONE 57' bzw. 'TONE 10'.

-TEXTZEILE-  
Max= 14 Buchstaben Länge  
APPEND? J/N  
N  
OK?



01*LBL "TEXTBC"	37*LBL 01	73 ST+ J	109 ACCOL	145 CLA
02*LBL 00	38 2.015	74 RDN	110 RDN	146 RDN
03 SF 12	39 STO 00	75 6	111 128	147 RTN
04 CF 13	40 "TEXT?"	76 ST* J	112 RCL 00	148*LBL 10
05 SF 28	41 AON	77 RDN	113 INT	149 127
06 CF 29	42 TONE 7	78 E1	114 +	150 X<>Y
07 FIX 0	43 STOP	79 ST* L	115 XEQ 00	151 X<=Y?
08 "--TEXTZEILE--"	44 AOFF	80 X<> L	116 XEQ 00	152 GTO 11
09 PRA	45 FC?C 23	81 ST+ J	117 RCL 01	153 -
10 CF 12	46 GTO 01	82 CLX	118 XEQ 00	154 1
11 "Max= 14 Buchsta"	47 PRA	83 X<> J	119 RCL 00	155 +
12 "Hben Länse"	48 XEQ 03	84 RTN	120 INT	156 .
13 PRA	49 X=0?	85*LBL 04	121 E-3	157 1
14 CLRG	50 GTO 13	86 RCL 00	122 +	158 BLDSPEC
15 "J"	51*LBL 02	87 INT	123*LBL 06	159 X<>Y
16 ASTO X	52 STO IND 00	88 ST+ 01	124 XEQ 07	160 BLDSPEC
17 ENTER↑	53 XEQ 03	89 RCL IND X	125 DSE X	161 RTN
18 SF 12	54 X=0?	90 DSE Y	126 GTO 06	162*LBL 11
19 "APPEND? J/N"	55 GTO 04	91*LBL 05	127 ADV	163 .
20 AVIEW	56 ISG 00	92 RCL IND Y	128 ADV	164 X<>Y
21 AON	57 GTO 02	93 +	129 CF 17	165 BLDSPEC
22 TONE 7	58 GTO 12	94 DSE Y	130 CLST	166 RTN
23 STOP	59*LBL 03	95 GTO 05	131 BEEP	167*LBL 12
24 AOFF	60 "↑↓****"	96 SF 00	132 ADV	168 "Text zu lang!"
25 PRA	61 RCL I	97 XEQ 00	133 ADV	169 TONE 0
26 ASTO X	62 STO I	98*LBL 00	134 ADV	170 PRA
27 X=Y?	63 CLX	99 RCL X	135 RTN	171 ADV
28 111	64 X<> J	100 256	136 GTO 00	172 GTO 01
29 111	65 SIGN	101 ST/ Z	137*LBL 07	173*LBL 13
30 X*Y?	66 CLX	102 MOD	138 RCL IND X	174 "Kein Text!"
31 95	67 X<> \	103 +	139*LBL 08	175 PRA
32 16	68 "↑***"	104 INT	140 XEQ 10	176 TONE 0
33 +	69 X<> I	105 FS?C 00	141 ARCL X	177 ADV
34 STO 01	70 X<> L	106 RTN	142 RDN	178 GTO 01
35 CF 12	71 X<> \	107 SF 17	143*LBL 09	179 END
36 CF 23	72 INT	108 .	144 OUTA	382 Bytes

## 2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "TEXTBc"

### Programmbedienung:

- SIZE: falls < 016 wird programmgesteuert 'SIZE 016' eingestellt.

XEQ "TEXTBc"



Die PRGM-Bedienung ist sonst gleich wie "TEXTBC"

## 2.3 Rechner HP 41 CV - Drucker, XF- und PM PRGM "TEXTbc"

### Programmbedienung: wie "TEXTBC"

- SIZE: könnte bei diesem Programm 000 sein.

XEQ "TEXTbc"



Die Listings der beiden Programme mit kurzen Erläuterungen stehen auf Seite 45. Die PRGM-Länge 'schrumpft' durch Moduleinsatz von 382 über 302 auf 201 Bytes.

01*LBL "TEXTbc"	29 74	57*LBL 02	85 ACCOL	113 ADV
02*LBL 00	30 X=Y?	58 ATOX	86 RDN	114 RTN
03 SIZE?	31 90	59 STO IND 00	87 127	115 GTO 00
04 16	32 37	60 ISG 00	88 RCL 00	116*LBL 05
05 X<Y?	33 +	61 GTO 02	89 INT	117 RCL IND X
06 PSIZE	34 STO 01	62 RCL 00	90 +	118*LBL 06
07 SF 12	35 CF 23	63 INT	91 XEQ 06	119 XTOA
08 CF 13	36*LBL 01	64 ST+ 01	92 XEQ 06	120*LBL 07
09 SF 28	37 2.001	65 RCL IND X	93 RCL 01	121 OUTA
10 CF 29	38 STO 00	66 DSE Y	94 XEQ 06	122 CLA
11 FIX 0	39 "TEXT?"	67*LBL 03	95 LASTX	123 RDN
12 "-TEXTZEILE-"	40 AON	68 RCL IND Y	96 1	124 RTN
13 PRA	41 TONE 7	69 +	97 -	125*LBL 08
14 CF 12	42 STOP	70 DSE Y	98 E3	126 "Kein Text!"
15 "Max= 14 Buchsta"	43 AOFF	71 GTO 03	99 /	127 PRA
16 "tben Länge"	44 FC?C 23	72 SF 00	100 2	128 TONE 0
17 PRA	45 GTO 01	73 XEQ 00	101 +	129 ADV
18 CLRG	46 PRA	74*LBL 00	102*LBL 04	130 GTO 01
19 SF 12	47 ALENG	75 RCL X	103 XEQ 05	131*LBL 09
20 "APPEND? J/N"	48 X=0?	76 256	104 ISG X	132 "Text zu lang!"
21 AVIEW	49 GTO 08	77 ST/ Z	105 GTO 04	133 PRA
22 AON	50 14	78 MOD	106 ADV	134 TONE 0
23 TONE 7	51 X<Y?	79 +	107 ADV	135 ADV
24 STOP	52 GTO 09	80 INT	108 CF 17	136 GTO 01
25 AOFF	53 X<>Y	81 FS?C 00	109 CLST	137 END
26 PRA	54 E3	82 RTN	110 BEEP	302 Bytes
27 CF 12	55 /	83 SF 17	111 ADV	
28 ATOX	56 ST+ 00	84 .	112 ADV	

Auch bei in diesem PRGM ist die Prüfsummenberechnung 'programmiert' - Z: 72-82.

01*LBL "TEXTbc"	16 AVIEW	31 "TEXT?"	46 RDN	61 RTN
02*LBL 00	17 AON	32 AON	47 +	62 GTO 00
03 SF 12	18 TONE 7	33 TONE 7	48 XTOA	63*LBL 02
04 CF 13	19 STOP	34 STOP	49 -1	64 "Kein Text!"
05 SF 28	20 AOFF	35 AOFF	50 AROT	65 PRA
06 CF 29	21 PRA	36 FC?C 23	51 ALENG	66 TONE 0
07 FIX 0	22 CF 12	37 GTO 01	52 -	67 ADV
08 "-TEXTZEILE-"	23 ATOX	38 PRA	53 BCKSM	68 GTO 01
09 PRA	24 74	39 RCL L	54 BCO	69*LBL 03
10 CF 12	25 X=Y?	40 ALENG	55 CLST	70 "Text zu lang!"
11 "Max= 14 Buchsta"	26 90	41 X=0?	56 CLA	71 PRA
12 "tben Länge"	27 38	42 GTO 02	57 BEEP	72 TONE 0
13 PRA	28 +	43 14	58 ADV	73 ADV
14 SF 12	29 STO L	44 X<Y?	59 ADV	74 GTO 01
15 "APPEND? J/N"	30*LBL 01	45 GTO 03	60 ADV	75 END 201 Bytes

Hier zeigt sich die Leistungsfähigkeit des Plottermoduls. Für die Berechnung der Prüfsumme wird nur der XROM-Befehl 'BCKSM' eingesetzt, und den Ausdruck der Barcodezeile besorgt die Funktion 'BCO', wobei die Anzahl der als Barcodes aus dem Alpharegister zu druckenden Zeichen, hier mit 'ALENG' ermittelt, als Negativwert im X-Register stehen muß. Die einzelnen Bytes für den BC-Druck werden auch nicht nacheinander gesendet, sondern der Ausdruck erfolgt direkt aus dem Alpharegister, weswegen keine Speicherregister für die 'Zwischenlagerung' und anschließende Einzeldefinition im Alpharegister für das Senden nötig sind.

### 3. Herstellen von Text-BC mit dem ThinkJet-Drucker HP-IL 2225B

Es wird hier noch einmal an die Ausführungen in Kapitel 1 Abschnitt IV.b.1. angeknüpft. Kernproblem eines Barcode-Programms für Drucker ohne BC-Option ist die Formatierung der Bausteine und der Datentransfer. Theoretisch könnten für synthetische Programmierung 256 ASCII-Zeichen vorkommen, so daß für jedes Byte eine entsprechende Bausteinkombination vorhanden sein müßte. Selbst wenn man die Textzeichen auf 1-126 beschränkte, würde noch ein unmöglich langes Programm entstehen. In dem nachfolgend vorgestellten Programm "TXTJBC" wird deswegen jedes Textbyte in Nybble zergliedert, wobei je nachdem wie der Baustein endet, mit 2x15 Kombinationen gearbeitet werden kann - (Ab LBL 15). Für die Datenübertragung Rechner-Drucker wird der File "BC" im erweiterten Speicher eröffnet. Alle Bausteine werden in einem einzigen Record akkumuliert und nach Fertigstellung der gesamten Barcode-Bausteinreihe dann mit der Funktion 'ARCLREC' zuerst ins Alpharegister und von dort anschließend 24-zeichenweise an den Druckerbuffer gesendet. Ganz allgemein zeigt sich bei diesen Datenmengen die Grenze der Arbeitsgeschwindigkeit des HP 41 sehr deutlich. Mit der vorliegenden Programmlösung dauert die Anfertigung einer 12-Zeichenzeile gut 3 Minuten. In Kapitel 5 anlässlich der Herstellung von PRGM-BC folgen noch einige Gedanken zu einer synthetischen Programmlösung zur Erhöhung der Arbeitsgeschwindigkeit.

#### 3.1 Rechner HP 41 CV - ThinkJet-Drucker, XF-Modul PRGM "TXTJBC"

##### Programmbedienung:

- SIZE: das Programm stellt nötigenfalls 'SIZE 015' ein.

XEQ "TXTJBC"



Nach Ausdruck der Kopfzeile, die gleichzeitig verschiedene Textmöglichkeiten des ThinkJet-Druckers zeigt, werden Sie gefragt:

- "APPEND? J/N": "J" = append - "N" = replace - 'R/S'
- "TEXT?": Sie können nun maximal 12 Textzeichen eingeben - 'R/S'

Bleibt das Alpharegister leer - Hinweis: "Kein Text!"

Sind mehr als 12 Zeichen vorhanden: "Text zu lang!"

Die Kommentarzeilen 02-04 im Programmlisting können bei Bedarf gelöscht werden.

—BARCODETEXTZEILE— Max= 12 Buchstaben Länge ASCII-Zeichensatz 32-125

Replace-Text: Alles klar??



01*LBL "TXTJBC"	17 XYY?	33 ACA	49 ACA	65 X=Y?
02 "!" Text-Barcodes"	18 PSIZE	34 "r"	50 "r"	66 SF 02
03 "!" auf ThinkJet-	19 2	35 OUTA	51 OUTA	67 X=Y?
04 "!" Drucker 2225B"	20 STO 14	36 "12"	52 "32-125"	68 90
05 "HP2225B"	21 SF 17	37 ACA	53 ACA	69 37
06 FINDID	22 "E&k1SE&dE&l0L"	38 "t"	54 PRBUF	70 +
07 SELECT	23 OUTA	39 OUTA	55 "E&dDE&k3S"	71 STO 01
08*LBL a	24 SF 28	40 " Buchsta"	56 OUTA	72 FS? 02
09 "BC"	25 CF 29	41 ACA	57 CF 02	73 "Append-
10 SF 25	26 FIX 0	42 "ben Lnge "	58 "APPEND? J/N"	74 FC? 02
11 23	27 "-BARCODETEXTZEI"	43 OUTA	59 AON	75 "Replace-
12 CRFLAS	28 "LE- "	44 "E&k2S"	60 TONE 7	76 "tText: "
13 CLFL	29 ACA	45 OUTA	61 STOP	77 ACA
14 CLRG	30 "E&k0S"	46 "ASCII-Zeichensa"	62 AOFF	78 "E&k0S"
15 SIZE?	31 OUTA	47 ACA	63 ATOX	79 OUTA
16 15	32 "Max= "	48 "tz "	64 74	80 CF 23

81*LBL 00	140 RCL X	199 BEEP	258*LBL 00	317 RTN
82 2.001	141 16	200 RTN	259 "t+pt"	318*LBL 00
83 STO 00	142 ST/ Z	201 GTO a	260 CF 01	319 "t+pt"
84 "TEXT?"	143 MOD	202*LBL 15	261 RTN	320 SF 01
85 AON	144 X<>Y	203 FC? 01	262*LBL 21	321 RTN
86 TONE 7	145 INT	204 GTO 00	263 FC? 01	322*LBL 27
87 STOP	146 15	205 "tpppp"	264 GTO 00	323 FC? 01
88 AOFF	147 +	206 SF 01	265 "t+pt"	324 GTO 00
89 FC?C 23	148 XEQ IND X	207 RTN	266 SF 01	325 "t+ppp"
90 GTO 00	149 X<>Y	208*LBL 00	267 RTN	326 SF 01
91 ACR	150 15	209 "t+ttt"	268*LBL 00	327 RTN
92 PRBUF	151 +	210 CF 01	269 "t+ttt"	328*LBL 00
93 ALENG	152 XEQ IND X	211 RTN	270 CF 01	329 "t+ttt"
94 X=0?	153 ALENG	212*LBL 16	271 RTN	330 CF 01
95 GTO 31	154 ST+ 14	213 FC? 01	272*LBL 22	331 RTN
96 12	155 APPCHR	214 GTO 00	273 FC? 01	332*LBL 28
97 X<Y?	156 RCL IND 00	215 "tpppt"	274 GTO 00	333 FC? 01
98 GTO 32	157 ISG 00	216 CF 01	275 "t+pt"	334 GTO 00
99 X<>Y	158 GTO 04	217 RTN	276 CF 01	335 "t+pt"
100 E3	159 FC? 01	218*LBL 00	277 RTN	336 CF 01
101 /	160 "tp"	219 "t+ttt"	278*LBL 00	337 RTN
102 ST+ 00	161 FS? 01	220 SF 01	279 "t+ttt"	338*LBL 00
103*LBL 01	162 "*t+"	221 RTN	280 SF 01	339 "t+ttt"
104 ATOX	163 ALENG	222*LBL 17	281 RTN	340 SF 01
105 STO IND 00	164 ST+ 14	223 FC? 01	282*LBL 23	341 RTN
106 ISG 00	165 APPCHR	224 GTO 00	283 FC? 01	342*LBL 29
107 GTO 01	166 "ttd0Er1280s"	225 "tpppt"	284 GTO 00	343 FC? 01
108 RCL 00	167 OUTA	226 CF 01	285 "t+ttt"	344 GTO 00
109 1	168 "E*rA"	227 RTN	286 CF 01	345 "t+ttt"
110 -	169 OUTA	228*LBL 00	287 RTN	346 CF 01
111 INT	170 1.024	229 "t+ttt"	288*LBL 00	347 RTN
112 ST+ 01	171 STO 00	230 SF 01	289 "tpppp"	348*LBL 00
113 RCL IND X	172*LBL 05	231 RTN	290 SF 01	349 "t+pt"
114 DSE Y	173 .	232*LBL 18	291 RTN	350 SF 01
115*LBL 02	174 SEEKPT	233 FC? 01	292*LBL 24	351 RTN
116 RCL IND Y	175 SF 17	234 GTO 00	293 FC? 01	352*LBL 30
117 +	176 "E*b"	235 "tpppt"	294 GTO 00	353 FC? 01
118 DSE Y	177 ARCL 14	236 SF 01	295 "t+ttt"	354 GTO 00
119 GTO 02	178 "tW"	237 RTN	296 SF 01	355 "t+ttt"
120 SF 00	179 OUTA	238*LBL 00	297 RTN	356 SF 01
121 XEQ 03	180 CF 17	239 "t+ttt"	298*LBL 00	357 RTN
122*LBL 03	181*LBL 06	240 CF 01	299 "t+pt"	358*LBL 00
123 RCL X	182 CLA	241 RTN	300 CF 01	359 "t+ttt"
124 256	183 SF 25	242*LBL 19	301 RTN	360 CF 01
125 ST/ Z	184 ARCLREC	243 FC? 01	302*LBL 25	361 RTN
126 MOD	185 FC? 25	244 GTO 00	303 FC? 01	362*LBL 31
127 +	186 GTO 07	245 "t+ttt"	304 GTO 00	363 "Kein Text!"
128 INT	187 OUTA	246 CF 01	305 "t+ttt"	364 PRA
129 FS?C 00	188 GTO 06	247 RTN	306 SF 01	365 TONE 0
130 RTN	189*LBL 07	248*LBL 00	307 RTN	366 ADV
131 E3	190 ISG 00	249 "t+ttt"	308*LBL 00	367 GTO a
132 ST/ 00	191 GTO 05	250 SF 01	309 "t+pt"	368*LBL 32
133 ISG 00	192 "E*rB"	251 RTN	310 CF 01	369 "Text zu lang!"
134 RDN	193 OUTA	252*LBL 20	311 RTN	370 PRA
135 "pp"	194 PRBUF	253 FC? 01	312*LBL 26	371 TONE 0
136 APPREC	195 .	254 GTO 00	313 FC? 01	372 ADV
137 SF 01	196 X<>F	255 "t+ttt"	314 GTO 00	373 GTO a
138*LBL 04	197 CLST	256 SF 01	315 "t+ttt"	374 END
139 CLA	198 CLA	257 RTN	316 CF 01	1053 Bytes

#### 4. Arbeiten mit Alpha-Textzeilen

Die beste Demonstration für das Arbeiten mit Alpha-Textzeilen ist ein Beispiel. Nehmen wir an, Sie hätten eine Aufstellung nach Monaten programmgesteuert anzufertigen, so können Sie die Monatsnamen als Programm-Textzeilen eingeben und über ein LBL aufrufen. Mit dem Lesestift und der Funktion 'WNDDTA' können Sie PRGM-Länge und -Laufzeit sehr verkürzen. Lesen Sie bitte das nachfolgende Programm ein, dann 'XEQ "AT"' und nach "MONAT?" die folgende sowie nach dem weiteren Stop "A....?", die letzte BC-Reihe abtasten.

PROGRAMM: "AT" Benötigte Register: 5 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



01\*LBL "AT"

Reihe 2: Zeilen 2-4



02 "MONAT?"

Reihe 3: Zeilen 5-9



03 CF 21

04 AVIEW

05 WNDDTA

06 AVIEW

07 WNDDTA

Reihe 4: Zeilen 9-12



08 SF 21

09 PRA

Reihe 5: Zeilen 12-12



XEQ "AT"



10 BEEP

11 GTO "AT"

12 END



Im Kapitel 3 kommt das Einlesen von Zahlen ins X-Register und das sequentielle Eingeben von Zahlen oder ASCII-Daten in Speicherregister hinzu.

#### b. Synthetische Programmierung: was ist das?

Unter synthetischer Programmierung versteht man mit Bezug auf den HP 41 eine Programmiermethode, welche eine Unzulänglichkeit im Prozessor dieses Rechners ausnutzt. Es entstehen so neue Befehle, welche die Programmiermöglichkeiten erweitern. Die nach dem Programmlisting "TEXTBC" erwähnte Routine LBL 03 ist ein klassisches Beispiel. Hier wird das Dezimal-Äquivalent des rechts im Alphatext stehenden ASCII-Zeichens ins X-Register gebracht. Dies ist ohne Module sonst nicht möglich. Die genaue Grundlage dieser Programmiermethode ist in dem in Abschnitt II. erwähnten 'Wickes' begründet. Eine Sammlung vieler Routinen findet man in dem ebenfalls erwähnten Buch von J.S. Dearing.

#### c. Synthetische Textzeilen

Synthetische Textzeilen sind Bytekombinationen, die normalerweise nicht in eine Programmzeile und über diese dann ins Alpha- oder ein Statusregister gebracht werden können. Es sind dies Textzeilen, die entsprechend dem Dezimalwert des gewünschten Byte in beliebiger Kombination zur Anfertigung einer BC-Reihe eingegeben und von dieser in eine Programmzeile eingelesen werden können. Alle Bytes von 0 bis 255 sind zugelassen.

#### 1. Aufbau, der Typindikator und die 8-Bit Prüfsumme

Der Aufbau und die Leitinformation ist völlig identisch mit der unter a. Alpha-Textzeilen 1. Abschnitt beschriebenen Zusammensetzung.

## 2. Herstellen von "SYNTEXTZEILEN" auf dem Thermodrucker

Diese Textzeilenbarcodes können wiederum mit drei verschiedenen Programmen hergestellt werden. Die Bedienung ist gleich.

### 2.1 Nur Rechner HP 41 CV - keine Module PRGM "SYNTBC"

#### Programmbedienung:

- SIZE: mindest 016

XEQ "SYNTBC"



SYNTEXTZEILE  
APPEND? J/N  
N  
MAX= 14 BYTE

BYTE 1 239 o  
BYTE 2 235 k  
BYTE 3 191 ?

- "APPEND? J/N": "J" = append - "N" = replace - 'R/S'
- "BYTE 1": Sie können nun den Dezimalwert des ersten Byte Ihrer synthetischen Textzeile, z.B. 239 eingeben - 'R/S'
- "BYTE 2": So können Sie bis maximal 14 Bytedezimalwerte eingeben. Nach Eingabe des 14. Byte löst 'R/S' den Druck der BC-Zeile aus. Soll die Zeile vorher 'abgebrochen' werden, ohne Eingabe 'R/S' drücken. Geben Sie nun noch Byte 235 und dann 191 ein. Bei Aufforderung Byte 4 nur 'R/S' drücken und die Zeile wird wie nebenstehend gedruckt.

Für den Start zu einer zweiten Zeile einfach 'R/S' drücken. Geben Sie nach Aufforderung "BYTE 1" nichts ein und drücken 'R/S', wird eine kurze bedeutungslose BC-Reihe ausgedruckt. Mit 'R/S' startet das Programm neu.

Lesen Sie die nebenstehende "SYNTEXTZEILE" ins Alpharegister oder als Programmzeile ein, so erscheint dreimal das Zeichen , die sogenannte 'Black Box'. Haben Sie die Zeile ins Alpharegister eingelesen, so drücken Sie die PRINT-Taste am Drucker. Als PRGM-Zeile führen Sie 'SST' und 'PRA' aus, und die Zeile "ok?" wird ausgedruckt.

Hier noch ein Tip: Für den Druck von Texten aus dem Alpharegister genügt der Befehl auch dann, wenn der Drucker nicht die primäre Einheit ist. 'PRA' wird nur vom Drucker verstanden. Ist ein anderes Gerät primäre Einheit, kann der Thermodrucker nicht in den Spaltenmodus gelangen, und es werden keine Barcodes ausgedruckt. Sie können dies mit der Herstellung der nebenstehenden BC-Reihe probieren. Definieren Sie ein anderes Gerät als primäre Einheit, werden keine Barcodes erstellt. Es erfolgt nur der programmierte Zeilenvorschub.



01*LBL "SYNTBC"	12 STO 00	23 ASTO X	34 ADV	45 FC? 22
02*LBL 00	13 "J"	24 X=Y?	35*LBL 01	46 GTO 04
03 SF 12	14 ASTO X	25 111	36 CF 12	47 STO IND 00
04 CF 13	15 ENTER†	26 111	37 CF 22	48 ACA
05 SF 28	16 "APPEND? J/N"	27 X*Y?	38 "BYTE "	49 SF 12
06 CF 29	17 AVIEW.	28 95	39 RCL 00	50 CLA
07 FIX 0	18 AON	29 16	40 1	51 ARCL X
08 "SYNTEXTZEILE"	19 TONE 7	30 +	41 -	52 "† "
09 PRA	20 STOP	31 STO 01	42 ARCL X	53 .
10 CLRG	21 AOFF	32 "MAX= 14 BYTE"	43 TONE 7	54 X<Y
11 2.015	22 PRA	33 PRA	44 PROMPT	55 X*0?

56 GTO 02	78*LBL 05	100 INT	122 CLST	144 RTN
57 I	79 RCL IND Y	101 +	123 BEEP	145*LBL 10
58 BLDSPEC	80 +	102 XEQ 00	124 ADV	146 127
59 .	81 DSE Y	103 XEQ 00	125 ADV	147 X<>Y
60 BLDSPEC	82 GTO 05	104 RCL 01	126 ADV	148 X<=Y?
61 GTO 03	83 SF 00	105 XEQ 00	127 RTN	149 GTO 11
62*LBL 02	84 XEQ 00	106 RCL 00	128 GTO 00	150 -
63 XEQ 10	85*LBL 00	107 INT	129*LBL 07	151 1
64*LBL 03	86 RCL X	108 1	130 SF 25	152 +
65 ARCL X	87 256	109 -	131 RCL IND X	153 .
66 FMT	88 ST/ Z	110 E3	132 X=0?	154 1
67 ACA	89 MOD	111 /	133 ACCOL	155 BLDSPEC
68 ADV	90 +	112 2	134 X=0?	156 X<>Y
69 ISG 00	91 INT	113 +	135 GTO 09	157 BLDSPEC
70 GTO 01	92 FS?C 00	114*LBL 06	136*LBL 08	158 RTN
71*LBL 04	93 RTN	115 XEQ 07	137 XEQ 10	159*LBL 11
72 ADV	94 SF 17	116 ISG X	138 ARCL X	160 .
73 ADV	95 .	117 GTO 06	139 RDN	161 X<>Y
74 INT	96 ACCOL	118 ADV	140*LBL 09	162 BLDSPEC
75 ST+ 01	97 RDN	119 ADV	141 OUTA	163 END
76 RCL IND X	98 127	120 CF 17	142 CLA	308 Bytes
77 DSE Y	99 RCL 00	121 CF 12	143 RDN	

Die Schritte 129 bis 135 senden das 'Null'-Byte - Hex 00. Die Schritte LBL 10 und LBL 11 für die Herstellung auch von ASCII-Zeichen >127 kennen wir bereits.

## 2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "SYNTBc"

Programmbedienung: wie "SYNTBC"

- SIZE: das Programm stellt, falls geringer, 'SIZE 016' ein.  
XEQ "SYNTBc"



01*LBL "SYNTBc"	26 X=Y?	51 "I "	76 MOD	101 XEQ 04
02*LBL 01	27 90	52 XTOA	77 +	102 ISG X
03 SIZE?	28 37	53 FMT	78 INT	103 GTO 03
04 16	29 +	54 ACA	79 FS?C 00	104 ADV
05 X<>Y?	30 STO 01	55 ADV	80 RTN	105 ADV
06 PSIZE	31 "MAX= 14 BYTE"	56 ISG 00	81 SF 17	106 CF 17
07 SF 12	32 PRA	57 GTO 00	82 .	107 CF 12
08 CF 13	33 ADV	58*LBL 01	83 ACCOL	108 BEEP
09 SF 28	34*LBL 00	59 ADV	84 RDN	109 ADV
10 CF 29	35 CF 12	60 ADV	85 127	110 ADV
11 FIX 0	36 CF 22	61 INT	86 RCL 00	111 ADV
12 "SYNTEXTZEILE"	37 "BYTE "	62 ST+ 01	87 INT	112 RTN
13 PRA	38 RCL 00	63 RCL IND X	88 +	113 GTO 01
14 CLRG	39 1	64 DSE Y	89 XEQ 05	114*LBL 04
15 2.015	40 -	65*LBL 02	90 XEQ 05	115 SF 25
16 STO 00	41 ARCL X	66 RCL IND Y	91 RCL 01	116 RCL IND X
17 "APPEND? J/N"	42 TONE 7	67 +	92 XEQ 05	117 X=0?
18 AVIEW	43 PROMPT	68 DSE Y	93 LASTX	118 ACCOL
19 AON	44 FC? 22	69 GTO 02	94 1	119*LBL 05
20 TONE 7	45 GTO 01	70 SF 00	95 -	120 XTOA
21 STOP	46 STO IND 00	71 XEQ 07	96 E3	121*LBL 06
22 ROFF	47 ACA	72*LBL 07	97 /	122 OUTA
23 PRA	48 SF 12	73 RCL X	98 2	123 CLA
24 ATOX	49 CLA	74 256	99 +	124 RDN
25 74	50 ARCL X	75 ST/ Z	100*LBL 03	125 END 251 Bytes

2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "SYNTbc"

Programmbedienung: wie "SYNTBC"

- SIZE: das Programm stellt bei < 'SIZE 016' ein.

XEQ "SYNTbc"



01*LBL "SYNTbc"	19 AON	37 "BYTE "	55 ADV	73 ISG X
02*LBL 01	20 TONE 7	38 RCL 00	56 ISG 00	74 GTO 02
03 SIZE?	21 STOP	39 1	57 GTO 00	75 RCL 00
04 16	22 AOFF	40 -	58*LBL 01	76 CHS
05 X>Y?	23 PRA	41 ARCL X	59 CLA	77 BCCKSM
06 PSIZE	24 ATOX	42 TONE 7	60 RCL 00	78 BCD
07 SF 12	25 74	43 PROMPT	61 INT	79 CF 12
08 CF 13	26 X=Y?	44 FC? 22	62 ST+ 01	80 CLST
09 SF 28	27 90	45 GTO 01	63 1	81 CLA
10 CF 29	28 36	46 STO IND 00	64 -	82 BEEP
11 FIX 0	29 +	47 ACA	65 E3	83 ADV
12 "SYNTEXTZEILE"	30 STO 01	48 SF 12	66 /	84 ADV
13 PRA	31 "MAX= 14 BYTE"	49 CLA	67 1	85 ADV
14 CLRG	32 PRA	50 ARCL X	68 +	86 RTN
15 2.015	33 ADV	51 " " "	69*LBL 02	87 GTO 01
16 STO 00	34*LBL 00	52 XTOA	70 RCL IND X	88 END
17 "APPEND? J/N"	35 CF 12	53 FMT	71 XTOA	191 Bytes
18 AVIEW	36 CF 22	54 ACA	72 RDN	

3. Arbeiten mit synthetischen Textzeilen

In erster Linie werden synthetische Textzeilen ganz sicher zur Manipulation und Steuerung der normalerweise unzugänglichen Statusregister des Rechners eingesetzt. Wenn auch landläufig von Text- oder ASCII-Zeilen gesprochen wird, so sind diese aus Alphazeichen zusammengesetzte Zeilen tatsächlich Bytekombinationen, die nicht zum Ausdruck von Textinformationen, sondern der Zustandsveränderung der 'Rechnerverwaltung' dienen. Hierzu zwei Beispiele zur Flag-Massen-Manipulation. Synthetik-Kenner können gleich zum nächsten Abschnitt übergehen, wenn sie auf die Vorteile der Barcodenutzung verzichten möchten.

3.1 Massensteuerung aller Flags

Jedes Rechnerregister kann 7 Bytes aufnehmen. Der Rechner hat insgesamt 00-55 = 56 Flags. Jedes Byte hat 8 Bit. Jedes Bit repräsentiert eines der 7x8=56 Flags. Flag nicht gesetzt - Bit = 0; Flag gesetzt - Bit = 1. Die Flags werden im Statusregister d verwaltet. Mit den synthetischen Befehlen 'STO d' oder 'X<>d' können wir den Zustand der Flags 'mit einem Schlag' verändern, und zwar auch solche Flags zwischen 30-55, die sich den normalen Funktionen 'SF\_\_' oder 'CF\_\_' verschließen. Soll die Gesamtinformation des Registers d verändert werden, so benötigt die Textzeile 7 Bytes Information im Alpharegister. Für die Steuerung werden die synthetischen Befehle 'RCL M' und 'STO d' = weitere 4 Byte benötigt, insgesamt also 11 Byte (ohne Textbyte). Demgegenüber steht die Einzel-Befehlseingabe mit 'SF' oder 'CF' 'FIX' 'ENG' 'SCI' 'ON' 'AON' 'AOFF' 'RAD' 'DEG' 'GRAD' 'PSE' usw. Die Flagmassensteuerung ist erst dann bytesparend, wenn diese aufgezählten Ein- oder Zweibytefunktionen allein mehr Speicherplatz benötigen würden. Dies muß abgewogen werden. Sollen sonst unzugängliche Flags verändert werden, kann hierauf natürlich keine Rücksicht genommen werden. Aus der Zusammenstellung 'Flag-Bedeutungen und Flag-Zustand' kann für den Aufbau einer BC-Reihe Bit für Bit der nötige oder gewünschte Flagzustand entnommen werden.

**FLAG-Bedeutungen und FLAG-Zustand**

00-10	Anwenderflags allgemeine Verwendung		
11	PRGM-Start nach Einschalten oder Einlesen	0 = 'XEQ'-Start	1 = Automat - Start
12	Thermodrucker 82162A	0 = normalbreit	1 = doppelbreit
13	Thermodrucker 82162A	0 = Großbuchstaben	1 = Kleinbuchstaben
14	Kartenleser MK 'PROTECTET'	0 = überschreiben no	1 = überschreiben ja
<u>15 16</u>	Andere Drucker ohne Modus-Schalter		
0 0	"MAN"		
0 1	"NORM"		
1 0	"TRACE"		
1 1	"TRACE MIT STACKAUSDRUCK"		
17	Druckerzeilen-Funktion	0 = Zeile beenden	1 = Zeile fortsetzen
18-19	Spätere Verwendung		
20	CCD-Modul 'ACAXY' Text- und Zeilenausdruck	0 = Text Zahl	1 = Text.....Zahl
21	Druckerfunktionen	0 = ignorieren	1 = Normalbetrieb
22	Tastenfeld Zahleneingabe	0 = keine Eingabe	1 = Eingabe erfolgt
23	Tastenfeld Alphaeingabe	0 = keine Eingabe	1 = Eingabe erfolgt
24	Bereichsfehler	0 = OUT OF RANGE	1 = Fehlerignoranz
25	Funktionsfehler	0 = ERROR-Meldung	1 = Fehlerignoranz
26	Akustisches Signal	0 = aus	1 = Beep und Tone an
27	"USER"-Funktion	0 = aus	1 = ein
28	Dezimalzeichen	0 = Komma	1 = Punkt
29	Zifferngruppierung 1.000,00 - 1000,00	0 = nein	1 = ja
30	CATalogfunktion (nur intern) normal = 0	0 = keine Ausführung	1 = CAT läuft gerade
31	TIME-Modul: Monat/Tag-Tag/Monat - Wahl	0 = 'MDY'	1 = 'DMY'
32	HP-IL Modul normal = 0	0 = 'AUTOIO'	1 = 'MANIO'
33	Development-Modul IL-Steuerung normal = 0	0 = 'CF 33'	1 = 'SF 33'
34	I/O-Modul normal = 0	0 = 'ADRON'	1 = 'ADROFF'
35	Modul Kassettenduplikation normal = 0	0 = kein Autostart	1 = Automat - Start
<u>36 37 38 39</u>	Stellenzahl		
0 0 0 0	= 0 Kommastellen bis		
1 0 0 1	= 9 Kommastellen		
<u>40 41</u>	Anzeigeformat		
0 0	SCI		
1 0	FIX		
0 1	ENG		
<u>42 43</u>	Winkelmodus		
0 0	DEG		
1 0	GRAD		
0 1	RAD		
44	Rechnereinschaltdauer normal = 0	0 = Aus nach 10 Min	1 = andauernd ein
45	Dateneingabe normal = 0	0 = beendet	1 = erfolgt gerade
46	Tastenfeld-Teilfolgeeingabe normal = 0	0 = abgeschlossen	1 = erfolgt gerade
47	SHIFT-Taste normal = 0	0 = aus	1 = gerade gedrückt
48	ALPHA-Modus	0 = aus	1 = ein
49	Batterie- oder Akkuzustand normal = 0	0 = gut	1 = zu schwach
50	Meldung und Anzeige (intern) normal = 0	0 = keine M/A	1 = aktiv z.B. VIEW
51	'SST' normal = 0	0 = 'XEQ'-PRGM-Lauf	1 = 'SST'- PRGM-Lauf
52	PRGM-Modus	0 = aus	1 = ein
53	Periferie Ein-/Ausgabebereitschaft n = 0	0 = nicht bereit	1 = Gerät ist bereit
54	'PSE' - Funktion	0 = keine PRGM-PSE	1 = gerade PRGM-PSE
55	Druckeranwesenheit	0 = keine Drucker	1 = Drucker ist da

Die Flags 00 bis 29 können mit den Funktionen 'SF\_\_' oder 'CF\_\_' verändert werden; alle übrigen Flags sind nur mit synthetischen Befehlen oder teils mit flagverändernden Funktionen zu steuern. Beispiele: 'MANIO' setzt Flag 32, 'RAD' löscht Flag 42 und setzt Flag 43. Mit diesen beiden Flags können drei Winkelmodi definiert werden. 'SF 42' und 'SF 43' wären nur synthetisch möglich.

Als Beispiel für eine praktische Anwendung erarbeiten wir die Zeilen 03 bis 05 aus dem Programm "BLDSBC" im nächsten Abschnitt dieses Kapitels.

Bytes für Register d

Dezimal	0	0	4	57	0	128	1
Binär	00000000	00000000	00000100	00111001	00000000	10000000	00000001
Flags	00-07	08-15	16-23	24-31	32-39	40-47	48-55

Gesetzt sind die Flags: 21 = Druckernormalbetrieb  
 26 = 'BEEP' und 'TONE' an  
 27 = 'USER' ein  
 28 = Dezimalzeichen - Punkt  
 31 = TIME-Modul 'DMY'  
 40 = Anzeigeformat 'FIX'  
 55 = Drucker 'anwesend'

SYNTEXTZEILE  
 APPEND? J/N  
 N  
 MAX= 14 BYTE

BYTE 1           4 α  
 BYTE 2           57 9  
 BYTE 3           0 \*  
 BYTE 4           128 \*  
 BYTE 5           1 \*

Kein TIME-Modul bei Flag 31 ist schadlos. Die Bytes für die synthetische Barcodereihe zur Flagsteuerung Register d 'auf einen Schlag' heißen: 0 0 4 57 0 128 1. Wir wollen nun zur Praxis kommen. Fertigen Sie zuerst mit einem der Programme "SYNT.." eine Barcodereihe an, indem Sie die Bytes 4 57 0 128 1 eingeben. Führende Null-Bytes können wegelassen werden. In diesem Fall nimmt die Zeile mit dem Textbyte also nur 6 Zeichen in Anspruch.

Lesen Sie nun das Programm "FS" ein, welches lediglich die zwei synthetischen Befehle 'RCL M' und 'STO d' beinhaltet.

PROGRAMM: "FS" Benötigte Register: 2 -

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-4



Wichtig! Schalten Sie jetzt erst in den PRGM-Modus; Sie sehen '01 LBL 'FS''. Lesen Sie nun die eben gefertigte Barcodezeile ein. Sie sehen '02 'I9'█'. Schalten Sie in den RUN-Modus zurück. Wir wollen nun erst noch einen 'wilden' Flagstatus an Ihrem Rechner herstellen. Führen Sie dazu bitte aus: 'SF 00' bis 'SF 04' - 'USER'-Modus ausschalten, sodann eingeben 'RAD' 'CF 26' (es 'BEEP't nicht mehr) 'ENG 9' 'CF 28' 'SF 29' 'SF 12' 'SF 13' und ins Alpha-Register: "ABC". Anschließend führen Sie dann über das Tastenfeld aus 'XEQ PRA'. Es wird gedruckt: **abc**. Schalten Sie nun in den Alpha-Modus und drücken Sie zusätzlich die Shift-Taste. Im Alpharegister steht noch ABC und das Rechnerindikatorfeld zeigt 'RAD' 'SHIFT' '0' '1' '2' '3' '4' und 'ALPHA'.

Nun kommt's! Führen Sie mit dem Lesestift 'XEQ "FS"' aus.

XEQ "FS"



Im X-Register sehen Sie 00 - darunter den 'USER'-Indikator, den wir ausgeschaltet hatten. Drücken Sie zu Kontrolle 'BEEP' - es geht wieder! Der ALPHA-Modus ist aus, die Flags 00 bis 04 sind gelöscht. Drücken Sie 'CLX' und es erscheint



eine '0' im 'FIX 0'-Format. Sie können sich auch noch überzeugen, daß die Flags 12 und 13 gelöscht sind. Im Alpharegister sehen Sie die synthetische Textzeile zur Flag-Massensteuerung. - Bei dieser Aktion ist gleich noch eine Fähigkeit von Barcodes zutage gekommen. Es lassen sich 'XEQ'-Anweisungen oder Befehle bei 'AON' ausführen. Dies ist über das Tastenfeld natürlich nicht möglich. Lassen Sie Programm "FS" bitte noch im Rechner.

3.2 Flagsteuerung mit 'STOFLAG'

Während das vorhergehende Beispiel keine Module erforderte, ist für die Ausführung der Funktion 'STOFLAG' das XF-Modul oder ein HP-41CX notwendig. Normalerweise wird diese Funktion zusammen mit 'RCLFLAG' benutzt. Der vorhandene Status aller Flags 00-43 kann mit 'RCLFLAG' aufgerufen, mit 'STO\_\_' in ein Register abgespeichert und später mit 'RCL\_\_' und 'STOFLAG' wieder installiert werden. Hier handelt es sich nun darum, einen eigenen Flagstatus zu kreieren, der gleich mit 'STOFLAG' alle Flags 00-43 verändert. Da es sich nur um diese Flags handelt, werden hier für eine Flag-Steuerzeile lediglich 44 Bit oder 5 1/2 Bytes bewegt. Diese füllen kein Register. Tatsächlich gehen der synthetischen Bytefolge für die Funktion 'STOFLAG' 1 1/2 Bytes zur Erkennung und Steuerung voraus. Es sind dies das Byte 31 und, falls die Flags 00-03 nicht gesetzt sind, das Byte 240. Analog zu unserem Beispiel aus dem vorigen Abschnitt sähe die Bit/Bytefolge dann so aus:

31 \*  
240 P  
0 +  
0 +  
67 C  
144 8  
8 A

Dez	31	240	0	0	67	144	8
Bin	00011111	11110000	00000000	00000000	01000011	10010000	00001000
Flag		00-03	04-11	12-19	20-27	28-35	36-43

Gesetzt sind alle Flags aus dem vorigen Beispiel bis auf Flag 55, welches nicht erreicht wird und unverändert bleibt. Fertigen Sie nun erst wieder die Barcodereihe wie nebenstehend an. Im Gegensatz zur Speicherung aller Flags ins Register d, bei der führende Null-Bytes weggelassen werden können, müssen hier wegen 31 240... stets alle 7 Bytes ausgedruckt werden. Es ist dementsprechend also abzuwägen, welche Lösung man aus Bytesparsicht einsetzt. Die Dezimalfolge ist: 31 240 0 0 67 144 8. Je nach dem ob Flags 00 bis 03 gesetzt sind, verändert sich der Wert von 240 bis hin zu 255.

Löschen Sie jetzt die Zeile 02 mit der 'alten' synthetischen Textzeile aus dem Programm "FS" und geben Sie stattdessen die eben angefertigte BC-Reihe ein. Löschen Sie sodann die Zeile 04 'STO d' und geben an deren Stelle als neue Zeile 04 'STOFLAG' ein. Vergewissern Sie sich, daß die SYNTextzeile zwischen 'LBL "FS"' und 'RCL M' steht und dann 'STOFLAG' folgt; andernfalls kann es bei Programmausführung unerwünschte Komplikationen geben. Wenn Sie möchten, können Sie noch einmal den 'wilden' Flagstatus herstellen und dann das abgeänderte PRGM "FS" erneut starten. Im X-Register sehen Sie dann Bytefolgen, die auch in Datenregister abgespeichert und erneut, wie nach 'RCLFLAG', verwendet werden können. Führen Sie 'RCLFLAG' 'X=Y?' aus - Antwort: yes.

Sofern Sie noch nie mit synthetischer Programmierung gearbeitet haben, mögen die beiden Beispiele Ihr Interesse geweckt haben. Nun werden synthetische Textzeilen nicht nur für die Flagsteuerung benutzt. Im Rahmen dieses Buches werden noch öfter solche Zeilen auch für andere Zwecke gebraucht. Wie supereinfach synthetische Befehle mit Barcodes direkt in Programmregister eingegeben werden können, wird in Kapitel 4 vorgestellt. Im nächsten Abschnitt werden wiederum SYNTexte für Fertigung von Druckersonderzeichen eingesetzt.



### III. Druckersonderzeichen ☺

Jeder HP-IL-Modulbesitzer kennt dieses lustige Druckersonderzeichen aus dem Bedienungshandbuch. Es ist durch Akkumulation von neun 2-stelligen Spaltendruckzahlen im Druckerbuffer entstanden und dann gedruckt worden. In einem PRGM würde diese Akkumulation mit 'ACCOL' und Ausdruck mit 'PRBUF' 34 Bytes beanspruchen. Die Eingabe mit Barcodes direkt in den Druckerbuffer ist nicht möglich.

Nun können Druckersonderzeichen auch noch mit der Funktion 'BLDSPEC' erzeugt werden. Allerdings erfordert die Eingabe für ein sinnvolles Zeichen stets 7 SD-Zahlen. Die Bytefolge für das im X-Register aufgebaute Zeichen kann anschließend mit 'ACSPEC' in den Druckerbuffer gebracht und gedruckt werden. Ist das Sonderzeichen länger wie oben ☺, so kann das ganze Zeichen in 2 oder mehr Teile zerlegt und jeder Teil mit 'ACSPEC' in den Druckerbuffer eingegeben werden; anschließend wird gedruckt. Von den 9 SD-Zahlen des 'lustigen Lächlers' dürfen also mit 'BLDSPEC' zunächst 7 eingegeben und mit 'ACSPEC' in den Buffer gebracht werden, dann können die restlichen 2 - und notwendig müssen weitere 5 SD-Zahlen '0' getastet werden, damit auch der 2. Zeichenteil dann vollständig mit 'ACSPEC' geladen und das Zeichen gedruckt werden kann. Dieser Vorgang würde in einem Programm 79 Bytes verbrauchen! - und doch bleibt 'BLDSPEC' indirekt und 'ACSPEC' direkt überlegen: mit Barcodes und einer weiteren Art von synthetischen Textzeilen. Hiermit werden dann nur 25 Bytes verbraucht. Eine andere Möglichkeit für die Anfertigung von Sonderzeichen ist der ESCAPE-Grafikmodus, der später noch bei längeren Sonderzeichen behandelt wird.

BLDSPEC  
↓



#### 1. Aufbau, Typindikator und Prüfsumme

Diese Textzeilen für Sonderzeichen entsprechen in der Leitinformation den synthetischen Textzeilen aus dem vorhergehenden Abschnitt. Abweichend ist lediglich, es gibt keine Appenzzeichen.

#### 2. Herstellen von Druckersonderzeichen

Es folgen wieder drei Herstellungsprogramme. Weil die 'BLDSPEC'-Bytes im X-Register tatsächlich Textzeilen sind wie wir noch sehen werden, könnten solche Textzeilen-BC auch mit den PRGM zur Herstellung einer SYNTextzeile angefertigt werden. Zur Umrechnung der 'BLDSPEC'-Zahlen in die entsprechenden Bytedezimalwerte und umgekehrt, stehen noch zwei weitere Programme "BLDSZ" und "DZBLDS" zur Verfügung.

##### 2.1 Rechner HP 41 CV - Drucker, keine Module PRGM "BLDSBC"

###### Programmbedienung:

- SIZE: mindestens 015

XEQ "BLDSBC"



- "BLDSPEC 1?": Sie werden gefragt ob ein 'BLDSPEC'-Zeichen mit 1x oder 2x7 SD-Zahlen aufgebaut werden soll. Geben Sie jetzt erst einmal '1' ein und drücken 'R/S'. Rechtsbündig wird gedruckt 'BLDSPEC' und im Display steht die Frage:
- "SD-Zahl 1 ?": Geben Sie jetzt die erste Spaltendruckzahl ein. Z.B. 120 und drücken Sie dann 'R/S'.
- "SD-Zahl 2 ?". Geben Sie nun die 2. SD-Zahl ein: 96 und drücken Sie 'R/S' usw. bis alle 7 Spaltendruckzahlen eingegeben sind; jetzt noch: 80 72 7 6 und 4. Nach der 7. Eingabe erfolgt keine weitere Abfrage mehr, es wird das eingegebene Sonderzeichen und daran anschlie-

Bend die entsprechende Barcodereihe gedruckt. Das Ergebnis steht auf Seite 55. Sie haben damit nun die Barcodes für die synthetische Textzeile dieses -Druckersonderzeichens angefertigt. Die weitere Verarbeitung folgt anschließend im Abschnitt 3.

01*LBL "BLDSBC"	39 ISG Z	77 ST+ ]	115 +	153 GTO 00
02*LBL 00	40 GTO 01	78 CLX	116 INT	154*LBL 07
03 "a9**"	41 ACSPEC	79 X<> ]	117 FS?C 00	155 SF 25
04 RCL [	42 FS? 01	80 RTN	118 RTN	156 RCL IND X
05 STO d	43 ADV	81*LBL 04	119 SF 17	157 X=0?
06 CLRG	44 FS? 01	82 FS? 00	120 .	158 ACCOL
07 2	45 ADV	83 GTO 04	121 ACCOL	159 X=0?
08 "BLDSPEC"	46 CLA	84 1.007	122 FS? 01	160 GTO 09
09 ACA	47 STO [	85 FC? 02	123 ""	161*LBL 08
10 ADV	48 XEQ 03	86 GTO 04	124 FS? 02	162 XEQ 10
11 "f 1'2?'"	49*LBL 02	87 STO 00	125 ""	163 ARCL X
12 TONE 7	50 STO IND 00	88 SF 00	126 XEQ 09	164 RDN
13 PROMPT	51 XEQ 03	89 GTO a	127 RDN	165*LBL 09
14 FC? 22	52 ISG 00	90*LBL 04	128 XEQ 08	166 OUTA
15 GTO 00	53 GTO 02	91 FS? 00	129 FS?C 01	167 CLA
16 X>Y?	54 GTO 04	92 CF 01	130 "w"	168 RTN
17 GTO 00	55*LBL 03	93 14.007	131 FS? 02	169*LBL 10
18 SF IND X	56 "f+!****"	94 FS? 02	132 "z"	170 127
19 8.014	57 RCL [	95 INT	133 XEQ 09	171 X<>Y
20 STO 00	58 STO [	96 RCL IND X	134 14.007	172 X<=Y?
21*LBL a	59 CLX	97 DSE Y	135*LBL 06	173 GTO 11
22 1.007	60 X<> ]	98*LBL 05	136 XEQ 07	174 -
23 FS? 00	61 SIGN	99 RCL IND Y	137 RDN	175 1
24 SF 01	62 CLX	100 +	138 DSE X	176 +
25 FS? 00	63 X<> \	101 DSE Y	139 GTO 06	177 .
26 8.014	64 "f****"	102 GTO 05	140 FS? 02	178 1
27 .	65 X<> [	103 FS? 01	141 INT	179 BLDSPEC
28 ENTERf	66 X<> L	104 119	142 FS?C 02	180 X<>Y
29*LBL 01	67 X<> \	105 FS? 02	143 GTO 06	181 BLDSPEC
30 CF 22	68 INT	106 126	144 ADV	182 RTN
31 "SD-ZAHL "	69 ST+ ]	107 +	145 ADV	183*LBL 11
32 ARCL Z	70 RDN	108 SF 00	146 CF 17	184 .
33 "f ?"	71 6	109 XEQ 04	147 CLST	185 X<>Y
34 TONE 7	72 ST* ]	110*LBL 04	148 BEEP	186 BLDSPEC
35 PROMPT	73 RDN	111 RCL X	149 ADV	187 END
36 FC? 22	74 E1	112 256	150 ADV	300 Bytes
37 GTO 01	75 ST/ L	113 ST/ Z	151 ADV	
38 BLDSPEC	76 X<> L	114 MOD	152 RTN	

Wir wollen nun erst noch eine Barcodereihe für ein doppeltbreites 'BLDSPEC'-Zeichen herstellen. Drücken Sie einfach 'R/S'. Im Display steht wieder die Frage: "BLDSPEC 1'2?". Geben Sie nun 2 ein - 'R/S' und anschließend die folgenden SD-Zahlen: 2 1 1 62 1 1 62. Nach der 7. Zahl läuft das PRGM eine kurze Zeit und fragt dann: "SD-Zahl 8?". Geben Sie jetzt die erste Zahl der 2. Zeichenhälfte, dann die zweite usw. ein. Es sind dies: 1 1 62 64 80 48 und 112. Nach der 14. Zahl wird unter BLDSPEC das Tierkreiszeichen Skorpion  gedruckt, für das Sie gerade die synthetische Textzeile in Barcodes erstellt haben. Heben Sie bitte diese beiden Barcodestreifen für Abschnitt 3 auf.

Das Pfeilzeichen stammt übrigens aus dem schon zitierten 'Wickes'. Das Tierkreiszeichen Skorpion ist von Heinz Dalkowski, dem Herausgeber der deutschen Ausgabe des 'Dearing' ertüfelt. In diesem Buch finden Sie auch die übrigen 11 Zeichen mit Dezimalwerten, die wir in Abschnitt 3 besprechen werden.

2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "BLDSBc"

Programmbedienung: wie "BLDSBC"

- SIZE: mindestens 015

XEQ "BLDSBc"



01*LBL "BLDSBc"	30 CF 22	59 ST+ 00	88 +	117 1.014
02*LBL 00	31 "SD-ZAHL "	60 ATOX	89 SF 00	118*LBL 06
03 "a9+*"	32 ARCL Z	61 ISG 00	90 XEQ 04	119 XEQ 07
04 RCL [	33 "f ?"	62 GTO 03	91*LBL 04	120 RDN
05 STO d	34 TONE 7	63 FS? 00	92 RCL X	121 ISG X
06 CLRG	35 PROMPT	64 GTO 04	93 256	122 GTO 06
07 2	36 FC? 22	65 8.014	94 ST/ Z	123 ADV
08 "BLDSPEC"	37 GTO 02	66 FC? 02	95 MOD	124 ADV
09 ACA	38 BLDSPEC	67 GTO 04	96 +	125 CF 17
10 ADV	39 ISG Z	68 STO 00	97 INT	126 CLST
11 "f 1'2?'"	40 GTO 02	69 SF 00	98 FS?C 00	127 BEEP
12 TONE 7	41 ACSPEC	70 GTO 01	99 RTN	128 ADV
13 PROMPT	42 FS? 01	71*LBL 04	100 SF 17	129 ADV
14 FC? 22	43 ADV	72 FS? 00	101 .	130 ADV
15 GTO 00	44 FS? 01	73 CF 01	102 ACCOL	131 RTN
16 XYY?	45 ADV	74 1.007	103 FS? 01	132 GTO 00
17 GTO 00	46 CLA	75 FS? 02	104 **	133*LBL 07
18 SF IND X	47 STO [	76 1.014	105 FS? 02	134 SF 25
19 1.007	48 ATOX	77 RCL IND X	106 **	135 RCL IND X
20 STO 00	49*LBL 03	78 ISG Y	107 XEQ 09	136 X=0?
21*LBL 01	50 STO IND 00	79*LBL 05	108 RDN	137 ACCOL
22 1.007	51 7	80 RCL IND Y	109 XEQ 08	138 X=0?
23 FS? 00	52 FS? 00	81 +	110 FS?C 01	139 GTO 09
24 SF 01	53 14	82 ISG Y	111 "u"	140*LBL 08
25 FS? 00	54 RCL 00	83 GTO 05	112 FS? 02	141 XTOA
26 8.014	55 INT	84 FS? 01	113 "Σ"	142*LBL 09
27 .	56 -	85 119	114 XEQ 09	143 OUTA
28 ENTER†	57 ALENG	86 FS? 02	115 1.007	144 CLA
29*LBL 02	58 -	87 126	116 FS?C 02	145 END 307 Bytes

Während in anderen Programmen mit XF-Modul die PRGM-Systematik erhalten blieb, und die Zeilen 55-80 z.B. aus "BLDSBC" einfach durch '-l' 'AROT' 'ATOX' ersetzt werden konnten, ist dies hier nicht möglich. Die SYNT-Zeilen können Nullbytes enthalten, welche verlorengehen, sobald sie allein ins Alpharegister gelangen. Die Zeilen 49-62 des vorstehenden PRGM berücksichtigen dies, indem mit 'ALENG' die verbliebene Alphazeichenzahl festgestellt und als indirekte Speicherregisteradresse verwendet wird. Zeile 06 'CLRG' setzt zu Anfang alle Register = 0.

2.3 Rechner HP 41 C - Drucker, XF- und Plotter-Modul PRGM "BLDSbc"

Programmbedienung: wie "BLDSBC"

- SIZE: bei diesem Programm ist 'SIZE 016' erforderlich

XEQ "BLDSbc"



Das Programmlisting folgt auf Seite 58. - In allen "BLDS.."-Programmen wird in Zeile 18 in Abhängigkeit ob "BLDSPEC" 1 oder 2 eingegeben wurde, Flag 1 oder 2 für die Steuerung der Textzeilenlänge benutzt. Im Programm "BLDSbc" wird die Plottermodul-Funktion 'BCREGX' für den Aufbau der synthetischen Textzeile ein-

01*LBL "BLDSbc"	20 STO 00	39 ACSPEC	58 GTO 04	77 XTOA
02*LBL 00	21*LBL 01	40 CLA	59 9.015	78 RCL 15
03 "α9+*	22 1.007	41 STO I	60 FC? 02	79 XTOA
04 RCL I	23 FS? 00	42 ATOX	61 GTO 04	80 15
05 STO d	24 8.014	43*LBL 03	62 STO 00	81*LBL 05
06 CLRG	25 .	44 STO IND 00	63 SF 00	82 1
07 2	26 ENTER↑	45 8	64 GTO 01	83 +
08 "BLDSPEC"	27*LBL 02	46 FS? 00	65*LBL 04	84 BCCKSM
09 ACA	28 CF 22	47 15	66 119	85 CHS
10 ADV	29 "SD-ZAHL "	48 RCL 00	67 FS? 02	86 BCO
11 "† 1'2?"	30 ARCL Z	49 INT	68 126	87 ADV
12 TONE 7	31 "† ?"	50 -	69 STO 01	88 ADV
13 PROMPT	32 TONE 7	51 ALENG	70 8	89 ADV
14 FC? 22	33 PROMPT	52 -	71 FS? 02	90 CF 01
15 GTO 00	34 FC? 22	53 ST+ 00	72 13	91 CLST
16 X?Y?	35 GTO 02	54 ATOX	73 BCREGX	92 BEEP
17 GTO 00	36 BLDSPEC	55 ISG 00	74 FC?C 02	93 RTN
18 SF IND X	37 ISG Z	56 GTO 03	75 GTO 05	94 GTO 00
19 2.000	38 GTO 02	57 FS?C 00	76 RCL 14	95 END 203 Bytes

gesetzt. Dadurch hat Flag 1 keine Aufgabe und kann höchstens als Kennzeichnung betrachtet werden. Die Funktion 'BCREGX' streikt bei mehr als 13 Register unregelmäßig. Deswegen müssen die beiden letzten Bytes (Zeilen 76-79) einzeln geholt werden.

### 3. Arbeiten mit Druckersonderzeichen-Barcodes

Die beiden Barcodestreifen aus dem vorigen Abschnitt haben Sie noch? - gut. Falls Sie das kleine Programm "FS" noch im Rechner haben, löschen Sie es bitte.



Lesen Sie dann Programm "FS" in der Ursprungsversion aus Abschnitt II.d.3.1 noch einmal neu ein, schalten in den PRGM-Modus und lesen dann die BC-Reihe BLDSPEC ein. Löschen Sie Zeile 04 'STO d' und geben sodann ein 'ACSPEC' und 'PRBUF'. Schalten Sie in den RUN-Modus zurück und führen Sie 'XEQ "FS"' aus. Das Zeichen wird linksbündig gedruckt, und im X-Register sehen Sie die Bytefolge 'BLDSPEC', die mit 'ACSPEC' in den Druckerbuffer gebracht und dann gedruckt wurde. Führen Sie 'PRX' aus - Ergebnis: "cβσ\*Ca". Schalten Sie jetzt in den Alpha-Modus und drücken die PRINT-Taste des Druckers - das Ergebnis ist:  $\alpha c \beta \sigma * C \alpha$ . Es ist dies die synthetische Textzeile des Pfeilzeichens, Zeile 02 im jetzigen PRGM "FS". Wie Sie wissen, können in den Stack oder sonstige Speicherregister maximal 6 Alphazeichen gespeichert werden. Tatsächlich faßt jedoch jedes Register 7 Zeichen. Damit der Prozessor erkennen kann, es handelt sich um eine Alphakette und nicht um eine Zahl, wird ein Code benötigt. Dieser Code ist die Dezimalzahl 16 bzw. 17 welcher vom Rechner bei Benutzung der Funktionen 'ASTO\_\_' oder 'ARCL\_\_' automatisch gesetzt oder aufgehoben wird. Wir brachten jedoch die SYNTextzeile mit dem synthetischen 'RCL M' ins X-Register. Dadurch kam exakt die Bytefolge in X an, die für das Pfeilzeichen und 'ACSPEC' benötigt wird. Bei 'RCL M' wird kein zusätzlicher Code gesetzt, sondern das erste Zeichen der Textzeile im Alpharegister wird als Code benutzt. Ist dies 16 oder 17 werden die folgenden Bytes als ASCII-Zeichen angezeigt. Die Funktion 'ARCL\_\_' hebt das Codebyte auf. Die Funktion 'ASTO\_\_' setzt immer 16. Löschen Sie das Alpharegister und führen aus: 'ARCL X' 'ASTO X' 'ACSPEC' 'ADV' - Ergebnis:  $\alpha$  - ein Pfeil ohne Spitze; aus 17 wurde 16. Dies ist der Grund, warum alle Dezimalzahlen für SD 64 in Spalte 1 mit 'ASTO X' verarbeitet fehlerhaft sind.

Diese Zusammenhänge lassen erkennen, daß zwischen den SD-Zahlen für die Funktion 'BLDSPEC' und der dezimalen Bytefolge der SYNTextzeile eine Wechselbeziehung besteht. Folglich muß es möglich sein, die Zahlen jeweils umzurechnen. Tatsächlich geht dies auch. Die SD-Zahlen 120 96 80 72 7 6 4 entsprechen unserer SYNTextzeile 17 227 5 9 1 195 4. Der genaue Aufbau einer synthetischen Textzeile für ein Druckersonderzeichen ist in den beiden erwähnten Büchern von Wickes und Dearing ausführlich beschrieben.

Die beiden folgenden Umrechnungsprogramme rechnen die SD-Zahlen in Dezimalwerte für SYNTextzeilen (ohne Textbyte) mit "BLDSDZ" oder eine Dezimalwertfolge in die SD-Zahlen mit "DZBLDS" um. Je nach dem was einem geläufiger ist, kann so die eine oder andere Methode benutzt werden. Beide PRGM benötigen keine Module.

### 3.1 Umrechnungsprogramm: "BLDSDZ"

#### Programmbedienung:

- SIZE: mindestens 015 erforderlich

XEQ "BLDSDZ"



**-UMRECHNUNG-**  
BLDSPECzahl → Dezimalzahl

Zeichen: ⚡

- "BLDSPEC 1 ?": Geben Sie die erste Spaltendruckzahl für Ihr 'BLDSPEC'-Zeichen ein; z.B. 120 - 'R/S'
- "BLDSPEC 2 ?": Geben Sie dann bei dieser und jeder weiteren Abfrage die folgenden Zahlen ein: 96 80 72 7 6 4. Nach Eingabe jeweils 'R/S' drücken. Sie erhalten den nebenstehenden Ausdruck der Umrechnung der BLDSPEC-Zahlen in die entsprechenden Dezimalwerte für eine SYNTextzeile. Für die Anfertigung einer entsprechenden Barcode-Reihe haben Sie nun die Möglichkeit eines der Programme "BLDSBC" - "...bc" oder für die Herstellung einer SYNTTEXTZEILE zu benutzen. Wir werden in dem Abschnitt 'Längere Druckersonderzeichen' die BC-Vorteile noch ausschöpfen. Der Gestaltung von Druckersonderzeichen sind so keine Grenzen gesetzt. Die verbliebene Arbeit besteht lediglich darin, das Sonderzeichen zu 'schöpfen'.

BLDSPEC: 120	Dez: 17
BLDSPEC: 96	Dez: 227
BLDSPEC: 80	Dez: 5
BLDSPEC: 72	Dez: 9
BLDSPEC: 7	Dez: 1
BLDSPEC: 6	Dez: 195
BLDSPEC: 4	Dez: 4

01*LBL "BLDSDZ"	25 PROMPT	49 1.007	73 X(> L	97 CLST
02*LBL 00	26 FC? 22	50 14.007	74 ST+ ]	98 RTN
03 "α9+""	27 GTO 01	51 GTO 04	75 CLX	99 GTO 00
04 RCL [	28 STO IND 00	52*LBL 03	76 X(> ]	100*LBL 05
05 STO d	29 BLDSPEC	53 "†+↓****"	77 RTN	101 X>Y?
06 CLRG	30 ISG 00	54 RCL [	78*LBL 04	102 GTO 05
07 SF 12	31 "+"	55 STO [	79 "BLDSPEC: "	103 "† "
08 "-UMRECHNUNG-"	32 ISG 2	56 CLX	80 9	104 GTO 06
09 PRA	33 GTO 01	57 X(> ]	81 RCL IND Z	105*LBL 05
10 CF 12	34 "Zeichen: "	58 SIGN	82 XEQ 05	106 X(>Y
11 "BLDSPECzahl → De"	35 ACA	59 CLX	83 ACA	107 RDN
12 "fzinalzahl"	36 ACSPEC	60 X(> \	84 FMT	108 99
13 PRA	37 PRBUF	61 "†****"	85 "Dez: "	109 X(>Y
14 ADV	38 ADV	62 X(> [	86 9	110 X>Y?
15 1.007	39 CLA	63 X(> L	87 RCL IND Y	111 GTO 06
16 STO 00	40 STO [	64 X(> \	88 XEQ 05	112 "† "
17 .	41 8.014	65 INT	89 ACA	113*LBL 06
18 ENTER†	42 STO 00	66 ST+ ]	90 ADV	114 ARCL X
19*LBL 01	43 XEQ 03	67 RDN	91 ISG Y	115 RDN
20 CF 22	44*LBL 02	68 6	92 "+"	116 RDN
21 "BLDSPEC "	45 STO IND 00	69 ST* ]	93 DSE Y	117 END
22 ARCL Z	46 XEQ 03	70 RDN	94 GTO 04	306 Bytes
23 "† ?"	47 ISG 00	71 E1	95 ADV	
24 TONE 7	48 GTO 02	72 ST* L	96 CLA	

### 3.2 Umrechnungsprogramm: DZBLDS"

#### Programmbedienung:

- SIZE: mindestens 016 nötig

XEQ "DZBLDS"



- "DEZ= 1 ?": Geben Sie nun in Umkehrung der Bedienung des Programms "BLDXDZ" die entsprechenden Dezimalwerte für die Umrechnung in BLDSPEC-Zahlen ein. Für die DEZ-Werte 17 227 5 9 1 195 und 4 erhalten Sie dann den nebenstehenden Ausdruck der Umrechnungszahlen für BLDSPEC-Werte.

#### -UMRECHNUNG-

Dezimalzahl+ BLDSPECzahl

Zeichen: \*

Dez: 17	BLDSPEC: 120
Dez: 227	BLDSPEC: 96
Dez: 5	BLDSPEC: 80
Dez: 9	BLDSPEC: 72
Dez: 1	BLDSPEC: 7
Dez: 195	BLDSPEC: 6
Dez: 4	BLDSPEC: 4

01+LBL "DZBLDS"	29 STO IND Y	57 X<>Y	85 1.007	113 MOD
02+LBL 00	30 RDN	58 INT	86 8.014	114 RCL 15
03 2	31 ISG X	59 ST+ 11	87+LBL 03	115 *
04 STO 00	32 GTO 01	60 RCL 06	88 "Dez: "	116 2
05 64	33 RCL 01	61 XEQ 15	89 9	117 ST* 00
06 STO 15	34 XEQ 15	62 STO 13	90 RCL IND Z	118 ST/ 15
07 CF 13	35 STO 08	63 X<>Y	91 XEQ 04	119 RDN
08 SF 12	36 RCL 02	64 INT	92 ACA	120 RTN
09 "-UMRECHNUNG-"	37 XEQ 15	65 ST+ 12	93 FMT	121+LBL 04
10 PRA	38 STO 09	66 RCL 07	94 "BLDSPEC: "	122 X<>Y?
11 CF 12	39 X<>Y	67 XEQ 15	95 9	123 GTO 04
12 "Dezimalzahl+ BL"	40 INT	68 STO 14	96 RCL IND Y	124 "+ "
13 "+BLDSPECzahl"	41 ST+ 08	69 X<>Y	97 XEQ 04	125 GTO 05
14 PRA	42 RCL 03	70 INT	98 ACA	126+LBL 04
15 ADV	43 XEQ 15	71 ST+ 13	99 ADV	127 X<>Y
16 SF 28	44 STO 10	72 "Zeichen: "	100 ISG Y	128 RTN
17 CF 29	45 X<>Y	73 ACA	101 "+	129 99
18 FIX 0	46 INT	74 8.014	102 ISG X	130 X<>Y
19 1.007	47 ST+ 09	75 RCL X	103 GTO 03	131 X<>Y?
20+LBL 01	48 RCL 04	76 .	104 ADV	132 GTO 05
21 CF 22	49 XEQ 15	77+LBL 02	105 CLA	133 "+ "
22 "DEZ= "	50 STO 11	78 RCL IND Z	106 CLST	134+LBL 05
23 ARCL X	51 X<>Y	79 BLDSPEC	107 RTN	135 ARCL X
24 "+ ?"	52 INT	80 ISG Z	108 GTO 00	136 RDN
25 TONE 7	53 ST+ 10	81 GTO 02	109+LBL 15	137 RDN
26 PROMPT	54 RCL 05	82 ACSPEC	110 RCL X	138 END
27 FC? 22	55 XEQ 15	83 PRBUF	111 RCL 00	308 Bytes
28 GTO 01	56 STO 12	84 ADV	112 ST/ Z	

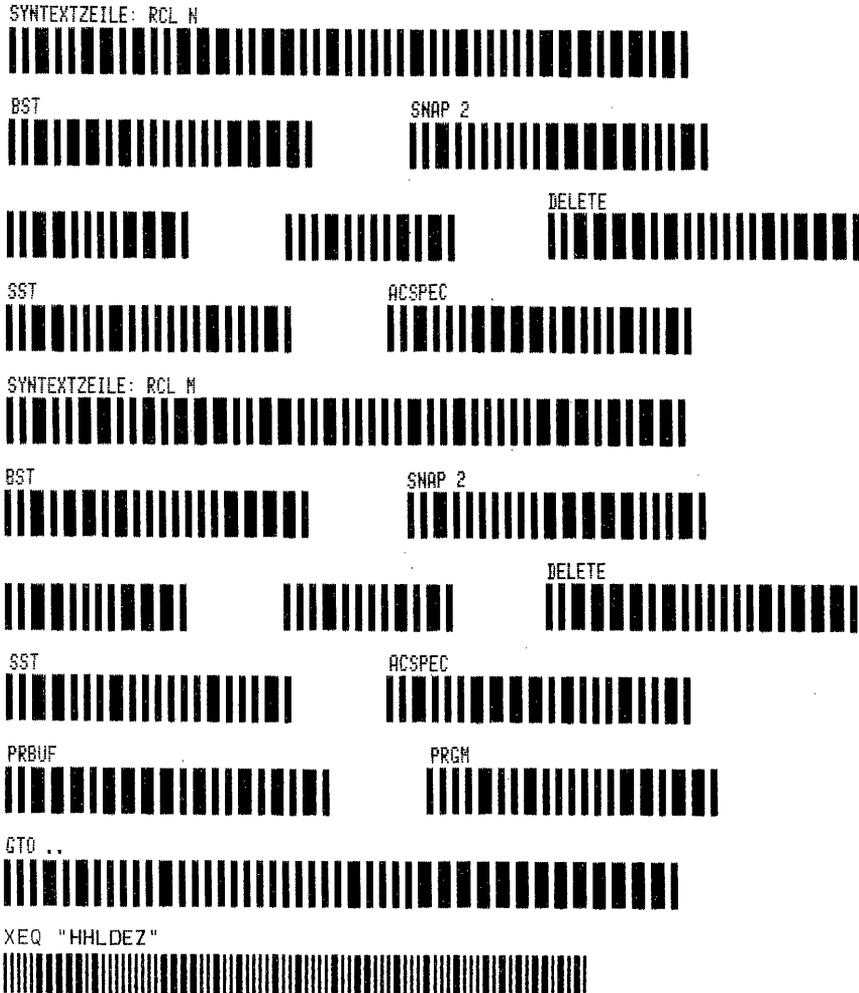
### 3.3 Längere Drucker Sonderzeichen

Der Druckerbuffer hat eine Kapazität von 101 ASCII-Zeichen. An normalbreiten Drucker Sonderzeichen nimmt der Buffer 14 auf, sodann wird gedruckt; doppeltbreit werden 15 Zeichen gespeichert, nach 12 Zeichen wird gedruckt, 3 verbleiben noch im Buffer. Für ganzzeilige, geschlossene Grafik ist deswegen die doppeltbreite Gestaltung angezeigt. Bei entsprechend ganzzeiliger Normalbreitgrafik sind Leerstellen erforderlich: . Es stehen zwei Druckermodi mit unterschiedlicher Arbeitsweise zur Verfügung, die kombiniert werden können: der 8-Bit-Modus und der Grafik-ESCAPE-Modus.

#### - Der 8-Bit-Modus

Längere Drucker Sonderzeichen entstehen in diesem Modus durch Akkumulation von 2 oder mehr 'BLDSPEC'-Einzelteilen. Für ein Sonderzeichen mit doppeltbrei-

tem 'BLDSPEC'-Aufbau kann eines der Programme "BLDS.." benutzt werden. Löschen Sie nun das PRGM "FS" und nehmen Sie den Barcodestreifen des Skorpionzeichens zur Hand. Schalten Sie in den PRGM-Modus und tasten Sie ein 'LBL "HHLDEZ"' und 'SF 12'; sodann lesen Sie die Skorpion-Barcodezeile ein. Nun tasten Sie mit dem Lesestift der Reihe nach die folgenden Barcodes einzeln ab:



Sie haben gerade mit Barcodes außer einigen anderen auch die beiden synthetischen Befehle 'RCL N' und 'RCL M' eingegeben. Von den 14 dezimalen ASCII-Zeichen der Skorpionbarcodes brachte der Befehl 'RCL N' die ersten 7 Zeichen ins X-Register, 'ACSPEC' lud sie als erste Hälfte in den Buffer, 'RCL M' brachte die zweite Zeichenhälfte nach X und 'ACSPEC' in den Buffer. 'PRBUF' druckte das Zeichen  $\mathfrak{M}$  linksbündig aus. Im Y-Register steht die 1. und im X-Register die 2. Hälfte der insgesamt 14 'BLDSPEC'-Zeichen, von denen jedoch nur jeweils 6 sichtbar sind. Die 14 ASCII-Zeichen der synthetischen Textzeile 03 stehen im Alpharegister. - Lesen Sie nun zum Zweck einer kleinen Programmweiterung die auf Seite 62 folgenden Barcodes wieder von links nach rechts einzeln ein. Diese Zusammenstellung zeigt Ihnen gleichzeitig, wie schnell mit BC gearbeitet wird.

Sie werden gewiß bemerkt haben, beim Einlesen muß man auf das Scrolling warten.



```

01*LBL "HHLDEZ"
02 SF 12
03 "0000" @ 000000
04 RCL I
05 ACSPEC
06 RCL I
07 ACSPEC
08 GTO "HHLDEZ"
09 PRBUF
10 END
    
```

Das Programm sieht nun wie nebenstehend aus. Es wurde mit den 'R/S'-Barcodes gestartet. Anhalten können Sie es nur, indem Sie die R/S-Taste am Rechner drücken. Mit den 'R/S'-Barcodes kann ein laufendes Programm nicht gestoppt werden. Lesen Sie während des Programmlaufs die 'R/S'-Barcodes erneut ein, so wird ein weiterer Startbefehl gespeichert. Hält das PRGM danach bei einem 'PROMPT'-Befehl an oder drücken Sie während des Programmlaufs die Stop-Taste, startet es sofort neu.

Während das Programm nun läuft oder lief, wurde ohne Druckerbefehl, 'GTO "HHLDEZ"' steht vor 'PRBUF', automatisch eine Zeile nach der anderen gedruckt. Im Doppeltbreit-Modus, Flag 12 ist gesetzt, sind pro Zeile genau 6 Doppelzeichen oder 12 Einzelzeichen möglich. Reiht man die Dezimalwerte der SD-Zahlen ununterbrochen aneinander, so kann eine ganze Druckzeile mit 6 synthetischen Barcodereihen von je 14 Dezimalcodes innerhalb eines Programms gestaltet werden. Die SYNTTextzeilen können entweder mit einem der PRGM "BLDS.."-BarCodes oder bei bekannten Dezimalwerten auch mit "SYNT.."-BarCodes angefertigt und eingelesen werden. Nötigenfalls können Sie die Umrechnungsprogramme benutzen. Ein Load-Bytes-Programm für den dezimalen Aufbau der SYNTTextzeilen wird nicht benötigt. Das Textbyte muß bei Einsatz von Barcodes nicht weiter beachtet werden; alle Dezimalcodezeilen verzichten deswegen auf eine Textbyteangabe.

Wir wollen jetzt die Möglichkeiten für längere Druckersonderzeichenbarcodes an 3 Beispielen und auch im ESCAPE-Grafikmodus weiter ergründen. Zunächst verbleiben wir noch im 8-Bit-Modus. Betrachten Sie die Herz-, Herz-, Lächlerzeile, welche mit DEZimalcodes in dem Programm "HHLDEZ" hergestellt wurde. Von dem Listing dieses neuen Programms haben Sie die Zeilen 01, 02 und 21-24 ja bereits im Rechner. Fertigen Sie nun erst die 6 synthetischen Barcodesreihen an.

01*LBL "HHLDEZ"	05 "00200 #01+co>>"	10 XEQ 00	16 CLA	22 ACSPEC
02 SF 12	06 XEQ 00	11 "00 000000"	17 CF 12	23 RCL I
03 "0000"	07 "000 >000+00"	12 XEQ 00	18 ADV	24 ACSPEC
04 XEQ 00	08 XEQ 00	13 "0+0+0+0"	19 RTH	25 END
	09 "0=0"	14 XEQ 00	20*LBL 00	138 Bytes
	+ 0 "	15 CLST	21 RCL \	

Dezimalwerte der Zeilen

- 03 = 16, 56, 251, 239, 143, 143, 142, 16, 56, 251, 239, 143, 143, 142
- 05 = 16, 64, 218, 56, 16, 32, 35, 16, 108, 128, 227, 239, 190, 62
- 07 = 16, 124, 112, 227, 239, 190, 62, 16, 124, 113, 3, 104, 224, 64
- 09 = 17, 1, 25, 178, 3, 143, 190, 17, 241, 241, 241, 195, 143, 190
- 11 = 17, 241, 241, 241, 196, 13, 163, 17, 2, 4, 4, 102, 200, 14
- 13 = 16, 125, 247, 199, 199, 199, 14, 16, 125, 247, 199, 199, 0

Löschen Sie die nicht benötigten Teile aus dem im Rechner vorhandenen Programm "HHLDEZ". Geben Sie dann die angefertigten SYNTTextbarcodes mit dem Lesestift, und die übrigen Befehle über das Tastenfeld ein. Ist alles ok? - Dann können Sie das Programm mit den BC 'XEQ "HHLDEZ"' Seite 61 oder 'GTO "HHLDEZ"' 'R/S' Seite 62 starten.

Erläuterungen zum Programm: Die Zeile 05 bewirkt beim Ausdruck eines Listings ein Umschalten in den Spaltenmodus. Es wird dann nicht weiter gelistet, sondern es werden Barcodes gedruckt. Die Zeile 05 im Listing auf Seite 62 ist im Alpha-Register 'nachgemacht'. Ab Zeile 06 wurde weiterge'LIST'et. Der STACK-Inhalt geht bei dieser Programmiermethode verloren. In den 8-Bit-Modus gelangt der Drucker nach dem Einschalten durch den Befehl 'ACSPEC' Zeile 22. Wir verlassen nun den 8-Bit-Modus und drucken die genau gleichen 'HHL'- Sonderzeichen im

- ESCAPE-Grafik-Modus

Der wesentliche Unterschied zum 8-Bit-Modus und der Akkumulation von 'BLDSPEC'- Zeichen im Druckerbuffer besteht darin, daß ähnlich wie mit der 'ACCOL'-Funktion eine beliebige Anzahl SD-Zahlen bis zum automatischen Bufferausdruck akkumuliert werden kann. Es sind dies hier keine Dezimalcodes, sondern die SD-Zahlen, deren Werte ja immer zwischen 0 und 127 liegen. Betrachten Sie das nachfolgende Listing, die Steuerzeile 05 und die 'BLDSPEC'-SD-Zahlen von 'HHELEC':

01*LBL "HHELEC"	06 OUTA	11 "1#000#00rE"	16 "<>#r#><>#r"	21 CF 12
02 SF 12	07 "r#><>#r#><>#r#"	12 OUTA	17 OUTA	22 END
03 FMT	08 "1#000#00rE"	13 "><>#r#><>#r#E#"	18 CF 17	136 Bytes
04 SF 17	09 OUTA	14 "1#000#00rE">	19 ADV	
05 "1#000#00rE") >11 "	10 "#><>#r#><>#r#E"	15 OUTA	20 CLA	

Dezimalwerte der Zeilen

Steuerzeile 05

253, 27, 42, 98, 56, 51, 71 (253 ist kein Textbyte, sondern nach 'FMT' in Zeile 03 der Steuercode für das Umschalten 8-Bit/ESCAPE-Modus)

'BLDSPEC'- SD-Zahlen der Zeilen

- 07 = 14, 31, 62, 124, 62, 31, 14, 14, 31, 62, 124, 62, 31, 14, 16
- 08 = Append 27, 35, 64, 64, 35, 27, 16, 14
- 10 = 31, 62, 124, 62, 31, 14, 14, 31, 62, 124, 62, 31, 14, 16, 27
- 11 = APPEND 35, 64, 64, 64, 35, 27, 16, 14, 31
- 13 = 62, 124, 62, 31, 14, 14, 31, 62, 124, 62, 31, 14, 16, 27, 35
- 14 = APPEND 64, 64, 64, 35, 27, 16, 14, 31, 62
- 16 = 124, 62, 31, 14, 14, 31, 62, 124, 62, 31, 14

Nur die Steuerzeile 05 erfordert wegen des Byte 253 eine SYNTText-BC-Reihe mit einem der PRGM "SYNT.."-BC. Alle anderen Zeilen können mit den 'Zweibyte- Funktions- und Druckerzeichen 0-127' mit dem Lesestift eingegeben werden. Es ist dabei natürlich möglich, tastbare Zeichen wie 62 = 'SHIFT TAN' über das Tastenfeld einzutasten. Sie haben bemerkt, wie hier im ESCAPE-Grafik-Modus die vollen 15 ASCII-Zeichen pro PRGM-Zeile und anschließend mit 'APPEND' alle 24 Zeichen des Alpharegisters vor einem 'OUTA' genutzt werden können. Dies Programm läuft schneller.

Erläuterungen zum Programm: Die Programmzeile 05 verursacht diesen chaotischen Zeilendruck und während des Listings einen Sprung auf Zeile 11; ab Zeile 06 ist deswegen mit 'LIST' fortgesetzt worden. Der STACK-Inhalt bleibt bei dieser Programmierversion unberührt. Zeile 03 'FMT' begibt den Drucker nach dem Einschalten in den 8-Bit-Modus, damit Zeile 05 anschließend ordnungsgemäß funktioniert. Ein großer Nachteil des ESCAPE-Grafik-Modus ist scheinbar, daß entgegen der Darstellung im Druckerhandbuch keine Nullen verarbeitet werden. Aber das ficht uns nicht an. Wir kombinieren beide Modi im:



Programm: "A127BC" - Benoetigte PRGM-Reg: 30

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 20-27



Reihe 6: Zeilen 27-35



Reihe 7: Zeilen 35-43



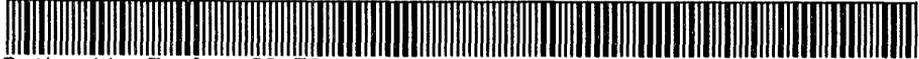
Reihe 8: Zeilen 44-52



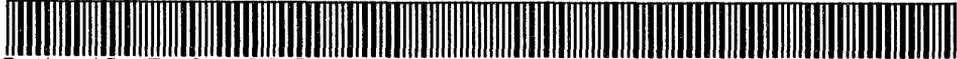
Reihe 9: Zeilen 53-60



Reihe 10: Zeilen 61-68



Reihe 11: Zeilen 69-78



Reihe 12: Zeilen 78-86



Reihe 13: Zeilen 87-96



Reihe 14: Zeilen 96-105



Reihe 15: Zeilen 106-109



Reihe 16: Zeilen 109-114



Reihe 17: Zeilen 114-114



Programm: "A127Bc" - Benoetigte PRGM-Reg: 25

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 27-34



Reihe 7: Zeilen 35-43



Reihe 8: Zeilen 44-53



Reihe 9: Zeilen 53-59



Reihe 10: Zeilen 60-69



Reihe 11: Zeilen 70-78



Reihe 12: Zeilen 79-82



Reihe 13: Zeilen 82-87



Reihe 14: Zeilen 87-87



Programm: "A127bc" - Benoetigte PRGM-Reg: 17

Reihe 1: Zeilen 1-3



Programm: "A127bc" - Fortsetzung

Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 27-34



Reihe 7: Zeilen 35-43



Reihe 8: Zeilen 44-47



Reihe 9: Zeilen 47-52



Reihe 10: Zeilen 52-52



Programm: "127Pbc" - Benoetigte PRGM-Reg: 76

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-6



Reihe 3: Zeilen 6-11



Reihe 4: Zeilen 12-15



Reihe 5: Zeilen 15-23



Reihe 6: Zeilen 23-31



Reihe 7: Zeilen 31-35



Programm: "127Pbc" - Fortsetzung

Reihe 8: Zeilen 35-40



Reihe 9: Zeilen 40-45



Reihe 10: Zeilen 45-46



Reihe 11: Zeilen 46-54



Reihe 12: Zeilen 55-59



Reihe 13: Zeilen 60-64



Reihe 14: Zeilen 64-67



Reihe 15: Zeilen 67-70



Reihe 16: Zeilen 71-76



Reihe 17: Zeilen 77-81



Reihe 18: Zeilen 81-85



Reihe 19: Zeilen 85-86



Reihe 20: Zeilen 86-94



Reihe 21: Zeilen 94-94



Reihe 22: Zeilen 94-95



Reihe 23: Zeilen 96-97



Reihe 24: Zeilen 97-98



Reihe 25: Zeilen 98-103



Programm: "127Pbc" - Fortsetzung

Reihe 26: Zeilen 103-106



Reihe 27: Zeilen 107-108



Reihe 28: Zeilen 109-116



Reihe 29: Zeilen 117-125



Reihe 30: Zeilen 126-133



Reihe 31: Zeilen 134-142



Reihe 32: Zeilen 142-148



Reihe 33: Zeilen 148-155



Reihe 34: Zeilen 155-161



Reihe 35: Zeilen 162-165



Reihe 36: Zeilen 165-170



Reihe 37: Zeilen 170-178



Reihe 38: Zeilen 178-186



Reihe 39: Zeilen 187-190



Reihe 40: Zeilen 191-198



Reihe 41: Zeilen 199-203



Programm: "A255BC" - Benoetigte PRGM-Reg: 36

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 10-10



Reihe 4: Zeilen 11-19



Reihe 5: Zeilen 20-23



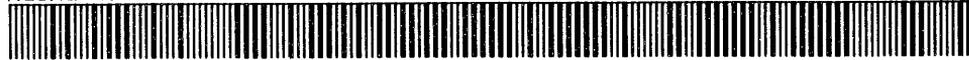
Reihe 6: Zeilen 23-30



Reihe 7: Zeilen 30-36



Reihe 8: Zeilen 36-42



Reihe 9: Zeilen 42-50



Reihe 10: Zeilen 50-56



Reihe 11: Zeilen 56-64



Reihe 12: Zeilen 64-70



Reihe 13: Zeilen 70-77



Reihe 14: Zeilen 77-87



Reihe 15: Zeilen 87-95



Reihe 16: Zeilen 95-104



Reihe 17: Zeilen 105-113



Programm: "A255BC" - Fortsetzung

Reihe 18: Zeilen 114-115



Reihe 19: Zeilen 115-123



Reihe 20: Zeilen 124-124



Programm: "A255Bc" - Benoetigte PRGM-Reg: 30

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 10-10



Reihe 4: Zeilen 11-19



Reihe 5: Zeilen 19-20



Reihe 6: Zeilen 21-29



Reihe 7: Zeilen 29-35



Reihe 8: Zeilen 35-42



Reihe 9: Zeilen 42-48



Reihe 10: Zeilen 48-56



Reihe 11: Zeilen 56-62



Reihe 12: Zeilen 62-69



Reihe 13: Zeilen 69-79



Programm: "A255Bc" - Fortsetzung

Reihe 14: Zeilen 79-88



Reihe 15: Zeilen 89-91



Reihe 16: Zeilen 91-96



Programm: "A255bc" - Benoetigte PRGM-Reg: 22

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 10-10



Reihe 4: Zeilen 11-19



Reihe 5: Zeilen 19-20



Reihe 6: Zeilen 21-29



Reihe 7: Zeilen 29-35



Reihe 8: Zeilen 35-41



Reihe 9: Zeilen 42-48



Reihe 10: Zeilen 48-57



Reihe 11: Zeilen 58-59



Reihe 12: Zeilen 59-64



Programm: "255Pbc" - Benoetigte PRGM-Reg: 79

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-6



Reihe 3: Zeilen 6-11



Reihe 4: Zeilen 12-15



Reihe 5: Zeilen 15-23



Reihe 6: Zeilen 23-31



Reihe 7: Zeilen 31-35



Reihe 8: Zeilen 35-40



Reihe 9: Zeilen 40-45



Reihe 10: Zeilen 46-46



Reihe 11: Zeilen 47-55



Reihe 12: Zeilen 56-60



Reihe 13: Zeilen 60-64



Reihe 14: Zeilen 64-67



Reihe 15: Zeilen 67-71



Reihe 16: Zeilen 71-77



Reihe 17: Zeilen 77-81



Programm: "255Pbc" - Fortsetzung

Reihe 18: Zeilen 81-85



Reihe 19: Zeilen 85-86



Reihe 20: Zeilen 87-94



Reihe 21: Zeilen 94-94



Reihe 22: Zeilen 94-96



Reihe 23: Zeilen 97-97



Reihe 24: Zeilen 97-98



Reihe 25: Zeilen 99-103



Reihe 26: Zeilen 103-107



Reihe 27: Zeilen 107-109



Reihe 28: Zeilen 109-115



Reihe 29: Zeilen 115-120



Reihe 30: Zeilen 120-125



Reihe 31: Zeilen 125-131



Reihe 32: Zeilen 132-136



Reihe 33: Zeilen 136-141



Reihe 34: Zeilen 141-149



Reihe 35: Zeilen 150-160



Programm: "255Pbc" - Fortsetzung

Reihe 36: Zeilen 161-167



Reihe 37: Zeilen 168-176



Reihe 38: Zeilen 176-183



Reihe 39: Zeilen 184-193



Reihe 40: Zeilen 193-199



Reihe 41: Zeilen 199-205



Reihe 42: Zeilen 205-212



Reihe 43: Zeilen 213-215



Programm: "TEXTBC" - Benoetigte PRGM-Reg: 55

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 11-11



Reihe 5: Zeilen 11-13



Reihe 6: Zeilen 13-19



Reihe 7: Zeilen 19-23



Reihe 8: Zeilen 24-29



Programm: "TEXTBC" - Fortsetzung

Reihe 9: Zeilen 30-37



Reihe 10: Zeilen 38-41



Reihe 11: Zeilen 42-48



Reihe 12: Zeilen 49-56



Reihe 13: Zeilen 56-60



Reihe 14: Zeilen 60-68



Reihe 15: Zeilen 68-73



Reihe 16: Zeilen 74-81



Reihe 17: Zeilen 82-90



Reihe 18: Zeilen 91-97



Reihe 19: Zeilen 98-105



Reihe 20: Zeilen 106-114



Reihe 21: Zeilen 115-121



Reihe 22: Zeilen 121-128



Reihe 23: Zeilen 129-138



Reihe 24: Zeilen 139-147



Reihe 25: Zeilen 148-157



Reihe 26: Zeilen 157-166



Programm: "TEXTBC" - Fortsetzung

Reihe 27: Zeilen 167-168



Reihe 28: Zeilen 168-174



Reihe 29: Zeilen 174-177



Reihe 30: Zeilen 178-179



Programm: "TEXTBc" - Benoetigte PRGM-Reg: 44

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-10



Reihe 3: Zeilen 11-12



Reihe 4: Zeilen 12-15



Reihe 5: Zeilen 15-16



Reihe 6: Zeilen 16-20



Reihe 7: Zeilen 20-23



Reihe 8: Zeilen 23-31



Reihe 9: Zeilen 31-37



Reihe 10: Zeilen 37-43



Reihe 11: Zeilen 44-50



Reihe 12: Zeilen 51-59



Programm: "TEXTbc" - Fortsetzung

Reihe 13: Zeilen 59-66



Reihe 14: Zeilen 67-73



Reihe 15: Zeilen 74-81



Reihe 16: Zeilen 82-90



Reihe 17: Zeilen 91-97



Reihe 18: Zeilen 98-105



Reihe 19: Zeilen 106-116



Reihe 20: Zeilen 117-126



Reihe 21: Zeilen 126-128



Reihe 22: Zeilen 128-132



Reihe 23: Zeilen 132-136



Reihe 24: Zeilen 137-137



Programm: "TEXTbc" - Benoetigte PRGM-Reg: 29

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 11-11



Programm: "TEXTbc" - Fortsetzung

Reihe 5: Zeilen 11-13



Reihe 6: Zeilen 13-15



Reihe 7: Zeilen 15-23



Reihe 8: Zeilen 23-30



Reihe 9: Zeilen 31-36



Reihe 10: Zeilen 37-43



Reihe 11: Zeilen 44-51



Reihe 12: Zeilen 52-62



Reihe 13: Zeilen 62-64



Reihe 14: Zeilen 65-70



Reihe 15: Zeilen 70-72



Reihe 16: Zeilen 73-75



Programm: "TXTJbc" - Benoetigte PRGM-Reg: 151

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-2



Reihe 3: Zeilen 3-3



Reihe 4: Zeilen 3-4



Programm: "TXTJbc" - Fortsetzung

Reihe 5: Zeilen 4-5



Reihe 6: Zeilen 5-11



Reihe 7: Zeilen 11-18



Reihe 8: Zeilen 19-22



Reihe 9: Zeilen 22-25



Reihe 10: Zeilen 26-27



Reihe 11: Zeilen 27-29



Reihe 12: Zeilen 30-32



Reihe 13: Zeilen 32-38



Reihe 14: Zeilen 38-41



Reihe 15: Zeilen 41-42



Reihe 16: Zeilen 43-46



Reihe 17: Zeilen 46-46



Reihe 18: Zeilen 47-52



Reihe 19: Zeilen 52-55



Reihe 20: Zeilen 55-58



Reihe 21: Zeilen 58-60



Reihe 22: Zeilen 60-68



Programm: "TXTJBe" - Fortsetzung

Reihe 23: Zeilen 69-73



Reihe 24: Zeilen 73-75



Reihe 25: Zeilen 76-78



Reihe 26: Zeilen 78-82



Reihe 27: Zeilen 83-89



Reihe 28: Zeilen 89-95



Reihe 29: Zeilen 96-103



Reihe 30: Zeilen 104-112



Reihe 31: Zeilen 112-119



Reihe 32: Zeilen 120-125



Reihe 33: Zeilen 126-134



Reihe 34: Zeilen 135-141



Reihe 35: Zeilen 142-150



Reihe 36: Zeilen 151-157



Reihe 37: Zeilen 158-162



Reihe 38: Zeilen 162-166



Reihe 39: Zeilen 166-168



Reihe 40: Zeilen 168-174



Programm: "TXTJBc" - Fortsetzung

Reihe 41: Zeilen 175-179



Reihe 42: Zeilen 180-187



Reihe 43: Zeilen 187-192



Reihe 44: Zeilen 193-201



Reihe 45: Zeilen 201-205



Reihe 46: Zeilen 206-211



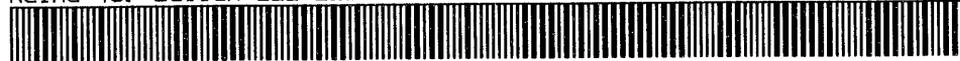
Reihe 47: Zeilen 212-216



Reihe 48: Zeilen 216-221



Reihe 49: Zeilen 222-226



Reihe 50: Zeilen 226-231



Reihe 51: Zeilen 232-235



Reihe 52: Zeilen 236-240



Reihe 53: Zeilen 241-245



Reihe 54: Zeilen 246-250



Reihe 55: Zeilen 251-255



Reihe 56: Zeilen 255-260



Reihe 57: Zeilen 260-265



Reihe 58: Zeilen 265-269



Programm: "TXTJBc" - Fortsetzung

Reihe 59: Zeilen 270-275



Reihe 60: Zeilen 275-279



Reihe 61: Zeilen 279-285



Reihe 62: Zeilen 285-289



Reihe 63: Zeilen 289-295



Reihe 64: Zeilen 295-299



Reihe 65: Zeilen 299-305



Reihe 66: Zeilen 305-309



Reihe 67: Zeilen 309-314



Reihe 68: Zeilen 315-319



Reihe 69: Zeilen 319-323



Reihe 70: Zeilen 324-328



Reihe 71: Zeilen 329-333



Reihe 72: Zeilen 333-337



Reihe 73: Zeilen 338-342



Reihe 74: Zeilen 342-346



Reihe 75: Zeilen 346-350



Reihe 76: Zeilen 351-355



Programm: "TXTJBC" - Fortsetzung

Reihe 77: Zeilen 355-359



Reihe 78: Zeilen 359-363



Reihe 79: Zeilen 363-368



Reihe 80: Zeilen 368-369



Reihe 81: Zeilen 369-374



Programm: "SYNTBC" - Benoetigte PRGM-Reg: 44

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 11-16



Reihe 5: Zeilen 16-20



Reihe 6: Zeilen 21-26



Reihe 7: Zeilen 27-32



Reihe 8: Zeilen 32-36



Reihe 9: Zeilen 36-42



Reihe 10: Zeilen 42-49



Reihe 11: Zeilen 49-57



Programm: "SYNTBC" - Fortsetzung

Reihe 12: Zeilen 58-65



Reihe 13: Zeilen 65-73



Reihe 14: Zeilen 74-81



Reihe 15: Zeilen 82-87



Reihe 16: Zeilen 88-96



Reihe 17: Zeilen 97-103



Reihe 18: Zeilen 104-113



Reihe 19: Zeilen 114-121



Reihe 20: Zeilen 121-131



Reihe 21: Zeilen 131-138



Reihe 22: Zeilen 139-148



Reihe 23: Zeilen 149-157



Reihe 24: Zeilen 158-163



Programm: "SYNTBc" - Benoetigte PRGM-Reg: 36

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-10



Reihe 3: Zeilen 11-12



Programm: "SYNTBc" - Fortsetzung

Reihe 4: Zeilen 12-17



Reihe 5: Zeilen 17-20



Reihe 6: Zeilen 20-28



Reihe 7: Zeilen 28-31



Reihe 8: Zeilen 31-37



Reihe 9: Zeilen 37-43



Reihe 10: Zeilen 44-50



Reihe 11: Zeilen 51-57



Reihe 12: Zeilen 57-66



Reihe 13: Zeilen 66-73



Reihe 14: Zeilen 73-81



Reihe 15: Zeilen 81-89



Reihe 16: Zeilen 89-96



Reihe 17: Zeilen 97-105



Reihe 18: Zeilen 106-115



Reihe 19: Zeilen 115-123



Reihe 20: Zeilen 124-125



Programm: "SYNTbc" - Benoetigte PRGM-Reg: 28

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-10



Reihe 3: Zeilen 11-12



Reihe 4: Zeilen 12-17



Reihe 5: Zeilen 17-20



Reihe 6: Zeilen 20-28



Reihe 7: Zeilen 28-31



Reihe 8: Zeilen 31-37



Reihe 9: Zeilen 37-43



Reihe 10: Zeilen 44-50



Reihe 11: Zeilen 51-57



Reihe 12: Zeilen 57-67



Reihe 13: Zeilen 68-76



Reihe 14: Zeilen 77-86



Reihe 15: Zeilen 87-88



Programm: "BLDSBC" - Benoetigte PRGM-Reg: 55

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 11-18



Reihe 5: Zeilen 18-22



Reihe 6: Zeilen 22-27



Reihe 7: Zeilen 27-31



Reihe 8: Zeilen 31-37



Reihe 9: Zeilen 37-44



Reihe 10: Zeilen 44-51



Reihe 11: Zeilen 51-56



Reihe 12: Zeilen 56-63



Reihe 13: Zeilen 63-68



Reihe 14: Zeilen 69-76



Reihe 15: Zeilen 77-84



Reihe 16: Zeilen 84-89



Reihe 17: Zeilen 90-94



Programm: "BLDSBC" - Fortsetzung

Reihe 18: Zeilen 95-102



Reihe 19: Zeilen 103-108



Reihe 20: Zeilen 109-115



Reihe 21: Zeilen 116-123



Reihe 22: Zeilen 124-129



Reihe 23: Zeilen 130-134



Reihe 24: Zeilen 134-140



Reihe 25: Zeilen 141-150



Reihe 26: Zeilen 151-159



Reihe 27: Zeilen 160-167



Reihe 28: Zeilen 168-177



Reihe 29: Zeilen 178-186



Reihe 30: Zeilen 187-187



Programm: "BLDSBc" - Benoetigte PRGM-Reg: 44

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



# Programm: "BLDSBc" - Fortsetzung

Reihe 4: Zeilen 11-18



Reihe 5: Zeilen 18-22



Reihe 6: Zeilen 23-27



Reihe 7: Zeilen 28-31



Reihe 8: Zeilen 32-37



Reihe 9: Zeilen 38-44



Reihe 10: Zeilen 45-53



Reihe 11: Zeilen 53-61



Reihe 12: Zeilen 62-66



Reihe 13: Zeilen 67-74



Reihe 14: Zeilen 74-77



Reihe 15: Zeilen 78-85



Reihe 16: Zeilen 85-90



Reihe 17: Zeilen 91-98



Reihe 18: Zeilen 99-106



Reihe 19: Zeilen 106-112



Reihe 20: Zeilen 112-116



Reihe 21: Zeilen 117-122



Programm: "BLDSbc" - Fortsetzung

Reihe 22: Zeilen 122-132



Reihe 23: Zeilen 133-141



Reihe 24: Zeilen 141-145



Programm: "BLDSbc" - Benoetigte PRGM-Reg: 29

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 11-18



Reihe 5: Zeilen 18-22



Reihe 6: Zeilen 23-28



Reihe 7: Zeilen 29-31



Reihe 8: Zeilen 31-37



Reihe 9: Zeilen 38-45



Reihe 10: Zeilen 46-54



Reihe 11: Zeilen 54-59



Reihe 12: Zeilen 59-66



Reihe 13: Zeilen 66-73



Programm: "BLDSbc" - Fortsetzung

Reihe 14: Zeilen 73-80



Reihe 15: Zeilen 81-90



Reihe 16: Zeilen 91-95



Programm: "BLDSDZ" - Benoetigte PRGM-Reg: 44

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-9



Reihe 4: Zeilen 10-11



Reihe 5: Zeilen 11-12



Reihe 6: Zeilen 12-17



Reihe 7: Zeilen 18-21



Reihe 8: Zeilen 22-27



Reihe 9: Zeilen 28-34



Reihe 10: Zeilen 34-36



Reihe 11: Zeilen 37-43



Reihe 12: Zeilen 43-49



Reihe 13: Zeilen 49-51



Programm: "BLDSDZ" - Fortsetzung

Reihe 14: Zeilen 52-55



Reihe 15: Zeilen 56-62



Reihe 16: Zeilen 62-70



Reihe 17: Zeilen 71-78



Reihe 18: Zeilen 79-81



Reihe 19: Zeilen 82-85



Reihe 20: Zeilen 86-92



Reihe 21: Zeilen 93-102



Reihe 22: Zeilen 102-109



Reihe 23: Zeilen 110-117



Reihe 24: Zeilen 117-117



Programm: "DZBLDS" - Benoetigte PRGM-Reg: 44

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 5-9



Reihe 3: Zeilen 9-12



Reihe 4: Zeilen 12-12



Reihe 5: Zeilen 12-14



Programm: "DZBLDS" - Fortsetzung

Reihe 6: Zeilen 14-19



Reihe 7: Zeilen 20-24



Reihe 8: Zeilen 24-31



Reihe 9: Zeilen 31-38



Reihe 10: Zeilen 39-47



Reihe 11: Zeilen 48-55



Reihe 12: Zeilen 56-65



Reihe 13: Zeilen 65-72



Reihe 14: Zeilen 72-74



Reihe 15: Zeilen 74-81



Reihe 16: Zeilen 82-86



Reihe 17: Zeilen 86-90



Reihe 18: Zeilen 91-94



Reihe 19: Zeilen 94-99



Reihe 20: Zeilen 100-108



Reihe 21: Zeilen 108-117



Reihe 22: Zeilen 117-124



Reihe 23: Zeilen 125-133



Programm: "DZBLDS" - Fortsetzung

Reihe 24; Zeilen 133-138



## Kapitel 3

### ZAHLEN UND SEQUENTIELLE DATEN

#### I. Zahlen

Auf den ersten Blick ist die Vielfalt der Zahlen und Anzeigeformate verwirrend. Wie soll es auch möglich sein, eine Zahl  $-3.778668641 \text{ E}-76$ , die als Programmzeile 16 Bytes beansprucht und deren zwei Ziffern 41 nicht einmal mehr im Rechnerdisplay Platz haben, noch in einer Barcodereihe darzustellen? 16 Bytes, dies stellten wir schon in Kapitel 1 fest, ist die maximale Länge einer BC-Reihe und zwar einschließlich Leitinformation. Müssen so lange Zahlen demnach verkürzt werden? Offenbar nicht, in ein Speicherregister passen sie ja vollständig hinein; die Lösung ist ganz einfach.

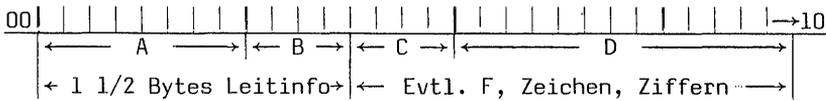
#### 1. Aufbau

Zahlenbarcodes weichen in ihrem Aufbau wegen der komprimierten Darstellung erheblich von anderen Typen ab. Im Gegensatz zu einer Programm-Barcodezeile, in der tatsächlich alle 16 Bytes der genannten Zahl nacheinander aufgeführt sind, folgt die Darstellung als BC-Reihe weitgehend dem internen Aufbau einer Zahl im Rechner. Jede Zahl, egal wie groß oder klein sie ist, besteht aus einer Folge von einzelnen Ziffern. Der Wert dieser Ziffern kann dabei immer nur zwischen 0 und 9 liegen, 10 ist bereits eine zweiziffrige Zahl.

In einem Nybble kann höchstens die Zahl 15 dargestellt werden, muß aber nicht! Man kann sich mit 0 bis 9 begnügen. Da immer nur ganze Bytes und keine Nybble verarbeitet werden, müßte die Zahl 9 in einem Byte so aussehen: 0000 1001 = 09; das erste Nybble in diesem Byte ist ungenutzt. Zählen wir eins weiter, so wird daraus: 0001 0000 = 10; es ist ersichtlich, bei den Zahlen handelt es sich um die hexadezimale Darstellung von dezimalen Ziffern innerhalb eines Bytes. Die dezimalen Ziffern 10 bis 14 (15 ist nicht erlaubt), werden für die Darstellung der Vorzeichen, des Dezimalpunktes und des Exponenten gebraucht. Außerdem wird zur neutralen Füllung nicht benötigter Nybble einer Zahl mit gerader Ziffernfolge ein 'Füllernybble' benötigt. Für den Aufbau einer Barcodesreihe stehen die folgenden Nybbleziffern zur Verfügung:

<u>Ziffern</u>		<u>Nybblecode</u>	
0		0000	
1		0001	
2		0010	
3		0011	
4		0100	
5		0101	
6		0110	
7		0111	
8		1000	
9		1001	
<u>Zeichen</u>	<u>Dezimal</u>		<u>Hexadezimal</u>
Füller	F = 10	1010	A
Dezpunkt	. = 11	1011	B
Plus	+ = 12	1100	C
Minus	- = 13	1101	D
Exponent	E = 14	1110	E
unzulässig	= 15	1111	

Die Dezimalzahl 99 würde demgemäß lauten: 10011001 = Dezimalwert des Byte 153. Dez 153 ist Hex = 99 - es werden hier also die Hexzahlen zu Dezimalzahlen.



A = 1 Byte 8-Bit Prüfsumme

B = 1. Nybble 4-Bit Typindikator

C = 2. Nybble 4 Bit: Besteht D aus einer ungeraden Anzahl Zeichen und Ziffern, z.B. -3.15 = 5, so nimmt C den Nybblecode für das erste Zeichen auf. Führt D eine gerade Anzahl Zeichen und Ziffern, so muß C mit dem Füller 'F' 'neutralisiert' werden.

D = Folge der Zeichen und Ziffern nybbleweise.

Noch eine weitere Auffüllung: Führt D nur eine Ziffer zwischen 0 und 9, so muß auch das erste Nybble des ersten Byte D mit dem Füller 'F' tatenlos gemacht werden, weil eine Barcodereihe dieses Typs aus mindestens drei Bytes bestehen muß. In diesem Fall lautet die Nybble/Bytefolge dann 'AA BF FD'.

### 1.1 Der Typindikator

Der Typindikator teilt dem Rechner lediglich mit, es ist eine Zahl ins X-Register einzulesen. Es gibt keine weiteren Alternativen. B = 6 - Zahl ins X-Register schreiben - binär 0110 0000 dezimal 96.

### 1.2 Die 8-Bit Prüfsumme

Die Prüfsumme ermittelt sich nach dem bekannten Verfahren aus der Addition von BC+D. Wir wollen die Prüfsumme der Barcodereihe für die erwähnte größere Zahl -3.778668641 E-76 berechnen und vorher die Bytes der Zahl aufbauen, deren Nybble folgende Werte haben:

HEX	6	A	D	3	B	7	7	8	6	6	8	6	4	1	E	D	7	6
	00000001	01101010	11010011	10110111	01111000	01100110	10000110	01000001	11101101	01110110								
DEZ		106	211	183	120	102	134	65	237	118								
			-	3	.	7	7	8	6	6	8	6	4	1	E	-	7	6

In den Ziffern der Hexadezimalzahlen ist unsere Beispielzahl unschwer zu erkennen. Aus der Umrechnung dieser Zahlen in Dezimalwerte, die ganz einfach Nybble 1x16+Nybble 2 erfolgen kann, ergibt sich die Bytefolge der Barcodereihe, aus der nun die Prüfsumme leicht zu berechnen ist: Addition Bytes 106 bis 118 ergibt 1276:256=4 Rest 252 - 4+252=256 - 256:256=1 Rest 0 - Prüfsumme = 1

Noch ein Wort zum Füllernybble und zum '+'-Zeichen. Wir haben festgestellt, daß bei einer einziffrigen Zahl, nehmen wir '1' = zwei Füllernybble die BC-Reihe auf insgesamt 3 Bytes füllen müssen. Führende Nullen werden vom Rechner bei Eingabe ins X-Register sofort eliminiert: '001' 'ENTER' = '1'. Es ergibt sich die Frage, warum ein besonderer Füller und nicht einfach '0' verwendet wird. Die Antwort finden Sie in diesem Barcode:



Lesen Sie Zeile ins X-Register ein, erhalten Sie entsprechend Ihrem gewählten Anzeigeformat '1'. Als Programmzeile eingelesen erhalten Sie '001'. Dies ist auch logisch, weil die Null eine Zahl ist, während der Füllercode neutralisiert. Das '+'-Zeichen ist offenbar innerhalb einer Barcodereihe bedeutungslos. Da ein '+'-Zeichen vom Rechner speziell nicht ausgewiesen wird, erscheint auch dies logisch. Warum trotzdem ein Code besteht bleibt unerfindlich. Die

folgenden beiden Barcodereihen demonstrieren dies. Sowohl die erste Zeile mit 5

HEX	9	6	6	C	2	B	E	C	1	2
DEZ		150		108		43		236		18
Zahl			+	2		.	E	+	1	2

als auch diese Zeile mit 4 Bytes Länge

HEX	3	3	6	2	B	E	1	2
DEZ		51		98		190		18
Zahl			2		.	E	1	2

lesen ins X-Register oder eine Programmzeile gleiche Zahlenwerte ein: 2. E 12. Die interne Zahlendarstellung im Rechner ist im 'Wickes' genau beschrieben. Unsere Beispielzahl -3.778668641 E-76 würde danach hexadezimal wie folgt lauten: 93 77 86 68 64 19 24. Auch hier erkennen wir wesentliche Teile der Ziffernfolge wieder. Es sind 7 Bytes, die genau ein Speicherregister füllen.

## 2. Herstellen von Zahlenbarcodes aus dem X-Register

Sie haben wieder die Wahl zwischen drei Programmen. Nirgends zeigt sich die Leistungsfähigkeit des Plottermoduls deutlicher, als bei diesen Programmen: "ZALNBC" benötigt 674 Bytes, während "ZALNbc" nur 39 Bytes verbraucht.

### 2.1 Nur Rechner HP 41 CV - Drucker, keine Module PRGM "ZALNBC"

#### Programmbedienung:

- SIZE: Das Programm prüft die 'SIZE'-Einstellung. Ist der 'SIZE' < 017, so werden Sie mit "XEQ SIZE 017" aufgefordert, mindestens diesen 'SIZE' einzustellen. Nach 'SIZE'-Einstellung drücken Sie dann 'R/S'.

XEQ "ZALNBC"



Besteht ein 'SIZE 017' oder größer, werden Sie nach 'XEQ' gleich gefragt: - "Zahl?": Geben Sie jetzt Ihre Zahl ein. Das Anzeigeformat kann beliebig sein. Es braucht nicht 'ENTER' gedrückt zu werden. Sie können das PRGM nach Eingabe Ihrer Zahl sofort mit 'R/S' starten. Gemäß Ihrem Anzeigeformat wird die Zahl rechtsbündig aus dem X-Register und anschließend werden die Barcodes gedruckt.

Bei Zahlen, deren Wert  $1 < -1$  ist, wird die führende Null bei Eingabe der Barcodes in eine Programmzeile fortgelassen: nicht 0.123 sondern .123 und nicht -0.123 sondern -.123. Im übrigen werden die Barcodes entsprechend der tatsächlichen Zahleneingabe und nicht des gerade vorhandenen Anzeigeformats erstellt. Auch die rechnerinterne Rundung bleibt unberücksichtigt. Hierzu zwei Beispiele: Geben Sie ein .9823224298 (nicht 0.98...) und drücken Sie 'ENTER' 'FIX 9'. In der Anzeige sehen Sie 0.982322430, da von den 10 eingegebenen Nachkommastellen wegen automatischer Rundung nur 9 angezeigt werden. Dabei wird aus ...242 = ...243 und aus ...429 = ...430. Trotzdem sind alle Ziffern im Rechner vorhanden. Führen Sie aus 10x, so sehen Sie 9.823224298. Geben Sie das zweite Beispiel .999999999 ein - 'ENTER'. Das Display zeigt 1.000000000. Auch hier bringt 10x alle im Rechner vorhandenen Neunen mit verschobener Kommastelle wieder zum Vorschein. Als Barcodes in eine Programmzeile eingelesen sehen Sie wieder die ursprüngliche Eingabe .999999999. Nichts wird verloren mit Barcodes.

01*LBL "ZALNBC"	59 48	117 CLX	175 106	233 X=Y?
02*LBL 00	60 X=Y?	118 X<> \	176 STO 01	234 GTO 04
03 17	61 GTO 02	119 "I+***"	177 .	235 RDN
04 *XEQ SIZE 017*	62 RDN	120 X<> I	178 RCL 15	236 48
05 DSE X	63 STO IND 00	121 X<> L	179 112	237 -
06 SF 25	64 DSE 00	122 X<> \	180 FS?C 00	238 16
07 RCL IND X	65 GTO 02	123 INT	181 48	239 *
08 FC? 25	66*LBL 00	124 ST+ J	182 +	240 ISG Y
09 TONE 7	67 XEQ 04	125 RDN	183 +	241 RCL IND Y
10 FC?C 25	68 RCL Z	126 6	184 STO 02	242 46
11 PROMPT	69 X=0?	127 ST* J	185 2	243 X=Y?
12 CLRG	70 CF 03	128 RDN	186 STO 00	244 GTO 00
13 RCL d	71 FC? 03	129 EI	187 GTO 09	245 RDN
14 STO I	72 GTO 02	130 ST* L	188*LBL 05	246 69
15 "I+**"	73 "0."	131 X<> L	189 XEQ 06	247 X=Y?
16 X<> I	74 E9	132 ST+ J	190 106	248 GTO 05
17 "**	75 *	133 CLX	191 STO 01	249 RDN
18 X<> I	76 INT	134 X<> J	192 RDN	250 48
19 STO \	77 FIX 0	135 RTN	193 FC? 00	251 -
20 ASTO 16	78 ARCL X	136*LBL 00	194 GTO 07	252 GTO 08
21 CF 00	79 LASTX	137 FS? 03	195 FS? 03	253*LBL 00
22 CF 01	80 FRC/	138 ISG 00	196 173	254 R↑
23 CF 03	81 EI	139 CF 01	197 FC? 03	255 R↑
24 15	82 *	140 .	198 160	256 11
25 STO 00	83 ARCL X	141 15	199 ISG 00	257 GTO 08
26 1	84*LBL 02	142 RCL 00	200 "**	258*LBL 04
27 "ZAHL?"	85 FS? 01	143 FS? 00	201 GTO 08	259 237
28 TONE 7	86 GTO 03	144 X=0?	202*LBL 06	260 STO IND 00
29 PROMPT	87 XEQ 04	145 CF 02	203 16	261 R↑
30 PRX	88 46	146 -	204 FS? 00	262 ISG X
31 ADV	89 X=Y?	147 1	205 1	263 ISG X
32 X<0?	90 SF 01	148 STO 00	206 2	264 GTO 06
33 SF 00	91 FS? 01	149 X=Y?	207 ST* L	265*LBL 05
34 SF 02	92 GTO 03	150 GTO 04	208 RDN	266 R↑
35 ABS	93 RDN	151 RDN	209 LASTX	267 R↑
36 X<Y?	94 48	152 FS? 00	210 FS? 00	268 14
37 SF 03	95 X=Y?	153 1	211 -	269*LBL 08
38 CLA	96 GTO 02	154 +	212 ABS	270 +
39 SF 28	97 SF 01	155 2	213 -	271 FS?C 02
40 CF 29	98 RDN	156 /	214 +	272 GTO 00
41 FIX 9	99 STO IND 00	157 FRC	215 .015	273 STO IND 00
42 ARCL X	100 DSE 00	158 X=0?	216 +	274 RDN
43 "I+**"	101 GTO 02	159 GTO 05	217 RCL IND X	275 ISG X
44 RCL J	102*LBL 03	160 .	218 RTN	276 GTO 06
45 X=0?	103 XEQ 04	161 XEQ 06	219*LBL 07	277-GTO 09
46 GTO 00	104 X=0?	162 FS? 00	220 ISG 00	278*LBL 06
47 CF 03	105 GTO 00	163 GTO 00	221 "**	279 RCL IND X
48 3	106 STO IND 00	164 FS? 03	222 46	280 GTO 07
49 CLA	107 DSE 00	165 61	223 X*Y?	281*LBL 00
50 ARCL Z	108 GTO 03	166 FC? 03	224 GTO 00	282 RCL IND 00
51*LBL 01	109 GTO 00	167 48	225 RCL Z	283 X<>Y
52 XEQ 04	110*LBL 04	168 GTO 00	226 ISG X	284 STO IND 00
53 STO IND 00	111 "I+I+****"	169*LBL 00	227 RCL IND X	285 RDN
54 RDN	112 RCL I	170 109	228 128	286 ISG Y
55 DSE 00	113 STO I	171 STO IND 00	229 GTO 00	287 GTO 07
56 DSE X	114 CLX	172 RDN	230*LBL 00	288*LBL 09
57 GTO 01	115 X<> J	173 GTO 07	231 RDN	289 RCL 00
58 XEQ 04	116 SIGN	174*LBL 04	232 69	290 RCL IND X

291 DSE Y	310 ACCOL	329 STO \	348 SF 25	367 GTO 04
292*LBL 10	311 RDN	330 "I**"	349 RCL IND X	368 -
293 RCL IND Y	312 128	331 X<> I	350 X=0?	369 1
294 +	313 RCL 00	332 STO \	351 ACCOL	370 +
295 DSE Y	314 +	333 "I*****"	352 X=0?	371 .
296 GTO 10	315 XEQ 05	334 X<> \	353 GTO 06	372 1
297 SF 02	316 XEQ 05	335 STO d	354*LBL 05	373 BLDSPEC
298 XEQ 00	317 RCL 00	336 CLX	355 XEQ 08	374 X<>Y
299*LBL 00	318 E-3	337 CLA	356 ARCL X	375 BLDSPEC
300 RCL X	319 *	338 CF 00	357 RDN	376 RTN
301 256	320 1	339 BEEP	358*LBL 06	377*LBL 04
302 ST/ Z	321 +	340 ADV	359 OUTA	378 .
303 MOD	322*LBL 11	341 ADV	360 CLA	379 X<>Y
304 +	323 XEQ 04	342 ADV	361 RDN	380 BLDSPEC
305 INT	324 ISG X	343 ADV	362 RTN	381 END
306 FS?C 02	325 GTO 11	344 ADV	363*LBL 08	674 Bytes
307 RTN	326 CLA	345 RTN	364 127	
308 SF 17	327 ARCL 16	346 GTO 00	365 X<>Y	
309 .	328 RCL d	347*LBL 04	366 X=Y?	

Anmerkungen zum Programm "ZALNBC": Die 'SIZE'-Prüfroutine Zeilen 02 bis 11 ist leicht abgewandelt dem bereits erwähnten 'Dearing' entnommen. Die Routine für das Speichern der Zeilen 13 - 20 und das Rückrufen, Zeilen 322 - 330 des Anzeigemodus entstammen dem gleichen Buch und sind ebenfalls leicht abgewandelt.

## 2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "ZALNBC"

Programmbedienung: Ist identisch mit "ZALNBC".

- SIZE: das Programm stellt, falls <= 'SIZE' 017 ein.

XEQ "ZALNBC"



01*LBL "ZALNBC"	26 CLA	51 GTO 02	76 FS? 01	101 FS? 03
02*LBL 00	27 SF 28	52*LBL 00	77 GTO 03	102 ISG 00
03 CLRG	28 CF 29	53 CF 03	78 RDN	103 CF 01
04 SIZE?	29 FIX 9	54 3	79 48	104 .
05 17	30 ARCL X	55*LBL 01	80 X=Y?	105 15
06 X>Y?	31 12	56 XEQ 04	81 GTO 02	106 RCL 00
07 PSIZE	32 ALENG	57 STO IND 00	82 SF 01	107 FS? 00
08 RCLFLAG	33 X>Y?	58 RCL Z	83 RDN	108 X=0?
09 STO 16	34 GTO 00	59 DSE 00	84 STO IND 00	109 CF 02
10 15	35 RCL Z	60 DSE X	85 DSE 00	110 -
11 STO 00	36 X=0?	61 GTO 01	86 GTO 02	111 1
12 CLST	37 CF 03	62 XEQ 04	87*LBL 03	112 STO 00
13 X<>F	38 FC? 03	63 48	88 XEQ 04	113 X=Y?
14 1	39 GTO 02	64 X=Y?	89 X=0?	114 GTO 04
15 "ZAHL?"	40 "0."	65 GTO 02	90 GTO 00	115 RDN
16 TONE 7	41 E9	66 RDN	91 STO IND 00	116 FS? 00
17 PROMPT	42 *	67 STO IND 00	92 DSE 00	117 1
18 PRX	43 INT	68 DSE 00	93 GTO 03	118 +
19 ADV	44 FIX 0	69*LBL 02	94 GTO 00	119 2
20 X<0?	45 ARCL X	70 FS? 01	95*LBL 04	120 /
21 SF 00	46 LASTX	71 GTO 03	96 -1	121 FRC
22 SF 02	47 FRC	72 XEQ 04	97 AROT	122 X=0?
23 ABS	48 E1	73 46	98 ATOX	123 GTO 05
24 X<Y?	49 *	74 X=Y?	99 RTN	124 .
25 SF 03	50 ARCL X	75 SF 01	100*LBL 00	125 XEQ 06

126 FS? 00	164 "**	202 16	240 GTO 06	278 +
127 GTO 00	165 GTO 00	203 *	241 GTO 09	279 XEQ 05
128 FS? 03	166*LBL 06	204 ISG Y	242*LBL 06	280 XEQ 05
129 61	167 16	205 RCL IND Y	243 RCL IND X	281 RCL 00
130 FC? 03	168 FS? 00	206 46	244 GTO 07	282 E-3
131 48	169 1	207 X=Y?	245*LBL 00	283 *
132 GTO 00	170 2	208 GTO 00	246 RCL IND 00	284 1
133*LBL 00	171 ST* L	209 RDN	247 X<>Y	285 +
134 109	172 RDN	210 69	248 STO IND 00	286*LBL 11
135 STO IND 00	173 LASTX	211 X=Y?	249 RDN	287 XEQ 04
136 RDN	174 FS? 00	212 GTO 05	250 ISG Y	288 ISG X
137 GTO 07	175 -	213 RDN	251 GTO 07	289 GTO 11
138*LBL 04	176 ABS	214 48	252*LBL 09	290 CLA
139 106	177 -	215 -	253 RCL 00	291 BEEP
140 STO 01	178 +	216 GTO 08	254 RCL IND X	292 ADV
141 .	179 .015	217*LBL 00	255 DSE Y	293 ADV
142 RCL 15	180 +	218 R↑	256*LBL 10	294 ADV
143 112	181 RCL IND X	219 R↑	257 RCL IND Y	295 ADV
144 FS?C 00	182 RTN	220 11	258 +	296 ADV
145 48	183*LBL 07	221 GTO 08	259 DSE Y	297 RCL 16
146 +	184 ISG 00	222*LBL 04	260 GTO 10	298 STOFAG
147 +	185 "**	223 237	261 SF 02	299 CLST
148 STO 02	186 46	224 STO IND 00	262 XEQ 00	300 RTN
149 2	187 X=Y?	225 R↑	263*LBL 00	301 GTO 00
150 STO 00	188 GTO 00	226 ISG X	264 RCL X	302*LBL 04
151 GTO 09	189 RCL Z	227 ISG X	265 256	303 SF 25
152*LBL 05	190 ISG X	228 GTO 06	266 SF / Z	304 RCL IND X
153 XEQ 06	191 RCL IND X	229*LBL 05	267 MOD	305 X=0?
154 106	192 128	230 R↑	268 +	306 ACCOL
155 STO 01	193 GTO 08	231 R↑	269 INT	307*LBL 05
156 RDN	194*LBL 00	232 14	270 FS?C 02	308 XTOR
157 FC? 00	195 RDN	233*LBL 00	271 RTN	309*LBL 06
158 GTO 07	196 69	234 +	272 SF 17	310 OUTA
159 FS? 03	197 X=Y?	235 FS?C 02	273 .	311 CLA
160 173	198 GTO 04	236 GTO 00	274 ACCOL	312 RDN
161 FC? 03	199 RDN	237 STO IND 00	275 RDN	313 END
162 160	200 48	238 RDN	276 128	527 Bytes
163 ISG 00	201 -	239 ISG X	277 RCL 00	

Erläuterungen zu "ZALNB.C.c": Das vorhandene Anzeigeformat wird gespeichert, und es wird 'FIX 9' gewählt. Sodann kopiert 'ARCL X' die Zahl ins Alpharegister von wo aus die Barcodereihe Ziffer für Ziffer formatiert wird. Nach dem Druck der BC wird das ursprüngliche Anzeigeformat wieder eingesetzt und es ist .END.

### 2.3 Rechner HP 41 C - Drucker, nur Plottermodul PRGM "ZALNbc"

Programmbedienung: genau wie "ZALNBC"

- SIZE: es wäre 'SIZE 000' möglich

XEQ "ZALNbc"



01*LBL "ZALNbc"	05 PROMPT	09 BCO	13 ADV	17 GTO 00
02*LBL 00	06 PRX	10 BEEP	14 ADV	18 END
03 "ZAHL?"	07 BCX	11 CLST	15 ADV	37 Bytes
04 TONE 7	08 CHS	12 CLA	16 RTN	

Die Plottermodul-Funktion 'BCX' erlaubt auch die Barcodefertigung von Kurztext, der vorher mit 'ASTO X' ins X-Register gebracht wurde. Die Anwendung dieser Möglichkeit ist praktisch jedoch nicht ergiebig, weil es sich bei diesem Text um genau den gleichen Barcodetyp handelt, welcher in Kapitel 2 besprochen wurde. Sofern einzelne Buchstaben mit 'BCX' gefertigt werden, sind sie identisch mit denen aus der Tabelle '0-255'. Das direkte Einlesen eines ASCII-Strings in das X-Register ist mit diesen BC nicht möglich.

### 3. Vielziffrige Zahlen in Programmen

Wir haben festgestellt, eine vielziffrige Zahl wie -3.77866841 E-76 verbraucht im Programmspeicher 16 Bytes, während die interne, 'komprimierte' Darstellung nur 7 Bytes verbraucht. Daraus ergibt sich die Frage, ob solche Zahlen auch in ein Programm bytesparend eingegeben werden können. Tatsächlich ist dies möglich. Das folgende Umrechnungsprogramm formt eine große Zahl in die entsprechenden 7 Bytes für einen Alphastring um, der dann mit 'RCL M' als entsprechende Zahl ins X-Register gelangt. Insgesamt werden für dieses Aktion benötigt: 7 Alphabytes + 1 Textbyte = 8 + 2 Bytes für 'RCL M' = 10 Bytes. Zahlen, die normal 11 und mehr Zeichen oder Ziffern haben, können so bis zu 6 Bytes sparend programmiert werden.

#### 3.1 Umrechnungsprogramm "ZXRADB"

##### Programmbedienung:

- SIZE: könnte 000 sein

XEQ "ZXRADB"



- "ZAHL?": Geben sie die große Zahl ein -3.778668641 E-76 - 'R/S' - Ergebnis: Sie erhalten den nebenstehenden Ausdruck. Die dezimalen Bytewerte 147 bis 36 können Sie nun mit einem der Programme "SYNT.." -BC in eine Barcodereihe fassen. Lesen Sie sodann die SYNTextzeile ein, setzen Sie aber vorher ein Alpha-LBL, und geben Sie anschließend 'RCL M' (z.B. aus PRGM "FS") ein. Starten Sie Ihr Programm und Sie sehen im X-Register bei 'FIX 9' -3.7786686-76. Führen Sie aus: 'ALPHA', evtl. 'CLA', dann 'ARCL X' 'ALPHA' 'PRA', und Sie erhalten: -3.778668641E-76 . Das Umrechnungsprogramm benutzt in Zeile 21 die umgekehrte synthetische Funktion nämlich 'STO M', um die Zahl als ASCII-String in das Alpharegister zu bringen. Anschließend werden dann die Barcodebytes formatiert.

##### -UMRECHNUNG-

Zahl X-Reg → AlphaDezByte

-3.778668641E-76

Byte 1= 147

Byte 2= 119

Byte 3= 134

Byte 4= 104

Byte 5= 100

Byte 6= 25

Byte 7= 36

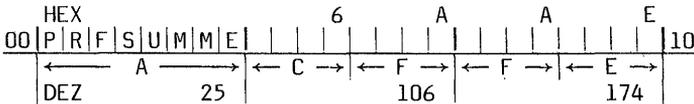
01*LBL "ZXRADB"	15 FC? 22	29 X<> ]	43 E1	57 "+= "
02*LBL 00	16 GTO 01	30 SIGN	44 ST* L	58 ARCL IND X
03 SF 12	17 CLA	31 CLX	45 X<> L	59 PRA
04 "-UMRECHNUNG-"	18 ARCL X	32 X<> \	46 ST+ ]	60 ISG X
05 PRA	19 PRA	33 "+****"	47 CLX	61 GTO 03
06 CF 12	20 CLA	34 X<> [	48 X<> ]	62 RCL Z
07 "Zahl X-Reg → Ale"	21 STO [	35 X<> L	49 STO IND Y	63 FIX 9
08 "hαDezByte"	22 7	36 X<> \	50 RDN	64 CLA
09 PRA	23 FIX 0	37 INT	51 ISE X	65 ADV
10*LBL 01	24*LBL 02	38 ST+ ]	52 GTO 02	66 RTN
11 FIX 9	25 "+↓****"	39 RDN	53 1.007	67 GTO 00
12 CF 22	26 RCL [	40 6	54*LBL 03	68 END
13 "ZAHL?"	27 STO [	41 ST* ]	55 "Byte "	101 Bytes
14 PROMPT	28 CLX	42 RDN	56 ARCL X	

- Noch eine kurze Betrachtung zur Programmierlösung:

Wenn mit 'RCL M' textformierte Bytes als Zahl ins X-Register geholt werden können, so ist es - wie wir gesehen haben - natürlich auch möglich eine Zahl mit 'STO M' zu Textbytes im Alpharegister zu machen und aus diesen 'ganz einfach' die Barcodereihe aufzubauen? So einfach, wie dies auf den ersten Blick zu sein scheint, ist es allerdings nicht. Eine Zahl .1000000001 würde nach 'STO M' die Dezimalbytefolge 1 0 0 0 0 25 153 oder HEX 01 00 00 00 00 19 99 ergeben. Als Barcodereihe müßte die Zahl lauten: DEZ 124 107 16 0 0 0 1 oder wieder hexadezimal 7C 6B 10 00 00 00 01. 'B' ist der Dezimalpunkt. Dies wirkt noch relativ einfach. Anders wird es mit unserer Beispielzahl -3.778668641 E-76. Mit 'STO M' ergibt sich DEZ 147 119 134 104 100 25 36 und HEX 93 77 86 68 64 19 24. Als Barcodes lauten die Bytes DEZ 1 106 211 183 120 102 134 65 237 118 oder wieder HEX 01 6A D3 B7 78 66 86 41 ED 76. Die Idee große Teile einer Zahl nach 'STO M' einfach übernehmen zu können ist nicht realistisch, weil dies Beispiel zeigt, auch hier wäre eine nybbleweise Ergänzung um Zeichen, Füller und das Zusammenfügen der Bytes für den Barcodedruck nötig. Eine genaue Betrachtung der HEX-'STO M'- und HEX-Barcodefolge verdeutlicht dies.

4. Kurzformexponenten

Wir haben bereits in Abschnitt I.1. festgestellt, das Symbol für den Exponenten ist die hexadezimale Zahl 'E'. Tatsächlich wird ja auch bei Eingabe des 'EEX'-Ponenten über das Tastenfeld in eine Programmzeile des 'E'-Symbol benutzt. Wird dabei eine Zahl wie 10000 mit der 'EEX'-Funktion eingegeben, so erscheint 1 E4. Gibt man nur 'EEX' ein, steht 1 E\_ in der Anzeige. Löscht man nun mit der Korrekturtaste die Eingabe, wird zuerst das 'E' und hernach mit zweitem 'CLX/CLA' die '1' gelöscht. Die Einbytefunktion DEZ 27 der Bytetabelle, nimmt also allein eingegeben zwei Bytes in Anspruch. Es stellt sich die Frage, ob die '1' vor dem 'E' denn überhaupt nötig ist? Wir untersuchen dies. Führen wir uns einmal den Aufbau vor:



Lesen Sie diese Barcodereihe in eine Programmzeile ein:



Sie sehen: 01 E. Führen Sie die Zeile mit 'SST' aus; Ergebnis - '1' im X-Register. Die '1' vor dem 'E' scheint überflüssig. Weitgehend stimmt dies auch jedoch nicht immer! Wer ein XF-Modul hat oder einen HP 41 CX führe folgendes aus: 'ALPHA' - Eingabe ins Alpharegister "1E4" - Umschalten RUN-Modus 'ANUM' - Ergebnis im X-Register: 10 000. - ... und nun: "E4" 'ANUM' - Ergebnis in X = 4. Vorsicht also mit Kurzformexponenten, die mit solchen PRGM-Sequenzen verarbeitet werden sollen. Es ergibt sich nun noch die Frage, ob es '-E', also '-1' geben kann. Dies ist ohne weiteres möglich. Lesen Sie ein und führen dann mit 'SST' aus:



Die Programmzeile zeigt '-E' und im X-Register erscheint '-1'. Praktisch möglich wären alle Exponenten zwischen 'E99' bis '-E99'. Da die Funktion 'CHS' jedoch genau wie das Minuszeichen ein Byte verbraucht, kann damit lediglich Papier beim Programmdruck eingespart werden. Bytes, so wie durch Wegfall der '1' vor dem 'E' werden nicht gespart. Tabellen von Kurzformexponenten, wie man sie gelegentlich sieht, beginnen meist mit 'E' und werden fortgesetzt mit 'E1'

'E-1' und so weiter. Es ist jedoch nicht sinnvoll Kurzformexponenten 'E2' < 'E-3' einzusetzen; der Wert 'E-1' = '.1' erfordert im PRGM-Speicher ein Byte mehr Speicherplatz. Für 'E1' = '10' ist es egal, ebenso für 'E-2' = '.01'. Bytesparend wird die Verwendung von Kurzformexponenten erst ab 'E2' = '100', bzw. 'E-3' = '.001'. Trotzdem beginnt das nachfolgende Programm "KFEXBC" bereits bei 'E1' bzw. 'E-1'. Wer mag, kann ja diese Formen dennoch verwenden.

4.1 Kurzformexponenten-Programm "KFEXBC"

Programmbedienung:

- SIZE: könnte 000 sein

XEQ "KFEXBC"



- "E ?": Sie können nun von 1 bis 99 oder -1 bis -99 alle Exponenten eingeben. Sofort bei einem unzulässigen Wert erfolgt Fehlerhinweis: "E ????" und Sie werden erneut aufgefordert "E ?". Ist die Barcodereihe gedruckt, so drücken Sie für eine weitere 'E'-Zahl 'R/S'.

01*LBL "KFEXBC"	30 106	59*LBL 04	88 "j"	117*LBL 09
02*LBL 00	31 +	60 RCL X	89 FC? 01	118 OUTA
03 CF 00	32 GTO 05	61 10	90 GTO 06	119 CLA
04 CF 01	33*LBL 01	62 MOD	91 XEQ 09	120 RTH
05 "E ?"	34 99	63 -	92 ""	121*LBL 10
06 9	35 X<>Y	64 LASTX	93 XEQ 09	122 127
07 CF 22	36 X?Y?	65 X<>Y	94 RDN	123 X<>Y
08 FIX 0	37 GTO 11	66 1.6	95 XEQ 08	124 X<=Y?
09 TONE 7	38 XEQ 04	67 *	96 GTO 07	125 GTO 12
10 PROMPT	39 FS? 00	68 +	97*LBL 06	126 -
11 FC? 22	40 GTO 02	69 RCL X	98 FS? 00	127 1
12 GTO 00	41 110	70 RTH	99 "n"	128 +
13 INT	42 +	71*LBL 05	100 XEQ 09	129 .
14 "E"	43 SF 00	72 RCL X	101 RDN	130 1
15 ACA	44 GTO 05	73 256	102 XEQ 08	131 BLDSPEC
16 ACX	45*LBL 02	74 ST/ Z	103*LBL 07	132 X<>Y
17 ADV	46 106	75 MOD	104 CLST	133 BLDSPEC
18 X=0?	47 237	76 +	105 CLA	134 RTH
19 GTO 11	48 +	77 INT	106 ADV	135*LBL 11
20 X<0?	49 +	78 ADV	107 ADV	136 "E ????"
21 SF 00	50 SF 01	79 SF 17	108 BEEP	137 ACA
22 ABS	51 GTO 05	80 .	109 ADV	138 TONE 0
23 X?Y?	52*LBL 03	81 ACCOL	110 ADV	139 ADV
24 GTO 01	53 200	82 RDN	111 RTH	140 GTO 00
25 FS? 00	54 +	83 ""	112 GTO 00	141*LBL 12
26 GTO 03	55 RCL X	84 FS? 01	113*LBL 08	142 .
27 224	56 110	85 ""	114 XEQ 10	143 X<>Y
28 +	57 +	86 XEQ 09	115 ARCL X	144 BLDSPEC
29 RCL X	58 GTO 05	87 XEQ 08	116 RDN	145 END 255 Bytes

Sofern jemand die Idee der negativen Kurzformexponenten aufgreifen möchte, kann das vorhandene Programm erweitert werden. Lesen Sie die folgende BC-Reihe in eine Programmzeile ein:



Sie sehen: 01 - E-99 . Führen Sie 'SST' 'FIX 2' 'PRX' aus: -1.00-99 \*\*\* . Die zusätzlichen Alternativen zum PRGM "KFEXBC" wären eine recht gute Erweiterung:

```

-E 1 bis -E 9 | A A | 6 D | E n |
-E -1 bis -E -9 | A A | 6 F | D E | D n |
-E 10 bis -E 99 | A A | 6 F | D E | n n |
-E-10 bis -E-99 | A A | 6 D | E D | n n |

```

Es sind: A A = Prüfsumme; D = "-"; E = Exponentsymbol und F = Füllernybble.

### 5. Arbeiten mit Zahlenbarcodes

Das Arbeiten mit Zahlenbarcodes ist, wie in Kapitel 2 Abschnitt II.a.4. beschrieben, auch in Kombination mit Textzeilen möglich. Lesen Sie bitte das kleine Programm "AZ" ein, alsdann 'XEQ "AZ"' und darauf bei "MONAT?" die erste und bei "JAHR?" die zweite der auf XEQ "AZ" folgenden Barcodes:

PROGRAMM = "AZ" Benötigte Register: 8 - nicht PV-geschützt

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-11



Reihe 4: Zeilen 12-17



Reihe 5: Zeilen 18-23



XEQ "AZ"



Ergebnis:  
MAI 1986

CLP "AZ"



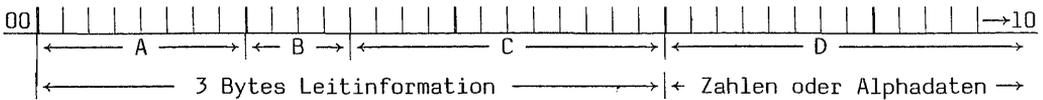
Nach diesem Beispiel können Sie das PRGM mit den BC CLP "AZ" wieder löschen.

## II. Sequentielle Daten

Dies ist der Barcodetyp für das aufeinanderfolgende Einlesen beliebiger Zahlen oder auf 6 ASCII-Zeichen beschränkte Alphadaten in Register. Die Lesestift-Funktion hierfür ist 'WNDDTX'. Die Reihenfolge der Register wird, wie beim Schleifenzähler, durch eine Steuerzahl in der Form bbb.eee bestimmt. Schlagen Sie dies nötigenfalls im Bedienungshandbuch zum Barcodelesestift noch einmal nach. Die Reihen-Folgenummer der Barcodereihe ist in der Reihe selbst verschlüsselt. Deswegen weicht der Aufbau dieser Barcodes von den Zahlen-BC ab.

### 1. Aufbau

Für den Aufbau der sequentiellen Datenbarcodes werden die gleichen Nybblecodes wie für Zahlenbarcodes verwendet, trotzdem ist alles anders bei diesen BC.



A = 1 Byte 8-Bit Prüfsumme

B = 1. Nybble 4-Bit Typindikator

C = 2. Nybble von Byte 2 und Byte 3 - 12 Bit Folgenummer

D = Besteht D aus einer ungeraden Anzahl Zeichen und Ziffern, so ist Nybble 1 vom 1. Byte D mit dem Füller zu neutralisieren. Führt D eine gerade Anzahl Ziffern/Zeichen entfällt ein Füller. Bei einer Anzahl ASCII-Zeichen wird kein Füller benötigt, weil jedes Zeichen immer ein Byte besetzt.

Eine BC-Reihe dieses Typs muß aus mindestens vier Bytes bestehen, siehe oben.

### 1.1 Der Typindikator

Dieser Typindikator teilt dem Rechner mit, ob Zahlen - B = 9 binär 1001... oder Alphanumerik - B = 10 binär 1010... eingelesen werden. Sofern diese Barcodes nicht als Folgedaten mit 'WNDDTX' in Register, sondern wie normale Zahlen- oder Textzeilen eingelesen werden, wird unter Mißachtung der Folgenummer eine Zahl ins X-Register und ein Alphastring bei Umschaltung ins Alpharegister eingelesen. Auch in eine PRGM-Zeile werden beide wie üblich und bei Mißachtung der Folgenummer übernommen.

Es gibt noch B = 11 - Alpha append. Praktisch ist dies ohne Bedeutung, weil die Alphanumerik für das Einlesen in Register bestimmt sind und sie mit 'ARCL\_\_' sowieso an den Alpha-Registerinhalt angehängt werden.

Sie werden ferner bemerkt haben, es ist bei B = 10 keine Angabe der Anzahl der Alphazeichen nötig, wie dies bei Textbarcodes erforderlich ist. Die vorhandene Anzahl Bytes, ob nun Ziffern und Zeichen oder ein Alphastring, wird einfach im Anschluß an die Folgenummer der Reihe angehängt; dabei ist die Anzahl ASCII-Zeichen keineswegs auf 6 beschränkt, sondern es können hier maximal 13 Bytes angehängt werden. Die Folgenummer wäre in diesem Fall auf 000 zu setzen. Dabei könnte dann bei B = 11 auch eine Append-Zeile entstehen. Die Verarbeitung von 'BLDSPEC'-Zeichen mit 2x7 Bytes ist damit dann allerdings nicht möglich. Dennoch, wer Text nur für das Alpharegister gebrauchen möchte, könnte sich ein kürzeres Programm für Texte mit diesem Barcodetyp schreiben.

### 1.2 Die Folgenummer

Die Folgenummer besteht aus 3 Nybble = 12 Bit. Im Gegensatz zu den Ziffern und Zeichen der eingegebenen Zahlen, von denen je zwei ein Byte ergeben, stellen die drei Nybble für die Folgenummer einen Zähler dar, der den logischen Notwendigkeiten des 8-Bit-Systems entspricht. Eine Zahl 999 müßte in 'D' dargestellt HEX lauten: A9 99 (A= HEX-Wert des Füllers) DEZ = 169 153. Die Zahl 999 als Folgenummer lautet dagegen: HEX (B) 3 E7 (B = Typindikator) - DEZ = 3 14 7 oder binär ..B.0011 11100111. Diese Folgenummer schlüsselt sich so auf: Die Zählung beginnt in Nybble 3 und geht hier bis maximal 15; dann läuft sie 'über' und setzt sich fort mit 1 (16x) in Nybble 2. Sie beginnt erneut in Nybble 3 bis wieder Überlauf erfolgt und 2 (16x) in Nybble 2 steht. So setzt sich die Zählung fort, bis schließlich 255 'voll' sind und Überlauf nach Nybble 1 mit 1 (256x) stattfindet. Ein HP 41 CV oder HP 41 C mit Quard-Modul hat mehr als 255 Register. Deswegen sind bei dieser Zählsystematik 3 Nybble erforderlich. Die höchste Folgenummer wäre übrigens  $(15 \times 256) + (15 \times 16) + 15 = 4095$ , die allerdings 'NONEXISTENT' ist. 999 ist die höchste Praxiszahl. Die Zahl 4095 werden wir jedoch noch bei den GTO-Barcodes wiedertreffen. Unsere Folgenummer berechnet sich:  $3 \times 256x = 768$  - Nybble 1 -  $14 \times 16x = 224$  - Nybble 2  $7x =$  zusammen 999.

### 1.3 Die 8-Bit Prüfsumme

Die Prüfsumme schließt wieder alle nachfolgenden Bytes ein, und sie wird nach dem inzwischen bekannten Rechengang ermittelt. Zwei Beispiele: Zahl '1' - Folgennummer 257. Die Bytefolge ohne Prüfsumme wäre:

HEX	9	1	0	1	A	1
	10010001	00000001	10100001			
DEZ	145		1	161		

Erklärung der HEX-Zahlen der Reihe nach:

9 = B - Zahl 9 16x	= 144		
1 = 256 1x	= 1	= 256	
0 = 16 0x	= 0	= 0	
1 = 1 1x	= 1	= 1	
A = Füller 16 10x	= 160		
1 = Zahl	= 1		
Für Prüfsummenermittlung:	307	Folgennummer =	257

=====

Die Prüfsumme errechnet sich aus  $307:256= 1$  Rest  $51 = 52$

Für Beispiel 2 ein Alphastring: "OK?" - Folgennummer '0'. Bytefolge ohne Prüfsumme:

HEX	A	0	0	0	4F	4B	3F
	10100000	00000000		0	K	?	
DEZ	160		0	79	75	63	

Die Prüfsumme errechnet sich aus  $160+79+75+63 = 377:256 = 1$  Rest  $121 = 122$ .

## 2. Herstellen von sequentiellen Daten-Barcodes

Wiederum gibt es drei Programme. Die Länge ist trotz gleicher Bedienung beim PRGM mit Plottermodul am geringsten. Dies Modul macht sich wirklich bezahlt.

### 2.1 Nur Rechner HP 41 CV - Drucker, keine Module PRGM "ZFDABC"

#### Programmbedienung:

- SIZE: Es wird geprüft ob 'SIZE => 023' ist, falls nicht erfolgt Aufforderung 'XEQ SIZE 023'.

XEQ "ZFDABC"



Anschließend, wenn der 'SIZE' entsprechend ist, wird gefragt:

- "FOLGENR= 0?": Die Folgennummer '0' wird immer dann angezeigt, wenn eine Barcodereihenfolge begonnen wird, weil diese stets mit '0' anfängt, egal welche Registernummer definiert ist. Diese Frage wird trotzdem gestellt, denn es ist möglich, eine Arbeit zur BC-Herstellung wurde unterbrochen und soll fortgesetzt werden. Sie könnten dann, nachdem vielleicht 22 BC-Reihen bereits angefertigt sind, bei dieser Frage mit 23 fortsetzen. Andernfalls drücken Sie 'R/S', und es wird mit '0' begonnen.
- "ZAHL?-ALPHA?": Geben Sie nun eine Zahl oder nach Umschaltung 'ALPHA' einen String von höchstens 6 Zeichen ein. Geben Sie mehr Alpha-Zeichen ein, werden diese ignoriert.

Rechtsbündig wird die Zahl entsprechend Ihrem Anzeigeformat wiederholt oder die Alphazeichen werden gedruckt. Anschließend wird die Barcodereihe erstellt. Danach wird gedruckt: Folgennummer n und im Display sehen Sie wiederum fragen: "ZAHL?-ALPHA?". Es können nun alle Daten eingegeben werden. Um die Folgennummer brauchen Sie sich nicht zu kümmern, sie wird automatisch weitergezählt.

Möchten Sie die Barcodeherstellung beenden, drücken Sie bei "ZAHL?-ALPHA?" ohne Eingabe 'R/S'; in der Anzeige steht dann '0'. Drücken Sie hierauf erneut 'R/S' erfolgt neuer Programmstart bei Folgenummer '0'.

01*LBL "ZF0ABC"	55 ABS	109 FIX 0	163 X<> L	217 DSE 00
02*LBL a	56 I	110 ARCL X	164 ST+ J	218 RDN
03 23	57 X<>Y	111 LASTX	165 CLX	219 INT
04 "XEQ SIZE 023"	58 X<Y?	112 FRC	166 X<> J	220 STO IND 00
05 DSE X	59 SF 04	113 EI	167 RTN	221 DSE 00
06 SF 25	60 CLA	114 *	168*LBL Y	222 "A"
07 RCL IND X	61 ARCL X	115 ARCL X	169 XEQ 13	223 FS? 03
08 FC? 25	62 "I**"	116*LBL 02	170 CLA	224 ASTO X
09 TONE 7	63 RCL J	117 FS? 01	171 CLST	225 FC? 03
10 FC?C 25	64 X=0?	118 GTO 03	172 RTN	226 9
11 PROMPT	65 GTO 00	119 XEQ 04	173 GTO a	227 STO IND 00
12 CLRG	66 SF 02	120 46	174*LBL 13	228 FS? 03
13 CLA	67 3	121 X=Y?	175 CLA	229 GTO 14
14 RCL d	68 CLA	122 SF 01	176 ARCL 22	230 3
15 STO I	69 ARCL Z	123 FS? 01	177 RCL d	231 RCL 00
16 "I+**"	70*LBL 01	124 GTO 03	178 STO \	232 +
17 X<> I	71 XEQ 04	125 RDN	179 "I+**"	233 E3
18 "**"	72 STO IND 00	126 48	180 X<> J	234 /
19 X<> I	73 RDN	127 X=Y?	181 STO \	235 20
20 STO \	74 DSE 00	128 GTO 02	182 "I+*****"	236 +
21 ASTO 22	75 DSE X	129 SF 01	183 X<> \	237 FC? 02
22 CF 22	76 GTO 01	130 RDN	184 STO d	238 GTO 07
23 "FOLGENR= 0 ?"	77 XEQ 04	131 STO IND 00	185 RTN	239 2
24 TONE 7	78 48	132 DSE 00	186*LBL 00	240 X<>Y
25 PROMPT	79 X=Y?	133 GTO 02	187 FS? 03	241*LBL 06
26*LBL b	80 GTO 02	134*LBL 03	188 GTO 05	242 RCL IND X
27 CF 29	81 RDN	135 XEQ 04	189 FC? 00	243 48
28 FIX 0	82 STO IND 00	136 X=0?	190 GTO 04	244 -
29 FS?C 22	83 DSE 00	137 GTO 00	191 45	245 STO IND Y
30 STO 21	84 GTO 02	138 STO IND 00	192 STO IND 00	246 RDN
31 "FOLGENR: "	85*LBL 00	139 DSE 00	193 DSE 00	247 DSE X
32 ARCL 21	86 RDN	140 GTO 03	194*LBL 04	248 DSE Y
33 ACA	87 FC? 23	141 GTO 00	195 RCL 00	249 GTO 06
34 ADV	88 XEQ 04	142*LBL 04	196 2	250 RCL IND X
35 20	89 FC? 23	143 "I+*****"	197 /	251 45
36 STO 00	90 GTO 00	144 RCL I	198 FRC	252 X*Y?
37 XEQ 13	91 AOFF	145 STO I	199 X=0?	253 GTO 00
38 CF 23	92 ASTO 20	146 CLX	200 GTO 05	254 "D"
39 "ZAHL?-ALPHA?"	93 CLA	147 X<> J	201 DSE 00	255 ASTO IND Z
40 TONE 7	94 ARCL 20	148 SIGN	202*LBL 05	256 DSE Z
41 PROMPT	95 ACA	149 CLX	203 RCL 21	257*LBL 00
42 FS? 23	96 ADV	150 X<> \	204 RCL X	258 "E"
43 GTO 00	97 ADV	151 "I+**"	205 256	259 ASTO IND Z
44 FC?C 22	98 SF 01	152 X<> I	206 ST/ Z	260 DSE Z
45 GTO Y	99 SF 03	153 X<> L	207 MOD	261 RCL Z
46 ACX	100 GTO 02	154 X<> \	208 RCL X	262*LBL 07
47 ADV	101*LBL 00	155 INT	209 16	263 RCL IND X
48 ADV	102 RDN	156 ST+ J	210 ST/ Z	264 46
49 "a)0+"	103 FC?C 04	157 RDN	211 MOD	265 X=Y?
50 RCL I	104 GTO 02	158 6	212 STO IND 00	266 XEQ 00
51 X<> d	105 "0."	159 ST* J	213 DSE 00	267 RDN
52 RDN	106 E9	160 RDN	214 RDN	268 45
53 X<0?	107 *	161 EI	215 INT	269 X=Y?
54 SF 00	108 INT	162 ST* L	216 STO IND 00	270 GTO 05

271 RDN	316 RDN	361 X<>Y	406 RCL IND Y	451 CLRG
272 X=0?	317 55	362 MOD	407 +	452 STO 22
273 GTO 04	318 +	363 ST- J	408 DSE Y	453 RDN
274 48	319 .	364 LASTX	409 GTO 11	454 STO 21
275 -	320 X<>Y	365 ST/ J	410 SF 02	455 CLX
276 STO IND Y	321 BLDSPEC	366 CLX	411 XEQ 00	456 CLA
277 RDN	322 STO IND Y	367 X<> J	412*LBL 00	457 GTO b
278 DSE X	323 RDN	368 X<>Y	413 RCL X	458*LBL 04
279 GTO 07	324 DSE X	369 29	414 256	459 SF 25
280 GTO 14	325 DSE Y	370 ST- Z	415 ST/ Z	460 RCL IND X
281*LBL 00	326 GTO 00	371 -	416 MOD	461 X=0?
282 "B"	327*LBL 04	372 .9	417 +	462 ACCOL
283 ASTO IND Z	328 FIX 0	373 ST* Z	418 INT	463 X=0?
284 RCL Z	329 RCL 00	374 *	419 FS?C 02	464 GTO 00
285 DSE X	330 FS? 03	375 INT	420 RTN	465*LBL 05
286 RCL IND X	331 GTO 00	376 X<>Y	421 SF 17	466 XEQ 04
287 ENTER†	332 .02	377 INT	422 .	467 ARCL X
288 RTN	333 +	378 16	423 ACCOL	468 RDN
289*LBL 04	334 1	379 *	424 RDN	469*LBL 00
290 "A"	335 STO 00	380 +	425 128	470 OUTA
291 ASTO IND Y	336 RDN	381 STO IND 00	426 RCL 00	471 CLA
292 GTO 14	337 GTO 09	382 RDN	427 +	472 RDN
293*LBL 05	338*LBL 00	383 ISG 00	428 XEQ 05	473 RTN
294 "D"	339 RCL 00	384 "+"	429 XEQ 05	474*LBL 04
295 ASTO IND Z	340 3	385 ISG X	430 RCL 00	475 127
296 RCL Z	341 +	386 GTO 09	431 E-3	476 X<>Y
297 DSE X	342 E3	387 FC?C 03	432 *	477 X<=Y?
298 GTO 07	343 /	388 GTO 04	433 1	478 GTO 04
299*LBL 14	344 +	389 INT	434 +	479 -
300 2	345 1	390 .02	435*LBL 12	480 1
301 RCL 00	346 STO 00	391 +	436 XEQ 04	481 +
302 3	347 RDN	392*LBL 10	437 ISG X	482 .
303 +	348*LBL 09	393 RCL IND X	438 GTO 12	483 1
304*LBL 08	349 CLA	394 STO IND 00	439 XEQ 13	484 BLDSPEC
305 RCL IND X	350 ARCL IND X	395 RDN	440 CF 00	485 X<>Y
306 9	351 ISG X	396 ISG 00	441 CF 01	486 BLDSPEC
307 X<Y?	352 GTO 00	397 "+"	442 CF 22	487 RTN
308 GTO 00	353 GTO 04	398 ISG X	443 BEEP	488*LBL 04
309 RDN	354*LBL 00	399 GTO 10	444 ADV	489 .
310 RDN	355 ARCL IND X	400*LBL 04	445 ADV	490 X<>Y
311 DSE X	356 "+Δ"	401 DSE 00	446 ADV	491 BLDSPEC
312 DSE Y	357 RCL I	402 RCL 00	447 ISG 21	492 END
313 GTO 00	358 E2	403 RCL IND X	448 "+"	889 Bytes
314 GTO 04	359 X<>Y	404 DSE Y	449 RCL 21	
315*LBL 00	360 STO J	405*LBL 11	450 RCL 22	

Die Routine Zeilen 354-380 entstammt leicht abgewandelt dem bereits erwähnten 'Dearing', und sie bewirkt die HEX-DEZ-Umwandlung. In diesem PRGM sind ein LBL und drei GTO's bzw. XEQ's enthalten, die nur synthetisch hergestellt werden können. Die Marke 'Y' Zeile 168 ist tatsächlich ein Langform-LBL (B 207 114) und nicht ein Alpha-LBL "Y". Dazu gehört das 'GTO Y' (B 208 0 114) Zeile 45. Die 'XEQ 13' Zeilen 37, 169 und 439 sind Langformaufführungen (B 224 0 13). Auch die 'GTO 14' - Zeilen 229, 280 und 292 sind Langform-Sprunganweisungen (B 208 0 14). Diese Verzweigungen 'springen' trotz ihres Kurzformaussehens über mehr als 112 Bytes; 'XEQ 13' auf 'LBL 13' Zeile 174 und 'GTO 14' auf 'LBL 14' Zeile 299. Trotzdem der praktische Nutzen angesichts der vielen verfügbaren LBL unseres HP 41 fragwürdig ist, werden diese synthetischen Produkte im Rahmen der Barcodes des Kapitel 4 mit behandelt.

2.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "ZFDABc"

Programmbedienung: identisch mit "ZFDABC"

- SIZE: das Programm stellt mindestens 'SIZE 023' ein - falls kleiner.

XEQ "ZFDABc"



01*LBL "ZFDABc"	51 ABS	101 SF 01	151 2	201 RDN
02*LBL a	52 1	102 SF 03	152 /	202 DSE X
03 SIZE?	53 X<Y	103*LBL 02	153 FRC	203 DSE Y
04 23	54 X>Y?	104 FS? 01	154 X=0?	204 GTO 06
05 X>Y?	55 GTO 00	105 GTO 03	155 GTO 05	205 RCL IND X
06 PSIZE	56 X=Y?	106 XEQ 04	156 DSE 00	206 45
07 CLRG	57 GTO 00	107 46	157*LBL 05	207 X*Y?
08 RCLFLAG	58 CLA	108 X=Y?	158 RCL 21	208 GTO 00
09 STO 22	59 FS?C 05	109 SF 01	159 RCL X	209 "D"
10 CF 22	60 "-"	110 FS? 01	160 256	210 ASTO IND Z
11 "FOLGENR= 0 ?"	61 "+0."	111 GTO 03	161 ST/ Z	211 DSE Z
12 TONE 7	62 E9	112 RDN	162 MOD	212*LBL 00
13 PROMPT	63 *	113 48	163 RCL X	213 "E"
14*LBL b	64 INT	114 X=Y?	164 16	214 ASTO IND Z
15 CF 29	65 FIX 0	115 GTO 02	165 ST/ Z	215 DSE Z
16 FIX 0	66 ARCL X	116 SF 01	166 MOD	216 RCL Z
17 FS?C 22	67 LASTX	117 RDN	167 STO IND 00	217*LBL 07
18 STO 21	68 FRC	118 STO IND 00	168 DSE 00	218 RCL IND X
19 "FOLGENR: "	69 E1	119 DSE 00	169 RDN	219 46
20 ARCL 21	70 *	120 GTO 02	170 INT	220 X=Y?
21 ACA	71 ARCL X	121*LBL 03	171 STO IND 00	221 XEQ 00
22 ADV	72 GTO 00	122 XEQ 04	172 DSE 00	222 RDN
23 20	73*LBL 05	123 X=0?	173 RDN	223 45
24 STO 00	74 SF 02	124 GTO 00	174 INT	224 X=Y?
25 RCL 22	75 3	125 STO IND 00	175 STO IND 00	225 GTO 05
26*STOFLAG	76*LBL 01	126 DSE 00	176 DSE 00	226 RDN
27 CF 23	77 XEQ 04	127 GTO 03	177 "A"	227 X=0?
28 "ZANL?-ALPHA?"	78 STO IND 00	128 GTO 00	178 FS? 03	228 GTO 04
29 TONE 7	79 RCL Z	129*LBL 04	179 ASTO X	229 48
30 PROMPT	80 DSE 00	130 -1	180 FC? 03	230 -
31 FS? 23	81 DSE X	131 AROT	181 9	231 STO IND Y
32 GTO 00	82 GTO 01	132 ATOX	182 STO IND 00	232 RDN
33 FC?C 22	83 XEQ 04	133 RTN	183 FS? 03	233 DSE X
34 GTO 13	84 48	134*LBL 13	184 GTO 14	234 GTO 07
35 ACX	85 X=Y?	135 RCL 22	185 3	235 GTO 14
36 ADV	86 GTO 02	136*STOFLAG	186 RCL 00	236*LBL 00
37 ADV	87 RDN	137 CLA	187 +	237 "B"
38 "E++B"	88 STO IND 00	138 CLST	188 E3	238 ASTO IND Z
39 RCL I	89 DSE 00	139 RTN	189 /	239 RCL Z
40*STOFLAG	90 GTO 02	140 GTO a	190 20	240 DSE X
41 RDN	91*LBL 00	141*LBL 00	191 +	241 RCL IND X
42 CLA	92 FC? 23	142 FS? 03	192 FC? 02	242 ENTER+
43 ARCL X	93 GTO 02	143 GTO 05	193 GTO 07	243 RTN
44 12	94 AOFF	144 FC? 00	194 2	244*LBL 04
45 ALENG	95 ASTO 20	145 GTO 04	195 X<Y	245 "A"
46 X>Y?	96 CLA	146 45	196*LBL 06	246 ASTO IND Y
47 GTO 05	97 ARCL 20	147 STO IND 00	197 RCL IND X	247 GTO 14
48 RCL Z	98 ACA	148 DSE 00	198 48	248*LBL 05
49 X<0?	99 ADV	149*LBL 04	199 -	249 "D"
50 SF 05	100 ADV	150 RCL 00	200 STO IND Y	250 ASTO IND Z

251 RCL Z	285 FS? 03	319 64	353*LBL 11	387 RCL 22
252 DSE X	286 GTO 00	320 -	354 RCL IND Y	388 STOFFLAG
253 GTO 07	287 .02	321 X>0?	355 +	389 CF 00
254*LBL 14	288 +	322 9	356 DSE Y	390 CF 01
255 2	289 1	323 X<0?	357 GTO 11	391 CF 22
256 RCL 00	290 STO 00	324 16	358 SF 04	392 BEEP
257 3	291 RDN	325 +	359 XEQ 00	393 ADV
258 +	292 GTO 09	326 +	360*LBL 00	394 ADV
259*LBL 00	293*LBL 00	327 FS?C 04	361 RCL X	395 ADV
260 RCL IND X	294 RCL 00	328 RTN	362 256	396 ISG 21
261 9	295 3	329 STO IND 00	363 ST/ Z	397 *-*
262 X<Y?	296 +	330 RDN	364 MOD	398 RCL 21
263 GTO 00	297 1 E3	331 ISG 00	365 +	399 RCL 22
264 RDN	298 /	332 *-*	366 INT	400 CLRG
265 RDN	299 +	333 ISG X	367 FS?C 04	401 STO 22
266 DSE X	300 1	334 GTO 09	368 RTN	402 RDN
267 DSE Y	301 STO 00	335 FC?C 03	369 SF 17	403 STO 21
268 GTO 08	302 RDN	336 GTO 04	370 .	404 CLX
269 GTO 04	303*LBL 09	337 INT	371 ACCOL	405 CLA
270*LBL 00	304 CLA	338 .02	372 RDN	406 GTO b
271 RDN	305 ARCL IND X	339 +	373 128	407*LBL 04
272 55	306 ISG X	340*LBL 10	374 RCL 00	408 SF 25
273 +	307 GTO 00	341 RCL IND X	375 +	409 RCL IND X
274 .	308 GTO 04	342 STO IND 00	376 XEQ 05	410 X=0?
275 X<Y	309*LBL 00	343 RDN	377 XEQ 05	411 ACCOL
276 BLDSPEC	310 ARCL IND X	344 ISG 00	378 RCL 00	412*LBL 05
277 STO IND Y	311 .	345 *-*	379 E-3	413 XTOA
278 RDN	312 SF 04	346 ISG X	380 *	414*LBL 00
279 DSE X	313 XEQ 00	347 GTO 10	381 1	415 OUTA
280 DSE Y	314*LBL 00	348*LBL 04	382 +	416 CLA
281 GTO 08	315 ATOX	349 DSE 00	383*LBL 12	417 RDN
282*LBL 04	316 16	350 RCL 00	384 XEQ 04	418 END
283 FIX 0	317 ST* Z	351 RCL IND X	385 ISG X	739 Bytes
284 RCL 00	318 RDN	352 DSE Y	386 GTO 12	

### 2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "ZFDAbc"

Programmbedienung: wie "ZFDABC"

- SIZE: könnte 000 sein.

XEQ "ZFDAbc"



01*LBL "ZFDAbc"	12 .	23 PROMPT	34 ASTO X	45 X<Y
02*LBL 00	13 "FOLGENR: "	24 FS? 23	35 CLA	46 SF 22
03 RCLFLAG	14 ARCL X	25 GTO 02	36 ARCL X	47 GTO 01
04 .	15 ACA	26 FC?C 22	37 ACA	48*LBL 04
05 CF 22	16 ADV	27 GTO 04	38 ADV	49 Rf
06 "FOLGENR= 0 ?"	17 RCL Z	28 ACX	39*LBL 03	50 STOFFLAG
07 TONE 7	18 STOFFLAG	29 ADV	40 BCXS	51 CLA
08 PROMPT	19 CF 23	30 GTO 03	41 CHS	52 CLX
09*LBL 01	20 X<Y	31*LBL 02	42 BCO	53 RTN
10 FIX 0	21 "ZAHL?-ALPHA?"	32 ROFF	43 ADV	54 GTO 00
11 FC?C 22	22 TONE 7	33 .	44 ADV	55 END 125 Bytes

- Hinweis zu "ZALNBC" im Vergleich zu "ZFDABC": Es wird angenommen, die Programme "ZALN.." -BC werden vorwiegend für die Anfertigung von Barcodes für direkte oder Programmzeilen-Eingabe benutzt. Bei diesen Programmen ist deswegen die führende Null vor dem Dezimaltrennzeichen weggelassen worden; also nicht 0.12, sondern nur .12. Die Programme "ZFDA.." -BC lesen dagegen die Zahlen vorwiegend in Register ein. Hier wurde deswegen im Interesse der Programmverkürzung auf das Entfernen der führenden Null verzichtet.

### 3. Arbeiten mit Barcodes für sequentielle Daten

Bei Eingabe von sequentiellen Daten können Zahlen und Alphadaten vermischt eingelesen werden. Die Regist-Steuerzahl bbb.eee und die Folgenummer der Reihe sind voneinander unabhängig, sie stehen jedoch in einer Wechselbeziehung. Die Folgenummer der ersten sequentiellen BC-Reihe muß immer mit Null beginnen. Die erste Registerzahl bbb kann irgendeine zulässige Adresse sein. Dies, und das ist wieder ein Vorteil von Barcodes, kann auch ein Register > 99 sein. Es muß hier nicht, wie bei Eingabe über das Tastenfeld, der indirekte Weg gewählt werden. Es kann direkt eingelesen werden.

Die sequentielle Einlesung bricht ab, wenn eee größer als bbb wird. Haben Sie eine größere Anzahl Register definiert als Folgebarcodes da sind, können Sie nach Einlesen der letzten Reihe die nächste Aufforderung "W: SC 10 nn" mit 'CLX/A' abbrechen. Umgekehrt ist es nicht so einfach. Haben Sie weniger Register definiert, als BC-Reihen vorhanden sind, so können Sie nach Abbruch der Eingabeschleife die richtige Anzahl Register erneut eingeben und mit 'SST' zur noch offenen Folgereihe gehen und anschließend die restlichen Daten einlesen. Eine gute Anwendungsmöglichkeit für diese Barcodes ist die Herstellung von sequentiellen Datenreihen für Steuerzahlen zu Schriftfahnenprogrammen wie dies im Buch von 'Dearing' beschrieben wird.

- Einschränkung für Alphadaten: Synthetische Textzeilen werden zwar als sequentielle Barcodes angefertigt, enthalten diese jedoch ASCII-Zeichen zwischen 240 bis 255 werden die Daten nicht oder nicht vollständig in Register eingelesen.

- Möglichkeiten mit der Plottermodul - Funktion 'BCREGX': Wer ein Plottermodul hat, kann für das Einlesen der Dezimalwerte in Register sequentielle Zahlenbarcodes verwenden. Die vorstehend beschriebene Beschränkung für Alphadaten kann damit gewissermaßen 'umschiff't werden. Es wird damit möglich, z.B. Dezimalwerte von 'BLDSPEC'-Zahlen, welche Bytes 240-255 enthalten, ohne den in Kapitel 2 beschriebenen Umweg über eine Programmzeile ins Alpha- oder Datenregister zu speichern. Nehmen wir als Beispiel den quergestellten Buchstaben "A" aus dem Programm für Querbeschriftung im 'Dearing'. Das dort vorgestellte Programm geht davon aus, daß die Bytekombinationen für die 'BLDSPEC'-Zeichen in Registern vorhanden sind. Die 'BLDSPEC'-Zahlen lauten: 65 65 127 65 65 34 28 - dezimal: 17 6 15 248 48 81 28. Eine solche synthetische Textzeile ließe sich nicht in das Alpha- oder ein Datenregister einlesen: 1. weil sie aus 7 Zeichen besteht und 2. weil ab Byte 248 die folgenden Bytes der Reihe nicht ins Register gelangen. Mit Barcodes ist die externe Speicherung aller 'BLDSPEC'-Zahlen und das Einlesen der Werte in alle 64 Register unproblematisch. Für jede einzelne Dezimalzahl von 17 bis 28 wird eine sequentielle Zahlen-BC-Reihe angefertigt, mit Folgenummern 0 bis 6. Die folgende 'BLDSPEC'-Zahlenkombination für die nächste Registereinlesung folgt dann anschließend.

Ein Einleseprogramm könnte so aussehen: Bitte 'SIZE 072' einstellen und das Programm "LD" Seite 113 einlesen.

XEQ "LD"





Programm: "ZALNBC" - Benoetigte PRGM-Reg: 97

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-4



Reihe 3: Zeilen 5-11



Reihe 4: Zeilen 12-17



Reihe 5: Zeilen 18-24



Reihe 6: Zeilen 24-30



Reihe 7: Zeilen 30-39



Reihe 8: Zeilen 39-45



Reihe 9: Zeilen 46-53



Reihe 10: Zeilen 53-59



Reihe 11: Zeilen 60-67



Reihe 12: Zeilen 67-73



Reihe 13: Zeilen 74-82



Reihe 14: Zeilen 83-89



Reihe 15: Zeilen 90-97



Reihe 16: Zeilen 97-104



Reihe 17: Zeilen 105-111



Programm: "ZALNBC" - Fortsetzung

Reihe 18: Zeilen 111-115



Reihe 19: Zeilen 116-121



Reihe 20: Zeilen 122-130



Reihe 21: Zeilen 130-138



Reihe 22: Zeilen 138-145



Reihe 23: Zeilen 146-156



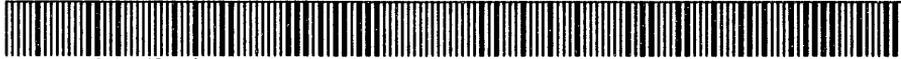
Reihe 24: Zeilen 157-164



Reihe 25: Zeilen 164-170



Reihe 26: Zeilen 171-179



Reihe 27: Zeilen 179-187



Reihe 28: Zeilen 188-194



Reihe 29: Zeilen 195-200



Reihe 30: Zeilen 200-207



Reihe 31: Zeilen 208-216



Reihe 32: Zeilen 217-224



Reihe 33: Zeilen 225-231



Reihe 34: Zeilen 232-240



Reihe 35: Zeilen 240-248



Programm: "ZALNBC" - Fortsetzung

Reihe 36: Zeilen 248-257



Reihe 37: Zeilen 257-264



Reihe 38: Zeilen 264-273



Reihe 39: Zeilen 273-280



Reihe 40: Zeilen 281-289



Reihe 41: Zeilen 290-297



Reihe 42: Zeilen 297-303



Reihe 43: Zeilen 304-312



Reihe 44: Zeilen 312-318



Reihe 45: Zeilen 319-327



Reihe 46: Zeilen 327-332



Reihe 47: Zeilen 333-337



Reihe 48: Zeilen 338-348



Reihe 49: Zeilen 348-355



Reihe 50: Zeilen 356-364



Reihe 51: Zeilen 365-374



Reihe 52: Zeilen 375-381



Programm: "ZALNBc" - Benoetigte PRGM-Reg: 76

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-11



Reihe 3: Zeilen 12-17



Reihe 4: Zeilen 18-26



Reihe 5: Zeilen 27-33



Reihe 6: Zeilen 34-40



Reihe 7: Zeilen 40-48



Reihe 8: Zeilen 49-56



Reihe 9: Zeilen 57-62



Reihe 10: Zeilen 63-70



Reihe 11: Zeilen 71-77



Reihe 12: Zeilen 77-85



Reihe 13: Zeilen 85-92



Reihe 14: Zeilen 92-99



Reihe 15: Zeilen 100-107



Reihe 16: Zeilen 107-116



Reihe 17: Zeilen 117-126



# Programm: "ZALNBc" - Fortsetzung

Reihe 18: Zeilen 126-132



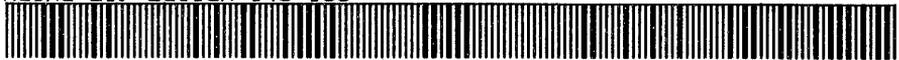
Reihe 19: Zeilen 133-139



Reihe 20: Zeilen 140-148



Reihe 21: Zeilen 149-156



Reihe 22: Zeilen 157-162



Reihe 23: Zeilen 162-169



Reihe 24: Zeilen 170-179



Reihe 25: Zeilen 179-186



Reihe 26: Zeilen 186-192



Reihe 27: Zeilen 193-201



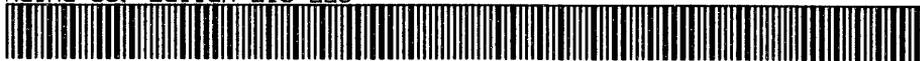
Reihe 28: Zeilen 202-209



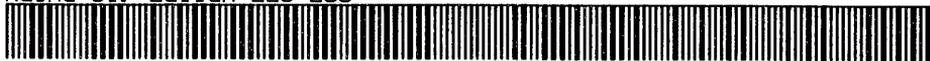
Reihe 29: Zeilen 210-218



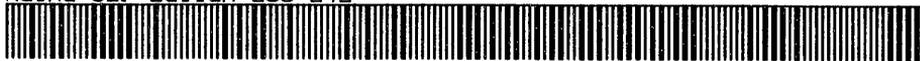
Reihe 30: Zeilen 219-226



Reihe 31: Zeilen 226-235



Reihe 32: Zeilen 235-242



Reihe 33: Zeilen 243-250



Reihe 34: Zeilen 251-259



Reihe 35: Zeilen 259-265



Programm: "ZALNbc" - Fortsetzung

Reihe 36: Zeilen 265-274



Reihe 37: Zeilen 274-280



Reihe 38: Zeilen 281-288



Reihe 39: Zeilen 289-298



Reihe 40: Zeilen 299-307



Reihe 41: Zeilen 308-313



Programm: "ZALNbc" - Benoetigte PRGM-Reg: 6

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-18



Programm: "ZXRADB" - Benoetigte PRGM-Reg: 26

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-4



Reihe 3: Zeilen 5-7



Reihe 4: Zeilen 7-8



Reihe 5: Zeilen 8-13



Programm: "ZXRADB" - Fortsetzung

Reihe 6: Zeilen 13-18



Reihe 7: Zeilen 19-25



Reihe 8: Zeilen 25-31



Reihe 9: Zeilen 32-36



Reihe 10: Zeilen 37-45



Reihe 11: Zeilen 45-52



Reihe 12: Zeilen 53-56



Reihe 13: Zeilen 56-61



Reihe 14: Zeilen 62-68



Programm: "KFEXBC" - Benoetigte PRGM-Reg: 37

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 10-17



Reihe 4: Zeilen 18-26



Reihe 5: Zeilen 26-32



Reihe 6: Zeilen 33-40



Reihe 7: Zeilen 40-46



Programm: "KFEXBC" - Fortsetzung

Reihe 8: Zeilen 47-53



Reihe 9: Zeilen 53-60



Reihe 10: Zeilen 61-69



Reihe 11: Zeilen 70-78



Reihe 12: Zeilen 79-86



Reihe 13: Zeilen 86-91



Reihe 14: Zeilen 91-97



Reihe 15: Zeilen 98-104



Reihe 16: Zeilen 105-114



Reihe 17: Zeilen 115-123



Reihe 18: Zeilen 124-133



Reihe 19: Zeilen 133-138



Reihe 20: Zeilen 139-145



Programm: "ZFDABC" - Benoetigte PRGM-Reg: 127

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-4



Reihe 3: Zeilen 4-10



# Programm: "ZFDABC" - Fortsetzung

Reihe 4: Zeilen 11-17



Reihe 5: Zeilen 18-23



Reihe 6: Zeilen 23-25



Reihe 7: Zeilen 26-31



Reihe 8: Zeilen 31-35



Reihe 9: Zeilen 35-39



Reihe 10: Zeilen 39-43



Reihe 11: Zeilen 43-49



Reihe 12: Zeilen 49-56



Reihe 13: Zeilen 57-64



Reihe 14: Zeilen 65-72



Reihe 15: Zeilen 72-78



Reihe 16: Zeilen 79-87



Reihe 17: Zeilen 87-94



Reihe 18: Zeilen 94-102



Reihe 19: Zeilen 103-109



Reihe 20: Zeilen 110-118



Reihe 21: Zeilen 118-124



Programm: "ZFDABC" - Fortsetzung

Reihe 22: Zeilen 125-132



Reihe 23: Zeilen 133-139



Reihe 24: Zeilen 140-143



Reihe 25: Zeilen 144-151



Reihe 26: Zeilen 151-157



Reihe 27: Zeilen 158-165



Reihe 28: Zeilen 166-173



Reihe 29: Zeilen 173-179



Reihe 30: Zeilen 180-183



Reihe 31: Zeilen 184-191



Reihe 32: Zeilen 191-200



Reihe 33: Zeilen 201-207



Reihe 34: Zeilen 208-215



Reihe 35: Zeilen 216-223



Reihe 36: Zeilen 223-229



Reihe 37: Zeilen 230-238



Reihe 38: Zeilen 239-247



Reihe 39: Zeilen 248-254



Programm: "ZFDABC" - Fortsetzung

Reihe 40: Zeilen 255-261



Reihe 41: Zeilen 262-269



Reihe 42: Zeilen 270-278



Reihe 43: Zeilen 278-284



Reihe 44: Zeilen 285-292



Reihe 45: Zeilen 292-299



Reihe 46: Zeilen 300-310



Reihe 47: Zeilen 311-318



Reihe 48: Zeilen 319-326



Reihe 49: Zeilen 327-334



Reihe 50: Zeilen 335-345



Reihe 51: Zeilen 346-354



Reihe 52: Zeilen 355-360



Reihe 53: Zeilen 361-369



Reihe 54: Zeilen 370-378



Reihe 55: Zeilen 379-386



Reihe 56: Zeilen 387-394



Reihe 57: Zeilen 394-401



Programm: "ZFDABC" - Fortsetzung

Reihe 58: Zeilen 402-409



Reihe 59: Zeilen 410-415



Reihe 60: Zeilen 416-425



Reihe 61: Zeilen 425-431



Reihe 62: Zeilen 431-439



Reihe 63: Zeilen 439-447



Reihe 64: Zeilen 447-454



Reihe 65: Zeilen 455-462



Reihe 66: Zeilen 463-470



Reihe 67: Zeilen 471-480



Reihe 68: Zeilen 481-490



Reihe 69: Zeilen 491-492



Programm: "ZFDABc" - Benoetigte PRGM-Reg: 106

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-10



Reihe 3: Zeilen 11-11



Reihe 4: Zeilen 12-18



# Programm: "ZFDABc" - Fortsetzung

Reihe 5: Zeilen 19-21



Reihe 6: Zeilen 21-28



Reihe 7: Zeilen 28-29



Reihe 8: Zeilen 30-36



Reihe 9: Zeilen 37-40



Reihe 10: Zeilen 41-48



Reihe 11: Zeilen 49-58



Reihe 12: Zeilen 59-65



Reihe 13: Zeilen 65-73



Reihe 14: Zeilen 74-80



Reihe 15: Zeilen 81-87



Reihe 16: Zeilen 88-95



Reihe 17: Zeilen 95-103



Reihe 18: Zeilen 104-110



Reihe 19: Zeilen 110-118



Reihe 20: Zeilen 118-125



Reihe 21: Zeilen 125-132



Reihe 22: Zeilen 132-140



Programm: "ZFDABc" - Fortsetzung

Reihe 23: Zeilen 141-147



Reihe 24: Zeilen 148-157



Reihe 25: Zeilen 158-164



Reihe 26: Zeilen 164-172



Reihe 27: Zeilen 172-179



Reihe 28: Zeilen 180-187



Reihe 29: Zeilen 188-196



Reihe 30: Zeilen 197-204



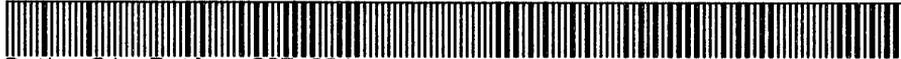
Reihe 31: Zeilen 204-211



Reihe 32: Zeilen 211-218



Reihe 33: Zeilen 219-226



Reihe 34: Zeilen 227-234



Reihe 35: Zeilen 235-241



Reihe 36: Zeilen 241-249



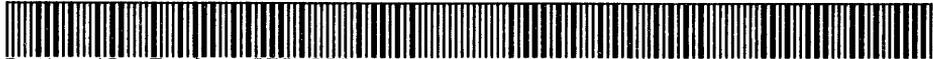
Reihe 37: Zeilen 249-257



Reihe 38: Zeilen 258-267



Reihe 39: Zeilen 267-276



Reihe 40: Zeilen 276-283



# Programm: "ZFDABc" - Fortsetzung

Reihe 41: Zeilen 284-292



Reihe 42: Zeilen 292-302



Reihe 43: Zeilen 303-310



Reihe 44: Zeilen 311-317



Reihe 45: Zeilen 318-327



Reihe 46: Zeilen 328-335



Reihe 47: Zeilen 335-342



Reihe 48: Zeilen 343-350



Reihe 49: Zeilen 351-358



Reihe 50: Zeilen 358-364



Reihe 51: Zeilen 365-373



Reihe 52: Zeilen 373-379



Reihe 53: Zeilen 380-387



Reihe 54: Zeilen 388-396



Reihe 55: Zeilen 396-403



Reihe 56: Zeilen 404-411



Reihe 57: Zeilen 412-418



Programm: "ZFDabc" - Benoetigte PRGM-Reg: 18

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-6



Reihe 3: Zeilen 6-13



Reihe 4: Zeilen 13-15



Reihe 5: Zeilen 16-21



Reihe 6: Zeilen 21-24



Reihe 7: Zeilen 25-32



Reihe 8: Zeilen 33-41



Reihe 9: Zeilen 42-50



Reihe 10: Zeilen 51-55



## Kapitel 4

### ANWEISUNGEN, BEFEHLE, FUNKTIONEN UND SYNTHETISCHES ZUM PROGRAMMIEREN

#### I. 1-Byte-Tastefeldfunktionen-Barcodes

Von diesen Funktionen gibt es nur 15 und zwar die Zahlen 0 bis 9, Dezimalzeichen = 10, 'EEX' = 11, 'CHS' = 12, und als einzige nicht programmierbare Funktion 'DEL\_\_' = 13 sowie 'R/S' = 14.

##### 1. Aufbau, Indikator und Spiegelprüfsumme

Der Aufbau und der Indikator dieses Typs ist durch die Spiegelprüfsumme festgelegt, welche bereits in Kapitel 1 besprochen wurde. Wir wiederholen: ohne Anfangs- und Endstriche lautet die BC-Reihe für z.B. '1' binär 1000 0001. Die Information ist in Nybble 2, hier = 0001 dargestellt; der Indikator und die Prüfsumme = 1000 in Nybble 1. Diese Bits sind spiegelbildlich zu Nybble 2 angeordnet. Nun gibt es pro Nybble die Zahl 15, sodaß eigentlich 16 Funktionen da sein könnten. 15 bewirkt die gleiche Funktion 'R/S-STOP' wie 14.

##### 2. Herstellen von 1-Byte-Tastefeldfunktionen auf dem Thermodrucker

Auch diese Programme sind dreifach vorhanden:

###### 2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "1BYTBC"

###### Programmbedienung:

- SIZE: könnte 000 sein.

XEQ "1BYTBC"



Rechtsbündig wird zuerst ausgedruckt: 1-B-TASTENBC - dann folgt die Frage: - "1 BYTE 0-14?": Geben Sie nun die Kennzahl der gewünschten 1-Byte-Funktion ein - 'R/S'. Geben Sie eine Zahl >15 ein erfolgt der Hinweis: "B nicht 0-14" und das Programm beginnt von vorn. Ansonsten wird die Kennzahl, die entsprechende Funktion und daran anschließend die BC-Reihe gedruckt.

Das Programmlisting steht auf Seite 131.

Einige Erläuterungen: In diesem Programm sind für die 0-14 Funktionen 0-14 Marken nötig. Bei Verwendung von 'LBL 00' bis 'LBL 14' würde die jetzige Startmarke für den BC-Ausdruck 'LBL 00' in Zeile 88 die Marke 15, also ein Langform-LBL werden. Diese Marke würde dann anstelle der jetzt gewählten 'GTO 00' Zeilen 31, 34, 37 usw. mit 'GTO 15' angesprungen werden. Dies erforderte einige Bytes mehr Speicherplatz als die jetzige Lösung, bei der einfach die Eingabekennzahl in den Zeilen 19 und 20 um '1' erhöht wird. So werden aus den Kennzahleingaben 0 = 'LBL 01' - 1 = 'LBL 02' usw. Vorteil, es können weitestgehend Kurzform-Marken verwendet werden.

###### 2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "1BYTBc"

###### Programmbedienung: genau wie "1BYTBC"

- SIZE: keine Datenregister erforderlich.

XEQ "1BYTBc"



01*LBL "1BYTBC"	29*LBL 01	57 153	85 ACA	113 ARCL X
02*LBL a	30 .	58 GTO 00	86 ADV	114*LBL 01
03 "1-B-TASTENBC"	31 GTO 00	59*LBL 11	87 126	115 OUTA
04 ACA	32*LBL 02	60 "."	88*LBL 00	116 CLA
05 ADV	33 129	61 ACA	89 ADV	117 RDN
06 CF 22	34 GTO 00	62 ADV	90 SF 17	118 RTN
07 FIX 0	35*LBL 03	63 90	91 .	119*LBL 02
08*LBL b	36 66	64 GTO 00	92 ACCOL	120 127
09 "1 BYTE 0-14?"	37 GTO 00	65*LBL 12	93 RDN	121 X<>Y
10 TONE 7	38*LBL 04	66 "EXX"	94 128	122 X<=Y?
11 PROMPT	39 195	67 ACA	95 XEQ 00	123 GTO 00
12 FC?C 22	40 GTO 00	68 ADV	96 XEQ 00	124 -
13 GTO b	41*LBL 05	69 219	97 ADV	125 1
14 ABS	42 36	70 GTO 00	98 ADV	126 +
15 CLA	43 GTO 00	71*LBL 13	99 CF 17	127 .
16 ARCL X	44*LBL 06	72 "CHS"	100 CLST	128 1
17 ACA	45 165	73 ACA	101 BEEP	129 BLDSPEC
18 ADV	46 GTO 00	74 ADV	102 ADV	130 X<>Y
19 I	47*LBL 07	75 60	103 ADV	131 BLDSPEC
20 +	48 102	76 GTO 00	104 RTN	132 RTN
21 SF 25	49 GTO 00	77*LBL 14	105 GTO a	133*LBL 00
22 GTO IND X	50*LBL 08	78 "e"	106*LBL 00	134 .
23 "B nicht 0-14"	51 231	79 ACA	107 SF 25	135 X<>Y
24 ACA	52 GTO 00	80 ADV	108 X=0?	136 BLDSPEC
25 ADV	53*LBL 09	81 189	109 ACCOL	137 END
26 TONE 0	54 24	82 GTO 00	110 X=0?	285 Bytes
27 ADV	55 GTO 00	83*LBL 15	111 GTO 01	
28 GTO a	56*LBL 10	84 "R/S-STOP"	112 XEQ 02	

01*LBL "1BYTbc"	25 ADV	49 GTO 00	73 ACA	97 ADV
02*LBL a	26 TONE 0	50*LBL 08	74 ADV	98 ADV
03 "1-B-TASTENBC"	27 ADV	51 231	75 60	99 CF 17
04 ACA	28 GTO a	52 GTO 00	76 GTO 00	100 CLST
05 ADV	29*LBL 01	53*LBL 09	77*LBL 14	101 BEEP
06 CF 22	30 .	54 24	78 "e"	102 ADV
07 FIX 0	31 GTO 00	55 GTO 00	79 ACA	103 ADV
08*LBL b	32*LBL 02	56*LBL 10	80 ADV	104 RTN
09 "1 BYTE 0-14?"	33 129	57 153	81 189	105 GTO a
10 TONE 7	34 GTO 00	58 GTO 00	82 GTO 00	106*LBL 00
11 PROMPT	35*LBL 03	59*LBL 11	83*LBL 15	107 SF 25
12 FC?C 22	36 66	60 "."	84 "R/S-STOP"	108 X=0?
13 GTO b	37 GTO 00	61 ACA	85 ACA	109 ACCOL
14 ABS	38*LBL 04	62 ADV	86 ADV	110 X=0?
15 CLA	39 195	63 90	87 126	111 GTO 01
16 ARCL X	40 GTO 00	64 GTO 00	88*LBL 00	112 XTOA
17 ACA	41*LBL 05	65*LBL 12	89 ADV	113*LBL 01
18 ADV	42 36	66 "EXX"	90 SF 17	114 OUTA
19 I	43 GTO 00	67 ACA	91 .	115 CLA
20 +	44*LBL 06	68 ADV	92 ACCOL	116 RDN
21 SF 25	45 165	69 219	93 RDN	117 END
22 GTO IND X	46 GTO 00	70 GTO 00	94 128	256 Bytes
23 "B nicht 0-14"	47*LBL 07	71*LBL 13	95 XEQ 00	
24 ACA	48 102	72 "CHS"	96 XEQ 00	

2.3 Rechner HP 41 C - Drucker, Plottermodul PRGM "1BYTbc"

Programmbedienung: wie "1BYTBC"

- SIZE: auch hier 000 zufriedenstellend

XEQ "1BYTbc"



01*LBL "1BYTbc"	21 +	41 **	61*LBL 11	81 ACA
02*LBL a	22 SF 25	42 GTO 00	62 ". "	82 ADV
03 CLST	23 GTO IND X	43*LBL 05	63 ACA	83 " "
04 "1-B-TASTENBC"	24 "B nicht 0-14"	44 "\$"	64 ADV	84 GTO 00
05 ACA	25 ADV	45 GTO 00	65 "Z"	85*LBL 15
06 ADV	26 ACA	46*LBL 06	66 GTO 00	86 "R/S-STOP"
07 CF 22	27 ADV	47 " "	67*LBL 12	87 ACA
08 FIX 0	28 TONE 0	48 GTO 00	68 "EXX"	88 ADV
09*LBL b	29 ADV	49*LBL 07	69 ACA	89 "Σ"
10 "1 BYTE 0-14?"	30 GTO a	50 "f"	70 ADV	90*LBL 00
11 TONE 7	31*LBL 01	51 GTO 00	71 "R/S-STOP"	91 -1
12 PROMPT	32 CLA	52*LBL 08	72 GTO 00	92 BCO
13 FC?C 22	33 GTO 00	53 **	73*LBL 13	93 CLST
14 GTO b	34*LBL 02	54 GTO 00	74 "CHS"	94 CLA
15 ABS	35 **	55*LBL 09	75 ACA	95 BEEP
16 CLA	36 GTO 00	56 "0"	76 ADV	96 ADV
17 ARCL X	37*LBL 03	57 GTO 00	77 "<"	97 RTN
18 "t "	38 "B"	58*LBL 10	78 GTO 00	98 GTO a
19 ACA	39 GTO 00	59 **	79*LBL 14	99 END
20 1	40*LBL 04	60 GTO 00	80 "+"	219 Bytes

Auf das XF-Modul kann bei diesem Programm verzichtet werden, weil die Bytes der Einbytebefehle - mit Barcodes 0-255 in den Programmspeicher eingelesen - direkt in das Alpharegister gebracht und von dort mit PM-Funktion 'BCO' als Barcodes gedruckt werden. Die ASCII-Zeichen entsprechen den Dezimalwerten der zugehörigen LBL in den Programmen "1BYTBC(c)"; Zeile 35 "1BYTbc" = Zeile 33 "1BYTBC".

Noch ein Wort zu 'R/S-STOP' - Diese Funktion benutzten wir bereits als 2-Byte-Ausführung in Kapitel 2. Auch die Einbyteversion kann kein laufendes Programm anhalten. Ansonsten trifft auch alles hier zu, was in Kapitel 2 gesagt wurde.

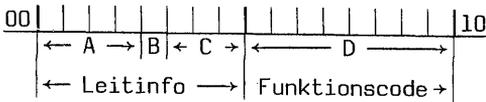
II. 2-Bytebarcodes - Tastenfeldfunktionen ohne Argument

In diesem Abschnitt werden alle mit 2-Bytebarcodes darstellbaren programmierbaren und nicht programmierbaren Tastenfeld- und XROM-Funktionen behandelt. Allen diesen Funktionen ist gemeinsam, daß sie entweder eine Einbytefunktion ohne Argument sind, z.B. 'BEEP' oder als Zweibytefunktion ein zugehöriges Argument anfordern, z.B. 'TONE\_'. Im Gegensatz hierzu stehen die Mehrbytebarcodes in Abschnitt III., die ein Argument bereits mit aufrufen. Beispiel: 'RCL\_' mit Zweibytebarcodes erfordert ein Argument wie 00 bis 99, X Y Z I L oder 'IND'. Dies muß dann als weitere BC-Reihe oder über das Tastenfeld eingegeben werden, während die Mehrbyte-BC im nächsten Abschnitt das Argument gleich beinhalten: 'RCL IND 09' ist dann eine Barcodereihe.

a. Programmierbare Rechnerfunktionen

Die Barcodes dieser Funktionen entsprechen bis auf die nicht programmierbaren Befehle dem 'CAT 3' des Rechners. Sie sind auch in den beiden Tastenfeldtafeln aufgeführt, welche dem Lesestift von HP beigegeben sind. Es wird angenommen, jeder Lesestiftbenutzer kennt diese dem Tastenfeld entsprechenden Barcodes.

## 1. Aufbau



A = 4-Bit Prüfsumme

B = 1-Bit Typindikator für 2-Byte-Nicht-XROM-Funktion

C = 3-Bit Indikator für diesen Zweibytetyp

D = Dezimalwert des Funktionscode

### 1.1 Der Typindikator

Der Typindikator ist auch hier im 2. Nybble des 1. Byte verschlüsselt. B auf Null gesetzt, teilt dem Rechner mit, es handelt sich um einen Nicht-XROM-2-Bytebarcode. C, ebenfalls 000 gibt bekannt, es handelt sich um Funktionen aus 'CAT 3'. D ist dann der Dezimalwert der entsprechenden Funktion aus der Byte-Tabelle.

### 1.2 Der Dezimalwert des Funktionscode

Der 'CAT 3' des Rechners beginnt bei Byte 64 = '+' und führt bis einschließlich Byte 143 = 'ADV' alle Einbyterechnerefunktionen auf. Es folgen auf 144 = 'RCL' zunächst bis 159 = 'TONE' 16 Zweibytefunktionen, die ein Argument erfordern. Die Bytes 160 bis 167 sind XROM-Funktionen, welche später behandelt werden. Es folgen weitere 6 Zweibytefunktionen von 168 bis 173. Das Dreibyte-'END' ist das Byte 192. Dann folgt auf 'verlorenem' Posten 'X<>\_', das Byte 206 und anschließend 'LBL' = 207. Mit 'GTO' = 208 und 'XEQ' = 224 ist die Liste erfüllt.

Das Byte 174 erfordert mit seiner Doppelfunktion zwingend ein Argument und wird im nächsten Abschnitt behandelt. Im Zusammenhang mit den aufgezählten Zweibyte-Funktionen wird für 'IND' eine eigene barcodierte Funktion eingesetzt, die anschließend behandelt wird.

### 1.3 Die 4-Bit Prüfsumme

Der Berechnungsgang für diese Prüfsumme ist deswegen sehr leicht, weil das Indikatornybble '0' ist. Der Dezimalwert des Funktionscode wird einfach durch 16 geteilt und der Rest hinzugezählt. Ist das Ergebnis >16 wird nochmal geteilt. Weil die Prüfsumme in Nybble 1 des ersten Byte enthalten ist, wird mit 16 multipliziert und es ist dies der Wert des 1. Byte. Beispiel: Funktion 'FC?' - Dezimalcode = 173:16=10 Rest 13 = 23:16=1 Rest 7 - Prüfsumme 8 x16 = Wert von Byte 1 = 128. Die zwei Bytes der Barcodereihe lauten: 128 173.

## 2. Herstellen von 2-Byte-Barcodes der programmierbaren Rechnerfunktionen

Die drei Programme folgen in der Systematik den bisherigen Programmen. Bei diesem Barcodetyp könnte das XF-Modul in ganz anderer Weise genutzt werden. Dies wird am Schluß des Programmlistings kurz betrachtet.

### 2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "2BYTBC"

#### Programmbedienung:

- SIZE: es wird nur das Register 00 benutzt

XEQ "2BYTBC"



- "FUNKTIONAME?": Geben Sie nun aus der untenstehenden Tabelle den Namen der wünschsten Funktion wie eine normale Alpha-Eingabe ein. Z.B. '10+X' = SHIFT 1, SHIFT 0, SHIFT ENTER+ und X. Dann 'R/S' drücken.
- "BYTEDEZIMAL?": Geben Sie nun den zugehörigen Dezimalcode ein, hier 87 'R/S'. Geben Sie irrtümllich einen nicht existierenden Dezimalcode ein, erfolgt der Hinweis: "BytDezNonexi".

Funktion	Dez										
+	64	CLA	135	FRC	185	P-R	78	SIN	89	X(=Y?	70
-	65	CLD	127	FS?	172	%	76	SIGN	122	X>Y?	69
*	66	CLRG	138	FS?C	170	%CH	77	SORT	82	X(<)	206
/	67	CLE	112	GRAD	130	PI	114	ST+	146	X(<)Y	113
1/X	96	CLST	115	GTO	208	PROMPT	142	ST-	147	XEQ	224
10+X	87	CLX	119	HMS	188	PSE	137	ST*	148	X!2	81
ABS	97	COS	90	HMS+	73	R↑	116	ST/	149	Y!X	83
ACOS	93	D-R	106	HMS-	74	R-D	187	STO	145		
ADV	143	DEC	95	HR	189	R-P	79	STOP	132		
AOFF	139	DEG	128	INT	184	RAD	129	TAN	91		
AON	140	DSE	151	ISG	158	RCL	144	TONE	159		
ARCL	155	END	192	LASTX	118	RDN	117	VIEW	152		
ASHF	136	ENG	158	LBL	207	RND	118	X=0?	103		
ASIN	92	ENTER↑	131	LN	80	RTN	133	X≠0?	99		
ASTO	154	E!X	85	LN1+X	181	SDEV	125	X<0?	102		
ATAN	94	E!X-1	88	LOG	86	SCI	157	X<=0?	123		
AVIEW	126	FACT	98	MEAN	124	SF	168	X>0?	100		
BEEP	134	FC?	173	MOD	75	Σ+	71	X=Y?	120		
CF	169	FC?C	171	OCT	111	Σ-	72	X*Y?	121		
CHS	84	FIX	156	OFF	141	ZREG	153	X/Y?	68		

Funktionsnamen und Bytedezimalwerte von Einbytebefehlen für Zweibytebarcodes.

01*LBL "2BYTBC"	27	STO 00	53	X<=Y?	79	129	105	OUTA
02*LBL 00	28	ACX	54	GTO 02	80	XEQ 04	106	CLA
03 FIX 0	29	X<=Y?	55	224	81	XEQ 04	107	RDN
04 CF 00	30	GTO 03	56	X*Y?	82	RCL 00	108	RTN
05 CF 23	31	159	57	GTO 03	83	XEQ 04	109*LBL 06	
06 SF 28	32	X<>Y	58*LBL 02		84	ADV	110	127
07 CF 29	33	X<=Y?	59	ADV	85	ADV	111	X<>Y
08 ADV	34	GTO 02	60	ADV	86	CF 17	112	X<=Y?
09 *FUNKTIONAME?"	35	167	61	16	87	CLST	113	GTO 07
10 TONE 7	36	X<>Y	62	RCL 00	88	BEEP	114	-
11 AON	37	X<=Y?	63	SF 00	89	ADV	115	1
12 STOP	38	GTO 03	64	XEQ 02	90	ADV	116	+
13 AOFF	39	173	65*LBL 02		91	RTN	117	.
14 FC? 23	40	X<>Y	66	RCL X	92	GTO 00	118	1
15 GTO 00	41	X<=Y?	67	16	93*LBL 03		119	BLDSPEC
16 ACA	42	GTO 02	68	ST/ Z	94	ADV	120	X<>Y
17 ADV	43	192	69	MOD	95	"BytDezNonexi"	121	BLDSPEC
18 63	44	X=Y?	70	+	96	TONE 0	122	RTN
19*LBL 01	45	GTO 02	71	INT	97	ACA	123*LBL 07	
20 CF 22	46	RDN	72	FS?C 00	98	ADV	124	.
21 *BYTEDEZIMAL?"	47	205	73	RTN	99	GTO 00	125	X<>Y
22 TONE 7	48	X<>Y	74	*	100*LBL 04		126	BLDSPEC
23 PROMPT	49	X<=Y?	75	SF 17	101	XEQ 06	127	END
24 FC?C 22	50	GTO 03	76	.	102	ARCL X	245	Bytes
25 GTO 01	51	208	77	ACCOL	103	RDN		
26 ABS	52	X<>Y	78	RDN	104*LBL 05			

## 2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "2BYTbc"

Programmbedienung: wie "2BYTEBC"

- SIZE: nur Register 00 erforderlich

XEQ "2BYTbc"



01*LBL "2BYTbc"	23 PROMPT	45 GTO 02	67 16	89 ADV
02*LBL 00	24 FC?C 22	46 RDN	68 ST/ Z	90 ADV
03 FIX 0	25 GTO 01	47 205	69 MOD	91 RTN
04 CF 00	26 ABS	48 X<>Y	70 +	92 GTO 00
05 CF 23	27 STO 00	49 X<=Y?	71 INT	93*LBL 03
06 SF 28	28 ACX	50 GTO 03	72 FS?C 00	94 ADV
07 CF 29	29 X<=Y?	51 208	73 RTN	95 "BytDezNonexi"
08 ADV	30 GTO 03	52 X<>Y	74 *	96 TONE 0
09 "FUNKTIONAME?"	31 159	53 X<=Y?	75 SF 17	97 ACA
10 TONE 7	32 X<>Y	54 GTO 02	76 .	98 ADV
11 ANM	33 X<=Y?	55 224	77 ACCOL	99 GTO 00
12 STOP	34 GTO 02	56 X*Y?	78 RDN	100*LBL 04
13 AOFF	35 167	57 GTO 03	79 129	101 XTOA
14 FC? 23	36 X<>Y	58*LBL 02	80 XEQ 04	102*LBL 05
15 GTO 00	37 X<=Y?	59 ADV	81 XEQ 04	103 OUTA
16 ACA	38 GTO 03	60 ADV	82 RCL 00	104 CLA
17 ADV	39 173	61 16	83 XEQ 04	105 RDN
18 63	40 X<>Y	62 RCL 00	84 ADV	106 END
19*LBL 01	41 X<=Y?	63 SF 00	85 ADV	215 Bytes
20 CF 22	42 GTO 02	64 XEQ 02	86 CF 17	
21 "BYTEDEZIMAL?"	43 192	65*LBL 02	87 CLST	
22 TONE 7	44 X=Y?	66 RCL X	88 BEEP	

- Kurze Betrachtung zu einer anderen Programmlösung mit XF-Modul: Die in der Liste der Einbytefunktionen aufgeführten Befehle und Dezimalcodes eignen sich bestens für das Anlegen eines ASCII-Files im erweiterten Speicher, z.B. mit Namen "2BF". In diesen File könnte jede Funktion als Record in der Form "+64" bis "Y↑X83" mit 'APPREC' eingegeben werden. Im Programm "2BYTEbc" könnte dann die Frage "BYTEDEZIMAL?" mit den zugehörigen Programmschritten wegfallen. Nach Eingabe des "FUNKTIONAME?", "+" oder "Y↑X" ins Alpharegister würde mit 'POSFL' und 'GETREC' die vollständige Kombination "+64" aus dem File "2BF" ins Alpharegister übertragen. Mit 'ANUM' könnte der Dezimalwert ins X-Register gebracht, und von dort weiter verarbeitet werden. Man würde bei dieser Lösung dann, wie gewohnt, nur mit der Funktionsbezeichnung arbeiten. Es ist leider nicht möglich, ASCII-Files als Barcodes zu erstellen und direkt in einen File einzulesen. Wohl können aber von den Funktionen Textzeilenbarcodes angefertigt werden und mit BC von der Funktion 'APPREC' nacheinander zügig in einen File eingelesen werden.

## 2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "2BYTbc"

Programmbedienung: wie "2BYTbc"

- SIZE: es werden keine Speicherregister benötigt

XEQ "2BYTbc"



### b. Die Funktion 'INDIREKT' als Barcode

Dieser Barcodetyp entspricht genau der indirekten Registeradressierung, um dort vorhandene Parameter für eine Zweibyt-Tastenfeldfunktion zu benutzen. Um einen Registerinhalt indirekt aufzurufen ist zu drücken 'RCL' 'SHIFT' und man sieht

01*LBL "2BYTbc"	17 ADV	33 GTO 02	49 GTO 03	65 CLST
02*LBL 00	18 63	34 167	50 208	66 CLA
03 FIX 0	19*LBL 01	35 X<>Y	51 X<>Y	67 BEEP
04 CF 00	20 CF 22	36 X<=Y?	52 X<=Y?	68 RTN
05 CF 23	21 "BYTEDEZIMAL?"	37 GTO 03	53 GTO 02	69 GTO 00
06 SF 28	22 TONE 7	38 173	54 224	70*LBL 03
07 CF 29	23 PROMPT	39 X<>Y	55 X*Y?	71 ADV
08 ADV	24 FC?C 22	40 X<=Y?	56 GTO 03	72 "BytDezNonexi"
09 "FUNKTIONAME?"	25 GTO 01	41 GTO 02	57*LBL 02	73 TONE 0
10 TONE 7	26 ABS	42 192	58 CLA	74 ACA
11 AON	27 ACX	43 X=Y?	59 XTOA	75 ADV
12 STOP	28 X<=Y?	44 GTO 02	60 -2	76 GTO 00
13 AOFF	29 GTO 03	45 RDN	61 BCCKSM	77 END
14 FC? 23	30 159	46 205	62 BCD	169 Bytes
15 GTO 00	31 X<>Y	47 X<>Y	63 ADV	
16 ACA	32 X<=Y?	48 X<=Y?	64 ADV	

'RCL IND\_\_' in der Anzeige. Das gleiche mit Einbyte- und INDirekt-Barcodes:

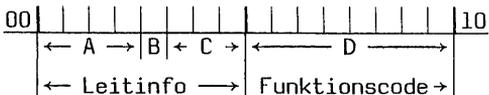


Die INDirekt-Barcodes können nicht wie die SHIFT-Taste benutzt werden. Diese Taste ist bei Barcode-Eingabe auch nicht nötig, weil geSHIFTete Funktionen wie 'GTO' direkt eingelesen werden. Barcodebeispiel der nächsten Reihe: 'GTO IND\_\_'



### 1. Aufbau, Typindikator, 4-Bit Prüfsumme und Funktionscode

Weil es sich nur um eine Funktion handelt, sind alle Kenndaten festgelegt.



A = 4-Bit Prüfsumme - 10

B = 1-Bit Indikator für 2-Byte-Nicht-XROM-Barcodes

C = 3-Bit auf 2 gesetzt für 'INDIREKT'

D = Dezimalwert 128 für den Funktionscode

Die Prüfsumme errechnet sich aus der Addition von B+C+D=130:16=8 Rest 2 also 10.

### 2. Herstellen von Barcodes für die 'IND'-Funktion

Alle drei Programme benötigen keine Datenregister. Sie brauchen mit den XEQ-Barcodes nur aufgerufen zu werden, und die Barcodes werden einschließlich 'Titel' erstellt.

#### 2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "INDBC"

XEQ "INDBC"



01*LBL "INDBC"	13 SF 17	25 CLST	37 CLA	49 1
02*LBL 00	14 .	26 BEEP	38 RDN	50 BLDSPEC
03 FIX 0	15 ACCOL	27 ADV	39 RTN	51 X<>Y
04 SF 28	16 162	28 ADV	40*LBL 03	52 BLDSPEC
05 CF 29	17 129	29 RTN	41 127	53 RTN
06 "IND"	18 XEQ 01	30 GTO 00	42 X<>Y	54*LBL 04
07 ACA	19 XEQ 01	31*LBL 01	43 X<=Y?	55 .
08 ADV	20 128	32 XEQ 03	44 GTO 04	56 X<>Y
09 128	21 XEQ 01	33 ARCL X	45 -	57 BLDSPEC
10 ACX	22 ADV	34 RDN	46 1	58 END
11 ADV	23 ADV	35*LBL 02	47 +	106 Bytes
12 ADV	24 CF 17	36 OUTA	48 .	

2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "INDBC"

XEQ "INDBC"



01*LBL "INDBC"	09 128	17 ACCOL	25 ADV	33*LBL 01
02*LBL 00	10 ACX	18 162	26 ADV	34 XTOA
03 FIX 0	11 ADV	19 129	27 CLST	35*LBL 02
04 SF 28	12 ADV	20 XEQ 01	28 BEEP	36 OUTA
05 CF 29	13 CLA	21 XEQ 01	29 ADV	37 CLA
06 "IND"	14 OUTA	22 128	30 ADV	38 RDN
07 ACA	15 SF 17	23 XEQ 01	31 RTN	39 END
08 ADV	16 .	24 CF 17	32 GTO 00	79 Bytes

2.3 Rechner HP 41 C - Drucker, nur Plottermodul PRGM "INDBC"

XEQ "INDBC"

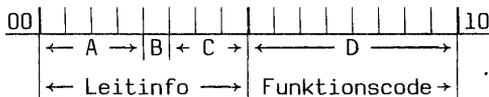


01*LBL "INDBC"	05 CF 29	09 128	13 BCO	17 CLA
02*LBL 00	06 "IND"	10 ACX	14 ADV	18 RTN
03 FIX 0	07 ACA	11 " "	15 ADV	19 GTO 00
04 SF 28	08 ADV	12 -2	16 CLST	20 END 45 Bytes

c. Nicht programmierbare Rechnerfunktionen ohne Argument

Es gibt 16 nicht programmierbare Rechnerfunktionen, die bis auf wenige ebenfalls dem 'CAT 3' entstammen. Solche, die ein Argument erfordern wie z.B. 'CAT' 'DELETE', 'SIZE' oder 'ASN' werden genau wie die Zweibytefunktionen allein aufgerufen; das Argument muß dann zusätzlich eingegeben oder eingelesen werden.

1. Aufbau



A = 4-Bit Prüfsumme

B = 1-Bit für Zweibyte-Nicht-XROM-BC

C = 3-Bit Indikator für diesen Typ

D = Dezimalwert des Funktionscode

### 1.1 Der Typindikator

Der Typindikator, wiederum im 2. Nybble von Byte 1 verschlüsselt, gibt dem Rechner bekannt: B = 0 - es ist ein Zweibyte-Nicht-XROM-BC; C = 4 - die Funktion ist nicht programmierbar; D = Dezimalwert entsprechend dem Funktionscode aus untenstehender Tabelle.

### 1.2 Der Dezimalcode der nicht programmierbaren Funktion

Als Funktionscode für diese nicht programmierbaren Befehle dienen die Bytes 0 bis 15 aus der Bytetabelle. Die Zuordnung finden Sie in der Auflistung bei der Programmbedienung zu "NPFBC".

### 1.3 Die 4-Bit Prüfsumme

Die Berechnung dieser Prüfsumme erfolgt nach gleichem Verfahren wie 'INDIREKT'. Beispiel: Funktion 'ASN' - Code = 15. Addition 15+4=19:16=1 Rest 3 - Prüfsumme = 4. Wert des ersten Byte: 4x16+4= 68 - die Bytes der Reihe lauten 68 15.



## 2. Herstellen von Barcodes nicht programmierbarer Rechnerfunktionen

Diese drei Programme entsprechen denen für die programmierbaren Funktionen. Der einzige Unterschied besteht im Indikator. Die programmierbaren Befehle können dabei eigenartigerweise ebensogut mit dem Indikator 4 hergestellt werden. Solche BC arbeiten genauso einwandfrei. Umgekehrt geht's nicht. NPFunktionen versagen den Dienst bei Indikator 0.

### Funktionsnamen und Bytedezimalwerte von nichtprogrammierbaren Rechnerfunktionen

<u>Funktion</u>	<u>DEZ</u>	<u>Funktion</u>	<u>DEZ</u>	<u>Funktion</u>	<u>DEZ</u>	<u>Funktion</u>	<u>DEZ</u>
CAT	0	CLP	4	SST	8	ALPHA	12
GTOZ	1	R/S	5	ON	9	PRGM	13
DEL	2	SIZE	6	PACK	10	USER	14
COPY	3	BST	7	DELETE	11	ASN	15

Erläuterung: 'GTOZ' = GTO,\_\_\_ - 'DEL' = Korrekturtaste und 'DELETE' = DEL\_\_\_.

### 2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "NPFBC"

#### Programmbedienung:

- SIZE: nur Register 00 wird belegt.

XEQ "NPFBC"



- "FUNKTIONAME?": Geben Sie hier aus der obenstehenden Auflistung Ihre Funktion in das Alpharegister ein. Sie brauchen nicht umzuschalten, der Rechner befindet sich im ALPHA-Modus. Anschließend 'R/S' drücken.

- "BYTEDEZIMAL?": Tasten Sie sodann den zugehörigen Dezimalcode ein - 'R/S'. Sollten Sie irrtümlich einen nicht zugelassenen Code eingeben, so erfolgt der Hinweis: "BytDezNonexi" und das Programm kehrt an den Anfang zurück.

### 2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "NPFBC"

#### Programmbedienung: und 'SIZE' wie "NPFBC"

XEQ "NPFBC"



2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "NPFbc"

Programmbedienung: wie "NPFBC"

- SIZE: 001 reicht aus

XEQ "NPFbc"



Anschließend folgen die drei PRGM-Listings und ein Kommentar zu "NPFbc".

01+LBL "NPFBC"	23 PROMPT	45 INT	67 RTN	89 RDN
02+LBL 00	24 FC?C 22	46 FS?C 00	68 GTD 00	90 RTN
03 FIX 0	25 GTD 01	47 RTN	69+LBL 03	91+LBL 07
04 CF 00	26 ABS	48 *	70 ADV	92 127
05 CF 23	27 STO 00	49 4	71 "BytDezNonexi"	93 X<Y
06 SF 28	28 ACX	50 +	72 TONE 0	94 X<=Y?
07 CF 29	29 X>Y?	51 SF 17	73 ACA	95 GTD 08
08 ADV	30 GTD 03	52 .	74 ADV	96 -
09 "FUNKTIONAME?"	31 ADV	53 ACCOL	75 GTD 00	97 1
10 TONE 7	32 ADV	54 RDN	76+LBL 04	98 +
11 AON	33 16	55 129	77 SF 25	99 .
12 STOP	34 4	56 XEQ 05	78 X=0?	100 1
13 AOFF	35 RCL 00	57 XEQ 05	79 ACCOL	101 BLDSPEC
14 FC? 23	36 +	58 RCL 00	80 X=0?	102 X<Y
15 GTD 00	37 SF 00	59 XEQ 04	81 GTD 06	103 BLDSPEC
16 ACA	38 XEQ 02	60 ADV	82+LBL 05	104 RTN
17 ADV	39+LBL 02	61 ADV	83 XEQ 07	105+LBL 08
18 15	40 RCL X	62 CF 17	84 ARCL X	106 .
19+LBL 01	41 16	63 CLST	85 RDN	107 X<Y
20 CF 22	42 ST/ Z	64 BEEP	86+LBL 06	108 BLDSPEC
21 "BYTEDEZIMAL?"	43 MOD	65 ADV	87 OUTA	109 END
22 TONE 7	44 +	66 ADV	88 CLA	209 Bytes

01+LBL "NPFbc"	19+LBL 01	37 SF 00	55 129	73 ACA
02+LBL 00	20 CF 22	38 XEQ 02	56 XEQ 05	74 ADV
03 FIX 0	21 "BYTEDEZIMAL?"	39+LBL 02	57 XEQ 05	75 GTD 00
04 CF 00	22 TONE 7	40 RCL X	58 RCL 00	76+LBL 04
05 CF 23	23 PROMPT	41 16	59 XEQ 04	77 SF 25
06 SF 28	24 FC?C 22	42 ST/ Z	60 ADV	78 X=0?
07 CF 29	25 GTD 01	43 MOD	61 ADV	79 ACCOL
08 ADV	26 ABS	44 +	62 CF 17	80 X=0?
09 "FUNKTIONAME?"	27 STO 00	45 INT	63 CLST	81 GTD 06
10 TONE 7	28 ACX	46 FS?C 00	64 BEEP	82+LBL 05
11 AON	29 X>Y?	47 RTN	65 ADV	83 XTOA
12 STOP	30 GTD 03	48 *	66 ADV	84+LBL 06
13 AOFF	31 ADV	49 4	67 RTN	85 OUTA
14 FC? 23	32 ADV	50 +	68 GTD 00	86 CLA
15 GTD 00	33 16	51 SF 17	69+LBL 03	87 RDN
16 ACA	34 4	52 .	70 ADV	88 END
17 ADV	35 RCL 00	53 ACCOL	71 "BytDezNonexi"	179 Bytes
18 15	36 +	54 RDN	72 TONE 0	

01*LBL "NPFbc"	11 AON	21 "BYTEDEZIMAL?"	31 XTOA	41*LBL 02
02*LBL 00	12 STOP	22 TONE 7	32 -2	42 "BytDezHexxi"
03 FIX 0	13 AOFF	23 PROMPT	33 BCKSM	43 ACA
04 CF 00	14 FC? 23	24 FC?C 22	34 BCO	44 ADV
05 CF 23	15 GTO 00	25 GTO 01	35 ADV	45 TONE 0
06 SF 28	16 ACA	26 ABS	36 ADV	46 GTO 00
07 CF 29	17 ADV	27 ACX	37 CLST	47 END
08 ADV	18 15	28 X>Y?	38 BEEP	118 Bytes
09 "FUNKTIONAME?"	19*LBL 01	29 GTO 02	39 RTN	
10 TONE 7	20 CF 22	30 "a"	40 GTO 00	

Zeile 30 führt das Byte 4, den Indikator. Zeile 31 bringt den Dezimalwert des Funktionscode nach ALPHA. Die 'BCKSM'-Funktion berechnet aus dem ABS-Wert in X (Zeile 32) für die so definierte Anzahl Bytes im Alpharegister die Prüfsumme. Steht nur ein Byte in Alpha, wie bei "2BYTbc", wird Indikator '0' angenommen. Stehen zwei Bytes in Alpha, wird die Prüfsumme aus beiden berechnet und das erste Byte, zunächst 4, gegen ein erstes Byte mit entsprechend um die Prüfsumme ergänztem Dezimalwert ausgetauscht.

#### d. Besondere Funktionen und deren Zweibytebarcodes

Unter II.a. programmierbare Rechnerfunktionen wurden die 'offiziellen' Befehle ab Byte 64 mit allen dazwischenliegenden 'Lücken' bis Byte 224 beschrieben. Dem Forscher stellt sich die Frage ob die anderen Bytes als Barcodes nicht auch etwas hergeben. Wir untersuchen dies und finden tatsächlich noch brauchbares. Die Bytes 00-28 ergeben als Zweibytebarcodes im wesentlichen gleiche Funktionen, wie die Einbyte-BC. 29-31 sind gewissermaßen 'Alpha-Diebe', die nur nach Vorbereitung des Register 'Q' zu gebrauchen sind. Dies Register wird im Abschnitt 'Der Q-Bote' mit beschrieben. Die Kurzform-Registerbefehle 32-63 zeigen sich ebenfalls unergiebig. Interessant wird es erst ab den 'XROM'.

#### 1.† -- Module Mathematik, Statistik und Vermessung

Diese bereits im Buch von W.C. Wickes "Synthetische Programmierung auf dem HP 41 C/CV" vorgestellte Funktion wird von K. Jarett in seinem "Synthetisches Programmieren auf dem HP 41 - leicht gemacht" wesentlich erweitert beschrieben. Sie ist nur für Benutzer der oben erwähnten Module von Interesse:



Dezimal: 160 160

K. Jarett beschreibt diese Funktion folgendermaßen: "Sie erscheint in der Anzeige in der Gestalt  $\uparrow$  und fordert Sie mit den Unterstrichen zur Eingabe zweier Ziffern auf. Für  $\uparrow$  00-99 erhalten Sie 'XROM 00,00' bis 'XROM 01,35'. Dabei entsprechen die Einträge 65-99 den Zahlen 'XROM 01,01' bis 'XROM 01,35'. Dies sind die für die 35 globalen Marken 'MATRIX' bis 'TANH' des Mathematikmoduls (die letzten 10 Marken werden nicht erfaßt) vorgewiesenen XROM-Zahlen bei Abwesenheit des Moduls. Im RUN-Modus haben Sie schnellen Zugriff auf diese Marken durch einfache Zahleneinträge. Im PRGM-Modus können Sie diese Marken als Unterprogramm Ihren Marken einfügen, und zwar auch in Abwesenheit des Moduls. Diese besondere Funktion nimmt sogar pseudo-indirekte Eingaben entgegen und hat damit einen doppelt so großen Wirkungsbereich: drücken Sie vor Zifferneingabe 'SHIFT' erhalten Sie zu  $\uparrow$  IND 00' bis  $\uparrow$  IND 99' die Zahlen 'XROM 02,00' bis 'XROM 03,35'. Darunter befinden sich für  $\uparrow$  IND 01' bis  $\uparrow$  IND 29' die zu den globalen Marken des Statistik-Moduls gehörigen Zahlen 'XROM 02,01' bis 'XROM 02,29' und für  $\uparrow$  IND 65' bis  $\uparrow$  IND 96' die zu den globalen Marken des Moduls für Vermessung gehörigen Zahlen 'XROM 03,01' bis 'XROM 03,32'. Für RUN- und PRGM-Modus gilt das oben Gesagte". - Zitat ende - geringfügig abgewandelt.

2. μ-- Module Finanzen, Netzwerkanalyse und Baustatik

μ--  Dezimal: 176 161

Auch diese Funktion wird von K. Jarett als Tastenzuweisung genannt: "Sie beschafft mit 'μ 00' bis 'μ 99' bzw. 'μ IND 00' bis 'μ IND 99' Zugang zu: 'XROM 04,00 - 04,63 / 05,00 - 05,35 / 06,00 - 06,63 und 07,00 - 07,55' und verhält sich in allen Punkten wie '↑\_\_'. Die Module Finanzen, Netzwerkanalyse und Baustatik können mit 'μ\_\_' so bequem genutzt werden, wie die Module für Statistik, Mathematik und Vermessung mit '↑\_\_'.

- Weitere bisher 'unbekannte Größen'

ØBEEP\_  Dezimal: 192 162

Der Eingabestrich fordert zur Alpha-Eingabe auf. Im RUN-Modus ist die Antwort auf alle möglichen Versionen 'NONEXISTENT'; im PRGM-Modus bewirkt eine Alpha-Eingabe 'XROM 08,00'.

XROM 12,00  Dezimal: 208 163

Dezimal 208 163 ruft im RUN-Modus 'NONEXISTENT' und im Alpha-Modus 'XROM 12,00' hervor - und zwar ohne zusätzliche Eingabe.

S\_  Dezimal: 224 164

Der Unterschied zu 'ØBEEP\_' besteht nur darin, daß 'XROM 16,00' nach Alpha-Eingabe aufgerufen wird.

XROM 20,00  Dezimal: 240 165

Ohne Eingabe reagiert dieser Barcode wie 208 163, jedoch mit 'XROM 20,00'.

---  Dezimal: 16 166

Bei Anwesenheit des Development-Moduls erscheinen im PRGM-Modus drei Trennstri- che, die auch in ein Programm zur Kennzeichnung eingelesen werden können, sonst erscheint 'XROM 24,00'.

Möglicherweise entdecken Sie mit diesen Barcodes und bestimmten Anwender-Modu- len oder Peripheriegeräten noch brauchbare Arbeitsverkürzungen, so wie diese mit der nächsten Funktion bereits von W.C Wickes vorgestellt wurde.

3. eØBEEP\_\_ - HP-IL-Funktionen

Diese Funktion ist als Tastenzuweisung zuerst von W.C. Wickes vorgestellt, dann von J.S. Dearing in seinem bereits erwähnten Buch erweitert und auch von K. Jarett behandelt worden. Sie gehört inzwischen zum synthetischen Repertoire für normale Programmierung aller Funktionen des HP-IL-Moduls. Die Arbeit ist leicht

eØBEEP\_\_  Dezimal: 32 167

Im RUN-Modus für Sofortausführung oder im PRGM-Modus zum Programmieren, die Barcodes einfach einlesen. Es erscheint 'eØBEEP\_\_'; dann die Eingabeauf- forderung mit dem numerischen Wert aus der auf Seite 142 stehenden Tabelle vor der Funktion beantworten. Z.B. für 'PRX' = 84. Aus dem '-MASS ST 1H' und dem '-CTL FNS' existieren Zahlen zwischen 00 - 41 entsprechend 'XROM 28,00-28,41'.

Die 'XROM 28,42-28,63' sind nicht belegt. Für '-PRINTER 2E' sind 'XROM 29,00-29,26' belegt. Dies entspricht den Zahleneinträgen 64-90. Danach kommen nur noch zwei: 92 = 'FINDID' - 'XROM 29,28', die doppelt vorhanden ist (# 28) und kurioserweise 94 = '7ISZI' aus dem Kartenleser, die wohl 'hängengeblieben' ist. Im Kartenleser hat diese Funktion die # 'XROM 30,30' hier 'XROM 29,30'. Die 'XROM 29,31-29,63' sind nicht belegt.

-Achtung. Zurückhaltung bei 03 'NEWM' und 71 'LIST' im RUN-Modus! Beide starten sofort, ohne Rücksicht und ohne Aufforderung zur Argumenteingabe. Dafür können diese beiden nicht programmierbaren Befehle über 'eGØBEEP\_\_' in gewissem Rahmen programmiert werden, wie weiter unten beschrieben wird. Die # 77 'PRP' nimmt eine kleine Sonderstellung ein. Auch dies erfahren Sie nachher. Was es mit den 'XROM'-Zahlen, den Dezimalzeilen 00, 25, 26, 64 und 90 der Tabelle auf sich hat, wird im Abschnitt f. 'XROM'-Funktionen beschrieben.

00 -MASS ST IH	20 WRTPV	40 STOPIO	60 XROM 28.60	80 PRREG
01 CREATE	21 WRTR	41 TRIGGER	61 XROM 28.61	81 PRREGX
02 DIR	22 WRTRX	42 XROM 28.42	62 XROM 28.62	82 PRE
03 NEWM	23 WRTS	43 XROM 28.43	63 XROM 28.63	83 PRSTK
04 PURGE	24 ZERO	44 XROM 28.44	64 -PRINTER 2E	84 PRX
05 READA	25 --	45 XROM 28.45	65 ACA	85 REGPLOT
06 READK	26 -CTL FNS	46 XROM 28.46	66 ACCHR	86 SKPCHR
07 READP	27 AUTOIO	47 XROM 28.47	67 ACCOL	87 SKPCOL
08 READR	28 FINDID	48 XROM 28.48	68 ACSPEC	88 STKPLOT
09 READRX	29 INA	49 XROM 28.49	69 ACX	89 FMT
10 READS	30 IND	50 XROM 28.50	70 BLSPEC	90 --
11 READSUB	31 INSTAT	51 XROM 28.51	71 LIST	91 XROM 29.27
12 RENAME	32 LISTEN	52 XROM 28.52	72 PRA	92 FINDID
13 SEC	33 LOCAL	53 XROM 28.53	73 XROM "PRAXIS"	93 XROM 29.29
14 SEEKR	34 MANTIO	54 XROM 28.54	74 PRBUF	94 7ISZI
15 UNSEC	35 OUTA	55 XROM 28.55	75 PRFLAGS	95 XROM 29.31
16 VERIFY	36 PWRDN	56 XROM 28.56	76 PRKEYS	96 XROM 29.32
17 WRTA	37 PWRUP	57 XROM 28.57	77 PRP	97 XROM 29.33
18 WRTK	38 REMOTE	58 XROM 28.58	78 XROM "PRPLOT"	98 XROM 29.34
19 WRTP	39 SELECT	59 XROM 28.59	79 XROM "PRPLOT"	99 XROM 29.35

- 'NEWM' und 'LIST' programmierbar - 'PRP' "NONEXISTENT"

Die Funktion 'NEWM' kann mit 'eGØBEEP 03' in den Programmspeicher eingelesen und ausgeführt werden. Steht 447 im X-Register, werden 167 Einträge definiert. Bei anderen größeren oder kleineren Zahlenwerten sind dies immer 7 Einträge. Eine so initialisierte Kassette wird normal beschrieben, und es kann wie üblich gearbeitet werden. Damit ist die programmgesteuerte Initialisierung einer Kassette ohne besondere Module möglich. Nach Abarbeiten von 'NEWM' läuft das Programm mit der nächsten Anweisung weiter.

Die Funktion 'LIST' kann ebenfalls in den Programmspeicher eingelesen werden.

01+LBL "LI"	Sie unterbricht ein laufendes Programm und listet ab	01+LBL "TI"
02 SF 00	der folgenden Zeile alle Befehle auf, bis 71 Zeilen	02 TIME
03 XEQ 00	gelistet sind oder ein 'END' angetroffen wird. Es soll-	03 " "
04+LBL 00	ten allerdings keine Null-Bytes (HEX 00) im Programm	04 ATIME
05 FS?C 00	vorhanden sein. Bei eingesetztem TIME-Modul oder Be-	05 " "
06 LIST	nutzung eines HP 41 CX führt die nebenstehende Routine	06 DATE
07 END	nur zum Ausdruck von Zeit und Datum wie zu Beginn eines	07 ADATE
	Programmlistings, und zwar innerhalb eines laufenden	08 PRA
	Programms. Dies erfordert 10 Bytes. Die gleiche Routine mit den	09 END
	üblichen Möglichkeiten programmiert, siehe rechts nebenstehend,	
	erfordert 17 Bytes. 'eGØBEEP 77' - 'PRP' funktioniert nicht, nur "NONEXISTENT".	

Im Abschnitt 6. Der 'Q-Bote' erfahren Sie mehr über 'PRP'. Die Bytes 168-174 der Bytetable haben dann wieder offizielle Funktionen und sind unexotisch.

4. -- 'GTO IND ' und 'XEQ IND '

Die nächste eigentlich nicht vorhandene Funktion basiert auf Byte 175 (SPARE). Als Barcode ruft es auf:



Man kann nun irgendeine Zahl zwischen 00-99 eingeben, und es wird im RUN-Modus ausgeführt oder als Programmzeile eingelesen 'GTO IND nn'. Drückt man nach '\_' die SHIFT-Taste so erscheint 'IND\_'. Die Eingabe einer Zahl zwischen 00-99 bewirkt jetzt 'XEQ IND nn'. Eigentlich sind die Funktionen 'GTO IND nn' und auch 'XEQ IND nn' dem Byte 174 aufgegeben. Dies wird bei den XEQ-Barcodes noch behandelt. Zweibytebarcodes der Bytes 176-191 und 193-204 erzeugen nichts brauchbares. 192 ist ja das offizielle 'END', 205 wird beim 'Q-Boten' besprochen. 206-208 sind offizielle Funktionen, 209-223 bringen nichts Ergiebigen. 224 ist das offizielle 'XEQ'-Byte; bis 239 hat dann nur das folgende Byte 229 Reize.

5. \$T+N IA\_\_\_ 'XEQ 00' bis 'XEQ e' = 000 - 255

Diese Funktion führt im RUN-Modus aus oder programmiert alle möglichen 'XEQ's für lokale Marken. Lesen Sie die Funktion möglichst im Programm-Modus ein:



(Außer Byte 229 bewirken auch die Bytes 231, 233 und 235 diesen Aufruf.) Sie können nun eine Zahl eingeben zwischen 000 = 'XEQ 00' bis 127 = 'XEQ e'. Die Zahlen 128-255 bewirken das gleiche wie 000-127. Auch sonst nur mit synthetischer Programmierung erreichbarer 'XEQ's' zwischen 112 = 'XEQ T' und Byte 122 = 'XEQ H' können mit diesen BC durch Eingabe der Byte-Dezimalzahl gemäß Bytetable programmiert werden. Wie die dazugehörigen 'LBL' und 'GTO's' entstehen, wird in Abschnitt e. beschrieben.

6. Der 'Q-Bote'

Der Q-Bote wurde von W.C. Wickes auch als Zweibytebarcode erstmals vorgestellt. K. Jarrett behandelt ihn ebenfalls. Das Anwenden von Barcodes macht ihn völlig überflüssig, weil 7 oder auch mehr beliebige Bytekombinationen ohne Vorbereitung direkt eingelesen werden können. Die Bytes 16-28 funktionieren als nicht programmierbare 2-Bytebarcodes alle als Q-Bote. Einige erfordern erst ein Argument. Wickes stellte 84/16 vor, wir benutzen 244 26, den Dezimalpunkt, der ja allein im PRGM-Speicher eine Null im X-Register ist.

Die Synthetik ist schnell erklärt: Bevor die ASCII-Zeichenkette eines LBL, GTO oder XEQ durch das zweite Drücken der Alphataste nach Eingabe abgeschlossen wird, bzw. eine Zahl vollständig eingegeben ist, werden die Zeichen im Register 'Q' gesammelt; sie werden dort in umgekehrter Folge dargestellt: 'NEBEL' als 'LEBEN'. Der Q-Bote holt nun den im Register 'Q' befindlichen Alphastring ab und bringt ihn in ein Programm, welches gerade editiert wird, als Umkehrtext ein. Das Zeichen des Q-Boten selbst, bei uns ".", steht dann in der Zeile vor dem überbrachten Text. Jede als 'LBL-', 'GTO'- oder 'XEQ'-'ALPHA', ob vorhanden oder 'NONEXISTENT' eingegebene Buchstaben/Zeichen-Kombination wird so in den Programmspeicher übertragen. Führen Sie nun im RUN-Modus aus, was nur den Zweck verfolgt, die Zeile nach 'Q' zu bringen: 'LBL''ALPHA' "A<>\*?%Σ" 'ALPHA'; es erscheint wieder ihr X-Inhalt. Jetzt 'GTO..' ausführen. Schalten Sie dann wieder in PRGM-Modus und lesen Sie Barcodeszeile Q-Bote auf Seite 144 oben ein. Sie sehen die Zeile 01 . ; gehen Sie nun mit 'SST' eine Zeile weiter, und es erscheint: 02 "A<>\*?%Σ" - nur der Alphastring ohne 'LBL' als reine

Q-Bote .



Dezimal: 244 26

Alphazeile im PRGM-Speicher, der vom Q-Boten aus dem Register 'Q' hierher gebracht wurde. Die Umkehrung fand zweimal statt: lx bei 'LBL''ALPHA' und noch einmal durch den Q-Boten, sodaß der Text korrekt steht. Nun läßt sich eine solche Zeile selbstverständlich auch direkt einlesen. Wenn es allerdings gelingt einen als Programmzeile nicht tastbaren Text nach 'Q' zu bringen, so kann dieser Text von maximal 7 Buchstaben mit Hilfe des Q-Boten in eine Programmzeile eingebracht werden. Löschen Sie nun die beiden Zeilen im Programmspeicher und lesen Sie das nachfolgende Kurzprogramm "Q" ein.

PROGRAMM = "Q" Benötigte Register: 2 - nicht PV-geschützt

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-4



Dieses Programm stellt Ihnen lediglich die beiden synthetischen Befehle 'RCL M' und 'STO Q' zur Verfügung. Werden nun ins Alpharegister maximal 7 Zeichen in rückwärtsgelesener Reihenfolgen eingegeben, so können diese anschließend mit 'RCL M' 'STO Q', wobei keine Umkehrung erfolgt, ins Register 'Q' geschrieben und von dort mit dem Q-Boten, der wieder umkehrt, in eine Programmzeile gebracht werden. Lesen Sie nun mit Hilfe der 'Zweibyte Funktions- und Druckerzeichen 0-127' die nicht direkt in Programmzeilen eintastbaren Zeichen DEZ 7 6 5 4 3 2 und 1 nacheinander ins Alpharegister ein. Bei 'SF 12' wird ausgedruckt: ↓Γ⊗α\*⊗\*. Führen Sie dann mit 'SST' = 'RCL M' und 'STO Q' aus, und schalten Sie in PRGM-Modus um. Tasten Sie sodann mit dem Lesestift die Barcodes 'Q-Bote .' ab. Es erscheint "04 ." und nach weiterem 'SST' sehen Sie die Zeile 05, die nach Umschalten in den RUN-Modus mit nochmals 'SST' ins Alpharegister geht und hier nach 'SF 12' den Spiegelausdruck der obigen Zeile ergibt: \*⊗←α⊗Γ↓. Auch diese Möglichkeit bleibt bei Anwendung von Barcodes unergiebig, weil alle nicht tastbaren Zeichen zwischen 1-126 mit den 'Zweibyte-Zeichen 0-127' auch in Programmzeilen bis zu 15 Bytes eingelesen werden können. Die Programme für 'SYNTEXTZEILEN' erschöpfen darüberhinaus jede Möglichkeit. Ein gewisser Vorteil bleibt nur für den Anwender, der keinen Drucker hat, um z.B. 7 ASCII-Zeichen Programmzeile aufzubauen, in denen Zeichen zwischen 128-255 vorkommen. Wir haben in Kapitel 2 gesehen, daß die Alphazeichen 0-255 replace oder append nicht als geschlossene Programmzeile, sondern nur einzeln einzugeben sind. Im Alpharegister können sie akkumuliert werden. Hier bietet der Q-Bote noch einen Vorteil. Soll z.B. eine 7 Bytes lange synthetische Textzeile, die Bytefolge 128 143 155 164 192 231 und 239 in ein Programm gebracht werden, so lesen Sie ins Alpharegister ein replace 239, append 231 192 164 155 143 128. Das Druckbild zeigt: ⊗⊗⊗⊗⊗⊗⊗. Nach 'GTO "Q"' und mit 'SST' ausgeführt 'RCL M' 'STO Q' - Umschalten in den PRGM-Modus, 'Q-Bote .'-BC einlesen, nochmals 'SST' sehen Sie eine Replace-Alpha Programmzeile mit 7 'Blackbox'. Bringen Sie diese Zeile mit nochmals 'SST' nach Alpha, so zeigt das Druckerbild bei 'SF 12' ⊗⊗⊗⊗⊗⊗⊗.

Lassen Sie uns nun auf die Bytes 29 = 'GTO''ALPHA'; 30 = 'XEQ''ALPHA'; 31 = 'SPARE' und 205 = 'LBL''ALPHA' zurückkommen. Dem Anwender ohne IL-Drucker erschließt sich bei entsprechender Vorbereitung des Register 'Q' die Möglichkeit, skurrile globale Marken mit dazugehörigen 'GTO''s oder 'XEQ''s aus nicht tastbaren Zeichen oder Kleinbuchstaben zu erzeugen. Mit Barcodes ist dies zwar viel bequemer, wie wir später sehen, doch soll dies hier trotzdem erwähnt werden.

Löschen Sie jetzt das 'END' des Programms "Q". Tasten Sie dann ins Alpharegister ein: ":",.,.,:" - Führen Sie sodann im RUN-Modus aus 'GTO "Q"', dann 3x 'SST', Umschalten in den PRGM-Modus; lesen Sie sodann diese Barcodes ein:

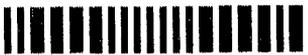
LBL " "  Dezimal: 160 205

Sie sehen: 'a\_'. Jetzt geben Sie zwei beliebige Ziffern ein, und anstelle dieser Zeile erscheint: 'LBL ":",.,.,:'. Es können so sämtliche Bytekombinationen einschließlich HEX 00-Bytes (natürlich nur mit Führungsbyte damit nichts verloren geht) als Marke zusammengebracht werden.

Mit den folgenden Barcodes entstehen 'GTO's und 'XEQ's für globale Abenteuer-Marken. Führen Sie erneut aus 'STO Q'. Gehen Sie auf die Zeile 'LBL ":",.,.,:'. Lesen Sie sodann diese Barcodes ein:

GTO " "  Dezimal: 224 29

Ohne Anforderung eines Arguments sehen Sie sofort 'GTO ":",.,.,:'. Führen Sie nun noch einmal 'STO Q' aus und gehen Sie dann zur Zeile 'GTO ":",.,.,:', wie vorstehend dargestellt. Lesen Sie alsdann die nachfolgenden Barcodes ein:

XEQ " "  Dezimal: 240 30

Auch hier wird sofort 'XEQ ":",.,.,:' in den PRGM-Speicher geschrieben. Sofern Sie alles richtig gemacht haben entsteht nach 'GTO..' mit 'PRP "Q"' der neben-  
01\*LBL "Q" stehende Programmausdruck. Fügen Sie zwischen Zeile 04 und 05  
02 RCL [ 'BEEP' ein und gehen dann mit 'SST' auf Zeile 06, schalten in den  
03 STO \_ RUN-Modus, drücken 'R/S' - so ertönt 'Dauer-BEEP'. Beweis: diese  
04\*LBL ":",.,.,:" skurrilen, globalen Marken arbeiten ganz normal. Sie können das  
05 GTO ":",.,.,:" Programm "Q" jetzt noch mit der Funktion 'eGØBEEP 77' = 'PRP'  
06 XEQ ":",.,.,:" ausdrucken. Bringen Sie ":",.,.,:" mit 'STO Q' erneut in das Regi-  
07 END ster 'Q' und lesen Sie dann im RUN-Modus diese Barcodes ein:

eGØBEEP 77  Dezimal: 32 167

Das Ergebnis ist: sofort wird das Programm, dessen Umkehrname im Register 'Q' steht, ausgedruckt. - Noch ein Letztes: Das 'SPARE'-Byte 31 bringt ebenfalls einen im Register 'Q' befindlichen Alphastring an sich. Als Programmzeile erscheint 'W "ALPHA"'. Was es mit diesem Byte genau auf sich hat, ist nebulös. Es scheint so, die Flags werden verändert. Jedenfalls ist beim Experimentieren Vorsicht geboten. Das Risiko geht bis zum GAU.

2 --  Dezimal: 16 31

Dies sind die Barcodes zu dem etwas nebulös erscheinenden 'SPARE'-Byte Nr. 31.

### 7. Byteschnapper 'Snap 2' bis 'Snap 9'

Die Bytes bis 239 geben nichts brauchbares her. 240-247 können als Bytehüpfen ursprünglicher 'Wickes-Synthetik' benutzt werden. Doch dann wird's 'bissig'!. Einige der Textbytes ab 248 können als wahre Byte-Schnapper bezeichnet werden, weil sie einfach die zunächst verfügbaren Bytes nachfolgender Zeilen richtig 'wegschnappen'!; und zwar Byte 248 'Snap 2', 249 'Snap 3', 252 'Snap 6', 254 'Snap 8' und 255 'Snap 9'. Die Ziffern bezeichnen jeweils die Anzahl Bytes welche weggesnapt werden. Für unseren Zweck genügt 'Snap 2'. Die Barcodes der anderen Snapper folgen anschließend. Möglicherweise werden damit bisher noch

unentdeckte Anwendungen gefunden. Ein Beispiel wird später schon einmal vorgeführt. Wir starten mit Barcodes in synthetische Gefilde: Geben Sie 01+LBL "T" das kleine Programm "T" ein; die 2x 'ENTER' sind nur Dummy zur Sicherheit. Das Byte 94 'ATAN' in Zeile 02 steht allein und stellt die Funktion 'ATAN' dar. Ginge ein Textbyte voraus - wir erinnern uns an Kapitel 1 - so wäre dies Byte das Zeichen "↑". Das Byte 50 ist allein die Funktion 'STO 02' und als Textzeichen die alphanumerische Ziffer "2". Führen Sie nun 'GTO. "T"' aus. In der Anzeige sehen Sie '01 LBL "T"'. Lesen Sie jetzt die folgenden Barcodes ein:

1CLP\_\_  
Snap 2  Dezimal: 128 248

Geben Sie dann über das Tastenfeld die Zahlen '8' und '3' ein und sofort passiert's. Das Textbyte 248 führt normalerweise 8 Bytes, die als Alphazeichen im Speicher erscheinen. Mit 'Snap 2' wird das Byte 248 allein in den Speicher eingegeben und erhält die Ziffernfolge 83 = "S" zugeteilt und zwar anstelle 2. Weil es jedoch 8 Alphazeichen binden muß, greift es sich nach insgesamt noch 4 vorhandenen 'Verfügungsbytes' (248+0,83,0,0,0,0) kurzerhand die 2 folgenden Einzelbytes 94 "↑" (ATAN) sowie 50 "2" und verleibt sie der Textzeile ein. Das

Programm sieht dann wie nebenstehend aus. Dieser Vorgang passiert an jeder beliebigen Stelle im PRGM-Speicher. Die anderen Schnapper binden dann jeweils 3, 6, 8 oder 9 im Speicher vorhandene Bytes. Große Vorsicht ist beim Experimentieren nahe am 'END' geboten, weil rücksichtslos auch das ständige .END. teils oder ganz weggesnapt wird! Dies beraubt den Prozessor dann jeder Orientierung und führt meist zum 'GAU'. Manchmal stellt jedoch ein 'GTO..' die Ordnung wieder her. Die weiteren Schnapper rufen folgende Display-Anzeigen auf:

2 \_\_  
Snap 3  Dezimal: 144 249

!Σ+ \_\_  
Snap 6  Dezimal: 192 252

> \_\_  
Snap 8  Dezimal: 224 254

2 \_\_  
Snap 9  Dezimal: 240 255

Und nun das angekündigte Beispiel: Wir verwenden 'Snap 9' um eine Replace-Textzeile entstehen zu lassen. Führen Sie aus: 'GTO "T"', 'PRGM', 'SST' und löschen Sie die Zeile 02 "\*S++++2" . Tasten Sie dann die nachfolgend im kleinen Kästchen stehenden Befehle ein, die als Programm natürlich sinnlos sind, weil sie ja in Text 'verwandelt' werden sollen. Die 'ENTER' ab Zeile 15 sind nur Dummy.

01+LBL "T"	05 OCT	09 FRC	13 OCT	17 ENTER†
02 ABS	06 X>0?	10 RND	14 R-D	18 ENTER†
03 PI	07 RCL 00	11 X>0?	15 ENTER†	19 ENTER†
04 X#0?	08 CLST	12 RCL 00	16 ENTER†	20 END

Führen Sie dann immer im PRGM-Modus aus 'GTO. "T"' - Sie sehen: 'LBL 01 "T"' 'PACK'en Sie jetzt. Tasten Sie sodann 4x 'ENTER' ein. Sie sehen 05 'ENTER'. Lesen Sie nun mit dem Lesestift 'Snap 9' ein, es erscheint 06 "2\_". Tasten Sie sodann die Ziffern 6 und 6 ein. Nach Entfernen der 'ENTER'-Dummy's sieht der Text so wie nebenstehend aus. Welcher genaue Editiervorgang hier zugrunde liegt, ist im 'Wickes' und im 'Jarett' über synthetische Programmierung beschrieben. Sehr viel ergiebiger ist allerdings der umgekehrte Vorgang, bei dem aus Textzeilen synthetische Befehle entwickelt werden.

e. Synthetische Befehle aus Textzeilen

Wir machen uns nun die Beißgier von 'SNAP 2' nutzbar. Löschen Sie zuerst die Zeile 02 "\*Barcode sind ok" aus dem Programm "T". Der Rechner steht auf 'LBL "T"'. Lesen Sie dann die folgende Barcode-Textzeile ein:



01+LBL "T" Diese Textzeile lautet dezimal:  
 02 "2+r" 244 = Das Textbyte für vier ASCII-Zeichen  
 03 ENTER↑ 50 = Die Zahl 2 als zweitem Dummy für 'SNAP 2'  
 04 ENTER↑ 208 = 'GTO' - in der Textzeile unsichtbar  
 05 END 0 = Das Byte für den Sprunglängeneintrag bei Dreibyte-GTO's  
 114 = r - später die Nachsilbe Y

Der Rechner steht jetzt noch auf der Textzeile, die Sie soeben eingelesen haben. Danach folgen aus der vorigen Version "T" noch einige 'Sicherheit-ENTER'. Kontrollieren Sie dies ruhig noch einmal. Das Programm sieht wie oben links abgedruckt aus. Der Rechner muß dann wieder auf der Textzeile stehen. Lesen Sie sodann die folgenden Barcodes nacheinander ein:



Sie sehen die Zeile 02 'GTO Y' - eine synthetisch erzeugte 'GTO'-Anweisung auf eine lokale Marke, die wie 'GTO A' arbeitet, jedoch nicht über das Tastenfeld 'beschafft' werden kann. Führen Sie 'BST' aus, Sie sehen wieder 'LBL "T"'. Lesen Sie dann die folgende BC-Reihe ein:



Tasten Sie mit dem Lesestift erneut die obigen Barcodes von 'BST' bis 'SST' ab. Im Programmspeicher erscheint: 02 'LBL Y'. Geben Sie nun über das Tastenfeld 'BEEP' ein, und führen Sie anschließend 'SST' aus; es erscheint 'GTO Y'. Nun lesen Sie Barcodes \$T+H IÄ... aus Abschnitt d.5. ein und fügen über das Tastenfeld die Ziffern 1,1,4 hinzu. Ergebnis: 05 'XEQ Y'. Schalten Sie jetzt 01+LBL "T" in den RUN-Modus zurück, 'PACK'en Sie und drücken dann 'R/S'. Erfolg: 02+LBL Y es 'BEEP't. Das Programm sieht jetzt wie nebenstehend aus. Mit 03 BEEP dieser Methode lassen sich aus Textzeilen sämtliche Zwei- oder Mehrbytefolgen, also auch alle synthetischen Befehle herstellen. Als Beispiel noch einmal 'RCL M'. Nachstehende Textbarcodes an beliebiger Programmstelle einlesen:

07 ENTER↑ Snap 2  
 08 END RCL M



Obige Barcodes 'BST' bis 'SST' abtasten und der Befehl 'RCL M' steht im Speicher. - Was ist hier nun geschehen? 'SNAP 2' hat sich als Kanibale erwiesen und einfach die beiden zunächst greifbaren Bytes, nämlich das Textbyte 243 und das Dummy-Byte 50 aus unserer Textzeile herausgesnapt. Nun mußten sich die beiden verwaist stehenden Bytes 144 und 117 von Alphazeichen in Befehle 'verwandeln'. Da Byte 144 Vorsilbenbyte ist, war es gezwungen, sich seinerseits mit dem nächsten Byte zu verbinden. Weil ein numerisches Register 117 direkt nicht existiert, ergriff 144 das normaler Programmierung verschlossene Statusregister M als Partner, und es entstand 'RCL M'. Genauso entstanden 'LBL Y' und 'GTO Y'.

1. Aufbau, Typindikator und 8-Bit Prüfsumme

Diese Textbarcodes entsprechen genau denen, welche in Kapitel 2 bei den Programmen für 'SYNTEXTZEILE' beschrieben sind. Hier wird allerdings nur :REPLACE benötigt.

2. Herstellen von BC für 'SNAP 2'- Programmierung auf dem Thermodrucker

Mit einem der Programme "SYNT.." -BC können solche Textzeilen für beliebige synthetische Bytekombinationen angefertigt und mit den Barcodes 'BST', 'SNAP 2', '8', '3', '<--' und 'SST' an beliebiger Programmstelle eingelesen werden. Trotzdem für die praktische Arbeit die meisten synthetischen Befehle am Ende dieses Abschnitts in alphabetischer Reihenfolge zusammengestellt sind, könnte gelegentlich Bedarf an weiteren Befehlen bestehen. Hierfür kann sehr bequem das nachfolgende Programm zur Herstellung auf dem Drucker benutzt werden.

2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "SNP2BC"

Programmbedienung:

-SIZE: es genügt 002

\*EQ "SNP2BC"



Zur Kennzeichnung welcher BC-Typ hergestellt wird erfolgt rechtsbündiger Ausdruck Snap 2.

- "SYNFUNKTION?": Geben Sie hier die Buchstaben, Zeichen oder Zahlen der synthetischen Funktion ein - 'R/S'
- "PRE↑POST↑": Tasten Sie jetzt den Vorsilbendecimalwert (Prefix) ein, und drücken Sie 'ENTER'. Geben Sie sodann den Bytedecimalwert der Nachsilbe (Postfix) ein und drücken 'R/S'. - Z.B. für 'RCL M' 144 'ENTER' 117 'R/S'. Falls Sie jeweils nichts eingeben, kehrt das Programm zur Frage zurück. Weitere Prüfungen sind im Interesse der PRGM-Kürze nicht vorgesehen. Es würden beispielsweise auch BC-SNAP 2-Textzeilen von sinnlosen Bytekombinationen erstellt. HEX 00-Bytes sind nicht zugelassen, seien Sie auch besonders vorsichtig mit allen Textbytes zwischen 240 und 255.

Die Flagsteuerzeile 03 im nachstehenden Listing führt die Bytes 4 41 128 129. Die Zeile 44 birgt das Byte 132, die Zeile 48 das Byte 115 und die Zeile 50 das Byte 50.

01*LBL "SNP2BC"	20 TONE 7	39 FS?C 00	58 CF 17	77 GTO 06
02*LBL 00	21 PROMPT	40 RTN	59 CLST	78 -
03 "a>*"	22 FC?C 22	41 SF 17	60 BEEP	79 1
04 RCL I	23 GTO 01	42 .	61 ADV	80 +
05 X<> d	24 ADV	43 ACCOL	62 ADV	81 .
06 "Snap 2"	25 STO 01	44 "	63 RTN	82 1
07 ACA	26 X<>Y	45 XEQ 04	64 GTO 00	83 BLDSPEC
08 ADV	27 STQ 00	46 RDN	65*LBL 03	84 X<>Y
09 "SYNFUNKTION?"	28 +	47 XEQ 03	66 XEQ 05	85 BLDSPEC
10 TONE 7	29 +	48 "s"	67 ARCL X	86 RTN
11 STOP	30 SF 00	49 XEQ 04	68 RDN	87*LBL 06
12 AOFF	31 XEQ 02	50 "2"	69*LBL 04	88 .
13 FC? 23	32*LBL 02	51 XEQ 04	70 OUTA	89 X<>Y
14 GTO 00	33 RCL X	52 RCL 00	71 CLA	90 BLDSPEC
15 ACA	34 256	53 XEQ 03	72 RTN	91 END
16 ADV	35 ST/ Z	54 RCL 01	73*LBL 05	104 Bytes
17 165	36 MOD	55 XEQ 03	74 127	
18*LBL 01	37 +	56 ADV	75 X<>Y	
19 "PRE↑POST↑"	38 INT	57 ADV	76 X<=Y?	

### 3. Herstellen der Tabellen für 'SNAP 2'- Programmierung auf dem Plotter

Wer einen Plotter hat, kann das nachfolgende Programm für eigene Tabellen im DIN A4-Format benutzen. Es könnten z.B. BC aller 128 'TONE' entstehen.

#### 3.1 Rechner HP 41 CV - Plotter, Plottermodul PRGM "S2Pbc"

##### Programmbedienung:

-SIZE: es muß mindestens 008 vorhanden sein

XEQ "S2Pbc"



Nach Abfragen "PEN 1", "PEN 2" und "PAPIER?" folgt die Frage "SYNFUNKTION?". Hier wird der Funktionsname, z.B. 'STO d' eingegeben - 'R/S'. Sodann steht "PRE↑POST↑": Hier wird der dezimale Vor-'ENTER'/Nachsilbenwert 'R/S' getastet.

01*LBL "S2Pbc"	44 "PAPIER?"	87 2	130 "BST"	172 RCL 05
02 AUTO10	45 TONE 7	88 PEN	131 LABEL	173 MOVE
03 "HP7470A"	46 PROMPT	89 RCL 07	132 XEQ 04	174 "<--"
04 FINDID	47 2	90 INT	133 1	175 LABEL
05 SELECT	48 PEN	91 RCL 05	134 PEN	176 XEQ 04
06 FIX 0	49 FRAME	92 MOVE	135 +	177 1
07 SF 28	50 .	93 "SYNFUNKTION?"	136 MOVE	178 PEN
08 CF 29	51 13	94 TONE 7	137 "	179 +
09 CLRG	52 MOVE	95 AON	↓"	180 MOVE
10 115	53 172	96 STOP	138 BC	181 " "
11 STO 01	54 13	97 ROFF	139 XEQ 03	182 BC
12 50	55 DRAW	98 LABEL	140 2	183 XEQ 03
13 STO 02	56 "CSSI.217,.35"	99 XEQ 04	141 PEN	184 2
14 4.11656	57 OUTA	100 1	142 41	185 PEN
15 STO 07	58 .	101 PEN	143 RCL 05	186 145
16 "PEN 1= .3 SWZ"	59 PEN	102 +	144 MOVE	187 RCL 05
17 TONE 7	60 "PEN 2= .5 SWZ"	103 MOVE	145 "SNAP 2"	188 MOVE
18 PROMPT	61 TONE 7	104 "PRE↑POST↑"	146 LABEL	189 "SST"
19 "PEN 2= .4 SWZ"	62 PROMPT	105 TONE 7	147 XEQ 04	190 LABEL
20 TONE 7	63 2	106 PROMPT	148 1	191 XEQ 04
21 PROMPT	64 PEN	107 STO 04	149 PEN	192 1
22 PINIT	65 4	108 RDW	150 +	193 PEN
23 11	66 9	109 STO 03	151 MOVE	194 +
24 263	67 MOVE	110 XEQ 02	152 ""	195 MOVE
25 12.5	68 SF 17	111 1.004	153 BC	196 "Δ"
26 184.5	69 "BARCODES FUWR S"	112 BCREGX	154 70	197 BC
27 LIMIT	70 "FYNTHETISC"	113 5	155 RCL 05	198 GTO 00
28 .	71 LABEL	114 BCKSM	156 1	199*LBL 02
29 252	72 "HES PROGRAMMIER"	115 BC	157 +	200 .
30 .	73 "PEN MIT SN"	116 ISG 07	158 MOVE	201 PDIR
31 PEN	74 LABEL	117 GTO 01	159 "δ"	202 LORG
32 172	75 "AP 2"	118 4.11656	160 BC	203 ACOS
33 SCALE	76 LABEL	119 STO 07	161 90	204 LDIR
34 18.03021	77 CF 17	120 13.5	162 RCL 05	205 RTN
35 12.331	78 .	121 ST+ 05	163 1	206*LBL 03
36 BCsize	79 PEN	122 ISG 06	164 +	207 "VS19"
37 XEQ 02	80 "PEN 2= .4 SWZ"	123 GTO 01	165 MOVE	208 OUTA
38*LBL 00	81 TONE 7	124 XEQ 03	166 ""	209 RTN
39 XEQ 03	82 PROMPT	125 2	167 BC	210*LBL 04
40 20	83 "CSSI.175,.275"	126 PEN	168 XEQ 03	211 "VS"
41 STO 05	84 OUTA	127 4	169 2	212 OUTA
42 .015	85*LBL 01	128 RCL 05	170 PEN	213 END
43 STO 06	86 XEQ 03	129 MOVE	171 117	598 Bytes

# BARCODES FÜR SYNTHETISCHES PROGRAMMIEREN MIT SNAP 2

ARCL a 155/123



ARCL d 155/126



ARCL N 155/118



ARCL P 155/120



ASTO c 154/125



ASTO M 154/117



ASTO O 154/119



DSE M 151/117



FC? IND M 173/245



FC? IND N 173/246



FC?C IND O 171/247



GTO IND M 174/117



ISG N 150/118



RCL b 144/124



RCL IND c 144/253



RCL e 144/127



ARCL c 155/125



ARCL M 155/117



ARCL O 155/119



ASTO a 154/123



ASTO d 154/126



ASTO N 154/118



ASTO P 154/120



DSE N 151/118



FC?C IND M 171/245



FC?C IND N 171/246



FIX IND M 156/245



ISG M 150/117



RCL a 144/123



RCL c 144/125



RCL d 144/126



RCL M 144/117



BST



SNAP 2



<--



SST



# BARCODES FÜR SYNTHETISCHES PROGRAMMIEREN MIT SNAP 2

RCL N 144/118



RCL P 144/120



RCL F 144/122



SCI IND N 157/246



SF IND N 168/246



ST- a 147/123



ST+ a 146/123



ST+ M 146/117



ST\* 0 148/119



ST/ N 149/118



ST0 a 145/123



ST0 c 145/125



ST0 e 145/127



ST0 M 145/117



ST0 0 145/119



ST0 Q 145/121



RCL 0 144/119



RCL Q 144/121



SCI IND d 157/245



SF IND M 168/245



SF IND 0 168/247



ST- 0 147/119



ST+ d 146/126



ST+ 0 146/119



ST/ M 149/117



ST/ 0 149/119



ST0 b 145/124



ST0 d 145/126



ST0 IND c 145/253



ST0 N 145/118



ST0 P 145/120



ST0 F 145/122



BST



SNAP 2



<--



SST



# BARCODES FÜR SYNTHETISCHES PROGRAMMIEREN MIT SNAP 2

STO IND a 145/251



TONE 4/14 159/14



TONE 6/26 159/26



TONE 7/47 159/47



TONE 4/64 159/64



TONE 0/70 159/70



TONE 5/75 159/75



TONE 3/83 159/83



TONE 0/100 159/100



X<>a 206/123



X<>c 206/125



X<>e 206/127



X<>N 206/118



X<>P 206/120



X<>T 206/122



X<> IND M 206/245



TONE 3/13 159/13



TONE 1/21 159/21



TONE 8/28 159/28



TONE 7/57 159/57



TONE 6/66 159/66



TONE 2/72 159/72



TONE 1/81 159/81



TONE 7/87 159/87



VIEW M 152/117



X<>b 206/124



X<>d 206/126



X<>M 206/117



X<>O 206/119



X<>Q 206/121



X<> IND c 206/253



X<> IND N 206/246



BST



SNAP 2



<--



SST

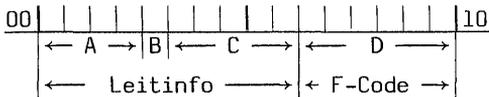


Das Programm für die Herstellung dieser Barcodes ist hinsichtlich der Aufteilung gegenüber dem Listing auf Seite 149 leicht abgewandelt. Das dort wiedergegebene Programm stellt 3 Befehle nebeneinander. Wem dies zu eng ist, und wer lieber die hier abgedruckte Ausführung erhalten möchte, muß die folgenden Zeilen abändern: Zeilen 14 und 118 in 21.09776; Zeile 127 in 10; Zeile 142 in 43; Zeile 154 in 72; Zeile 161 in 92; Zeile 171 in 116 und Zeile 186 in 141. Außerdem sollte der Plotter das einzige Gerät in der Schleife sein.

f. XROM-Funktionen

Die 'eXternen' ROM = XROM-Befehle betreffen alle Funktionen der Periferigeräte, bzw. der -Module. Es sind ausschließlich Zweibytefunktionen. Wenn ein Programm Befehle aus periferen Einheiten enthält, und die Einheit entfernt wird, erscheint im PRGM-Speicher 'XROM nn,nn'. Alle diese Funktionen sind in der Byte-tabelle zwischen dezimal 160 bis 167 enthalten und werden hieraus abgeleitet. Deswegen besteht eine Wechselbeziehung zwischen Zweibyte-XROM-Barcodes, z.B. 'PRX' und den Mehrbyte-Direktausführungs-BC 'XEQ' 'ALPHA' "PRX", welche im nächsten Abschnitt behandelt werden.

1. Aufbau



A = 4-Bit Prüfsumme

B = 1-Bit Indikator für XROM-Funktion

C = 5-Bit Kennzahl für die ROM-Identifikation

D = 6-Bit ROM-Funktionscode

1.1 Der Typindikator und die übrigen Kenndaten

Der Typindikator teilt dem Rechner mit: B = 1 - es handelt sich um einen Zweibyte XROM-Barcode; C = der vor dem Komma einer XROM-Zahl stehende Dezimalwert für die Periferie-Identifikation, Beispiel: 'PRX' = XROM 29,20 und D = der nach dem Komma stehende Funktionscode, in diesem Beispiel XROM 29,20.

1.2 Die XROM-Identifikationsnummer und der Funktionscode

Wie Sie wissen, können in einer HP-IL-Schleife maximal 30 Geräte adressiert werden. Eine gewisse Parallele ist in der XROM-Identifikationsnummer zu sehen; es können dies ja sogar 30 verschiedene Geräte sein. In den 5 Bit von C kann maximal die Zahl 31 dargestellt werden. Die niedrigstmögliche Geräteidentität wäre 1, und die höchstmögliche 31. In den 6 Bit des Feldes D ist die Funktionsnummer dargestellt. Mit 6 Bit kann höchstens die Zahl 63 erreicht werden. Dadurch ist die Zahl der möglichen Befehle pro Periferiegerät auf 0-63 begrenzt. Demnach ist die niedrigste XROM-# = 00,00 und die höchste = 31,63. Wegen der Schleifenbegrenzung auf 30 Geräte könnten theoretisch 30x64 = 1920 XROM-Befehle zusätzlich zu den Rechnerbefehlen verschlüsselt werden; eine Anzahl, die bei weitem nicht ausgeschöpft ist.

Nun mag Ihnen bereits aufgefallen sein, die Funktionsnummern beginnen stets mit 01 - z.B. die Druckerfunktionen aus dem IL-Modul mit XROM 29,01 oder Kassettenlaufwerk mit XROM 28,01. Was ist aber mit 29,00 oder 28,00? Diese Nummern werden nicht für Befehle, sondern meistens für die Gerätebenennung im CAT 2 benutzt. Barcodes dieser XROM-# bewirken im RUN-Modus nicht, schalten vom ALPHA in den RUN-Modus um und können wie ein normaler Befehl ohne Wirkung in eine Programmzeile eingelesen werden, wo sie im Bedarfsfall als Kennzeichnung die-

nen könnten. In einem laufenden PRGM werden sie überlesen. Das kleine Programm "PRX" demonstriert dies. Es druckt fortlaufend den Inhalt des X-Registers aus. Falls Sie bei der Eingabe des Befehls 'PRX' über das Tastenfeld XEQ "PRX" erhalten sollten, nehmen Sie doch einfach Barcodes.

XROM 29,00		<pre> 01*LBL "PRX" 02 -PRINTER 2E 03 PRX 04 GTO "PRX" 05 END         </pre>
------------	-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------

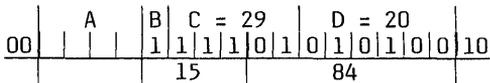
Nicht belegte XROM-# wie z.B. IL-Modul/Drucker 29,26 können zwar mit BC in den Programmspeicher eingelesen werden, führen jedoch logischerweise bei Programm-  
lauf zu 'NONEXISTENT'. Die Möglichkeit, XROM-Befehle bei nicht vorhandenem Per-  
iferiergerät einzulesen, wird im Abschnitt Arbeiten mit Zweibytebefehlen be-  
schrieben. Manchmal 'fehlen' zwischendurch XROM-# wie bei Auflistung zum IL-Mod-  
dul XROM 28,25 und 28,26. Tatsächlich sind diese nur anders belegt. XROM 28,25

<pre> 01*LBL "--CTL" 02 -- 03 -CTL FNS 04 END         </pre>	<pre> 01*LBL "--CTL" 02 XROM 28,25 03 XROM 28,26 04 END         </pre>
--------------------------------------------------------------	------------------------------------------------------------------------

sind Trennstriche und 28,26 birgt den Namen der fol-  
genden Befehlesgruppe. Sie können sich mit einem der  
XROM-PRGM die BC anfertigen und einlesen oder den BC-  
Befehl 'eGØBEEP 25' bzw. '...26' ausführen. - Hier  
noch ein kleines Autorengeheimnis: wie konnten die  
XROM-Zeilen im PRGM-Listing rechts ohne vorhandenes IL-Modul, denn nur dann er-  
scheinen ja die XROM-#, gedruckt werden? Links steht das Listing mit IL-Modul.

### 1.3 Die 4-Bit Prüfsumme

Die andersartige Aufteilung bei C und D darf uns nicht verwirren. Tatsächlich  
bestehen ja B/C und D, wie bei allen anderen Zweibytebefehlen auch, aus 12 Bit,  
3 Nybble oder 1 1/2 Bytes. Es müssen nun lediglich die Bits der XROM-Zahlen so  
abgeteilt werden, daß sie die Dezimalwerte ihrer Nybble ergeben. Im Beispiel  
'PRX' 29,20 sähe der Aufbau binär so aus:



Nybble 2 von Byte eins = 15 + Byte zwei = 84 ergibt 99 : 16 = 6 Rest 3 - Prüf-  
summe ist 9; 16x = 144 + 15 = Wert von Byte eins: 159. Dezimalfolge der Bar-  
codereihe lautet 159 84. Um zu einer programmierbaren Lösung zu kommen muß die  
Nybble-Abteilung berechenbar sein. Was muß passieren? Das Bit B mit dem Stel-  
lenwert 2<sup>5</sup> = 32 innerlab der Bitfolge B und C muß zunächst zu C hinzugerechnet  
werden, was 61 ergibt. Sodann müssen die Bits ab Stelle 2<sup>2</sup> abgeteilt werden =  
61 : 4 = 15,25. Damit ist das Nybble zwei von Byte eins = 1111 - 15 isoliert.  
Der Rest 0,25 stellt die zu Byte zwei Nybble eins hinzuzurechnenden Stellen  
dar. Es sind die 0,25 x 4 = 1. Da dies die 2 hoch 6. Stelle = 64 ist, muß der  
Restwert, hier 1, mit 64 multipliziert und dann dem Wert von D, hier 20, hinzu-  
gerechnet werden, was 84 ergibt. Die erhaltenen Dezimalwerte 15 + 84 ergeben  
die Ausgangszahl 99 für die Prüfsummenberechnung.

## 2. Herstellen von Barcodes für XROM-Funktionen auf dem Thermodrucker

Für die Barcode-Herstellung gibt es vier Programme; drei wie bekannt für den  
Thermodrucker und ein Programm für den Plotter 7470A. Der oben beschriebene  
Rechengang kann in allen Programmen mit 'SST' nachvollzogen werden.

### 2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "XROMBC"

#### Programmbedienung:

- SIZE: es muß mindestens 'SIZE 003' eingestellt sein.

Der Ausdruck erfolgt rechtebündig, so kann das Druckpapier zweimal durchlaufen.

XEQ "XROMBC"



- "XR-FUNKTION?": Geben Sie hier die Buchstabenfolge der gewünschten Funktion ein - z.B. "PRX" und drücken Sie 'R/S'.
  - "XROM-NR? N,NN?": Geben Sie nun die zugehörige XROM-Nr. ein, in der Form nn,nn - für 'PRX' also 29,20 - 'R/S'.
- Keines der Programme prüft, ob eine XROM-Nr. einer bestimmten Funktion entspricht, bzw. ob diese Nummer mit einer Funktion belegt ist. Für jede Eingabe zwischen 00,00 bis 31,63 wird der demgemäße Barcode ausgedruckt. Geben Sie Werte >31 oder >63 ein erfolgt Fehlermeldung: "XROM >31,63".

01*LBL "XROMBC"	28 " "	55 X<=Y?	82 RCL 00	109*LBL 06
02*LBL 00	29 10	56 GTO 03	83 XEQ 05	110 OUTA
03 "a1x"	30 XYY?	57 MOD	84 RCL 01	111 CLA
04 RCL [	31 "f0"	58 Rf	85 XEQ 04	112 RDN
05 X<> d	32 ARCL 00	59 +	86 ADV	113 RTN
06 CLST	33 ACA	60 STO 01	87 ADV	114*LBL 07
07 ADV	34 ADV	61 ADV	88 ADV	115 127
08 "XR-FUNKTION?"	35 RDN	62 +	89 CLST	116 X<>Y
09 TONE 7	36 32	63 SF 00	90 CLA	117 X<=Y?
10 STOP	37 X<=Y?	64 XEQ 02	91 RTN	118 GTO 08
11 AOFF	38 GTO 03	65*LBL 02	92 GTO 00	119 -
12 FC? 23	39 +	66 RCL X	93*LBL 03	120 1
13 GTO 00	40 RCL X	67 16	94 "XROM > 31,63"	121 +
14 ACA	41 4	68 ST/ Z	95 ACA	122 ,
15 ADV	42 ST/ Z	69 MOD	96 ADV	123 1
16 "XROM"	43 MOD	70 +	97 TONE 0	124 BLDSPEC
17 ACA	44 64	71 INT	98 GTO 00	125 X<>Y
18 "f-NR? N,NN"	45 *	72 FS?C 00	99*LBL 04	126 BLDSPEC
19*LBL 01	46 X<>Y	73 RTN	100 SF 25	127 RTN
20 TONE 7	47 INT	74 16	101 X=0?	128*LBL 08
21 PROMPT	48 STO 00	75 *	102 ACCOL	129 ,
22 FC? 22	49 Rf	76 ST+ 00	103 X=0?	130 X<>Y
23 GTO 01	50 FRC	77 SF 17	104 GTO 06	131 BLDSPEC
24 ABS	51 E2	78 ,	105*LBL 05	132 END
25 STO 00	52 *	79 ACCOL	106 XEQ 07	247 Bytes
26 ENTERf	53 INT	80 129	107 ARCL X	
27 INT	54 64	81 XEQ 05	108 RDN	

## 2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "XROMBc"

### Programmbedienung:

- SIZE und Bedienung wie "XROMBC"

XEQ "XROMBc"



Das Programmlisting finden Sie auf Seite 156.

## 2.3 Rechner HP 41 C - Drucker, Plottermodul PRGM "XROMbc"

Programmbedienung und SIZE wie vorstehend

XEQ "XROMbc"



01*LBL "XROMbc"	24 ABS	47 INT	70 +	93*LBL 03
02*LBL 00	25 STO 00	48 STO 00	71 INT	94 *XROM > 31.63*
03 "α1X"	26 ENTER↑	49 R↑	72 FS?C 00	95 ACA
04 RCL [	27 INT	50 FRC	73 RTN	96 ADV
05 X(>) d	28 " "	51 E2	74 16	97 TONE 0
06 CLST	29 10	52 *	75 *	98 GTO 00
07 ADV	30 X>Y?	53 INT	76 ST+ 00	99*LBL 04
08 *XR-FUNKTION?"	31 "I-0"	54 64	77 SF 17	100 SF 25
09 TONE 7	32 ARCL 00	55 X(<=Y?	78 ,	101 X=0?
10 STOP	33 ACA	56 GTO 03	79 ACCOL	102 ACCOL
11 AOFF	34 ADV	57 MOD	80 129	103 X=0?
12 FC? 23	35 RDN	58 R↑	81 XEQ 05	104 GTO 06
13 GTO 00	36 32	59 +	82 RCL 00	105*LBL 05
14 ACA	37 X<=Y?	60 STO 01	83 XEQ 05	106 XTOA
15 ADV	38 GTO 03	61 ADV	84 RCL 01	107*LBL 06
16 *XROM"	39 +	62 +	85 XEQ 04	108 OUTA
17 ACA	40 RCL X	63 SF 00	86 ADV	109 CLA
18 "I-NR? N,NN"	41 4	64 XEQ 02	87 ADV	110 RDN
19*LBL 01	42 ST/ Z	65*LBL 02	88 ADV	111 END
20 TONE 7	43 MOD	66 RCL X	89 CLST	217 Bytes
21 PROMPT	44 64	67 16	90 CLA	
22 FC? 22	45 *	68 ST/ Z	91 RTN	
23 GTO 01	46 X(>Y	69 MOD	92 GTO 00	

01*LBL "XROMbc"	17 ACA	33 ACA	49 R↑	65 BCO
02*LBL 00	18 "I-NR? N,NN"	34 ADV	50 FRC	66 ADV
03 "α1X"	19*LBL 01	35 RDN	51 E2	67 ADV
04 RCL [	20 TONE 7	36 32	52 *	68 BEEP
05 X(>) d	21 PROMPT	37 X<=Y?	53 INT	69 CLST
06 CLST	22 FC? 22	38 GTO 02	54 64	70 CLA
07 ADV	23 GTO 01	39 +	55 X(<=Y?	71 RTN
08 *XR-FUNKTION?"	24 ABS	40 RCL X	56 GTO 02	72 GTO 00
09 TONE 7	25 STO 00	41 4	57 MOD	73*LBL 02
10 STOP	26 ENTER↑	42 ST/ Z	58 R↑	74 *XROM > 31.63*
11 AOFF	27 INT	43 MOD	59 +	75 ACA
12 FC? 23	28 " "	44 64	60 STO 01	76 ADV
13 GTO 00	29 10	45 *	61 E-3	77 TONE 0
14 ACA	30 X>Y?	46 X(>Y	62 BCREGX	78 GTO 00
15 ADV	31 "I-0"	47 INT	63 -2	79 END
16 *XROM"	32 ARCL 00	48 STO 00	64 BCKSM	167 Bytes

### 3. Herstellen von XROM-Tabellen mit dem Plotter

Die am Schluß dieses Abschnitts stehenden Tabellen mit den XROM-Funktionen und zugehörigen Nummern sind alle mit dem nachfolgenden Programm hergestellt. Wer mit seinem Plotter solche Tabellen ganz oder teilweise oder für andere Peripheriegeräte bzw. Module im DIN A 4 - Format erstellen möchte, kann dafür dies Programm einsetzen.

#### 3.1 Rechner HP 41 CV - Plotter, Plottermodul PRGM "XRMPbc"

##### Programmbedienung.

- SIZE: es wird mindestens 'SIZE 008' erwartet.

Der Plotter sollte möglichst das einzige Schleifengerät sein. In diesem Fall ist es zweckmäßig Zeile 02 'AUTOIO' gegen den Befehl 'MANIO' zu ersetzen.

XEQ "XRMPbc"



- "PEN 1= .3 SWZ": fordert Sie auf diesen Stift einzusetzen. - 'R/S'
- "PEN 2= .4 SWZ": ebenso
- "PAPIER?": Nun müssen Sie Plotterpapier einlegen.
- "PEN 2= .5 SWZ": Stiftwechsel für andere Beschriftung.  
Nach der Umrandung wird im Kopffeld geschrieben: FUNKTIONEN MIT XROM-NR: DES  
dann folgt die Eingabeaufforderung:
- "M-NAME 11 BS": geben Sie den Namen des Moduls mit max. 11 Buchstaben ein.
- "FUNKTION?": es folgen die Buchstaben des Funktionsnamens.
- "XROM-NR.? N,NN": fordert die zugehörige XROM-Nr. an - z.B. 29,20.

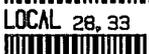
Sind alle Funktionen geplottet, so geben Sie für 'PLOT ENDE' bei der Frage nach der Funktion nichts ein, sondern drücken gleich 'R/S'. Im Rechnerdisplay steht "PLOT ENDE", Sie hören 'BEEP' und das Papier wird 'ausgeworfen'. Bei diesem Programm, mit dem konzentriertes Arbeiten erlaubt ist, wurde deswegen auf weitergehende Fehlermeldungen verzichtet.

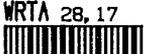
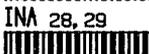
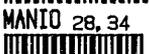
Die Zeilen 32, 52, 82, 90, 108 und 114 sind HP-GL-Befehlsfolgen, welche mit 'OUTA' aus dem Alpharegister gesendet werden. Sie steuern die Zeichengeschwindigkeit bzw. die Ausführung der Schrifthöhe oder -breite.

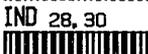
01*LBL "XRMPbc"	36 20	71 AON	106 CF 28	141 +
02 AUTOIO	37 STO 05	72 STOP	107 FIX 2	142 STO 01
03 "HP7470A"	38 .015	73 ROFF	108 "SI.125,.206"	143 E-3
04 FINDID	39 STO 02	74 "I-MODULS"	109 OUTA	144 BCREGX
05 SELECT	40 "PAPIER?"	75 LABEL	110 " "	145 2
06 FIX 0	41 TONE 7	76 .	111 ARCL X	146 BCKSM
07 SF 20	42 PROMPT	77 PEN	112 LABEL	147 SF 25
08 CF 29	43 2	78 "PEN 2= .4 SWZ"	113 SF 28	148 BC
09 7.13532	44 PEN	79 TONE 7	114 "VS"	149 CF 25
10 STO 07	45 FRAME	80 PROMPT	115 OUTA	150 ISG 07
11 "PEN 1= .3 SWZ"	46 .	81*LBL 01	116 RCL 07	151 GTO 01
12 TONE 7	47 13	82 "VS19"	117 INT	152 7.13532
13 PROMPT	48 MOVE	83 OUTA	118 RCL 05	153 STO 07
14 "PEN 2= .4 SWZ"	49 167	84 2	119 1	154 13.5
15 TONE 7	50 13	85 PEN	120 +	155 ST+ 05
16 PROMPT	51 DRAW	86 RCL 07	121 MOVE	156 XEQ 04
17 PINIT	52 "SI.225,.35"	87 INT	122 RCL Z	157 ISG 02
18 11	53 OUTA	88 RCL 05	123 ENTER↑	158 GTO 01
19 263	54 .	89 MOVE	124 INT	159 GTO 00
20 15	55 PEN	90 "SI.125,.275"	125 32	160*LBL 04
21 182	56 "PEN 2= .5 SWZ"	91 OUTA	126 +	161 .
22 LIMIT	57 TONE 7	92 CF 23	127 RCL X	162 PDIR
23 .	58 PROMPT	93 "FUNKTION?"	128 4	163 LORG
24 252	59 2	94 TONE 7	129 ST/ Z	164 ACOS
25 .	60 PEN	95 AON	130 MOD	165 LDIR
26 PEN	61 7	96 STOP	131 64	166 RTH
27 167	62 9	97 ROFF	132 *	167*LBL 05
28 SCALE	63 MOVE	98 FC?C 23	133 X<Y	168 "PTOT ENDE"
29 18.03021	64 SF 17	99 GTO 05	134 INT	169 BEEP
30 12.331	65 "FUNKTIONEN MIT "	100 LABEL	135 STO 00	170 AVIEW
31 BCSIZE	66 "XROM-NR. "	101 1	136 R↑	171 .
32 "VS19"	67 LABEL	102 PEN	137 FRC	172 .
33 OUTA	68 "DES "	103 "XROM-NR.? N,NN"	138 E2	173 PEN
34 XEQ 04	69 LABEL	104 TONE 7	139 *	174 MOVE
35*LBL 00	70 "M-NAME 11 BS"	105 PROMPT	140 R↑	175 END 500 B

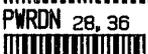
FUNKTIONEN MIT XROM-NR. DES

HP-IL-MODULS

ACA 29, 01  
  
BLDSPEC 29, 06  
  
PRFLAGS 29, 11  
  
PRREG 29, 16  
  
REGPLOT 29, 21  
  
CREATE 28, 01  
  
READK 28, 06  
  
READSUB 28, 11  
  
VERIFY 28, 16  
  
WRTR 28, 21  
  
FINDID 28, 28  
  
LOCAL 28, 33  
  
REMOTE 28, 38  

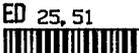
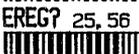
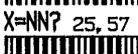

ACCHR 29, 02  
  
LIST 29, 07  
  
PRKEYS 29, 12  
  
PRREGX 29, 17  
  
SKPCHR 29, 22  
  
DIR 28, 02  
  
READP 28, 07  
  
RENAME 28, 12  
  
WRTA 28, 17  
  
WRTRX 28, 22  
  
INA 28, 29  
  
MANIO 28, 34  
  
SELECT 28, 39  


ACCOL 29, 03  
  
PRA 29, 08  
  
PRP 29, 13  
  
PRE 29, 18  
  
SKPCOL 29, 23  
  
NEWM 28, 03  
  
READR 28, 08  
  
SEC 28, 13  
  
WRTK 28, 18  
  
WRTS 28, 23  
  
IND 28, 30  
  
OUTA 28, 35  
  
STOPIO 28, 40  


ACSPEC 29, 04  
  
PRAXIS 29, 09  
  
PRPLOT 29, 14  
  
PRSTK 29, 19  
  
STKPLOT 29, 24  
  
PURGE 28, 04  
  
READRX 28, 09  
  
SEEKR 28, 14  
  
WRTP 28, 19  
  
ZERO 28, 24  
  
INSTAT 28, 31  
  
PWRDN 28, 36  
  
TRIGGER 28, 41  


ACX 29, 05  
  
PRBUF 29, 10  
  
PRPLOT 29, 15  
  
PRX 29, 20  
  
FMT 29, 25  
  
READA 28, 05  
  
READS 28, 10  
  
UNSEC 28, 15  
  
WRTPV 28, 20  
  
AUTOIO 28, 27  
  
LISTEN 28, 32  
  
PWRUP 28, 37  


## FUNKTIONEN MIT XROM-NR. DES X/FUNKTIONS-MODULS

ALENG 25, 01 	ANUM 25, 02 	APPCHR 25, 03 	APPREC 25, 04 	ARCLREC 25, 05 
AROT 25, 06 	ATOX 25, 07 	CLFL 25, 08 	CLKEYS 25, 09 	CRFLAS 25, 10 
CRFLD 25, 11 	DELCHR 25, 12 	DELREC 25, 13 	EMDIR 25, 14 	FLSIZE 25, 15 
GETAS 25, 16 	GETKEY 25, 17 	GETP 25, 18 	GETR 25, 19 	GETREC 25, 20 
GETRX 25, 21 	GETSUB 25, 22 	GETX 25, 23 	INSCR 25, 24 	INSREC 25, 25 
PASN 25, 26 	PCLPS 25, 27 	POSA 25, 28 	POSFL 25, 29 	PSIZE 25, 30 
PURFL 25, 31 	RCLFLAG 25, 32 	RCLPT 25, 33 	RCLPTA 25, 34 	REGMOVE 25, 35 
REGSWAP 25, 36 	SAVEAS 25, 37 	SAVEP 25, 38 	SAVER 25, 39 	SAVERX 25, 40 
SAVEX 25, 41 	SEEKPT 25, 42 	SEEKPTA 25, 43 	SIZE? 25, 44 	STOFLAG 25, 45 
X<F 25, 46 	XTOA 25, 47 	ASROOM 25, 49 	CLRGX 25, 50 	ED 25, 51 
EMDIRX 25, 52 	EMROOM 25, 53 	GETKEYX 25, 54 	RESZFL 25, 55 	EREG? 25, 56 
X=NN? 25, 57 	XNN? 25, 58 	X<NN? 25, 59 	X<=NN? 25, 60 	X>NN? 25, 61 
X>=NN? 25, 62 				

Die Funktionen ab Befehl 'ASROOM' sind nicht im X/F-Modul sondern im HP 41 CX enthalten, der ja auch alle Funktionen des X/F-Moduls beinhaltet. - Der Plotter stellt verschiedene Zeichen nicht oder mit anderen Symbolen dar. Der Befehl 'EREG?' lautet korrekt: '⌋REG?', der Befehl 'XNN?' lautet: 'X≠NN?'.

FUNKTIONEN MIT XROM-NR. DES

TIME-MODULS

ADATE 26, 01



ALMCAT 26, 02



ALMNOW 26, 03



ATIME 26, 04



ATIME24 26, 05



CLK12 26, 06



CLK24 26, 07



CLKT 26, 08



CLKTD 26, 09



CLOCK 26, 10



CORRECT 26, 11



DATE 26, 12



DATE+ 26, 13



DDAYS 26, 14



DMY 26, 15



DOW 26, 16



MDY 26, 17



RCLAF 26, 18



RCLSW 26, 19



RUNSW 26, 20



SETAF 26, 21



SETDATE 26, 22



SETIME 26, 23



SETSW 26, 24



STOPSW 26, 25



SW 26, 26



T+X 26, 27



TIME 26, 28



XYZALM 26, 29



ON 26, 30



CLALMA 26, 31



CLALMX 26, 32



CLRALMS 26, 33



RCLALM 26, 34



SWPT 26, 35



Die Funktionen ab Befehl 'CLALMA' sind nicht im TIME-Modul sondern im HP 41 CX enthalten, der im übrigen auch alle Funktionen des TIME-Moduls beinhaltet.

## FUNKTIONEN MIT XROM-NR. DES MAGNETKARTENLESERS

MRG 30, 01



RDTA 30, 02



ROTX 30, 03



RSUB 30, 04



VER 30, 05



WALL 30, 06



WDTA 30, 07



WDTAX 30, 08



WPRV 30, 09



WSTS 30, 10



7CLREG 30, 11



7DSP0 30, 12



7DSP1 30, 13



7DSP2 30, 14



7DSP3 30, 15



7DSP4 30, 16



7DSP5 30, 17



7DSP6 30, 18



7DSP7 30, 19



7DSP8 30, 20



DSP9 30, 21



7DSP1 30, 22



7DSZ 30, 23



7DSZI 30, 24



7ENG 30, 25



7FIX 30, 26



7GSBI 30, 27



7GTOI 30, 28



7ISZ 30, 29



7ISZI 30, 30



7P↔S 30, 31



7PRREG 30, 32



7PRSTK 30, 33



7PRTX 30, 34



7RCLE 30, 35

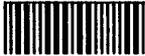


7SCI 30, 36



## FUNKTIONEN MIT XROM-NR. DES BARCODE-LESESTIFTS

WDDTA 27, 01



WDDTX 27, 02



WDLNK 27, 03



WNSUB 27, 04



WNDCN 27, 05



WNTST 27, 06



FUNKTIONEN MIT XROM-NR. DES

PLOTTER-MODULS

CLIPUU 17,01



FRAME 17,06



LABEL 17,11



LORG 17,16



MOVE 17,21



PLOT 17,26



SETGU 17,31



XAXIS 17,36



PDIR 18,02



PLTUXY 18,07



BCA 18,12



BCREGX 18,17



CXSIZE 17,02



GCLEAR 17,07



LDIR 17,12



LTYPE 17,17



PEN 17,22



PLREGX 17,27



SETUU 17,32



XAXISO 17,37



PRCL 18,03



PLANOT 18,08



BCAA 18,13



BCSIZE 18,18



CSIZED 17,03



IDRAW 17,08



LIMIT 17,13



LTYPED 17,18



PENDN 17,23



RATIO 17,28



TICLEN 17,33



YAXIS 17,38



NEWPLOT 18,04



Y? 18,09



BCCKSM 18,14



BCX 18,19



DGTIZE 17,04



IMOVE 17,09



LOCATD 17,14



LXAXIS 17,19



PENUP 17,24



RPLOT 17,29



UNCLIP 17,34



YAXISO 17,39



REPLOT 18,05



X? 18,10



BCD 18,15



BCXS 18,20



DRAW 17,05



IPLLOT 17,10



LOCATE 17,15



LYAXIS 17,20



PINIT 17,25



SCALE 17,30



WHERE 17,35



PCLBUF 18,01



PLINIT 18,06



BC 18,11



BCP 18,16



# FUNKTIONEN MIT XROM-NR. DES EXTEND. I/O-MODULS

COPYFL 23, 01



MCOPYPV 23, 06



ATOXR 23, 12



YTOAX 23, 17



DEVT 23, 23



INAE 23, 28



NLOOP 23, 33



OUTAN 23, 38



POLLE 23, 43



XFER 23, 48



ADROFF 23, 54



SEND 23, 59



DIRX 23, 02



MVERIFY 23, 07



ATOXX 23, 13



AID 23, 19



FINDAID 23, 24



INAN 23, 29



NOTREM 23, 34



OUTXB 23, 39



POLLUNC 23, 44



XFERC 23, 49



ADRON 23, 55



TAD 23, 60



FLLENG 23, 03



ALENGIO 23, 09



XTOAL 23, 14



CLRDEV 23, 20



ID 23, 25



INXB 23, 30



OUTAC 23, 35



OUTP 23, 40



RCLSEL 23, 45



XFERCL 23, 50



DOL 23, 56



UNL 23, 61



FLTYPE 23, 04



ANUMDEL 23, 10



XTOAR 23, 15



CLRLOOP 23, 21



INAC 23, 26



INP 23, 31



OUTACL 23, 36



POLL 23, 41



SRQ? 23, 46



XFERE 23, 51



DDT 23, 57



UNT 23, 62



MCOPY 23, 05



ATOXL 23, 11



X<>FIO 23, 16



DEVL 23, 22



INACL 23, 27



LOCK 23, 32



OUTAE 23, 37



POLLD 23, 42



STAT 23, 47



XFERN 23, 52



LAD 23, 58



# FUNKTIONEN MIT XROM-NR. DES DEVELOPMENT-MODULS

A-BUF 22, 01



A=BUF? 22, 03



ASIZE? 24, 02



BSIZE? 22, 08



CF33 22, 13



FRNS? 22, 18



IDY 22, 21



LAD 22, 26



NRD 22, 30



OR 24, 12



PRFRMS 22, 39



RG-BUFX 22, 42



SAI 22, 45



SF33 22, 50



UNL 22, 55



X-AR 24, 19



Y-AX 24, 20



A-XL 24, 03



AAD 22, 04



BININ 24, 06



BUF-AX 22, 09



CMD 22, 14



GET 22, 19



IFCR? 22, 22



LPD 22, 27



NRE 22, 31



OUTBIN 22, 33



PT= 22, 36



RG-BUF? 22, 43



SCOPE 22, 46



SRQR? 22, 51



UNT 22, 56



X-BUF 22, 59



A-XR 24, 04



AAU 22, 05



BINVIEW 24, 07



BUF-RGX 22, 10



DDL 22, 15



GTL 22, 20



IFC 22, 23



NIPT 22, 28



OCTIN 24, 13



OUTBINY 22, 34



PT? 22, 37



ROMCHKX 24, 15



SDA 22, 47



SST 22, 52



WFRM 22, 57



X<>FLAG 22, 61



A-XX 24, 05



AIPT 22, 06



BIT? 24, 08



BUF-XA 22, 11



DOT 22, 16



HEXIN 24, 09



INBIN 22, 24



MONITOR 22, 29



OCTVIEW 24, 14



OUTBUFX 22, 35



REN 22, 40



ROTXY 24, 18



SDC 22, 48



TAD 22, 53



WREG 22, 58



X-BUF? 22, 60



A=BUF? 22, 02



AND 24, 01



BSIZE? 22, 07



BUF-XB 22, 12



FRAY? 22, 17



HEXVIEW 24, 10



INBUFX 22, 25



NOT 24, 11



ORAV? 22, 32



PRBYTES 22, 38



RFRM 22, 41



RREG 22, 44



SDI 22, 49



TCT 22, 54



X-AL 24, 18



XOR 24, 17



# FUNKTIONEN MIT XROM-NR. DES -W&W CCD A-MODULS

B? 9, 01



SEED 9, 06



?IJA 9, 12



CNRM 9, 17



IJ=A 9, 22



M/ 9, 27



MOVE 9, 32



R>+ 9, 37



RSUM 9, 42



1CMP 9, 48



Cb 9, 53



S< 9, 58



XOR 9, 63



CAS 9, 02



SORT 9, 07



C<>C 9, 13



CSUM 9, 18



M+ 9, 23



MAX 9, 28



PIV 9, 33



R>- 9, 38



SUM 9, 43



2CMP 9, 49



NOT 9, 54



S> 9, 59



CLB 9, 03



>C+ 9, 09



C+ 9, 14



DIM 9, 19



M- 9, 24



MAXAB 9, 29



R-PR 9, 34



R>R? 9, 39



SUMAB 9, 44



AND 9, 50



OR 9, 55



Sb 9, 60



RNDM 9, 04



>R+ 9, 10



C- 9, 15



FNRM 9, 20



M\* 9, 25



MDTM 9, 30



R-QR 9, 35



RMAXAB 9, 40



SWAP 9, 45



bC? 9, 51



R< 9, 56



UNS 9, 61



SAS 9, 05



?IJ 9, 11



CMAXAB 9, 16



IJ= 9, 21



M\*M 9, 26



MTN 9, 31



R<R 9, 36



RNRM 9, 41



YC+C 9, 46



bS? 9, 52



R> 9, 57



WSIZE 9, 62



# FUNKTIONEN MIT XROM-NR. DES CCD-I/O FNS-MODULS

ABSP 11, 01



ACAXY 11, 02



ACLX 11, 03



ARCLE 11, 04



ARCLH 11, 05



ARCLI 11, 06



CLA- 11, 07



F/E 11, 08



INPT 11, 09



PMTA 11, 10



PMTH 11, 11



PMTK 11, 12



PRAXY 11, 13



PRL 11, 14



VIEWH 11, 15



XTOAH 11, 16



A+ 11, 18



A+B 11, 19



A- 11, 20



A-A 11, 21



DCD 11, 22



PC->RTN 11, 23



PC>X 11, 24



PEEKB 11, 25



PEEKR 11, 26



PHD 11, 27



POKEB 11, 29



POKER 11, 30



PPLNG 11, 31



X>PC 11, 32



X>RTN 11, 33



XR>RTN 11, 34



GETB 11, 36



GETK 11, 37



MRGK 11, 38



SAVEB 11, 39



SAVEK 11, 40



SORTFL 11, 41



g. XROM-Nummern von synthetischen Befehlen gibt es nicht

In seinem grundlegenden Buch "Synthetische Programmierung auf dem HP-41C/CV", deutsche Ausgabe, 2. erweiterte Auflage, Seiten 68, 69 beschreibt W.C.Wickes: "Synthetische XROM-Codes können in derselben Weise entschlüsselt werden, wie gewöhnliche XROM's, ...um beispielsweise den zu 'RCL M' gehörigen XROM-Code zu bestimmen, schreiben wir den Hexadezimal-Code '90 75' für 'RCL M' binär auf, gruppieren die letzten 12 Bits zu zwei Zahlen je 6 Bits und wandeln diese in Dezimalzahlen um:

hexadezimal:	9	0	7	5
binär:	1001	0000	0111	0101
dezimal:		01		53

Deshalb gilt 'RCL M' = 'XROM 01,53' und ähnlich 'STO M' = 'XROM 05,53'.

Solch Irrtum mag dadurch zustande gekommen sein, daß dieser mit einem Tastenzuweisungsprogramm 'eingeschmuggelte' Zweibytebefehl auf entsprechenden Tastendruck auch tatsächlich XROM 01,53 anzeigt und anschließend 'RCL M' ausgeführt oder programmiert wird. Tatsächlich ist für die Anzeige XROM 01,53 jedoch in erster Linie Byte 2, hier 117, zuständig und sofern Nybble 2 von Byte 1 einen Wert größer oder gleich 1 hat, wird dieser nach den beschriebenen Regeln hinzugerechnet. So entsteht 'STO M' = XROM 05,53. Ist Nybble 2 jedoch 0000, wird jede Bytekombination aus der Spalte 0 der Bytetabelle, Bytes 0 bis 240 als 1. Byte und 117 als 2. Byte mit XROM 01,53 angezeigt und zwar egal, ob das 1. Byte ein Vorsilbenbyte ist oder nicht. Dies ist aus zwei Gründen auch sofort einzu- sehen:

- Es ist kein Gerät mit der Identität-# 01 und einem Befehl 53 angeschlossen.
- Jedes Dezimalbyte 0-240 ergibt binär Zahlen, deren Nybble 2 0000 ist.

Einige Proben mit 117 als zweitem Byte:

32 RCL 00	bytedezimal:	32		117
	binär:	0010	0000 01	11 0101
	XR-dezimal:			01 53
128 DEG	bytedezimal:	128		117
	binär:	1000	0000 01	11 0101
	XR-dezimal:			01 53
208 GTO M	bytedezimal:	208		117
	binär:	1101	0000 01	11 0101
	XR-dezimal:			01 53

Trotz unterschiedlicher Bytekombinationen ergibt sich stets XROM 01,53. Das Abarbeiten der Befehle erfolgt, indem zuerst der Vorsilbenplatz und anschließend der Nachsilbenplatz gecheckt wird. Enthält der Vorsilbenplatz im Tastenregister einen Einbytebefehl, wird dieser ausgeführt und der nachfolgende Befehl ignoriert. Ist Platz eins eine Vorsilbe, so werden beide Befehle zum Segen der synthetischen Programmierung auch dann ausgeführt, wenn sie als XEQ-Befehl über das Tastenfeld oder als Barcodes verweigert werden. Hier fehlt zum Glück eine rechnerinterne Prüfung. Die XROM-Befehle sind, wie wir bereits in Abschnitt f. gehört haben, mit den Direktausführungsbytes 160 bis 167 verschlüsselt. Rechnet man demgemäß die XROM-# 01,53 in eine XEQ-# um, so ergibt sich = 160 167. (Es handelt sich nicht um den Befehl XEQ selbst, sondern um eine Zweibytefunktion eines externen Gerätes.) Das Umrechnungsprogramm lernen wir im folgenden Abschnitt kennen. Legen wir diese Bytekombination auf eine USER-Taste, so wird deren Betätigung mit 'NONEXISTENT' beantwortet. Dies ist logisch, weil ja 'XEQ 160 167' in einer periferen Einheit vorhanden sein müßte, die nicht exi-

tiert. Wäre sie vorhanden, würde der Textname der Funktion gemäß CAT erscheinen wie z.B. 'XROM 29,20' - 'PRX' - 'XEQ 167 84'. Es ist somit klar, XROM-Befehle oder hieraus abgeleitete XEQ-Funktionen von synthetischen Befehlen gibt es nicht. Aus gleichem Grund kann es keine Barcodes davon geben. Was soll der Prozessor auch mit Barcodes 'XROM 01,53' wo doch die Periferie fehlt und im Rechner 16 Möglichkeiten bestehen? Wie es mit Barcodes 'RCL M' - 'XEQ 144 117' steht, wird im nächsten Abschnitt besprochen.

Treten wir nun noch den praktischen Beweis an: das nachfolgende kleine Programm "XROMKA" verzichtet wegen kürzestmöglicher Ausführung auf das Setzen der Tastenzuweisungsflags. Deswegen müssen zuvor Strohmännbytes eingegeben sein. Belegt werden die Tasten 11( $\Sigma$ ), 12(1/x), 13( $\sqrt{x}$ ) und 14(LOG). Lesen Sie jetzt die auf Seite 169 stehenden 5 Barcodes 'ASN "% 11' bis 'GTO..'; Ihre evtl. auf diesen Tasten liegenden bisherigen Zuweisungen werden dadurch überschrieben.  
ASN "% 11 Zur Kontrolle können Sie den Drucker-Mode-Schalter auf 'NORM' schalten.  
ASN "% 12 Sie erhalten dann nebenstehenden Begleitausdruck. Anschließend  
ASN "% 13 lesen Sie dann bitte das folgende Programm "XROMKA" ein. Achtung: Dies  
ASN "% 14 hochsynthetische Programm darf nicht mit 'SST' durchgeführt werden, GTO.. weil dies mit Sicherheit zu dem ärgerlichen 'MEMORY LOST' führt.

PROGRAMM: "XROMKA" Benötigte Register: 8 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-4



Reihe 3: Zeilen 5-8



Reihe 4: Zeilen 8-11



Reihe 5: Zeilen 12-18



Lesen Sie sodann die folgenden Funktionen ein:

GTO .. XEQ "XROMKA"



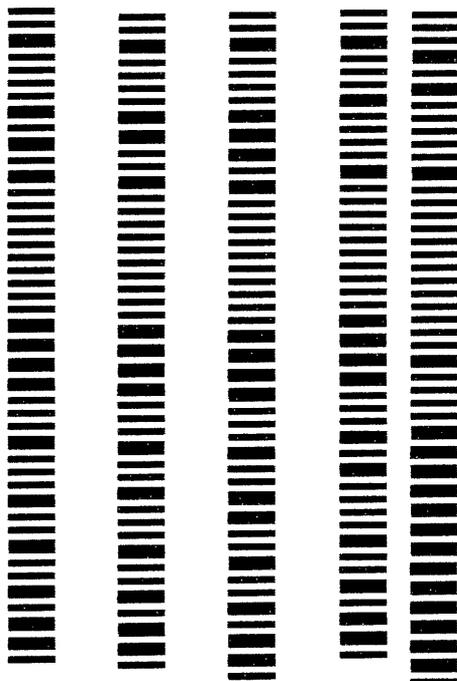
XEQ "PRKEYS"



Dieser Befehl ist die Ausführung 'XEQ'ALPHA'"PRKEYS"' (Byte 30 + Alphastring), der gemeinsam mit 'XEQ 167 76' (bewirkt ebenfalls 'PRKEYS') im folgenden Abschnitt behandelt wird. Die kürzeste Form ist 'XROM 29,12' wie wir bereits gesehen haben. Sie erhalten den links stehenden Ausdruck plus Ihre sonst noch bestehenden anderen Zuweisungen. Führen Sie nun noch einmal 'GTO..' aus, schalten in den PRGM-Mode und tasten ein: 'LBL "T"'. Drücken Sie dann nacheinander im USER-Modus die Tasten 11, 12, 13 und 14. Es erscheint jedesmal der Code 'XROM 01,53' dann 'RCL M', 'XROM 01,53' (entsprechend 160 117), 'GTO M' und 'DEG'. Schalten Sie Drucker-Mode nun wieder auf 'MAN'. Ein Ausdruck des Programms "T" steht rechts und zeigt die im Programm vorhandenen Befehle. Auf der nächsten Seite ist der Inhalt der beiden Tastenzuweisungsregister, welche mit dem Programm "XROMKA" gefüllt wurden, dargestellt. Weiter finden Sie einige Anmerkungen zu Zeilen des KA-Programms.

USER KEYS:		01*LBL "T"
11 XROM 01,53	Ausdruck plus Ihre sonst noch bestehenden anderen Zuweisungen.	02 RCL [
12 XROM 01,53	Führen Sie nun noch einmal 'GTO..' aus,	03 XROM 01,53
13 XROM 01,53	schalten in den PRGM-Mode und tasten ein: 'LBL "T"'. Drücken Sie dann nacheinander im USER-Modus die Tasten 11, 12, 13 und 14. Es erscheint jedesmal der Code 'XROM 01,53' dann 'RCL M', 'XROM 01,53' (entsprechend 160 117), 'GTO M' und 'DEG'. Schalten Sie Drucker-Mode nun wieder auf 'MAN'. Ein Ausdruck des Programms "T" steht rechts und zeigt die im Programm vorhandenen Befehle. Auf der nächsten Seite ist der Inhalt der beiden Tastenzuweisungsregister, welche mit dem Programm "XROMKA" gefüllt wurden, dargestellt. Weiter finden Sie einige Anmerkungen zu Zeilen des KA-Programms.	04 GTO [
14 XROM 01,53		05 DEG
		06 END

ASN "% 11 ASN "% 12 ASN "% 13 ASN "% 14 GTO ..



01*LBL "XROMKA"	Blick auf KA-Register
02 "u"uR"	-----
03 RCL c	00 240 KA-Register 01
04 RCL I	01 208 GTO
05 "µ++"	02 117 M
06 ASTO c	03 33 Code für Ta 13
07 STO 00	04 128 DEG
08 "u!uI"	05 117 M = "tot"
09 X<>Y	06 49 Code für Ta 14
10 RCL I	07 240 KA-Register 00
11 "µ++"	08 144 RCL
12 ASTO c	09 117 M
13 STO 01	10 01 Code für Ta 11
14 X<>Y	11 160 XEQ-Code= "tot"
15 STO c	12 117 M = "tot"
16 CLST	13 17 Code für Ta 12
17 CLA	
18 END	

Es lauten die Zeilen des Programms "XROMKA" wie folgt: 02 DEZ: 240, 144, 117, 1, 160, 117, 17 - 05 DEZ: 12, 0, 0 - 08 DEZ: 240, 208, 117, 33, 128, 117, 49 - 11 DEZ: 12, 0, 1. Dezi-mal 240 ist der Code für das KA-Regi-ster. Zunächst finden Sie in Register 00 die Bytefolge gemäß Zeile 02, so-dann in Register 01 diejenige von Zei-le 08. 05 und 11 sind Steuerzeilen.

CLP "XROMKA"



#### h. Arbeiten mit Tastenfeld- und XROM-Barcodes

Das Arbeiten mit diesen Barcodes ist weitgehend identisch mit Tastenfeldeingaben, nur wesentlich schneller. Funktionen wie 'MOD'; 'INT' oder 'DEC', die mit XEQ über das Tastenfeld recht bedächtig arbeiten, werden mit Barcodes praktisch verzögerungsfrei ausgeführt. Tastenbelegungen häufig benutzter Funktionen entfallen, was Programmspeicher freigibt.

Genau wie die 'Zweibyte Funktions- und Druckerzeichen' ASCII 37, 42, 43, 45, 47 im RUN-Modus die Funktionen %, \*, +, -, / ausführen, können die Funktionen 64+, 65-, 66\*, 67/ und 76% ins Alpharegister eingelesen werden. Alle übrigen schalten vom ALPHA- in den RUN-Modus und führen den Befehl aus. Barcodes vom 'END' bewirken im RUN-Modus nichts, im PRGM-Modus wird ein 'END' gesetzt wie über das Tastenfeld.

#### - Nicht programmierbare Funktionen durch die Hintertür

Genau wie bei 'eGØBEEP' beschrieben können mit XROM-Barcodes alle nicht programmierbaren Befehle in den Programmspeicher eingebracht werden: Gerät oder Modul entfernen, Barcodes der XROM-Funktion einlesen, Gerät wieder anschließen oder Modul einstecken und der Funktionsname steht in Programmzeile. Ob eine solche an sich nicht programmierbare Funktion trotzdem voll oder eingeschränkt nutzbar ist, muß man herausfinden. Die Tabellen der XROM-Funktionen Seiten 158 bis 166 beinhalten auch die XROM-Nummern der nicht programmierbaren Befehle.

Diese besondere Arbeitsweise ist mit den XROM-Barcodes auch sonst möglich. Wie Sie wissen, werden mit den PRGM "XROM.." -BC von den eingegebenen Zahlen in je-

dem Fall die entsprechenden Barcodes angefertigt und auch eingelesen. Existiert das Gerät und die Funktion wird sie im RUN-Modus sofort ausgeführt; im PRGM-Modus wird der CATalog verzeichnete Funktionsname in die PRGM-Zeile geschrieben. Existiert das Gerät oder die Funktion nicht, wird im RUN-Modus 'NONEXISTENT' aufgerufen, im PRGM-Modus jedoch immer die XROM-Zahl in die Programmzeile geschrieben und zwar völlig unabhängig davon ob es ein Gerät mit dieser Identitätsnummer und der aufgerufenen Funktion überhaupt gibt oder nicht. Alle XROM-Zahlen von 00,00 bis 31,63 können mit Barcodes eingelesen werden. Sobald nun ein Gerät mit der IDI-# und dem Befehl angeschlossen wird, ist der Klartextbefehl zu lesen und die Funktion ist verfügbar. So ist es mit XROM-BC möglich Programme aufgrund von Listings in den Rechner einzugeben auch, wenn das Periferiegerät und dessen CAT mit dem Klartextbefehl nicht angeschlossen ist. Von so ohne Periferie eingegebenen Programmen können auch ohne weiteres PRGM-Barcodes angefertigt und später wieder eingelesen werden. Das Abspeichern auf andere Massenspeicher ist ebenfalls erlaubt. Diese Art der Programmeingabe ist nur mit Barcodes, neuerdings auch mit dem CCD-Modul, möglich. Kleine Probe: führen Sie 'GTO..' aus, entfernen Sie den Stecker des IL-Moduls, lesen Sie in den Programmspeicher ein



und Sie sehen 'XROM 29,20'. Schalten Sie den Rechner aus, schließen das IL-Modul wieder an, Schalten den Rechner ein und in den PRGM-Modus: 'PRX'. Löschen Sie dann den Befehl, damit er nicht unbeabsichtigt hängen bleibt.

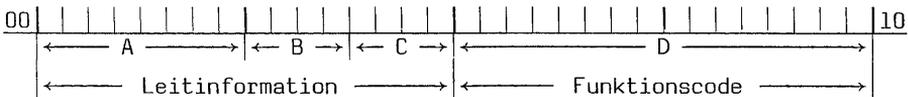
**III. Mehrbytebarcodes - Tastenfeldfunktionen mit und ohne Argument**

Während die Barcodes Abschnitt II. von den Zweibytefunktionen nur die Vorsilbe aufrufen, können bei diesem Barcodetyp Vorsilbe und Nachsilbenargument in eine BC-Zeile 'gegossen' werden. Es ist dabei ohne weiteres möglich, auch 'IND'-Funktionen oder Zweibyte-XROMbefehle einzugeben. Auch nicht programmierbare Rechner- oder XROM-Funktionen können mit Argument als BC angefertigt werden. Beispiele: 'CLP' "XROMKA" oder 'LIST 027'.

a. Programmierbare sowie nicht programmierbare Rechner- und XROM-Funktionen

Alle Barcodes dieser Funktionen entsprechen den tastbaren oder mit 'XEQ' einzugebenden Befehlen, im wesentlichen 'CAT 2' und 'CAT 3' des Rechners. Die Zweibyte-rechnerbefehle erfordern dabei immer ein numerisches Argument. Nicht programmierbare Rechnerfunktionen wie 'BST' oder 'PACK' sind eingeschlossen. Solche, die ein Argument erfordern wie 'LIST' oder 'COPY' sind hier ausgenommen. Dafür gibt es separate Programme. Sofern diese Funktionen als XEQ-Barcodes eingegeben werden, arbeiten sie wie solche ohne Argument: 'LIST\_\_'.

1. Aufbau



- A = 8-Bit Prüfsumme
- B = 4-Bit Typindikator
- C = 4-Bit ungenutzt
- D = ein oder zwei Bytes Funktionscode

In der Praxis bestehen diese Barcodes aus drei oder vier Bytes, je nachdem ob ein Argument belegt ist oder nicht. Da es sich hier immer um ein numerisches Argument handelt, werden Bytes zwischen 00 bis 99 oder IND 00 bis 99 gefordert.



### 3. Das Umrechnungsprogramm "XR-XEQ"

#### Programmbedienung.

- SIZE: begnügt sich mit 003

XEQ "XR-XEQ"



- "XR-XEQ-FNKT?": geben Sie die Buchstabenfolge des XROM-Befehls ein, z.B. "PRX" - 'R/S'
- "XROM-NR? N,NN": nun geben Sie die XROM-Zahlen ein, hier 29,20 - 'R/S'. Das Ergebnis ist: **PRX**

XROM 29,20 = XEQ 167/84

Den kleinen Selbstschutz und den Fehlerhinweis "XROM >31,63" gibt es hier auch.

01*LBL "XR-XEQ"	18 ACA	35 RDN	52 *	69 "I/"
02*LBL 00	19 "I-NR? N,NN"	36 32	53 INT	70 ARCL 01
03 "α1X"	20*LBL 01	37 X<=Y?	54 64	71 ACA
04 RCL I	21 TONE 7	38 GTO 02	55 X<=Y?	72 ADV
05 X<> d	22 PROMPT	39 MOD	56 GTO 02	73 BEEP
06 CLST	23 FC? 22	40 RCL X	57 MOD	74 CLST
07 ADV	24 GTO 01	41 4	58 R↑	75 CLA
08 "XR-XEQ FNKT?"	25 ABS	42 ST/ Z	59 +	76 RTN
09 TONE 7	26 STO 00	43 MOD	60 STO 01	77 GTO 00
10 STOP	27 ENTER↑	44 64	61 160	78*LBL 02
11 AOFF	28 INT	45 *	62 ST+ 00	79 "XROM > 31.63"
12 FC? 23	29 " "	46 X<>Y	63 FIX 0	80 ACA
13 GTO 00	30 10	47 INT	64 " = XEQ "	81 ADV
14 SF 12	31 X>Y?	48 STO 00	65 ACA	82 TONE 0
15 PRA	32 "I-0"	49 R↑	66 FMT	83 GTO 00
16 CF 12	33 ARCL 00	50 FRC	67 CLA	84 END
17 "XROM"	34 ACA	51 E2	68 ARCL 00	186 Bytes

### 4. Herstellen von XEQ-Funktionsbarcodes

Die drei Programme sind wegen einer Reihe Zuordnungsabfragen etwas länger, und die durch Moduleinsatz erreichbaren Verkürzungen von 590 über 560 bis 494 Bytes nicht so bedeutend. Barcodes von nicht programmierbaren Funktionen ohne Argument wie 'R/S' oder 'PACK' können mit diesen Programmen angefertigt werden. Ebenso BC des Bytes 174, welches zwei Funktionen bewirkt: 'GTO IND' wird ausgeführt, wenn das Argument zwischen 00 bis 99 oder für STACK und L zwischen 112 und 116 liegt. 'XEQ IND' erfordert die gleichen Zahlen +128. Beispiele: 'GTO IND 00' = 174 0 - 'XEQ IND 00' = 174 128.

#### 4.1 Nur Rechner HP 41 CV - Drucker, keine Module PRGM "XEQBC"

##### Programmbedienung:

- SIZE: wäre mit 'SIZE 002' zufrieden

XEQ "XEQBC"



- "FUNKTIONNAME?": geben Sie die Buchstaben/Zeichenfolgen der Funktion ein. Z.B. "XEQ IND 17" - 'R/S'
- "DEZ BYTE 1 ?": hier geben Sie den Dezimalwert des 1. Byte ein. Ist dies eine Einbytefunktion, z.B. 'R/S' - geben Sie 5 ein - 'R/S', und die Barcodes werden gedruckt. Ist es eine Vorsilbe, z.B. 174 werden Sie gefragt:
- "DEZ BYTE 2?": nun geben Sie die Nachsilbe ein, z.B. für 'XEQ IND 17' = 145 (17+128) und drücken 'R/S', die Barcodereihe wird ausgeführt.

Geben Sie einen Dezimalwert ein, der in diesem Zusammenhang nicht existiert, erfolgt Fehlermeldung: "DezBytNonexi" - anschließend wird erneut gefragt: "FUNKTIONAME?" und Sie können mit richtiger Eingabe wiederholen. Falls Sie ein Byte einer nicht programmierbaren Funktion für die ein eigenes PRGM besteht eingeben, erfolgt Fehlerhinweis: "Sep Programm" und danach "FUNKTIONAME?". Ist eine Barcodesreihe angefertigt, so drücken Sie für eine zweite BC-Herstellung einfach 'R/S'.

XEQ-Tabelle der Rechnerfunktionen - XROM-Befehle entsprechen XEQ 160-167

Funktion	Dez	Funktion	Dez								
+	64	CF__	169	FC?C__	171	MOD	75	RTN	133	X#0?	99
-	65	CHS	84	FIX _	156	OCT	111	SVED	125	X<0?	102
*	66	CLA	135	FRC	105	OFF	141	SCI _	157	X<=0?	123
/	67	CLD	127	FS?__	172	ON	9	SF__	168	X>0?	100
1/X	96	CLRG	138	FS?C__	170	P-R	78	E+	71	X=Y?	120
10^X	87	CLE	112	GRAD	130	PACK	10	E-	72	X#Y?	121
ABS	97	CLST	115	GTO__	200	PRGM	13	SREG__	153	X<Y?	68
ACOS	93	CLX	119	GTO IND_	174	%	76	SIN	89	X<=Y?	70
ADV	143	COS	90	HMS	100	%CH	77	SIGN	122	X>Y?	69
ALPHA	12	D-R	106	HMS+	73	PI	114	SQRT	82	X<>_	206
ADFF	139	DEC	95	HMS-	74	PROMPT	142	SST	8	X<>Y	113
ADN	140	DEG	128	HR	109	PSE	137	ST+__	146	XEQ 160-167	
ARCL__	155	DSE__	151	INT	104	R†	116	ST-__	147	XEQ IND_	174
ASHF	136	END	192	ISG__	150	R-D	107	ST*__	148	XEQ__	224
ASIN	92	ENG _	158	LASTX	110	R-P	79	ST/_	149	X†2	81
ASTO__	154	ENTER†	131	LBL__	207	RAD	129	STO__	145	Y†X	83
ATAN	94	E†X	85	LN	80	RCL__	144	STOP	132		
AVIEW	126	E†X-1	88	LN1+X	101	RDN	117	TAN	91		
BEEP	134	FACT	98	LOG	86	RND	110	TOPE	159		
BST	7	FC?__	173	MEAN	124	R/S	5	USER	14		

Zu den Vorsilbenbytes sind die folgenden Nachsilben möglich:

Vorsilben	Nachsilben	Vorsilben	Nachsilben
144-155	00- 99 112-116 IND 128-227 IND 240-244	156-159	0- 9 IND 128-227 IND 128-227
168-171	00- 29 IND 128-227 IND 240-244	172-173	00- 55 IND 128-227 IND 240-244
174 'GTO IND'	00- 99 112-116	206	00- 99 112-116
'XEQ IND'	128-227 240-244		IND 128-227 IND 240-244
207, 208, 224	00- 99 102-111 123-127	<u>Es sind:</u> 112-116	X, Y, Z, T, L

Alle diese Funktionen entsprechen den tastbaren Eingaben. Weil dies so ist, und die synthetischen Funktionen nicht eingetastet werden können, verweigert der Rechner die Annahme synthetischer BC-Befehle wie z.B. 'RCL M' dezimal 144 117.

01+LBL "XEQBC"	04 RCL I	07 CLRG	10 TONE 7	13 FC? 23
02+LBL 00	05 X<> d	08 ADV	11 STOP	14 GTO 00
03 "d)+"	06 CLST	09 "FUNKTIONAME?"	12 ADFF	15 ACA

16 ADV	74 X=Y?	132 X<>Y	190 127	248 RCL 01
17 15	75 GTO 17	133 X=Y?	191 X<>Y	249 FS? 02
18+LBL 01	76 RDN	134 SF 01	192 X<=Y?	250 XEQ 03
19 "DEZ BYTE 1 ?"	77 207	135+LBL 17	193 GTO 20	251 CF 17
20 TONE 7	78 X=Y?	136 SF 00	194 227	252 ADV
21 PROMPT	79 GTO 08	137+LBL 08	195 X<>Y	253 ADV
22 FC?C 22	80 RDN	138 99	196 X<=Y?	254 CLST
23 GTO 01	81 208	139 XEQ 09	197 CF 01	255 BEEP
24 ABS	82 X=Y?	140 X<=Y?	198 X<=Y?	256 ADV
25 STO 00	83 GTO 08	141 CF 01	199 GTO 19	257 ADV
26 ACX	84 RDN	142 X<=Y?	200 239	258 RTN
27 X<=Y?	85 224	143 GTO 19	201 X<>Y	259 GTO 00
28 GTO 02	86 X=Y?	144 FS?C 00	202 X<=Y?	260+LBL 20
29 30	87 GTO 08	145 GTO 10	203 GTO 20	261 ADV
30 X<>Y	88 GTO 20	146 101	204 244	262 "BezBytNonexi"
31 X<=Y?	89+LBL 07	147 X<>Y	205 X<>Y	263 TONE 0
32 GTO 07	90 GTO 07	148 X<=Y?	206 X<=Y?	264 ACA
33 63	91+LBL 02	149 GTO 20	207 CF 01	265 ADV
34 X<>Y	92 X=Y?	150 111	208 X<=Y?	266 GTO 00
35 X<=Y?	93 GTO 07	151 X<>Y	209 GTO 19	267+LBL 03
36 GTO 20	94 4	152 X<=Y?	210 GTO 20	268 SF 25
37 143	95 X<>Y	153 GTO 19	211+LBL 19	269 X=0?
38 X<>Y	96 X<=Y?	154 122	212 FS?C 01	270 ACCOL
39 X<=Y?	97 GTO 07	155 X<>Y	213 GTO 20	271 X=0?
40 GTO 19	98 6	156 X<=Y?	214 ADV	272 GTO 05
41 155	99 X=Y?	157 GTO 20	215 ADV	273+LBL 04
42 X<>Y	100 GTO 07	158 127	216 64	274 XEQ 06
43 X<=Y?	101 X<>Y	159 X<>Y	217 RCL 00	275 ARCL X
44 GTO 16	102 GTO 19	160 X<=Y?	218 +	276 RDN
45 159	103+LBL 03	161 GTO 19	219 RCL 01	277+LBL 05
46 X<>Y	104 9	162 GTO 20	220 +	278 OUTA
47 X<=Y?	105 GTO 06	163+LBL 09	221 SF 03	279 CLA
48 GTO 03	106+LBL 04	164 "DEZ BYTE 2 ?"	222 XEQ 02	280 RDN
49 167	107 29	165 TONE 7	223+LBL 02	281 RTN
50 X<>Y	108 GTO 06	166 PROMPT	224 RCL X	282+LBL 06
51 X<=Y?	109+LBL 05	167 FC?C 22	225 256	283 127
52 GTO 15	110 55	168 GTO 09	226 ST/ Z	284 X<>Y
53 171	111+LBL 06	169 ABS	227 MOD	285 X<=Y?
54 X<>Y	112 XEQ 09	170 STO 01	228 +	286 GTO 07
55 X<=Y?	113 X<=Y?	171 "/"	229 INT	287 -
56 GTO 04	114 GTO 19	172 10	230 FS?C 03	288 1
57 173	115 GTO 18	173 X>Y?	231 RTN	289 +
58 X<>Y	116+LBL 07	174 "H"	232 SF 17	290 ,
59 X<=Y?	117 ADV	175 RDN	233 ,	291 1
60 GTO 05	118 "Sep Programm"	176 ARCL X	234 ACCOL	292 BLDSPEC
61 174	119 TONE 9	177 ACA	235 RDN	293 X<>Y
62 X=Y?	120 ACA	178 SF 02	236 130	294 BLDSPEC
63 GTO 17	121 ADV	179 RTN	237 FS? 02	295 RTN
64 RDN	122 GTO 00	180+LBL 10	238 1	296+LBL 07
65 192	123+LBL 15	181 111	239 FS? 02	297 ,
66 X=Y?	124 XEQ 09	182 X<>Y	240 +	298 X<>Y
67 GTO 19	125 255	183 X<=Y?	241 XEQ 04	299 BLDSPEC
68 RDN	126 X<>Y	184 GTO 20	242 XEQ 04	300 END
69 205	127 X<=Y?	185 116	243 "0"	590 Bytes
70 X=Y?	128 GTO 19	186 X<>Y	244 XEQ 05	
71 GTO 07	129 GTO 20	187 X<=Y?	245 RCL 00	
72 RDN	130+LBL 16	188 GTO 19	246 XEQ 04	
73 206	131 153	189+LBL 18	247 FS? 02	

4.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "XEQBc"

Programmbedienung: und auch SIZE wie "XEQBC"

XEQ "XEQBc"



01*LBL "XEQBc"	52 GTO 15	103*LBL 03	154 122	205 X<>Y
02*LBL 00	53 171	104 9	155 X<>Y	206 X<=Y?
03 "a)+"	54 X<>Y	105 GTO 06	156 X<=Y?	207 CF 01
04 RCL I	55 X<=Y?	106*LBL 04	157 GTO 20	208 X<=Y?
05 X<> d	56 GTO 04	107 29	158 127	209 GTO 19
06 CLST	57 173	108 GTO 06	159 X<>Y	210 GTO 20
07 CLRG	58 X<>Y	109*LBL 05	160 X<=Y?	211*LBL 19
08 ADV	59 X<=Y?	110 55	161 GTO 19	212 FS?C 01
09 "FUNKTIONNAME?"	60 GTO 05	111*LBL 06	162 GTO 20	213 GTO 20
10 TONE 7	61 174	112 XEQ 09	163*LBL 09	214 ADV
11 STOP	62 X=Y?	113 X<=Y?	164 "DEZ BYTE 2 ?"	215 ADV
12 AOFF	63 GTO 17	114 GTO 19	165 TONE 7	216 64
13 FC? 23	64 RDN	115 GTO 18	166 PROMPT	217 RCL 00
14 GTO 00	65 192	116*LBL 07	167 FC?C 22	218 +
15 ACA	66 X=Y?	117 ADV	168 GTO 09	219 RCL 01
16 ADV	67 GTO 19	118 "Sep Programm"	169 ABS	220 +
17 15	68 RDN	119 TONE 9	170 STO 01	221 SF 03
18*LBL 01	69 205	120 ACA	171 "/*"	222 XEQ 02
19 "DEZ BYTE 1 ?"	70 X=Y?	121 ADV	172 10	223*LBL 02
20 TONE 7	71 GTO 07	122 GTO 00	173 X>Y?	224 RCL X
21 PROMPT	72 RDN	123*LBL 15	174 "I=0"	225 256
22 FC?C 22	73 206	124 XEQ 09	175 RDN	226 ST/ Z
23 GTO 01	74 X=Y?	125 255	176 ARCL X	227 MOD
24 ABS	75 GTO 17	126 X<>Y	177 ACA	228 +
25 STO 00	76 RDN	127 X<=Y?	178 SF 02	229 INT
26 ACX	77 207	128 GTO 19	179 RTN	230 FS?C 03
27 X<=Y?	78 X=Y?	129 GTO 20	180*LBL 10	231 RTN
28 GTO 02	79 GTO 08	130*LBL 16	181 111	232 SF 17
29 30	80 RDN	131 153	182 X<>Y	233 ,
30 X<>Y	81 208	132 X<>Y	183 X<=Y?	234 ACCOL
31 X<=Y?	82 X=Y?	133 X=Y?	184 GTO 20	235 RDN
32 GTO 07	83 GTO 08	134 SF 01	185 116	236 130
33 63	84 RDN	135*LBL 17	186 X<>Y	237 FS? 02
34 X<>Y	85 224	136 SF 00	187 X<=Y?	238 1
35 X<=Y?	86 X=Y?	137*LBL 00	188 GTO 19	239 FS? 02
36 GTO 20	87 GTO 00	138 99	189*LBL 18	240 +
37 143	88 GTO 20	139 XEQ 09	190 127	241 XEQ 04
38 X<>Y	89*LBL 07	140 X<=Y?	191 X<>Y	242 XEQ 04
39 X<=Y?	90 GTO 07	141 CF 01	192 X<=Y?	243 "0"
40 GTO 19	91*LBL 02	142 X<=Y?	193 GTO 20	244 XEQ 05
41 155	92 X=Y?	143 GTO 19	194 227	245 RCL 00
42 X<>Y	93 GTO 07	144 FS?C 00	195 X<>Y	246 XEQ 04
43 X<=Y?	94 4	145 GTO 10	196 X<=Y?	247 FS? 02
44 GTO 16	95 X<>Y	146 101	197 CF 01	248 RCL 01
45 159	96 X<=Y?	147 X<>Y	198 X<=Y?	249 FS? 02
46 X<>Y	97 GTO 07	148 X<=Y?	199 GTO 19	250 XEQ 03
47 X<=Y?	98 6	149 GTO 20	200 239	251 ADV
48 GTO 03	99 X=Y?	150 111	201 X<>Y	252 ADV
49 167	100 GTO 07	151 X<>Y	202 X<=Y?	253 CF 17
50 X<>Y	101 X<>Y	152 X<=Y?	203 GTO 20	254 CLST
51 X<=Y?	102 GTO 19	153 GTO 19	204 244	255 BEEP

256 ADV	261 ADV	266 GTO 00	271 X=0?	276 OUTA
257 ADV	262 "DezBytNonexi"	267*LBL 03	272 GTO 05	277 CLA
258 RTN	263 TONE 0	268 SF 25	273*LBL 04	278 RDN
259 GTO 00	264 ACA	269 X=0?	274 XTOA	279 END
260*LBL 20	265 ADV	270 ACCOL	275*LBL 05	560 Bytes

#### 4.3 Rechner HP 41 CV - Drucker, XF- und Plottermodul PRGM "XEQbc"

##### Programmbedienung:

- SIZE: auch hier ist 002 ausreichend. Sonstige Bedienung genau wie "XEQBC".

XEQ "XEQbc"



01*LBL "XEQbc"	44 GTO 16	87 GTO 08	130*LBL 16	173 X>Y?
02*LBL 00	45 159	88 GTO 20	131 153	174 "+0"
03 "a)+"	46 X<>Y	89*LBL 07	132 X<>Y	175 RDN
04 RCL I	47 X<=Y?	90 GTO 07	133 X=Y?	176 ARCL X
05 X<> d	48 GTO 03	91*LBL 02	134 SF 01	177 ACA
06 CLST	49 167	92 X=Y?	135*LBL 17	178 SF 02
07 CLRG	50 X<>Y	93 GTO 07	136 SF 00	179 RTN
08 ADV	51 X<=Y?	94 4	137*LBL 08	180*LBL 10
09 "FUNKTIONAME?"	52 GTO 15	95 X<>Y	138 99	181 111
10 TONE 7	53 171	96 X<=Y?	139 XEQ 09	182 X<>Y
11 STOP	54 X<>Y	97 GTO 07	140 X<=Y?	183 X<=Y?
12 AOFF	55 X<=Y?	98 6	141 CF 01	184 GTO 20
13 FC? 23	56 GTO 04	99 X=Y?	142 X<=Y?	185 116
14 GTO 00	57 173	100 GTO 07	143 GTO 19	186 X<>Y
15 ACA	58 X<>Y	101 X<>Y	144 FS?C 00	187 X<=Y?
16 ADV	59 X<=Y?	102 GTO 19	145 GTO 10	188 GTO 19
17 15	60 GTO 05	103*LBL 03	146 101	189*LBL 18
18*LBL 01	61 174	104 9	147 X<>Y	190 127
19 "DEZ BYTE 1 ?"	62 X=Y?	105 GTO 06	148 X<=Y?	191 X<>Y
20 TONE 7	63 GTO 17	106*LBL 04	149 GTO 20	192 X<=Y?
21 PROMPT	64 RDN	107 29	150 111	193 GTO 20
22 FC?C 22	65 192	108 GTO 06	151 X<>Y	194 227
23 GTO 01	66 X=Y?	109*LBL 05	152 X<=Y?	195 X<>Y
24 ABS	67 GTO 19	110 55	153 GTO 19	196 X<=Y?
25 STO 00	68 RDN	111*LBL 06	154 122	197 CF 01
26 ACX	69 205	112 XEQ 09	155 X<>Y	198 X<=Y?
27 X<=Y?	70 X=Y?	113 X<=Y?	156 X<=Y?	199 GTO 19
28 GTO 02	71 GTO 07	114 GTO 19	157 GTO 20	200 239
29 30	72 RDN	115 GTO 18	158 127	201 X<>Y
30 X<>Y	73 206	116*LBL 07	159 X<>Y	202 X<=Y?
31 X<=Y?	74 X=Y?	117 ADV	160 X<=Y?	203 GTO 20
32 GTO 07	75 GTO 17	118 "Sep Programm"	161 GTO 19	204 244
33 63	76 RDN	119 TONE 9	162 GTO 20	205 X<>Y
34 X<>Y	77 207	120 ACA	163*LBL 09	206 X<=Y?
35 X<=Y?	78 X=Y?	121 ADV	164 "DEZ BYTE 2 ?"	207 CF 01
36 GTO 20	79 GTO 08	122 GTO 00	165 TONE 7	208 X<=Y?
37 143	80 RDN	123*LBL 15	166 PROMPT	209 GTO 19
38 X<>Y	81 208	124 XEQ 09	167 FC?C 22	210 GTO 20
39 X<=Y?	82 X=Y?	125 255	168 GTO 09	211*LBL 19
40 GTO 19	83 GTO 08	126 X<>Y	169 ABS	212 FS?C 01
41 155	84 RDN	127 X<=Y?	170 STO 01	213 GTO 20
42 X<>Y	85 224	128 GTO 19	171 "/*"	214 "0"
43 X<=Y?	86 X=Y?	129 GTO 20	172 10	215 RCL 00

216 XTOR	221 I	226 CLST	231 RTN	236 TONE 0
217 RCL 01	222 +	227 CLA	232 GTO 00	237 ACA
218 FS? 02	223 BCKSM	228 BEEP	233+LBL 20	238 ADV
219 XTOR	224 CHS	229 ADV	234 ADV	239 GTO 00
220 ALENG	225 BCO	230 ADV	235 "BezByt.Nonexi"	240 END 494 Bytes

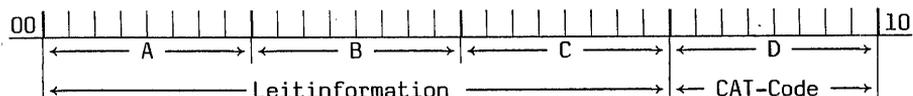
b. Nicht programmierbare Rechner- und XROM-Funktionen mit Argument

Es handelt sich um die Funktionen 'CAT', 'GTO.', 'GTO..', 'DELETE', 'COPY', 'CLP', 'SIZE', 'ASN' (im Rechner), 'LIST', 'PRP', 'NEWM' (im IL-Modul) und 'PLNG' (im CCD-Modul). Weil die Barcodetypindikatoren und die Argumente unterschiedlich sind, teils Zahlen, teils ASCII-Zeichen oder beides, gibt es für jeden Befehl ein eigenes Programm - wie bekannt in dreifach.

1. Die CATALOG-Funktion 'CAT'

Von dieser Funktion gibt es je nach Rechner oder bei eingesetztem CCD-Modul unterschiedliche Definitionsziffern. Die Programme rufen CAT 0 bis 6 auf. Wer CAT nn Barcodes auch vom CCD-Modul - CAT 7 bis F benötigt, kann die Programme leicht abändern.

1.1 Aufbau



- A = 8-Bit Prüfsumme
- B = 8-Bit Typindikator
- C = 8-Bit Identifikationscode
- D = 8-Bit CAtalog-Code

1.2 Der Typindikator, Identifikations- und CAtalog-Code

Der Typindikator sagt dem Rechner: B = 64 - es ist ein XEQ-BC. C = 0 - die CAT-Funktion soll ausgeführt werden, und D = 0 bis 6 - der definierte CAT ist auszuführen.

1.3 Die 8-Bit Prüfsumme

Da C = Null ist, berechnet sich diese Prüfsumme sehr einfach aus 64 + CAT-Code.

1.4 Herstellen von 'CAT' - 0 bis 6 Barcodes

Die drei Programme sind wegen der einfachen Alternativen nur kurz.

1.4.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "CATBC"

Programmbedienung:

-SIZE: ist mit 000 zufrieden.

XEQ "CATBC"



- "CAT-NR: 0-6?": CAtalog-# eingeben; Fehlermeldung erfolgt nicht. Geben Sie eine CAT-# >6 ein, Rückkehr zu: "CAT-NR: 0-6?".

01+LBL "CATBC"	04 CF 00	07 TONE 7	10 X?Y?	13 GTO 00
02+LBL 00	05 6	08 PROMPT	11 GTO 00	14 X=0?
03 CF 22	06 "CAT-NR: 0-6?"	09 ABS	12 FC? 22	15 SF 00

16 SF 12	27 SF 17	38 FS? 00	49 RTN	60 ARCL X
17 "CAT "	28 ,	39 XEQ 03	50 GTO 00	61 OUTA
18 FIX 0	29 ACCOL	40 FC?C 00	51+LBL 01	62 RTN
19 ARCL X	30 XEQ 01	41 XEQ 02	52 OUTA	63+LBL 03
20 ACA	31 ""	42 CF 17	53 CLA	64 SF 25
21 ADV	32 XEQ 01	43 CLST	54 RTN	65 ,
22 ADV	33 XEQ 02	44 ADV	55+LBL 02	66 ACCOL
23 RCL X	34 RDN	45 ADV	56 RDN	67 END
24 64	35 "0"	46 BEEP	57 ,	135 Bytes
25 +	36 XEQ 01	47 ADV	58 X<>Y	
26 CF 12	37 XEQ 03	48 ADV	59 BLD\$PEC	

#### 1.4.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "CATbc"

Programmbedienung: und SIZE wie "CATBC"

XEQ "CATbc"



01+LBL "CATbc"	14 X=0?	27 SF 17	40 FC?C 00	53 CLA
02+LBL 00	15 SF 00	28 ,	41 XEQ 02	54 RTN
03 CF 22	16 SF 12	29 ACCOL	42 CLST	55+LBL 02
04 CF 00	17 "CAT "	30 XEQ 01	43 CF 17	56 RDN
05 6	18 FIX 0	31 ""	44 ADV	57 XTOA
06 "CAT-NR: 0-6?"	19 ARCL X	32 XEQ 01	45 ADV	58 OUTA
07 TONE 7	20 ACA	33 XEQ 02	46 BEEP	59 RTN
08 PROMPT	21 ADV	34 RDN	47 ADV	60+LBL 03
09 ABS	22 ADV	35 "0"	48 ADV	61 SF 25
10 X<>Y?	23 RCL X	36 XEQ 01	49 RTN	62 ,
11 GTO 00	24 64	37 XEQ 03	50 GTO 00	63 ACCOL
12 FC? 22	25 +	38 FS? 00	51+LBL 01	64 END
13 GTO 00	26 CF 12	39 XEQ 03	52 OUTA	131 Bytes

#### 1.4.3 Rechner HP 41 C - Drucker, Plottermodul PRGM "CATbc"

Programmbedienung: und SIZE wie "CATBC" (XF-Modul nicht erforderlich)

XEQ "CATbc"

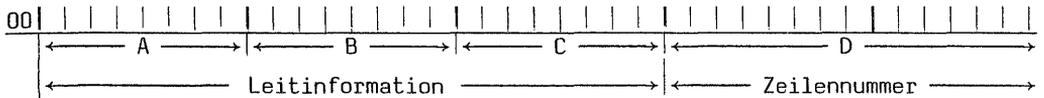


01+LBL "CATbc"	08 ABS	15 FIX 0	22 CF 12	29 RTN
02+LBL 00	09 X<>Y?	16 ARCL X	23 BCO	30 GTO 00
03 CF 22	10 GTO 00	17 ACA	24 CLST	31 END
04 6	11 FC? 22	18 "00"	25 CLA	74 Bytes
05 "CAT-NR: 0-6?"	12 GTO 00	19 ARCL X	26 ADV	
06 TONE 7	13 SF 12	20 -4	27 BEEP	
07 PROMPT	14 "CAT "	21 BCCK\$H	28 ADV	

## 2. Die Funktionen 'GTO.' und 'GTO..'

Barcodes z.B. von 'GTO.015' positionieren das Arbeitsprogramm auf Zeile 15 genau wie beim Betätigen über das Tastensfeld und zwar sowohl im RUN-, PRGM- und ALPHA-Modus. Der Rechner bleibt im Alpha-Modus. Nur das Appendzeichen muß erneut angefertigt werden, weil eine momentane Zeile durch das Einlesen endet! Die Funktionen 'GTO. "ALPHA"' sind als Barcodes leider nicht vorgesehen.

### 2.1 Aufbau



A = 8-Bit Prüfsumme

B = 8-Bit Typindikator

C = 8-Bit Identifikationscode

D = 2 Bytes Zeilennummer

### 2.2 Der Typindikator und die Zeilennummer

Es wird dem Rechner mitgeteilt: B = 64 - ein XEQ-BC ist auszuführen; C = 1 - es soll eine GTOZeilennummer-Funktion ausgeführt werden; D = zwei Bytes, ganz genau nur 12 Bit Zeilennummer. Diese Zeilennummer berechnet sich nach dem gleichen, in Kapitel 3 II. 1.2 Folgenummer besprochenen Verfahren. Bis einschließlich 255 - alle Bits von Byte zwei sind 1 - wird nur in Byte zwei gezählt. Bei 256 erfolgt 'Überlauf' in Nybble 2 von Byte eins. Dies bedeutet hier 1x 256 - Byte 2 ist dann Null. Zeile 257 ist 1x 256 (Byte 1) + 1 (Byte 2). Da Nybble 1 von Byte eins ungenutzt ist, ergibt sich als höchstmögliche rechnerische Zeilennummer  $15 \times 256 + 255 = 3840 + 255 = 4095$ . Die Eingabe dieser Zeilennummer bewirkt 'GTO..'.

### 2.3 Die 8-Bit Prüfsumme

Die Prüfsumme ist einfach errechnet: z.B. für 'GTO..'. B= 64 + C= 1 ergibt 65 + Byte eins von D= 15, macht 80 + Byte zwei von D= 255 - Ergebnis:  $335 : 256 = 1$  Rest 79 - Prüfsumme = 80.

### 2.4 Herstellen von 'GTO.'- und 'GTO..'-Barcodes

Die drei Programme unterscheiden sich lediglich in ihrer Länge und im 'SIZE'.

#### 2.4.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "GTOZBC"

##### Programmbedienung

- SIZE: es wäre 001 ausreichend

XEQ "GTOZBC"



- "ZEILENUMMER?": nun tasten Sie die gewünschte Zeilennummer zwischen Null, das ist gleich 'GTO.000' und maximal 1999, das ist die höchstmögliche Rechnerzeilennummer oder  $4095 = 'GTO..'$  ein.

Geben Sie eine Zeilennummer größer 1999 ein, erfolgt die Fehler-Kurzmeldung: "Z-Nr zu groß". Anschließend neu: "ZEILENUMMER?".

Das Listing dieses Programms finden Sie auf Seite 180.

#### 2.4.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "GTOZBc"

##### Programmbedienung

-SIZE: und Bedienung genau wie "GTOZBC"

XEQ "GTOZBc"



Das Listing auch dieses Programms finden Sie auf Seite 180. Die Zeile Nummer 48 ist in beiden Programmen das Byte 132. Zeilen 52 und 54 sind Bytes 64 und 1.

01*LBL "GTOZBC"	25 1999	49 XEQ 06	73 ADV	97 RTN
02 CF 12	26 X<Y?	50 RDN	74 TONE 0	98*LBL 07
03 CF 13	27 GTO 02	51 XEQ 05	75 GTO 00	99 127
04 SF 20	28*LBL 01	52 "0"	76*LBL 03	100 X<Y?
05 CF 29	29 "1."	53 XEQ 06	77 RCL X	101 X<Y?
06 FIX 0	30*LBL 02	54 "-"	78 256	102 GTO 00
07*LBL 00	31 ACA	55 XEQ 06	79 ST/ Z	103 -
08 4095	32 ADV	56 RDN	80 MOD	104 1
09 "ZEILENUMMER?"	33 ADV	57 RCL 00	81 X<Y?	105 +
10 TONE 7	34 X<Y?	58 XEQ 04	82 INT	106 .
11 PROMPT	35 GTO 01	59 RDN	83 RTN	107 1
12 ABS	36 XEQ 03	60 XEQ 04	84*LBL 04	108 BLDSPEC
13 "GTO ."	37 STO 00	61 CLST	85 SF 25	109 X<Y?
14 X=Y?	38 65	62 CLA	86 X=0?	110 BLDSPEC
15 GTO 01	39 +	63 ADV	87 ACCOL	111 RTN
16 100	40 X<Y?	64 ADV	88 X=0?	112*LBL 00
17 X<Y?	41 STO Z	65 BEEP	89 GTO 06	113 .
18 "1-0"	42 +	66 ADV	90*LBL 05	114 X<Y?
19 X<Y?	43 XEQ 03	67 ADV	91 XEQ 07	115 BLDSPEC
20 10	44 +	68 RTN	92 ARCL X	116 END
21 X<Y?	45 SF 17	69 GTO 00	93 RDN	227 Bytes
22 "1-0"	46 .	70*LBL 01	94*LBL 06	
23 X<Y?	47 ACCOL	71 "Z-Nr zu groß"	95 OUTA	
24 ARCL X	48 "	72 ACA	96 CLA	

01*LBL "GTOZbc"	20 10	39 +	58 XEQ 04	77 RCL X
02 CF 12	21 X<Y?	40 X<Y?	59 RDN	78 256
03 CF 13	22 "1-0"	41 STO Z	60 XEQ 04	79 ST/ Z
04 SF 20	23 X<Y?	42 +	61 CLST	80 MOD
05 CF 29	24 ARCL X	43 XEQ 03	62 CLA	81 X<Y?
06 FIX 0	25 1999	44 +	63 ADV	82 INT
07*LBL 00	26 X<Y?	45 SF 17	64 ADV	83 RTN
08 4095	27 GTO 02	46 .	65 BEEP	84*LBL 04
09 "ZEILENUMMER?"	28*LBL 01	47 ACCOL	66 ADV	85 SF 25
10 TONE 7	29 "1."	48 "	67 ADV	86 X=0?
11 PROMPT	30*LBL 02	49 XEQ 06	68 RTN	87 ACCOL
12 ABS	31 ACA	50 RDN	69 GTO 00	88 X=0?
13 "GTO ."	32 ADV	51 XEQ 05	70*LBL 01	89 GTO 06
14 X=Y?	33 ADV	52 "0"	71 "Z-Nr zu groß"	90*LBL 05
15 GTO 01	34 X<Y?	53 XEQ 06	72 ACA	91 XTOA
16 100	35 GTO 01	54 "-"	73 ADV	92*LBL 06
17 X<Y?	36 XEQ 03	55 XEQ 06	74 TONE 0	93 OUTA
18 "1-0"	37 STO 00	56 RDN	75 GTO 00	94 CLA
19 X<Y?	38 65	57 RCL 00	76*LBL 03	95 END 197 Bytes!

### 2.4.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "GTOZbc"

#### Programmbedienung

-SIZE: es werden keine Datenregister benötigt. Sonstige Bedienung wie "GTOZBC".

XEQ "GTOZbc"



Listing dieses Programms finden Sie auf Seite 181. Zeile 41 faßt die Zeilen 52 und 54 der vorhergehenden Programme zusammen. Ansonsten ist alles gleich.

01*LBL "GTOZbc"	14 X=Y?	27 GTO 02	40 INT	53 RTH
02 CF 12	15 GTO 01	28*LBL 01	41 "0:"	54 GTO 00
03 CF 13	16 100	29 "t."	42 XTOA	55*LBL 01
04 SF 28	17 X>Y?	30*LBL 02	43 X<>Y	56 "Z-Nr zu groß"
05 CF 29	18 "f0"	31 ACA	44 XTOA	57 ACA
06 FIX 0	19 X<>Y	32 ADV	45 -5	58 ADV
07*LBL 00	20 10	33 X>Y?	46 BCCKSH	59 TONE 0
08 4095	21 X>Y?	34 GTO 01	47 BCD	60 GTO 00
09 "ZEILENUMMER?"	22 "f0"	35 RCL X	48 CLST	61 END
10 TONE 7	23 X<>Y	36 256	49 CLA	142 Bytes
11 PROMPT	24 ARCL X	37 ST/ Z	50 ADV	
12 ABS	25 1999	38 MOD	51 BEEP	
13 "GTO ."	26 X<>Y	39 X<>Y	52 ADV	

### 3. Die Funktion 'DELETE' - Löschen einer Anzahl PRGM-Zeilen

Barcodes dieser Funktion 'DELnnn' löschen nur im PRGM-Modus ab der momentanen Position die definierte Anzahl Programmzeilen. Einlesen im RUN- oder ALPHA-Modus wird ignoriert.

#### 3.1 Aufbau, Typindikator, Zeilenanzahl und 8-Bit Prüfsumme

Bis auf den Identifikations-Code 'C', der hier auf 2 gesetzt ist, sind alle übrigen Daten im Aufbau gleich mit denen von 'GTO.' Der Zeilennummer entspricht hier die Anzahl der zu löschenden Programmzeilen. Die Zahl 4095 spielt bei diesem Typ natürlich keine Rolle.

#### 3.2 Herstellen von 'DEL ' Barcodes

Naturgemäß sind die drei Programme für die Herstellung dieses Typs denen der 'GTO.'-Barcodes sehr ähnlich, weil ja nur der Idi-Code abweicht.

##### 3.2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "DELBC"

###### Programmbedienung:

-SIZE: 001 ist ausreichend

XEQ "DELBC"



- DEL-ANZAHL?": Falls Sie Null eingeben, wird dies abgelehnt, und die Frage wiederholt. Geben Sie zwischen 1 - 1999 ein. Ab 2000 erfolgt Irrtumshinweis: "DEL\_ zu groß".

##### 3.2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "DELBC"

XEQ "DELBC"



##### 3.2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "DELbc"

XEQ "DELbc"



Programmbedienungen sind identisch mit "DELBC". SIZE könnte bei "DELbc" 000 sein. Die drei Listing folgen auf der nächsten Seite. Die Zeile 43 führt das Byte 132, Zeile 47 das Byte 64 und Zeile 49 das Byte 2. Zeile 36 die Byte 64 2.

01*LBL "DELBC"	24 1999	47 "0"	70 GTO 00	93*LBL 06
02 CF 12	25 X<Y	48 XEQ 05	71*LBL 02	94 127
03 CF 13	26 ACA	49 "X"	72 RCL X	95 X<Y
04 SF 28	27 ADV	50 XEQ 05	73 256	96 X<Y?
05 CF 29	28 ADV	51 RDN	74 ST/ Z	97 GTO 07
06 FIX 0	29 X>Y?	52 RCL 00	75 MOD	98 -
07*LBL 00	30 GTO 01	53 XEQ 03	76 X<Y	99 1
08 "DEL-ANZAHL?"	31 XEQ 02	54 RDN	77 INT	100 +
09 TONE 7	32 STO 00	55 XEQ 03	78 RTN	101 .
10 PROMPT	33 66	56 CLST	79*LBL 03	102 1
11 X=0?	34 +	57 CLA	80 SF 25	103 BLDSPEC
12 GTO 00	35 X<Y	58 ADV	81 X=0?	104 X<Y
13 ABS	36 STO Z	59 ADV	82 ACCOL	105 BLDSPEC
14 "DEL "	37 +	60 BEEP	83 X=0?	106 RTN
15 100	38 XEQ 02	61 ADV	84 GTO 05	107*LBL 07
16 X>Y?	39 +	62 ADV	85*LBL 04	108 .
17 "+0"	40 SF 17	63 RTN	86 XEQ 06	109 X<Y
18 X<Y	41 .	64 GTO 00	87 ARCL X	110 BLDSPEC
19 10	42 ACCOL	65*LBL 01	88 RDN	111 END
20 X>Y?	43 ""	66 "DEL_ zu groß"	89*LBL 05	213 Bytes
21 "+0"	44 XEQ 05	67 ACA	90 OUTA	
22 X<Y	45 RDN	68 ADV	91 CLA	
23 ARCL X	46 XEQ 04	69 TONE 0	92 RTN	

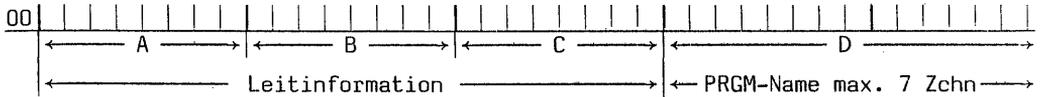
01*LBL "DELbc"	19 10	37 +	55 XEQ 03	73 256
02 CF 12	20 X>Y?	38 XEQ 02	56 CLST	74 ST/ Z
03 CF 13	21 "+0"	39 +	57 CLA	75 MOD
04 SF 28	22 X<Y	40 SF 17	58 ADV	76 X<Y
05 CF 29	23 ARCL X	41 .	59 ADV	77 INT
06 FIX 0	24 1999	42 ACCOL	60 BEEP	78 RTN
07*LBL 00	25 X<Y	43 ""	61 ADV	79*LBL 03
08 "DEL-ANZAHL?"	26 ACA	44 XEQ 05	62 ADV	80 SF 25
09 TONE 7	27 ADV	45 RDN	63 RTN	81 X=0?
10 PROMPT	28 ADV	46 XEQ 04	64 GTO 00	82 ACCOL
11 X=0?	29 X>Y?	47 "0"	65*LBL 01	83 X=0?
12 GTO 00	30 GTO 01	48 XEQ 05	66 "DEL_ zu groß"	84 GTO 05
13 ABS	31 XEQ 02	49 "X"	67 ACA	85*LBL 04
14 "DEL "	32 STO 00	50 XEQ 05	68 ADV	86 XTOA
15 100	33 66	51 RDN	69 TONE 0	87*LBL 05
16 X>Y?	34 +	52 RCL 00	70 GTO 00	88 OUTA
17 "+0"	35 X<Y	53 XEQ 03	71*LBL 02	89 CLA
18 X<Y	36 STO Z	54 RDN	72 RCL X	90 END 183 Bytes

01*LBL "DELbc"	13 ABS	25 X<Y	37 XTOA	49 GTO 00
02 CF 12	14 "DEL "	26 ACA	38 X<Y	50*LBL 01
03 CF 13	15 100	27 ADV	39 XTOA	51 "DEL_ zu groß"
04 SF 28	16 X>Y?	28 X>Y?	40 -5	52 ACA
05 CF 29	17 "+0"	29 GTO 01	41 BCKSM	53 ADV
06 FIX 0	18 X<Y	30 RCL X	42 BCO	54 TONE 0
07*LBL 00	19 10	31 256	43 CLST	55 GTO 00
08 "DEL-ANZAHL?"	20 X>Y?	32 ST/ Z	44 CLA	56 END
09 TONE 7	21 "+0"	33 MOD	45 ADV	128 Bytes
10 PROMPT	22 X<Y	34 X<Y	46 BEEP	
11 X=0?	23 ARCL X	35 INT	47 ADV	
12 GTO 00	24 1999	36 "0X"	48 RTN	

#### 4. Die Funktion 'COPY' von XROM-Programmen in den Rechnerspeicher

Bei diesen Barcodes ist das Argument erstmalig ein Alphastring, nämlich der zu kopierende Programmname mit maximal 7 ASCII-Zeichen.

##### 4.1 Aufbau



- A = 8-Bit Prüfsumme
- B = 8-Bit Typindikator
- C = 8-Bit Identifikations-Code
- D = Maximal 7 Byte Programm-Name

##### 4.2 Der Typindikator, Identifikations-Code und Programm-Name

Wiederum werden dem Rechner Daten mitgeteilt: B = 64 - es ist ein XEQ-Barcode; C = 3 - der Code für die 'COPY'-Funktion und D = der Programm-Name, bestehend aus 1 bis 7 ASCII-Zeichen.

##### 4.3 Die 8-Bit Prüfsumme

Die Prüfsumme setzt sich in diesem Fall zusammen aus  $B+C = 67$  + aller Byte-Dezimalwerte der ASCII-Zeichen des Programmnamens. Z.B. "WNDTST" aus dem Lese-stift. Die Prüfsumme errechnet sich:  $67 (B+C) + 87 + 78 + 68 + 84 + 83 + 84 = 552:256 = 2 \text{ Rest } 39$  - Prüfsumme = 41.

##### 4.4 Herstellen von 'COPY' - Barcodes

Die drei Programme sind wieder sehr einfach zu bedienen.

###### 4.4.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "COPYBC"

###### Programmbedienung:

-SIZE: hier ist mindestens 008 erforderlich.

XEQ "COPYBC"



- "COPY-NAME?": Geben Sie die Buchstabenfolge des zu kopierenden XROM-Programm-Namen ein, z.B. "WNDTST" - 'R/S'

Geben Sie mehr als 7 Buchstaben ein: "Name zu lang". Falls Sie keinen Namen eingeben (Alpharegister leer): "Kein Name!". Diese Fehlermeldungen sparen Papier. Es wird hingegen nicht geprüft, ob ein gültiger XROM-Name eingegeben wurde. Es werden von jedem beliebigen Alphastring 1 - 7 Zeichen BC gefertigt.

###### 4.4.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "COPYBc"

###### Programmbedienung:

-SIZE und Bedienung wie "COPYBC".

XEQ "COPYBc"



Die Listings der beiden Programme folgen auf der nächsten Seite. Zeile 105 in "COPYBC" ist das Byte 64 und Zeile 107 das Byte 3 für B bzw. C. In "COPYBc" ist das Byte 64 in Zeile 85 und das Byte 3 in Zeile 87 enthalten. Im Gegensatz zu PRGM "COPYbc" beide Zeile 26 - müssen sie hier einzeln abgearbeitet werden.

01*LBL "COPYBC"	35 GTO 13	69*LBL 03	103 XEQ 08	137 127
02*LBL 00	36*LBL 01	70 RCL 00	104 XEQ 08	138 X<>Y
03 CF 12	37 STO IND 00	71 INT	105 "0"	139 X<=Y?
04 CF 13	38 XEQ 02	72 RCL IND X	106 XEQ 09	140 GTO 11
05 FIX 0	39 X=0?	73 DSE Y	107 "+"	141 -
06 CLRG	40 GTO 03	74 GTO 04	108 XEQ 09	142 1
07 CF 23	41 ISG 00	75 GTO 05	109 RCL 00	143 +
08 "COPY-NAME?"	42 GTO 01	76*LBL 04	110 INT	144 .
09 TONE 7	43*LBL 02	77 RCL IND Y	111*LBL 06	145 1
10 AON	44 "↑↓****"	78 +	112 XEQ 07	146 BLDSPC
11 PROMPT	45 RCL [	79 DSE Y	113 DSE X	147 X<>Y
12 AOFF	46 STO [	80 GTO 04	114 GTO 06	148 BLDSPC
13 FC? 23	47 CLX	81*LBL 05	115 ADV	149 RTN
14 GTO 00	48 X<> ]	82 67	116 ADV	150*LBL 11
15 RCL \	49 SIGN	83 +	117 CF 17	151 .
16 X=0?	50 CLX	84 SF 00	118 CLST	152 X<>Y
17 GTO 12	51 X<> \	85 XEQ 05	119 BEEP	153 BLDSPC
18 ASTO 00	52 "↑***"	86*LBL 05	120 ADV	154 RTN
19 ASHF	53 X<> [	87 RCL X	121 ADV	155*LBL 12
20 ASTO 01	54 X<> L	88 256	122 ADV	156 ACA
21 "COPY "	55 X<> \	89 ST/ Z	123 RTN	157 ADV
22 ARCL 00	56 INT	90 MOD	124 GTO 00	158 "Name zu lang"
23 ARCL 01	57 ST+ ]	91 +	125*LBL 07	159 GTO 02
24 "↑"	58 RDN	92 INT	126 RCL IND X	160*LBL 13
25 ACA	59 6	93 FS?C 00	127*LBL 08	161 "Kein Name!"
26 ADV	60 ST+ ]	94 RTN	128 XEQ 10	162*LBL 02
27 ADV	61 RDN	95 SF 17	129 ARCL X	163 ACA
28 CLA	62 E1	96 .	130 RDN	164 ADV
29 ARCL 00	63 ST+ L	97 ACCOL	131*LBL 09	165 TONE 0
30 ARCL 01	64 X<> L	98 RDN	132 OUTA	166 GTO 00
31 1.007	65 ST+ ]	99 130	133 CLA	167 END
32 STO 00	66 CLX	100 RCL 00	134 RDN	325 Bytes
33 XEQ 02	67 X<> ]	101 INT	135 RTN	
34 X=0?	68 RTN	102 +	136*LBL 10	

01*LBL "COPYBc"	22 14	43 GTO 01	64 SF 00	85 "0"
02*LBL 00	23 ALENG	44*LBL 02	65 XEQ 05	86 XEQ 09
03 CF 12	24 X>Y?	45 -1	66*LBL 05	87 "+"
04 CF 13	25 GTO 10	46 AROT	67 RCL X	88 XEQ 09
05 FIX 0	26 ACA	47 ATOX	68 256	89 RCL 00
06 CLRG	27 ADV	48 RTN	69 ST/ Z	90 INT
07 CF 23	28 ADV	49*LBL 03	70 MOD	91*LBL 06
08 "COPY-NAME?"	29 CLA	50 RCL 00	71 +	92 XEQ 07
09 TONE 7	30 ARCL 00	51 INT	72 INT	93 DSE X
10 AON	31 ARCL 01	52 RCL IND X	73 FS?C 00	94 GTO 06
11 PROMPT	32 1.007	53 DSE Y	74 RTN	95 ADV
12 AOFF	33 STO 00	54 GTO 04	75 SF 17	96 ADV
13 FC? 23	34 XEQ 02	55 GTO 05	76 .	97 CF 17
14 GTO 00	35 X=0?	56*LBL 04	77 ACCOL	98 CLST
15 ASTO 00	36 GTO 11	57 RCL IND Y	78 RDN	99 BEEP
16 ASHF	37*LBL 01	58 +	79 130	100 ADV
17 ASTO 01	38 STO IND 00	59 DSE Y	80 RCL 00	101 ADV
18 "COPY "	39 XEQ 02	60 GTO 04	81 INT	102 ADV
19 ARCL 00	40 X=0?	61*LBL 05	82 +	103 RTN
20 ARCL 01	41 GTO 03	62 67	83 XEQ 08	104 GTO 00
21 "↑"	42 ISG 00	63 +	84 XEQ 08	105*LBL 07

106 RCL IND X	111 CLA	116 ADV	121*LBL 02	126 END
107*LBL 08	112 RDN	117 "Name zu lang"	122 ACA	254 Bytes
108 XTOA	113 RTN	118 GTO 02	123 ADV	
109*LBL 09	114*LBL 10	119*LBL 11	124 TONE 0	
110 OUTA	115 ACA	120 "Kein Name!"	125 GTO 00	

#### 4.4.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "COPYbc"

##### Programmbedienung:

-SIZE: bei diesem PRGM nur 002 - Bedienung wie "COPYBC"

XEQ "COPYbc"



01*LBL "COPYbc"	12 ASHF	23 X=Y?	34 BCO	45 "Name zu lang"
02*LBL 00	13 ASTO 01	24 GTO 02	35 CLST	46 GTO 03
03 CF 23	14 "COPY "	25 ACA	36 CLA	47*LBL 02
04 "COPY-NAME?"	15 ARCL 00	26 "0+"	37 ADV	48 "Kein Name!"
05 TONE 7	16 ARCL 01	27 ARCL 00	38 BEEP	49*LBL 03
06 AON	17 "L"	28 ARCL 01	39 ADV	50 ACA
07 STOP	18 14	29 ALENG	40 RTN	51 ADV
08 AOFF	19 ALENG	30 1	41 GTO 00	52 TONE 0
09 FC? 23	20 XYY?	31 +	42*LBL 01	53 GTO 00
10 GTO 00	21 GTO 01	32 CHS	43 ACA	54 END
11 ASTO 00	22 7	33 BCCKSM	44 ADV	131 Bytes

#### 5. Die Funktion 'CLP ' - Löschen von Programmen

Dies ist die Funktion 'CLP "xyz"', die einen Programm-Namen von 1 bis 7 Zeichen anfordert. Wird kein Programm-Name eingegeben, funktionieren die Barcodes wie 'CLP""'; es wird das Programm gelöscht, auf welches der Rechner positioniert ist.

##### 5.1 Aufbau, Typindikator, Identifikations-Code, PRGM-Name und Prüfsumme

Bis auf den Identifikations-Code der hier 4 ist, entsprechen alle übrigen Daten denen von "COPYBC". Der Programm-Name läßt Variationen zu, deren LBL-Anfertigung ohne Einsatz von synthetischer Programmierung im Abschnitt über Barcodes für 'GTO/XEQ/LBL "ALPHA"' noch behandelt wird. Das normaler Eingabe unzugängliche 'LBL ":ok!,.."' kann mit 'CLP ":ok!,.."' - Barcodes gelöscht werden.

##### 5.2 Herstellen von 'CLP ' - Barcodes

Barcodes zur Löschung von Programmen mit so exotischen Namen können mit allen drei Programmen "CLP.."-BC angefertigt werden.

##### 5.2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "CLPBC"

##### Programmbedienung:

-SIZE: müßte wenigstens 008 sein.

XEQ "CLPBC"



- "PRGM-NAME?": Geben Sie hier den noch so 'wilden' Namen unter Verwendung von Barcodes aus den Einzelzeichen-Tabellen ein - 'R/S'  
 Falls Sie keinen Namen eingeben und gleich 'R/S' drücken wird 'CLP""' gedruckt.

Ist der PRGM-Name hingegen >7 Zeichen: "Name zu lang" und Rückkehr zum Start.

01*LBL "CLPBC"	34 X=0?	67 ST+ ]	100 RDN	133*LBL 09
02*LBL 00	35 STO 00	68 CLX	101 130	134 OUTA
03 CF 12	36 X=0?	69 X<> ]	102 RCL 00	135 CLA
04 CF 13	37 GTO 05	70 RTN	103 INT	136 RDN
05 FIX 0	38*LBL 01	71*LBL 03	104 +	137 RTN
06 CLRG	39 STO IND 00	72 RCL 00	105 XEQ 08	138*LBL 10
07 CF 23	40 XEQ 02	73 INT	106 XEQ 08	139 127
08 *PRGM-NAME?"	41 X=0?	74 RCL IND X	107 "0"	140 X<>Y
09 TONE 7	42 GTO 03	75 DSE Y	108 XEQ 09	141 X(=Y?
10 AON	43 ISG 00	76 GTO 04	109 "a"	142 GTO 11
11 PROMPT	44 GTO 01	77 GTO 05	110 XEQ 09	143 -
12 AOFF	45*LBL 02	78*LBL 04	111 RCL 00	144 1
13 FC? 23	46 "+-+*****"	79 RCL IND Y	112 INT	145 +
14 CLA	47 RCL [	80 +	113*LBL 06	146 .
15 RCL \	48 STO [	81 DSE Y	114 XEQ 07	147 1
16 X=0?	49 CLX	82 GTO 04	115 DSE X	148 BLDSPEC
17 GTO 12	50 X<> ]	83*LBL 05	116 GTO 06	149 X<>Y
18 ASTO 00	51 SIGN	84 60	117 ADV	150 BLDSPEC
19 ASHF	52 CLX	85 +	118 ADV	151 RTN
20 ASTO 01	53 X<> \	86 SF 00	119 CF 17	152*LBL 11
21 "CLP ""	54 "+****"	87 XEQ 00	120 CLST	153 .
22 ARCL 00	55 X<> [	88*LBL 00	121 BEEP	154 X<>Y
23 ARCL 01	56 X<> L	89 RCL X	122 ADV	155 BLDSPEC
24 "-"	57 X<> \	90 256	123 ADV	156 RTN
25 ACA	58 INT	91 ST/ Z	124 ADV	157*LBL 12
26 ADV	59 ST+ ]	92 MOD	125 RTN	158 ACA
27 ADV	60 RDN	93 +	126 GTO 00	159 ADV
28 CLA	61 6	94 INT	127*LBL 07	160 "Name zu lang"
29 ARCL 00	62 ST* ]	95 FS?C 00	128 RCL IND X	161 ACA
30 ARCL 01	63 RDN	96 RTN	129*LBL 08	162 ADV
31 1.007	64 E1	97 SF 17	130 XEQ 10	163 TONE 0
32 STO 00	65 ST* L	98 .	131 ARCL X	164 GTO 00
33 XEQ 02	66 X<> L	99 ACCOL	132 RDN	165 END 309 Bytes

### 5.2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "CLPbc"

Programmbedienung: und SIZE wie "CLPBC"

XEQ "CLPbc"



### 5.2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "CLPbc"

Programmbedienung: wie "CLPBC" -SIZE: kommt mit 002 aus.

XEQ "CLPbc"



Die Listings der Programme folgen auf der nächsten Seite. Im Programm "CLPbc" können die Zeilen 79 und 80 nach 'ACCOL' entfallen. Anstelle der Zeile 79 muß 'RDN' eingefügt werden. Es entsteht dann die gleiche Zeilenfolge wie im vorstehenden Programm "CLPBC" Zeilen 97 bis 104.

### 6. Die Funktion 'SIZE' - Einstellen einer Registerzuweisung

Diese Funktion erfordert wieder ein numerisches Dreistellenargument 000 - 319.

01*LBL "CLPbc"	26 ADV	51 RCL 00	76 SF 17	101 BEEP
02*LBL 00	27 ADV	52 INT	77 .	102 ADV
03 CF 12	28 CLA	53 RCL IND X	78 ACCOL	103 ADV
04 CF 13	29 ARCL 00	54 DSE Y	79 "E1"	104 ADV
05 FIX 0	30 ARCL 01	55 GTO 04	80 XEQ 09	105 RTN
06 CLRG	31 1.007	56 GTO 05	81 130	106 GTO 00
07 CF 23	32 STO 00	57*LBL 04	82 RCL 00	107*LBL 07
08 "PRGM-NAME?"	33 XEQ 02	58 RCL IND Y	83 INT	108 RCL IND X
09 TONE 7	34 X=0?	59 +	84 +	109*LBL 08
10 AOM	35 STO 00	60 DSE Y	85 XEQ 08	110 XTOA
11 PROMPT	36 X=0?	61 GTO 04	86 XEQ 08	111*LBL 09
12 AOFF	37 GTO 05	62*LBL 05	87 "e"	112 OUTA
13 FC? 23	38*LBL 01	63 68	88 XEQ 09	113 CLA
14 CLA	39 STO IND 00	64 +	89 "a"	114 RDN
15 RCL \	40 XEQ 02	65 SF 00	90 XEQ 09	115 RTN
16 X=0?	41 X=0?	66 XEQ 00	91 RCL 00	116*LBL 10
17 GTO 10	42 GTO 03	67*LBL 00	92 INT	117 ACA
18 ASTO 00	43 ISG 00	68 RCL X	93*LBL 06	118 ADV
19 ASHF	44 GTO 01	69 256	94 XEQ 07	119 "Name zu lang"
20 ASTO 01	45*LBL 02	70 ST/ Z	95 DSE X	120 ACA
21 "CLP ""	46 -1	71 MOD	96 GTO 06	121 ADV
22 ARCL 00	47 AROT	72 +	97 ADV	122 TONE 0
23 ARCL 01	48 ATOX	73 INT	98 ADV	123 GTO 00
24 "t""	49 RTN	74 FS?C 00	99 CF 17	124 END
25 ACA	50*LBL 03	75 RTN	100 CLST	241 Bytes

01*LBL "CLPbc"	11 AOFF	21 ACA	31 +	41 GTO 00
02*LBL 00	12 FC? 23	22 13	32 CHS	42*LBL 01
03 CF 12	13 CLA	23 ALENG	33 BCCKSM	43 ADV
04 CF 13	14 ASTO 00	24 X>Y?	34 BCO	44 "Name zu lang"
05 CF 23	15 ASHF	25 GTO 01	35 CLST	45 ACA
06 FIX 0	16 ASTO 01	26 "ea"	36 CLA	46 ADV
07 "PRGM-NAME?"	17 "CLP ""	27 ARCL 00	37 ADV	47 TONE 0
08 TONE 7	18 ARCL 00	28 ARCL 01	38 BEEP	48 GTO 00
09 AOM	19 ARCL 01	29 ALENG	39 ADV	49 END
10 STOP	20 "t""	30 1	40 RTN	113 Bytes

### 6.1 Aufbau, Typindikator, Identifikations-Code, SIZE-Anzahl und 8-Bit Prüfsumme

Nur der Identifikations-Code, hier = 6, unterscheidet den Aufbau und die übrigen Daten sowie die Berechnung der Prüfsumme von "GTOZ.."- bzw. "DEL.."-BC.

### 6.2 Herstellen von 'SIZE ' - Barcodes

Die drei Programme ähneln denen der anderen PRGM mit Argumenteingabe natürlich.

#### 6.2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "SIZEBC"

##### Programmbedienung:

-SIZE: es wird nur Register 00 benötigt.

XEQ "SIZEBC"



- "SIZE\_\_?": Geben Sie die gewünschte SIZE-Zahl, je nach Rechner 000-319 ein. Ein größerer Wert führt zum Hinweis: "SIZE >319" und zum Neustart des PRGM's.

01*LBL "SIZEBC"	23 X<Y	45 "e"	67 TONE 0	89 CLA
02 CF 12	24 ACA	46 XEQ 05	68 GTO 00	90 RTN
03 CF 13	25 ADV	47 "f"	69*LBL 02	91*LBL 06
04 SF 28	26 ADV	48 XEQ 05	70 RCL X	92 127
05 CF 29	27 X>Y?	49 RDN	71 256	93 X<Y
06 FIX 0	28 GTO 01	50 RCL 00	72 SF/ Z	94 X<Y?
07*LBL 00	29 XEQ 02	51 XEQ 03	73 MOD	95 GTO 07
08 "SIZE...?"	30 STO 00	52 RDN	74 X<Y	96 -
09 TONE 7	31 70	53 XEQ 03	75 INT	97 1
10 PROMPT	32 +	54 CLST	76 RTN	98 +
11 ABS	33 X<Y	55 CLA	77*LBL 03	99 .
12 "SIZE "	34 STO Z	56 ADV	78 SF 25	100 1
13 100	35 +	57 ADV	79 X=0?	101 BLDSPEC
14 X>Y?	36 XEQ 02	58 BEEP	80 ACCOL	102 X<Y
15 "f0"	37 +	59 ADV	81 X=0?	103 BLDSPEC
16 X<Y	38 SF 17	60 ADV	82 GTO 05	104 RTN
17 10	39 .	61 RTN	83*LBL 04	105*LBL 07
18 X>Y?	40 ACCOL	62 GTO 00	84 XEQ 06	106 .
19 "f0"	41 "	63*LBL 01	85 ARCL X	107 X<Y
20 X<Y	42 XEQ 05	64 "SIZE > 319"	86 RDN	108 BLDSPEC
21 ARCL X	43 RDN	65 ACA	87*LBL 05	109 END
22 319	44 XEQ 04	66 ADV	88 OUTA	206 Bytes

6.2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "SIZEbc"

Programmbedienung: und SIZE wie "SIZEBC"

XEQ "SIZEbc"



01*LBL "SIZEbc"	19 "f0"	37 +	55 CLA	73 MOD
02 CF 12	20 X<Y	38 SF 17	56 ADV	74 X<Y
03 CF 13	21 ARCL X	39 .	57 ADV	75 INT
04 SF 28	22 319	40 ACCOL	58 BEEP	76 RTN
05 CF 29	23 X<Y	41 "	59 ADV	77*LBL 03
06 FIX 0	24 ACA	42 XEQ 05	60 ADV	78 SF 25
07*LBL 00	25 ADV	43 RDN	61 RTN	79 X=0?
08 "SIZE...?"	26 ADV	44 XEQ 04	62 GTO 00	80 ACCOL
09 TONE 7	27 X>Y?	45 "e"	63*LBL 01	81 X=0?
10 PROMPT	28 GTO 01	46 XEQ 05	64 "SIZE > 319"	82 GTO 05
11 ABS	29 XEQ 02	47 "f"	65 ACA	83*LBL 04
12 "SIZE "	30 STO 00	48 XEQ 05	66 ADV	84 XTOA
13 100	31 70	49 RDN	67 TONE 0	85*LBL 05
14 X>Y?	32 +	50 RCL 00	68 GTO 00	86 OUTA
15 "f0"	33 X<Y	51 XEQ 03	69*LBL 02	87 CLA
16 X<Y	34 STO Z	52 RDN	70 RCL X	88 END
17 10	35 +	53 XEQ 03	71 256	176 Bytes
18 X>Y?	36 XEQ 02	54 CLST	72 ST/ Z	

6.2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "SIZEbc"

Programmbedienung: wie "SIZEBC" -SIZE: könnte 000 sein.

XEQ "SIZEbc"



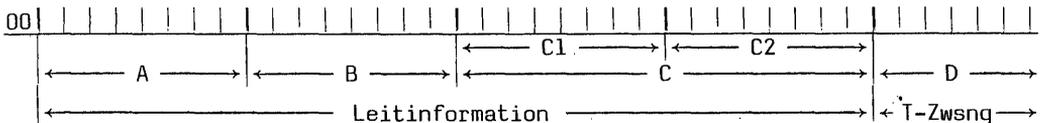
Das Listing für dies Programm mit XF- und Plottermodul folgt auf Seite 189.

01*LBL "SIZEbc"	12 *SIZE "	23 X<>Y	34 *@"	45 RTH
02 CF 12	13 100	24 ACA	35 XTOA	46 GTO 00
03 CF 13	14 X<>Y?	25 ADV	36 X<>Y	47*LBL 01
04 SF 28	15 "+0"	26 X<>Y?	37 XTOA	48 *SIZE > 319"
05 CF 29	16 X<>Y	27 GTO 01	38 -5	49 ACA
06 FIX 0	17 10	28 RCL X	39 BCCKSM	50 ADV
07*LBL 00	18 X<>Y?	29 256	40 BCD	51 TONE 0
08 *SIZE___?"	19 "+0"	30 ST/ Z	41 CLX	52 GTO 00
09 TONE 7	20 X<>Y	31 MOD	42 BEEP	53 END
10 PROMPT	21 ARCL X	32 X<>Y	43 ADV	120 Bytes
11 ABS	22 319	33 INT	44 ADV	

### 7. Die Funktion 'ASN ' - Tastenzuweisungen

Dieser Barcodetyp ist von den nichtprogrammierbaren Rechnerfunktionen mit Argument sicherlich der interessanteste. Außer dem Argument, dem Namen einer Funktion oder eines Programms, welches dann im Rechner vorhanden sein muß, ist auch noch die Taste einzubeziehen, welcher die Zuweisung aufgegeben werden soll.

#### 7.1 Aufbau



A = 8-Bit Prüfsumme

B = 1 Byte Typindikator

C = 2 Byte: C1 = Identifikations-Code und  
C2 = Tasten-Code

D = 1 bis 7 ASCII-Zeichen Tastenbelegung oder 'ASN "" nn'.

#### 7.2 Der Typindikator, Identifikations- und Tastencode

Diese Daten teilen dem Rechner mit: B = 64 - ein XEQ-Barcode ist auszuführen. C1 = 15 - der Identifikationscode für 'ASN ' ; C2 = Tastencode entsprechend Belegung über das Tastenfeld, also 'LN' = 15 oder 'VIEW' = -84.

#### 7.3 Definition und Berechnung des Tastencode

Die Tastencodes sind in der Bedienungsanleitung zum HP 41 definiert. Weil dies die Hexadezimalwerte der für Barcodes zu verarbeitenden dezimalen Tastencodes sind, ist die Berechnung nichts weiter als eine Umrechnung von z.B.  $\Sigma + \text{HEX } 11$  in DEZ 17. Das Verfahren ist sehr einfach:  $11:10 = 1 \times 16 + \text{Rest } 1 = 17$ . Für umgeschaltete Tasten, z.B. -11 wird der so berechnete Code analog zur 'IND'-Funktion um 128 erhöht, für -11 also 145. Da -81 nach dieser Methode den Code 257 ergeben würde, eine Zahl, die im 8-Bit-System nicht existiert, werden diese Codes MOD 256 genommen, für -81 demzufolge 1.

#### 7.4. Die 8-Bit Prüfsumme

Die Prüfsumme berechnet sich aus  $B+C1$  - hier stets  $64+15 = 79$  + C2 + Einzeldezimalwerte der nachfolgenden ASCII-Zeichen. Beispiel: 'PRX' soll auf die Taste 11 gelegt werden, so ist die Rechnung:  $79 (B+C1) + 17 (C2) + 80 (P) + 82 (R) + 88 (X)$  ergibt 346; dies geteilt durch 256 macht 1 Rest 90 - Prüfsumme = 91. Sie sehen, die Berechnungsverfahren für die Prüfsummen sind immer sehr ähnlich.

### 7.5 Herstellen von ASN-Barcodes

Mit den drei Programmen können Barcodes für die Tastenzuweisung sämtlicher Rechnerfunktionen ohne Argument erstellt werden, die auch über das Tastenfeld eingegeben werden können. Darüber hinaus lassen sich die abenteuerlichsten nicht tastbaren Programmnamen zuweisen, wie schon bei 'CLP..'BC erwähnt. D ist immer ein Alphastring, der im Rechner-CATalog oder als Alpha-LBL im Rechner verfügbar sein muß. XROM-Zuweisungen ohne angeschlossene Periferie sind nicht möglich. Der Versuch wird mit 'NONEXISTENT' beantwortet. Dies sicherlich deswegen, weil die Funktion 'ASN ' den String catalogabhängig buchstabiert. Ein einmal eingelesener XROM-Befehl bleibt auch bei nicht vorhandener Periferie erhalten; es erscheint die XROM-Nr. Gerät wieder da, Befehl sofort verfügbar.

01*LBL "ASNBC"	46 X<Y	91 X=0?	136 RCL 00	181*LBL 09
02*LBL 00	47 5	92 GTO 05	137 +	182 RCL IND X
03 "a9+"	48 X<Y?	93 ISG 00	138 SF 00	183*LBL 10
04 RCL [	49 GTO 15	94 GTO 03	139 XEQ 07	184 XEQ 12
05 X<) d	50 RDN	95*LBL 04	140*LBL 07	185 ARCL X
06 "FUNKTION?"	51 X<Y	96 "f-↓****"	141 RCL X	186 RDN
07 TONE 7	52 GTO 02	97 RCL [	142 256	187*LBL 11
08 STOP	53*LBL 01	98 STO [	143 ST/ Z	188 OUTA
09 AOFF	54 RDN	99 CLX	144 MOD	189 CLA
10 FC? 23	55 8	100 X<) I	145 +	190 RDN
11 CLA	56 X<Y?	101 SIGN	146 INT	191 RTN
12 RCL \	57 GTO 15	102 CLX	147 FS?C 00	192*LBL 12
13 X=0?	58 RDN	103 X<) \	148 RTN	193 127
14 GTO 14	59 X<Y	104 "f****"	149 SF 17	194 X<Y
15 ASTO 00	60 4	105 X<) [	150 .	195 X<=Y?
16 ASHF	61 X<Y?	106 X<) L	151 ACCOL	196 GTO 13
17 ASTO 01	62 GTO 15	107 X<) \	152 RDN	197 -
18 "ASN "	63 RDN	108 INT	153 131	198 1
19 ARCL 00	64 X<Y	109 ST+ J	154 RCL 00	199 +
20 ARCL 01	65*LBL 02	110 RDN	155 INT	200 .
21 "f--"	66 16	111 6	156 +	201 1
22 ACA	67 *	112 ST* J	157 XEQ 10	202 BLDSPEC
23 "TASTENCODE?"	68 +	113 RDN	158 XEQ 10	203 X<Y
24 TONE 7	69 .	114 EI	159 "e"	204 BLDSPEC
25 PROMPT	70 X<Y	115 ST* L	160 XEQ 11	205 RTN
26 ACX	71 FS?C 00	116 X<) L	161 "f"	206*LBL 13
27 ADV	72 128	117 ST+ J	162 XEQ 11	207 .
28 INT	73 +	118 CLX	163 RCL 00	208 X<Y
29 X<0?	74 256	119 X<) J	164 XEQ 10	209 BLDSPEC
30 SF 00	75 MOD	120 RTN	165 RCL 00	210 RTN
31 ABS	76 STO 00	121*LBL 05	166 INT	211*LBL 15
32 RCL X	77 ADV	122 RCL 00	167*LBL 00	212 "Code ??"
33 10	78 CLA	123 INT	168 XEQ 09	213 GTO 01
34 ST/ Z	79 ARCL 00	124 RCL IND X	169 DSE X	214*LBL 14
35 MOD	80 ARCL 01	125 DSE Y	170 GTO 00	215 ACA
36 X=0?	81 1.007	126 GTO 06	171 ADV	216 ADV
37 GTO 15	82 STO 00	127 GTO 07	172 ADV	217 "Funktion (<=?"
38 X<Y	83 XEQ 04	128*LBL 06	173 CF 17	218*LBL 01
39 INT	84 X=0?	129 RCL IND Y	174 CLST	219 ACA
40 X=0?	85 STO 00	130 +	175 BEEP	220 ADV
41 GTO 15	86 X=0?	131 DSE Y	176 ADV	221 TONE 0
42 3	87 GTO 07	132 GTO 06	177 ADV	222 GTO 00
43 X<Y?	88*LBL 03	133*LBL 07	178 ADV	223 END
44 GTO 01	89 STO IND 00	134 79	179 RTN	415 Bytes
45 RDN	90 XEQ 04	135 +	180 GTO 00	

7.5.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "ASNBC"

Programmbedienung:

-SIZE: es ist mindestens 009 vonnöten.

XEQ "ASNBC"



- "FUNKTION?": Geben Sie die Zeichenfolge des Zuweisungsnamens ein - 'R/S'. Soll die Funktion 'ASN' erstellt werden, ohne Eingabe gleich 'R/S' drücken.
- "TASTENCODE?": Hier geben Sie den gewünschten Tastencode zwischen 11 bis -84 ein - 'R/S'. Es können auch die Tasten 31/-31 belegt werden. Für eine zweite Barcodereihe einfach 'R/S' drücken, nicht neu mit XEQ starten.

Es wird nicht geprüft, ob eine Funktion vorhanden oder zulässig ist. BC werden von jeder ASCII-Folge erstellt. Bei unzulässigem Tastencode: "Code ??". Bei mehr als 7 ASCII-Zeichen: "Funktion <=?" und danach jeweils Neustart von vorn.

Es folgt hier nun erst das Listing des Programms "ASNbc" mit XF-Modul. Die Bedienung und die XEQ-Barcodes folgen auf der nächsten Seite. Diese Programme können alle mit 'SST' durchgeführt werden - man sieht dann was passiert.

01*LBL "ASNbc"	38 X<>Y	75 MOD	112*LBL 07	149 GTO 08
02*LBL 00	39 INT	76 STO 08	113 79	150 ADV
03 "α9+"	40 X=0?	77 ADV	114 +	151 ADV
04 RCL [	41 GTO 15	78 CLA	115 RCL 08	152 CF 17
05 X<> d	42 3	79 ARCL 00	116 +	153 CLST
06 "FUNKTION?"	43 X<Y?	80 ARCL 01	117 SF 00	154 BEEP
07 TONE 7	44 GTO 01	81 1.007	118 XEQ 07	155 ADV
08 STOP	45 RDN	82 STO 00	119*LBL 07	156 ADV
09 AOFF	46 X<>Y	83 XEQ 04	120 RCL X	157 ADV
10 FC? 23	47 5	84 X=0?	121 256	158 RTN
11 CLA	48 X<Y?	85 STO 00	122 ST/ Z	159 GTO 00
12 RCL \	49 GTO 15	86 X=0?	123 MOD	160*LBL 09
13 X=0?	50 RDN	87 GTO 07	124 +	161 RCL IND X
14 GTO 12	51 X<>Y	88*LBL 03	125 INT	162*LBL 10
15 ASTO 00	52 GTO 02	89 STO IND 00	126 FS?C 00	163 XTOA
16 ASHF	53*LBL 01	90 XEQ 04	127 RTN	164*LBL 11
17 ASTO 01	54 RDN	91 X=0?	128 SF 17	165 OUTA
18 "ASH "	55 8	92 GTO 05	129 .	166 CLA
19 ARCL 00	56 X<Y?	93 ISG 00	130 ACCOL	167 RDN
20 ARCL 01	57 GTO 15	94 GTO 03	131 RDN	168 RTN
21 "t"	58 RDN	95*LBL 04	132 131	169*LBL 15
22 ACA	59 X<>Y	96 -1	133 RCL 00	170 "Code ??"
23 "TASTENCODE?"	60 4	97 AROT	134 INT	171 GTO 01
24 TONE 7	61 X<Y?	98 ATOX	135 +	172*LBL 12
25 PROMPT	62 GTO 15	99 RTN	136 XEQ 10	173 ACA
26 ACX	63 RDN	100*LBL 05	137 XEQ 10	174 ADV
27 ADV	64 X<>Y	101 RCL 00	138 "0"	175 "Funktion <=?"
28 INT	65*LBL 02	102 INT	139 XEQ 11	176*LBL 01
29 X<0?	66 16	103 RCL IND X	140 "f"	177 ACA
30 SF 00	67 *	104 DSE Y	141 XEQ 11	178 ADV
31 ABS	68 +	105 GTO 06	142 RCL 08	179 TONE 0
32 RCL X	69 .	106 GTO 07	143 XEQ 10	180 GTO 00
33 10	70 X<>Y	107*LBL 06	144 RCL 00	181 END
34 ST/ Z	71 FS?C 00	108 RCL IND Y	145 INT	342 Bytes
35 MOD	72 128	109 +	146*LBL 08	
36 X=0?	73 +	110 DSE Y	147 XEQ 09	
37 GTO 15	74 256	111 GTO 06	148 DSE X	

7.5.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "ASNbc"

Programmbedienung:

-SIZE: und übrige Bedienung wie "ASNBC"

XEQ "ASNbc"



7.5.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "ASNbc"

Programmbedienung: wie "ASNBC" - SIZE: es wird hier nur 002 benötigt.

XEQ "ASNbc"



01*LBL "ASNbc"	23 ACA	45 GTO 01	67 16	89 CLA
02*LBL 00	24 "TASTENCODE?"	46 RDN	68 *	90 ADV
03 "α9+"	25 TONE 7	47 X<Y	69 +	91 BEEP
04 RCL [	26 PROMPT	48 5	70 .	92 ADV
05 STO d	27 ACX	49 X<Y?	71 X<Y	93 RTN
06 "FUNKTION?"	28 ADV	50 GTO 03	72 FS?C 00	94 GTO 00
07 TONE 7	29 INT	51 RDN	73 128	95*LBL 03
08 STOP	30 X<0?	52 X<Y	74 +	96 "Code ??"
09 AOFF	31 SF 00	53 GTO 02	75 256	97 GTO 05
10 FC?C 23	32 ABS	54*LBL 01	76 MOD	98*LBL 04
11 CLA	33 RCL X	55 RDN	77 ADV	99 ACA
12 ASTO 00	34 10	56 8	78 "0?"	100 ADV
13 ASHF	35 ST/ Z	57 X<Y?	79 XTOA	101 "Funktion (<=7"
14 ASTO 01	36 MOD	58 GTO 03	80 ARCL 00	102*LBL 05
15 "ASN "	37 X=0?	59 RDN	81 ARCL 01	103 ACA
16 ARCL 00	38 GTO 03	60 X<Y	82 ALENG	104 ADV
17 ARCL 01	39 X<Y	61 4	83 1	105 TONE 0
18 "t"	40 INT	62 X<Y?	84 +	106 GTO 00
19 13	41 X=0?	63 GTO 03	85 BCCKSM	107 END
20 ALENG	42 GTO 03	64 RDN	86 CHS	213 Bytes
21 X<Y?	43 3	65 X<Y	87 BCO	
22 GTO 04	44 X<Y?	66*LBL 02	88 CLST	

7.6 Tastenbelegung im Vergleich zu Barcodebenutzung

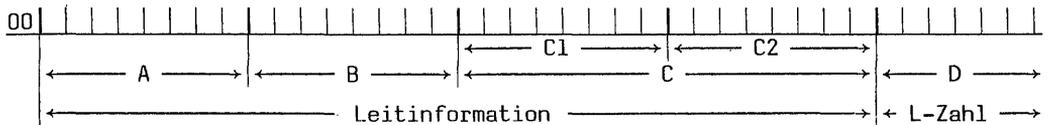
Der Sinn von Tastenzuweisungen ist es, häufig gebrauchte Funktionen, die nicht direkt, sondern nur mit 'XEQ "ALPHA"' eingegeben werden können, dennoch auf Tastendruck verfügbar zu haben. Von allen 'offiziellen' Rechner- oder Periferiefunktionen ohne und mit Argument können Barcodes für den gleichen Zweck benutzt werden und zwar ohne zusätzlichen Speicherbedarf. Tastenfeldschablonen oder sonstige Notizen, welche Tasten mit welchen Funktionen belegt sind, entfallen; die Barcodes sind ja alle einwandfrei beschriftet. Die Umschaltung 'USER'/'NORMAL'-Modus entfällt ebenso, wie das unbeabsichtigte Aufrufen einer 'USER'-Funktion wenn die normale Tastenfunktion gebraucht wird. Suchzeiten nach der USER-Funktion entfallen, Barcodes sind schneller. Die Programmierung synthetischer Befehle ist mit Barcodes sehr einfach und schnell, wie wir gesehen haben. Die synthetische Zuweisung von normalen Rechnerfunktionen wie 'ST+IND X' ist nicht nötig - mit Barcodes ist dies nur ein 'Federstrich'. Bleibt eigentlich nur noch die Sofortausführung synthetischer Befehle, die ja außer bei Einsatz des CCD-Moduls über das Tastenfeld nicht möglich ist. Braucht man nur wenige Befehle, kann die Zuweisung auf eine Taste bequem sein. Weil synthetische

Funktionen mit Barcodes nicht auf Tasten gelegt werden können, muß diese Zuweisung mit einem "KA" (Key Assignment)-Programm vorgenommen werden. Dies ist ein recht aufwendiger Vorgang. Wie ein synthetisches Tastenfeld für Sofortausführung vieler synthetischer Befehle mit Barcodes simuliert wird, kommt später.

### 8. Die Funktion 'LIST' - Ausdruck von Programmzeilen

Die LIST-Funktion ist die erste aus einem periferen Gerät, u.z. dem IL-Modul. Bei dieser und weiteren nicht programmierbaren 'Außenbordfunktionen' ist der Aufbau der Barcodes deswegen geringfügig erweitert.

#### 8.1 Aufbau



A = 8-Bit Prüfsumme

B = 1 Byte Typindikator

C = 2 Bytes 'XEQ'/'XROM'-Identifikation

D = 2 Bytes (genau 1 1/2 Bytes) - Anzahl zu listender Zeilen

#### 8.2 Der Typindikator und die XEQ-Identifikation

Der Rechner wird über folgendes informiert: B = 64 für 'XEQ'-Barcodes. C = die aus der XROM-Zahl für 'LIST' 29,07 mit dem Programm "XR-XEQ" berechneten beiden Bytes für XEQ Barcodes C1 = 167 und C2 = 71.

#### 8.3 Die 'LIST' - Anzahl

Die Anzahl der aufzulistenden Zeilen wird genau wie 'SIZE' oder 'DEL' berechnet. Bis 255 zählt der Rechner aus Byte D2, dann Überlauf nach Nybble 2 von Byte D1. Es können maximal 1999 Zeilen definiert werden. Dies ist auch die höchste Anzahl, die über das Tastenfeld mit der Tastenfolge 'XEQ' 'ALPHA' "LIST" 'ALPHA' 'EEX 999' eingegeben werden kann.

#### 8.4 Die 8-Bit Prüfsumme

Weil die Werte der Bytes B, C1 und C2 feststehen, es sind dies  $64+167+71 = 302$ , brauchen nur die beiden Dezimalwerte der Bytes D2 und D1 hinzugerechnet zu werden, um die Ausgangszahl für die Prüfsummenberechnung zu erhalten. Beispiel: es sollen 308 Zeilen gelistet werden;  $308:256 = 1$  (Wert D1) Rest 52 (Wert D2). Die Prüfsumme:  $302 (B+C1+C2) + 53 (D2+D1) = 355:256 = 1$  Rest 99 - Prüfsumme = 100. Die vollständige Bytefolge der Barcodes lautet dann: A = 100, B = 64, C1 = 167, C2 = 71, D1 = 1 und D2 = 52.

#### 8.5 Herstellen von 'LIST' - Barcodes

Der Aufbau dieser drei Programme entspricht dem von 'DEL..' oder 'GTOZ..' -BC.

##### 8.5.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "LISTBC"

Programmbedienung:

-SIZE: würde mit 001 auskommen.

XEQ "LISTBC"



- "LIST \_\_\_?": Sie können nun die Anzahl der zu listenden Programmzeilen eingeben. Das Programm muß auf die Zeile positioniert sein, ab der gelistet werden soll.

Geben Sie Null ein, wird neu gestartet; geben Sie keine Zahl ein, wird die Zahl genommen, welche zufällig im X-Register steht. Ist die Anzahl >1999 erfolgt der Hinweis: "LIST <= 1999" - anschließend startet das Programm selbsttätig neu.

01*LBL "LISTBC"	24 1999	47 RDN	70 "LIST <= 1999"	93*LBL 05
02 CF 12	25 X<>Y	48 XEQ 04	71 ACA	94 OUTA
03 CF 13	26 ACA	49 "0"	72 ADV	95 CLA
04 SF 28	27 ADV	50 XEQ 05	73 TONE 0	96 RTN
05 CF 29	28 ADV	51 " "	74 GTO 00	97*LBL 06
06 FIX 0	29 X>Y?	52 XEQ 05	75*LBL 02	98 127
07*LBL 00	30 GTO 01	53 "C"	76 RCL X	99 X<>Y
08 "LIST____?"	31 XEQ 02	54 XEQ 05	77 256	100 X<=Y?
09 TONE 7	32 STO 00	55 RDN	78 ST/ Z	101 GTO 07
10 PROMPT	33 302	56 RCL 00	79 MOD	102 -
11 X=0?	34 +	57 XEQ 03	80 X<>Y	103 1
12 GTO 00	35 X<>Y	58 RDN	81 INT	104 +
13 ABS	36 STO Z	59 XEQ 03	82 RTN	105 .
14 "LIST "	37 +	60 CLST	83*LBL 03	106 1
15 100	38 XEQ 02	61 CLA	84 SF 25	107 BLDSPEC
16 X>Y?	39 +	62 ADV	85 X=0?	108 X<>Y
17 "+0"	40 XEQ 02	63 ADV	86 ACCOL	109 BLDSPEC
18 X<>Y	41 +	64 BEEP	87 X=0?	110 RTN
19 10	42 SF 17	65 ADV	88 GTO 05	111*LBL 07
20 X>Y?	43 .	66 ADV	89*LBL 04	112 .
21 "+0"	44 ACCOL	67 RTN	90 XEQ 06	113 X<>Y
22 X<>Y	45 **	68 GTO 00	91 ARCL X	114 BLDSPEC
23 ARCL X	46 XEQ 05	69*LBL 01	92 RDN	115 END 223 Bytes

### 8.5.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "LISTBc"

Programmbedienung: und SIZE wie "LISTBC"

XEQ "LISTBc"



01*LBL "LISTBc"	20 X>Y?	39 +	58 RDN	77 256
02 CF 12	21 "+0"	40 XEQ 02	59 XEQ 03	78 ST/ Z
03 CF 13	22 X<>Y	41 +	60 CLST	79 MOD
04 SF 28	23 ARCL X	42 SF 17	61 CLA	80 X<>Y
05 CF 29	24 1999	43 .	62 ADV	81 INT
06 FIX 0	25 X<>Y	44 ACCOL	63 ADV	82 RTN
07*LBL 00	26 ACA	45 **	64 BEEP	83*LBL 03
08 "LIST____?"	27 ADV	46 XEQ 05	65 ADV	84 SF 25
09 TONE 7	28 ADV	47 RDN	66 ADV	85 X=0?
10 PROMPT	29 X>Y?	48 XEQ 04	67 RTN	86 ACCOL
11 X=0?	30 GTO 01	49 "0"	68 GTO 00	87 X=0?
12 GTO 00	31 XEQ 02	50 XEQ 05	69*LBL 01	88 GTO 05
13 ABS	32 STO 00	51 " "	70 "LIST <= 1999"	89*LBL 04
14 "LIST "	33 302	52 XEQ 05	71 ACA	90 XTOA
15 100	34 +	53 "C"	72 ADV	91*LBL 05
16 X>Y?	35 X<>Y	54 XEQ 05	73 TONE 0	92 OUTA
17 "+0"	36 STO Z	55 RDN	74 GTO 00	93 CLA
18 X<>Y	37 +	56 RCL 00	75*LBL 02	94 END
19 10	38 XEQ 02	57 XEQ 03	76 RCL X	193 Bytes

8.5.3 Rechner HP 41 C - XF- und Plottermodul PRGM "LISTbc"

Programmbedienung: wie "LISTBC" - SIZE: 000

XEQ "LISTbc"



01*LBL "LISTbc"	13 ABS	25 X<Y	37 XTOA	49 GTO 00
02 CF 12	14 "LIST "	26 ACA	38 X<Y	50*LBL 01
03 CF 13	15 100	27 ADV	39 XTOA	51 "LIST <= 1999"
04 SF 28	16 X<Y?	28 X<Y?	40 -6	52 ACA
05 CF 29	17 "I-0"	29 GTO 01	41 BCKSM	53 ADV
06 FIX 0	18 X<Y	30 RCL X	42 BCO	54 TONE 0
07*LBL 00	19 10	31 256	43 CLST	55 GTO 00
08 "LIST____?"	20 X<Y?	32 ST/ Z	44 CLA	56 END
09 TONE 7	21 "I-0"	33 MOD	45 ADV	129 Bytes
10 PROMPT	22 X<Y	34 X<Y	46 BEEP	
11 X=0?	23 ARCL X	35 INT	47 ADV	
12 GTO 00	24 1999	36 "0 G"	48 RTN	

Hier zunächst Listing "PRPBC". Die Funktion 'PRP' wird ab Seite 196 behandelt.

01*LBL "PRPBC"	37 SF 01	73*LBL 03	109 "0"	145*LBL 09
02*LBL 00	38 X=0?	74 RCL 00	110 XEQ 09	146 OUTA
03 CF 12	39 GTO 05	75 INT	111 " "	147 CLA
04 CF 13	40*LBL 01	76 RCL IND X	112 XEQ 09	148 RDN
05 FIX 0	41 STO IND 00	77 DSE Y	113 "M"	149 RTN
06 CLRG	42 XEQ 02	78 GTO 04	114 XEQ 09	150*LBL 10
07 CF 23	43 X=0?	79 GTO 05	115 FS? 01	151 127
08 "PRGM-NAME?"	44 GTO 03	80*LBL 04	116 DSE 00	152 X<Y
09 TONE 7	45 ISG 00	81 RCL IND Y	117 "+"	153 X<Y?
10 AON	46 GTO 01	82 +	118 RCL 00	154 GTO 11
11 PROMPT	47*LBL 02	83 DSE Y	119 INT	155 -
12 AOFF	48 "I-0****"	84 GTO 04	120 SF 25	156 1
13 FC? 23	49 RCL [	85*LBL 05	121 X=0?	157 1
14 CLA	50 STO [	86 300	122 ACCOL	158 .
15 FC? 23	51 CLX	87 +	123 FS?C 01	159 1
16 SF 01	52 X< ]	88 SF 00	124 GTO 00	160 BLDSPEC
17 RCL \	53 SIGN	89 XEQ 00	125*LBL 06	161 X<Y
18 X=0?	54 CLX	90*LBL 00	126 XEQ 07	162 BLDSPEC
19 GTO 12	55 X< \	91 RCL X	127 DSE X	163 RTN
20 ASTO 00	56 "I****"	92 256	128 GTO 06	164*LBL 11
21 ASHF	57 X< [	93 ST/ Z	129*LBL 00	165 .
22 ASTO 01	58 X< L	94 MOD	130 ADV	166 X<Y
23 "PRP "	59 X< \	95 +	131 ADV	167 BLDSPEC
24 ARCL 00	60 INT	96 INT	132 CF 17	168 RTN
25 ARCL 01	61 ST+ ]	97 FS?C 00	133 CLST	169*LBL 12
26 "I-0"	62 RDN	98 RTN	134 BEEP	170 ACA
27 ACA	63 6	99 SF 17	135 ADV	171 ADV
28 ADV	64 ST* ]	100 .	136 ADV	172 "Name zu lang"
29 ADV	65 RDN	101 ACCOL	137 RTN	173 ACA
30 CLA	66 EI	102 RDN	138 GTO 00	174 ADV
31 ARCL 00	67 ST* L	103 131	139*LBL 07	175 TONE 0
32 ARCL 01	68 X< L	104 RCL 00	140 RCL IND X	176 GTO 00
33 1.007	69 ST+ ]	105 INT	141*LBL 08	177 END
34 STO 00	70 CLX	106 +	142 XEQ 10	335 Bytes
35 XEQ 02	71 X< ]	107 XEQ 00	143 ARCL X	
36 X=0?	72 RTN	108 XEQ 00	144 RDN	

## 9. Die Funktion 'PRP' - Ausdruck von Programmlistings

Mit den nachfolgenden Programmen werden 'PRP'-Barcodes zum Ausdruck bestimmter PRGM-Listings hergestellt. Die IL-Modul-Funktion erfordert ein Argument von 1-7 ASCII-Zeichen. Wie bei 'CLP' und 'ASN' schon erwähnt, können auch hier die verwegensten ASCII-Kombinationen, die sonst nicht tastbar sind, eingegeben werden. Im Abschnitt d. über das Arbeiten mit Barcodes aus diesem Kapitel finden Sie dann Beispiele.

### 9.1 Aufbau, Typindikator, Identifikations-Code und Prüfsumme

Der Aufbau entspricht prinzipiell dem von "LIST.."-BC mit folgenden Änderungen: C2 = 77; D = 1 bis 7 ASCII-Zeichen - der Programm-Name. Die Prüfsumme ermittelt sich wie bekannt, z.B. für 'PRP "LISTBC"':  $308 (B:64+C1:167+C2:77) + 76 + 73 + 83 + 84 + 66 + 67 = 757:256 = 2 \text{ Rest } 245 - \text{Prüfsumme} = 247.$

### 9.2 Herstellen von 'PRP'-Barcodes

Die drei Programme sind ähnlich wie 'LIST..'- oder auch 'CLP..'-BC.

#### 9.2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "PRPBC"

##### Programmbedienung:

-SIZE: es ist 008 erforderlich.

XEQ "PRPBC"



- "PRGM-NAME?": Geben Sie hier den Namen des auszudruckenden Programms ein. Geben Sie nichts ein und drücken gleich 'R/S' wird 'PRP' erstellt. Geben Sie mehr als 7 ASCII-Zeichen ein - Fehlermeldung: "Name zu lang". Die Funktion 'PRP' druckt das Programm aus, auf das der Rechner gerade positioniert ist.

01*LBL "PRPbc"	28 ADV	55 RCL IND X	82 131	109 ADV
02*LBL 00	29 ADV	56 DSE Y	83 RCL 00	110 ADV
03 CF 12	30 CLA	57 GTO 04	84 INT	111 CF 17
04 CF 13	31 ARCL 00	58 GTO 05	85 +	112 CLST
05 FIX 0	32 ARCL 01	59*LBL 04	86 XEQ 00	113 BEEP
06 CLRG	33 1.007	60 RCL IND Y	87 XEQ 08	114 ADV
07 CF 23	34 STO 00	61 +	88 "0"	115 ADV
08 "PRGM-NAME?"	35 XEQ 02	62 DSE Y	89 XEQ 09	116 RTN
09 TONE 7	36 X=0?	63 GTO 04	90 "	117 GTO 00
10 RDN	37 SF 01	64*LBL 05	91 XEQ 09	118*LBL 07
11 PROMPT	38 X=0?	65 308	92 "M"	119 RCL IND X
12 ROFF	39 GTO 05	66 +	93 XEQ 09	120*LBL 08
13 FC? 23	40*LBL 01	67 SF 00	94 FS? 01	121 XTOA
14 CLA	41 STO IND 00	68 XEQ 00	95 DSE 00	122*LBL 09
15 FC? 23	42 XEQ 02	69*LBL 00	96 "+"	123 OUTA
16 SF 01	43 X=0?	70 RCL X	97 RCL 00	124 CLA
17 RCL \	44 GTO 03	71 256	98 INT	125 RDN
18 X=0?	45 ISG 00	72 ST/ Z	99 SF 25	126 RTN
19 GTO 10	46 GTO 01	73 MOD	100 X=0?	127*LBL 10
20 ASTO 00	47*LBL 02	74 +	101 ACCOL	128 ACA
21 ASHF	48 -1	75 INT	102 FS?C 01	129 ADV
22 ASTO 01	49 AROT	76 FS?C 00	103 GTO 00	130 "Name zu lang"
23 "PRP "	50 ATOX	77 RTN	104*LBL 06	131 ACA
24 ARCL 00	51 RTN	78 SF 17	105 XEQ 07	132 ADV
25 ARCL 01	52*LBL 03	79 .	106 DSE X	133 TONE 0
26 "f"	53 RCL 00	80 ACCOL	107 GTO 06	134 GTO 00
27 ACA	54 INT	81 RDN	108*LBL 00	135 END 262 Bytes

9.2.2. Rechner HP 41 C - Drucker, XF-Modul PRGM "PRPbc"

Programmbedienung: und SIZE wie "PRPBC"

XEQ "PRPbc"



Das Listing zu diesem Programm finden Sie auf Seite 196. Zeile 90 = Byte 167.

9.2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "PRPbc"

Programmbedienung: wie "PRPBC" - SIZE: kommt mit 002 aus.

XEQ "PRPbc"



01+LBL "PRPbc"	12 STOP	23 ARCL 00	34 1	45 GTO 00
02+LBL 00	13 AOFF	24 ARCL 01	35 +	46+LBL 01
03 CF 12	14 FC? 23	25 "+"	36 CHS	47 ACA
04 CF 13	15 CLA	26 ACA	37 BCKSM	48 ADV
05 FIX 0	16 RCL \	27 ADV	38 BCO	49 "Name zu lans"
06 CLRG	17 X+0?	28 "0 M"	39 CLA	50 ACA
07 CLST	18 GTO 01	29 ARCL 00	40 CLST	51 ADV
08 CF 23	19 ASTO 00	30 ARCL 01	41 ADV	52 TONE 0
09 "PRGM-NAME?"	20 ASHF	31 FC? 23	42 BEEP	53 GTO 00
10 TONE 7	21 ASTO 01	32 XTOA	43 ADV	54 END
11 ADN	22 "PRP ""	33 ALENG	44 RTN	121 Bytes

10. Die Funktion 'NEWM' - Initialisierung einer Kassette oder Diskette

'NEWM' ist die letzte nicht programmierbare Funktion mit numerischem Argument.

10.1. Aufbau, Typindikator, Identifikation und Anzahl der Einträge

Bis auf die 'XEQ/XROM'-Identifikation, die hier 167 03 ist, und die Anzahl der Einträge mit maximal 447, sind alle übrigen Merkmale gleich wie 'LIST..' -BC.

10.2. Herstellen von 'NEWM'-Barcodes

Wiederum kann zwischen drei Programmen gewählt werden. Listings auf Seite 198.

10.2.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "NEWMBC"

Programmbedienung:

-SIZE: 001 ist ausreichend.

XEQ "NEWMBC"



- "NEWM \_\_\_?": Sie können nun die Anzahl Einträge zwischen 001 bis 447 eingeben. 'R/S' - Bei Eingabe 000 wird neu gefragt, bei mehr als 447 Einträgen erscheint: "NEWM <= 447" und das Programm startet neu mit der Frage: "NEWM \_\_\_?".

10.2.2 Rechner HP 41 C - Drucker, XF-Modul PRGM "NEWMbc"

Programmbedienung: und SIZE wie "NEWMBC".

XEQ "NEWMbc"



Die Kurzdarstellung des Programms "NEWMbc" mit XF- und PM folgt auf Seite 199.

01*LBL "NEWMBC"	24 447	47 "0"	70 ADV	93 CLA
02 CF 12	25 X<>Y	48 XEQ 05	71 TONE 0	94 RTN
03 CF 13	26 ACA	49 "	72 GTO 00	95*LBL 06
04 SF 28	27 ADV	50 XEQ 05	73*LBL 02	96 127
05 CF 29	28 ADV	51 "+"	74 RCL X	97 X<>Y
06 FIX 0	29 XYY?	52 XEQ 05	75 256	98 X<=Y?
07*LBL 00	30 GTO 01	53 RDN	76 ST/ Z	99 GTO 07
08 "NEWM____?"	31 XEQ 02	54 RCL 00	77 MOD	100 -
09 TONE 7	32 STO 00	55 XEQ 03	78 X<>Y	101 1
10 PROMPT	33 234	56 RDN	79 INT	102 +
11 X=0?	34 +	57 XEQ 03	80 RTN	103 .
12 GTO 00	35 X<>Y	58 CLST	81*LBL 03	104 1
13 ABS	36 STO Z	59 CLA	82 SF 25	105 BLDSPC
14 "NEWM "	37 +	60 ADV	83 X=0?	106 X<>Y
15 100	38 XEQ 02	61 ADV	84 ACCOL	107 BLDSPC
16 XYY?	39 +	62 BEEP	85 X=0?	108 RTN
17 "f0"	40 SF 17	63 ADV	86 GTO 05	109*LBL 07
18 X<>Y	41 .	64 ADV	87*LBL 04	110 .
19 10	42 ACCOL	65 RTN	88 XEQ 06	111 X<>Y
20 XYY?	43 "	66 GTO 00	89 ARCL X	112 BLDSPC
21 "f0"	44 XEQ 05	67*LBL 01	90 RDN	113 END
22 X<>Y	45 RDN	68 "NEWM <= 447"	91*LBL 05	216 Bytes
23 ARCL X	46 XEQ 04	69 ACA	92 OUTA	

01*LBL "NEWMbc"	20 XYY?	39 +	58 CLST	77 MOD
02 CF 12	21 "f0"	40 SF 17	59 CLA	78 X<>Y
03 CF 13	22 X<>Y	41 .	60 ADV	79 INT
04 SF 28	23 ARCL X	42 ACCOL	61 ADV	80 RTN
05 CF 29	24 447	43 "	62 BEEP	81*LBL 03
06 FIX 0	25 X<>Y	44 XEQ 05	63 ADV	82 SF 25
07*LBL 00	26 ACA	45 RDN	64 ADV	83 X=0?
08 "NEWM____?"	27 ADV	46 XEQ 04	65 RTN	84 ACCOL
09 TONE 7	28 ADV	47 "0"	66 GTO 00	85 X=0?
10 PROMPT	29 XYY?	48 XEQ 05	67*LBL 01	86 GTO 05
11 X=0?	30 GTO 01	49 "	68 "NEWM <= 447"	87*LBL 04
12 GTO 00	31 XEQ 02	50 XEQ 05	69 ACA	88 XTOA
13 ABS	32 STO 00	51 "+"	70 ADV	89*LBL 05
14 "NEWM "	33 234	52 XEQ 05	71 TONE 0	90 OUTA
15 100	34 +	53 RDN	72 GTO 00	91 CLA
16 XYY?	35 X<>Y	54 RCL 00	73*LBL 02	92 END
17 "f0"	36 STO Z	55 XEQ 03	74 RCL X	186 Bytes
18 X<>Y	37 +	56 RDN	75 256	
19 10	38 XEQ 02	57 XEQ 03	76 ST/ Z	

01*LBL "NEWMbc"	13 ABS	25 X<>Y	37 XTOA	49 GTO 00
02 CF 12	14 "NEWM "	26 ACA	38 X<>Y	50*LBL 01
03 CF 13	15 100	27 ADV	39 XTOA	51 "NEWM <= 447"
04 SF 28	16 XYY?	28 XYY?	40 -6	52 ACA
05 CF 29	17 "f0"	29 GTO 01	41 BCKSM	53 ADV
06 FIX 0	18 X<>Y	30 RCL X	42 BCO	54 TONE 0
07*LBL 00	19 10	31 256	43 CLST	55 GTO 00
08 "NEWM____?"	20 XYY?	32 ST/ Z	44 CLA	56 END
09 TONE 7	21 "f0"	33 MOD	45 ADV	126 Bytes
10 PROMPT	22 X<>Y	34 X<>Y	46 BEEP	
11 X=0?	23 ARCL X	35 INT	47 ADV	
12 GTO 00	24 447	36 "0"	48 RTN	

10.2.3 Rechner HP 41 C - Drucker, XF- und Plottermodul PRGM "NEWMbc"

Programmbedienung: wie "NEWMBC" - SIZE: könnte 000 sein.

XEQ "NEWMbc"



11. Die Funktion 'PLNG ' - Feststellen einer PRGM-Länge (CCD-Modul)

Bei dieser nicht programmierbaren Funktion mit einem aus 1 bis 7 ASCII-Zeichen bestehenden Argument handelt es sich um einen Befehl aus dem CCD-Modul, welches von Mitgleidern des Computer Club Deutschland entwickelt wurde. Wer hierzu Barcodes benötigt, kann das Programm "PRPBC" sehr leicht entsprechend abwandeln: Zeile 01 = 'LBL "PLNGBC"'; Zeile 23 = "PLNG"; Zeile 86 = 446; Zeile 111 = 162 ; Zeile 113 = 220 . Die anderen Programme können entsprechend modifiziert werden. Die XROM-Zahl dieser Funktion ist 11,28 - daraus leiten sich die XEQ-Bytes 162 220 ab. Die Eingaben können mit BC aus Kapitel 2 erfolgen.

Programmbedienung: und SIZE dann wie "PRPBC":

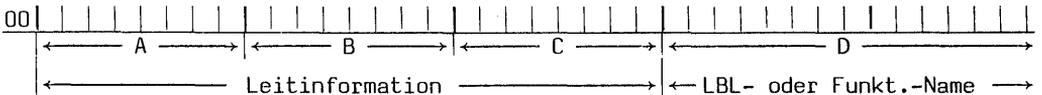
XEQ "PLNGBC"



c. Barcodes von 'GTO/XEQ/LBL "ALPHA"' - zur Direktausführung

Diese Barcodes erstellen globale Marken, z.B. 'LBL "GXLABC"', aber auch exotische Varianten wie 'LBL\*+\*+\*+\*' deren 1 bis 7 ASCII-Zeichen beliebige Kombinationen (außer 00) aus der Bytetabelle sein können. Wer solche Marken herstellen wollte, konnte das nur mit synthetischer Programmierung und bei Einsatz eines "LB"-Programms bewerkstelligen. Mit Barcodes ist auch dies viel einfacher. Die 'GTO'- und 'XEQ'-Barcodes verzweigen zu den 'ALPHA'-LBL bzw. führen sie aus. Die 'XEQ'-Barcodes führen auch sämtliche programmierbaren Funktionen aus Peripheriegeräten aus, sofern diese kein Argument erfordern. Rechnerbefehle, die Tastenfeld oder über die Tasten 'XEQ " ALPHA"' ausgeführt werden, verweigern diese Barcodes; Beispiel: Tastenfeld 'XEQ "ALPHA"%' wird ausgeführt, BC nicht.

1. Aufbau



A = 8-Bit Prüfsumme

B = 1 Byte Typindikator

C = 1 Byte Identifikations-Code

D = 1 bis 7 ASCII-Zeichen LBL- oder Funktions-Name

2. Der Typindikator mit Identifikations-Code

Diese Daten teilen dem Rechner mit: B = 64 - ein XEQ-BC ist auszuführen; C = hier gibt es drei Möglichkeiten: C = 29 für 'GTO "ALPHA"' - C = 30 für 'XEQ "ALPHA"' und C = 205 für 'LBL "ALPHA"'. Genaugenommen ist C in keinem dieser Fälle ein Identifikations-Code, sondern eine Art Vorsilbe für den nachfolgenden Alphastring. Weil die entsprechenden Funktionen jedoch diese Bytes gleichzeitig identifizieren, wollen wir die Bezeichnung Identifikations-Code gelten lassen. Außerdem ist es ja noch so, daß die Barcodes ohne genaue Definition des Codes einwandfrei funktionieren. Das Ganze ist wohl eine Ungereimtheit der Bytestabelle, die ja verschiedene solcher Verworrenheiten erkennen läßt.

### 3. Die 8-Bit Prüfsumme

Wieder wird die Prüfsumme aus der Addition der einzelnen Bytedezimalwerte B bis D berechnet. Beispiel: 'LBL "Wunsch" = 64 + 205 + 87 + 117 + 110 + 115 + 99 + 104 = 901:256 = 3 Rest 133 - Prüfsumme: 136.

### 4. Herstellen von 'GTO/XEQ/LBL "ALPHA" - Barcodes

Die drei Programme verdeutlichen noch einmal die Leistungsfähigkeit des XF- und des Plottermoduls. Das Programm "GXLABc" mit beiden Modulen benötigt fast nur die Hälfte der Byteanzahl gegenüber dem Programm "GXLABC" ohne Module.

#### 4.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "GXLABC"

##### Programmbedienung:

-SIZE: es ist 008 erwünscht.

XEQ "GXLABC"



- " G X L ?": Wählen Sie hier erst durch Eingabe von G = 'GTO' - X = 'XEQ' oder L = 'LBL' die gewünschte 'Vorsilbenfunktion' - 'R/S'
- "FUNKTS-NAME?": Geben Sie jetzt die ASCII-Folge der Marke oder Funktion ein. Geben Sie jeweils nichts ein, werden die Fragen wiederholt. Ist der Funktions-Name länger als 7 ASCII-Zeichen: "Name zu lang". HEX 00 Bytes sind unzulässig.

01*LBL "GXLABC"	34 AON	67*LBL 02	100*LBL 04	133 ACCOL
02*LBL 00	35 PROMPT	68 STO IND 00	101 RCL 00	134 RDN
03 "α)*"	36 AOFF	69 XEQ 03	102 INT	135 130
04 RCL [	37 FC? 23	70 X=0?	103 RCL IND X	136 RCL 00
05 STO d	38 GTO 01	71 GTO 04	104 DSE Y	137 INT
06 CLRG	39 RCL \	72 ISG 00	105 GTO 05	138 +
07 " G X L ?"	40 X*0?	73 GTO 02	106 GTO 06	139 XEQ 09
08 TONE 7	41 GTO 13	74*LBL 03	107*LBL 05	140 XEQ 09
09 PROMPT	42 ASTO 00	75 "†+↓*****"	108 RCL IND Y	141 "0"
10 AOFF	43 ASHF	76 RCL [	109 +	142 XEQ 10
11 FC?C 23	44 ASTO 01	77 STO [	110 DSE Y	143 FS? 00
12 GTO 00	45 FS? 00	78 CLX	111 GTO 05	144 "*"
13 71	46 *GTO ""	79 X<> ]	112*LBL 06	145 FS? 01
14 XEQ 03	47 FS? 01	80 SIGH	113 FS? 00	146 "€"
15 X=Y?	48 *XEQ ""	81 CLX	114 93	147 FS? 02
16 SF 00	49 FS? 02	82 X<> \	115 FS? 01	148 ""
17 00	50 "LBL ""	83 "†****"	116 94	149 XEQ 10
18 X=Y?	51 ARCL 00	84 X<> [	117 FS? 02	150 RCL 00
19 SF 01	52 ARCL 01	85 X<> L	118 269	151 INT
20 RDN	53 "†""	86 X<> \	119 +	152*LBL 07
21 76	54 ACA	87 INT	120 SF 03	153 XEQ 00
22 X=Y?	55 ADV	88 ST+ ]	121 XEQ 06	154 DSE X
23 SF 02	56 ADV	89 RDN	122*LBL 06	155 GTO 07
24 FS? 00	57 CLA	90 6	123 RCL X	156 ADV
25 GTO 01	58 ARCL 00	91 ST* ]	124 256	157 ADV
26 FS? 01	59 ARCL 01	92 RDN	125 ST/ Z	158 CF 17
27 GTO 01	60 1.007	93 E1	126 MOD	159 CLST
28 FS? 02	61 STO 00	94 ST* L	127 +	160 BEEP
29 GTO 01	62 XEQ 03	95 X<> L	128 INT	161 ADV
30 GTO 00	63 X=0?	96 ST+ ]	129 FS?C 03	162 ADV
31*LBL 01	64 STO 00	97 CLX	130 RTN	163 ADV
32 *FUNKTS-NAME?"	65 X=0?	98 X<> ]	131 SF 17	164 RTN
33 TONE 7	66 GTO 06	99 RTN	132 .	165 GTO 00

166*LBL 00	174 CLA	182 -	190 RTN	198 ADV
167 RCL IND X	175 RDN	183 1	191*LBL 12	199 "Name zu lang"
168*LBL 09	176 RTN	184 +	192 .	200 ACA
169 XEQ 11	177*LBL 11	185 .	193 X<>Y	201 ADV
170 ARCL X	178 127	186 1	194 BLDSPEC	202 TONE 0
171 RDN	179 X<>Y	187 BLDSPEC	195 RTN	203 GTO 00
172*LBL 10	180 X<=Y?	188 X<>Y	196*LBL 13	204 END
173 OUTA	181 GTO 12	189 BLDSPEC	197 ACA	405 Bytes

#### 4.2 Rechner HP 41 CV - Drucker, XF-Modul PRGM "GXLABc"

##### Programmbedienung:

-SIZE: und Bedienung wie "GXLABC"

XEQ "GXLABc"



Für die Identifikation um welche Funktion - 'GTO', 'XEQ' oder 'LBL' - es sich handelt, werden in allen drei Programmen die Flags 00, 01 und 02 benutzt. Vorher wird der Dezimalwert des ASCII-Zeichens verglichen. - In den Zeilen 123 bis 128 werden dann die zugehörigen Bytes 29, 30 bzw. 205 eingesetzt.

01*LBL "GXLABc"	34 AON	67*LBL 02	100 SF 03	133 XEQ 08
02*LBL 00	35 PROMPT	68 STO IND 00	101 XEQ 06	134 DSE X
03 "a)+"	36 AOFF	69 XEQ 03	102*LBL 06	135 GTO 07
04 RCL I	37 FC? 23	70 X=0?	103 RCL X	136 ADV
05 STO d	38 GTO 01	71 GTO 04	104 256	137 ADV
06 CLRG	39 RCL \	72 ISG 00	105 ST/ Z	138 CF 17
07 " G X L ?"	40 X=0?	73 GTO 02	106 MOD	139 CLST
08 TONE 7	41 GTO 11	74*LBL 03	107 +	140 CLA
09 PROMPT	42 ASTO 00	75 -1	108 INT	141 BEEP
10 AOFF	43 ASHF	76 AROT	109 FS?C 03	142 ADV
11 FC?C 23	44 ASTO 01	77 RDN	110 RTN	143 ADV
12 GTO 00	45 FS? 00	78 ATOX	111 SF 17	144 ADV
13 71	46 *GTO **	79 RTN	112 .	145 RTN
14 XEQ 03	47 FS? 01	80*LBL 04	113 ACCOL	146 GTO 00
15 X=Y?	48 *XEQ **	81 RCL 00	114 RDN	147*LBL 08
16 SF 00	49 FS? 02	82 INT	115 130	148 RCL IND X
17 88	50 *LBL **	83 RCL IND X	116 RCL 00	149*LBL 09
18 X=Y?	51 ARCL 00	84 DSE Y	117 INT	150 XTOR
19 SF 01	52 ARCL 01	85 GTO 05	118 +	151*LBL 10
20 RDN	53 *+*	86 GTO 06	119 XEQ 09	152 OUTA
21 76	54 ACA	87*LBL 05	120 XEQ 09	153 CLA
22 X=Y?	55 ADV	88 RCL IND Y	121 *0*	154 RDN
23 SF 02	56 ADV	89 +	122 XEQ 10	155 RTN
24 FS? 00	57 CLA	90 DSE Y	123 FS? 00	156*LBL 11
25 GTO 01	58 ARCL 00	91 GTO 05	124 *+*	157 ACA
26 FS? 01	59 ARCL 01	92*LBL 06	125 FS? 01	158 ADV
27 GTO 01	60 1.007	93 FS? 00	126 *E*	159 "Name zu lang"
28 FS? 02	61 STO 00	94 93	127 FS? 02	160 ACA
29 GTO 01	62 XEQ 03	95 FS? 01	128 **	161 ADV
30 GTO 00	63 X=0?	96 94	129 XEQ 10	162 TONE 0
31*LBL 01	64 STO 00	97 FS? 02	130 RCL 00	163 GTO 00
32 *FUNKTS-NAME?*	65 X=0?	98 269	131 INT	164 END
33 TONE 7	66 GTO 06	99 +	132*LBL 07	334 Bytes

4.3 Rechner HP 41 CV - Drucker, XF- und Plottermodul PRGM "GXLABc"

Programmbedienung: wie "GXLCB" -SIZE: kommt mit 002 aus.

XEQ "GXLABc"



01*LBL "GXLABc"	19 SF 01	37 FC? 23	55 ADV	73 CF 17
02*LBL 00	20 RDH	38 GTO 01	56 "0"	74 CLST
03 "c)*"	21 76	39 RCL \	57 FS? 00	75 BEEP
04 RCL I	22 X=Y?	40 X*0?	58 "I*"	76 CLA
05 STO d	23 SF 02	41 GTO 02	59 FS? 01	77 ADV
06 CLRG	24 FS? 00	42 ASTO 00	60 "I-E"	78 RTN
07 " G X L ?"	25 GTO 01	43 ASHF	61 FS? 02	79 GTO 00
08 TONE 7	26 FS? 01	44 ASTO 01	62 "I"	80*LBL 02
09 PROMPT	27 GTO 01	45 FS? 00	63 ARCL 00	81 ACA
10 ROFF	28 FS? 02	46 "GTO ""	64 ARCL 01	82 ADV
11 FC?C 23	29 GTO 01	47 FS? 01	65 ALENG	83 "Name zu lang"
12 GTO 00	30 GTO 00	48 "XEQ ""	66 I	84 ACA
13 71	31*LBL 01	49 FS? 02	67 +	85 ADV
14 ATOX	32 "FUNKTS-NAME?"	50 "LBL ""	68 BCKCKSH	86 TONE 0
15 X=Y?	33 TONE 7	51 ARCL 00	69 CHS	87 GTO 00
16 SF 00	34 AON	52 ARCL 01	70 BCO	88 END
17 88	35 PROMPT	53 "I""	71 ADV	206 Bytes
18 X=Y?	36 ROFF	54 ACA	72 ADV	

d. Arbeiten mit Mehrbytebarcodes

Ein wichtiger Vorteil der Ein-, Zwei- und Mehrbytebarcodes mit oder ohne Argument ist die Einsparung von Tastenzuweisungen. Solche beim Programmieren oder sonstigem Arbeiten öfter benutzte Funktionen, die viele Tastenbetätigungen bei Einzeleingabe erfordern, z.B. 'XEQ 'ALPHA' "BLDSPEC" 'ALPHA', sind einfach und schnell zu handhaben. Funktionen mit Argument wie 'RCL IND X' oder 'STO IND 00' können als Tastenbelegung ohnehin nur synthetisch mit Hilfe eines entsprechenden "KA"-Programms erzeugt werden. Barcodes sind hier eindeutig überlegen. Das synthetische Programmieren ist mit Barcodes und 'SNAP 2' keineswegs auf die in den Tabellen zusammengestellten Befehle beschränkt. Im Programm "ZFDABC" in Kapitel 3 ist ein 'LBL Y' als lokale Marke und ein 'GTO Y', wie unter II.e. beschrieben, verwendet. Angesichts der vielen möglichen Marken des Rechners sind solche LBL sicherlich nur extravagant. Anders ist dies hingegen bei den Kurzform-LBL 00 bis 14, die von einem synthetisch zum Langform gemachten GTO 00-14 über mehr als 112 Bytes angesprochen werden. Beim LBL wird ein Byte eingespart. Ebenso arbeiten zur Langform gemachte XEQ's einwandfrei.

Mit den Programmen "GXLA.." -BC können 'GTO/XEQ/LBL' von 1 bis 7 ASCII-Zeichen beliebiger Bytekombination hergestellt werden. Sofern eine solche Marke maximal 6 Zeichen hat, und diese ins Alpharegister gebracht werden können, ist es möglich, die Zeichenkette mit 'ASTO X' ins X-Register zu bringen und dann die Funktionen 'GTO IND X' oder 'XEQ IND X' auszuführen. Bei 7 Zeichen versagt diese Methode. Hier besteht dann totale Abhängigkeit von Barcodes oder man muß jedesmal 'CAT 1' bemühen. Synthetisch könnte man mit 'SNAP 2' ohne weiteres auch Marken in den Programmspeicher bringen, die bis 10 ASCII-Zeichen lang sind; doch werden solche LBL nicht ausgeführt, weil stets Register Q eingeschaltet wird und dort nun einmal nur 7 Bytes Platz haben. So müssen wir uns mit skurilen Marken wie 'LBL "\*\*\*\*\*"' oder 'GTO "\*\*\*\*\*"' bzw. 'XEQ "\*\*\*\*\*"' zufrieden geben. Es ist hier sieben mal das Byte 34 verwendet worden, das mit Hilfe der 'Zweibyte Funktions- und Druckerzeichen 0-127' eingebracht wurde.

e. Das Rätselprogramm "Wunsch"

In diesem Programm ist ein Wunsch des Autors verborgen, dessen Erfüllung Sie Ihr ganzes Leben begleiten möge. Mit Barcodes ist die Lösung einfach. Sie brauchen nur anzuwenden, was wir besprochen haben. Lesen Sie nun das Programm ein.

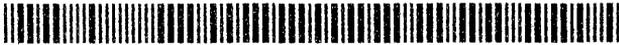
1. Nur Rechner HP 41 C - Drucker, keine Module PRGM "Wunsch"

Programmbedienung:

-SIZE: wir wünschen nur 002. Sodann lesen Sie nacheinander diese Barcodes ein:



XEQ "Wunsch"



Nach 'XEQ "Wunsch"' erhalten Sie den Ausdruck: Wunsch:  
Rätsel ungelöst

Des Rätsels Lösung besteht darin, eine einzige Zeile dieses Programms durch eine als Barcodes hergestellte andere Zeile zu ersetzen. Danach muß



ausgeführt werden. Haben Sie alles richtig gemacht, snappen Sie's mit Sicherheit. Das Programm hat dann ein Byte eingebüßt. Die erneute Ausführung von 'XEQ "Wunsch"' druckt die Lösung aus. Viel Spaß beim Enträtseln des Programms.

01+LBL "Wunsch"	13 SCI 5	25 RCL 01	37 ACA	49 FIX 2
02 "Wunsch:"	14 LOG	26 FC? 37	38 PRBUF	50+LBL 00
03 PRA	15 FRC	27 PRA	39 CLST	51 TONE 5
04 FIX 2	16 LNI+X	28 "Rätsel "	40 CF 21	52 TONE 2
05 CLST	17 HMS	29 ACA	41 "--ALLES KLAR--"	53 GTO 00
06 LNI+X	18 RCL 00	30 "un"	42 FS? 39	54+LBL 01
07 INT	19 X>Y?	31 FC? 40	43 CLA	55 TONE 0
08 PI	20 PI	32 ACA	44 AVIEW	56 TONE 7
09 FS? 38	21 X<0?	33 "gelöst"	45 SF 21	57 TONE ↑
10 STO 00	22 OCT	34 ACA	46 FS? 38	58 TONE 5
11 STO 01	23 HMS	35 " bravo!!"	47 GTO 01	59 GTO 01
12 FC? 36	24 X=0?	36 FS? 38	48 FC? 36	60 END 142 Bytes

f. Simuliertes synthetisches Tastenfeld mit PRGM "WDL" und SYNFKTION-Barcodes

Bei den ASN-Programmen stellten wir fest, synthetische Befehle können mit Barcodes nicht auf eine Taste gelegt werden. Barcodes von 'XEQ RCL M' werden ebenfalls nicht ausgeführt. Sind Barcodes denn nun bei der Direktausführung synthetischer Befehle im Hintertreffen? Keineswegs: zwar nicht ganz so elegant wie auf Tastendruck, es sind zwei kurze BC-Reihen abzutasten, dafür aber bei weitem umfangreicher ist auch die Sofortausführung synthetischer Befehle mit Barcodes möglich. Bei fast 80 synthetischen Befehlen müßten sämtliche Tastenbelegungsmöglichkeiten erschöpft werden, und es blieben noch Synthetiks, die nicht zur Verfügung stünden. Die Sache würde außerdem ca. 35 Tastenzuweisungsregister im Programmspeicher beanspruchen. Abgesehen von diesen Belegungen müßte ein umfangreiches Verzeichnis der USER-Funktionen erstellt werden. Dies alles entfällt bei der Anwendung von Barcodes aus den nachfolgenden Tabellen; sie sind alphabetisch geordnet, beanspruchen ungleich weniger Speicherplatz und stellen so gut wie alle gebräuchlichen synthetischen Befehle zur Verfügung. Darüberhinaus können mit den folgenden Programmen alle 'fehlenden' BC angefertigt werden.

1. Rechner HP 41 C - Drucker, keine Module PRGM "WDL"

Programmbedienung:

-SIZE: kann 000 sein. Die übrigen Zusammenhänge werden anschließend ausführlich besprochen. Lesen Sie nun 'XEQ "WDL"' ein.

XEQ "WDL"



- "SYNFUNKTION?": Frage nach der einzulesenden synthetischen Funktion. Lesen Sie nun die nachstehenden Barcodes ein:

TONE 26



Sie haben soeben nach dem üblichen kurzen BC-Eingabe-Piep den mit Barcodes ausgeführten synthetischen 'TONE 26' gehört, der im Programmspeicher als 'TONE 6' angezeigt wird, der jedoch tiefer und wesentlich länger ist als 'TONE 6'.

So einfach lassen sich alle in den nachfolgenden Tabellen aufgeführten synthetischen Befehle ausführen: 'XEQ "WDL"' (ist zu diesem Zweck auf jeder Seite unten rechts vorhanden) - Displayaufforderung: "SYNFUNKTION?" anschließend gewünschten synthetischen Befehl abtasten, der sofort ausgeführt wird. Doch große Vorsicht! Sollten Sie in den Künsten der synthetischen Programmierung noch nicht sehr bewandert sein, könnte Ihnen unsachgemäßes Experimentieren ein 'MEMORY LOST' oder gar einen GAU größeren Ausmaßes einbringen, der dann nur durch Entfernen der gesamten Periferie, aller Module und Herausnehmen der Stromversorgung für einige Minuten behoben werden könnte. Seien Sie besonders vorsichtig mit allen 'STO'-Befehlen und auch allen Austauschbefehlen 'X<>\_'. Über Ihre eventuellen Wissenslücken helfen die Bücher von W.C.Wickes oder auch K.Jarett über dies Gebiet der Programmierung hinweg.

Das Programm "WDL" sollte als letztes im Programmspeicher sein. Es benutzt die Lesestiftfunktion 'WNDLNK', welche mit "W: SCN SUBLNK" auffordert, ein Unterprogramm einzulesen. Im PRGM "WDL" wird diese Funktion mit 'AVIEW' durch die Aufforderung: "SYNFUNKTION?" überlagert. Erfolgt nun eine Unterprogrammeingabe, so wird zunächst das Programm "WDL" mit einem 'END' abgeschlossen, sodann wird das Unterprogramm unmittelbar an das PRGM "WDL" angefügt. Das Unterprogramm seinerseits schließt mit einem '.END.' ab. Nach dem Einlesen des

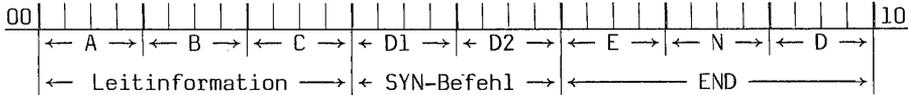
01*LBL "WDL"	Programms "WDL" und der ersten Eingabe eines Unterprogramms sieht
02 CF 21	das Programm mit Unterprogramm wie nebenstehend aufgelistet aus.
03 "SYNFUNKTION?"	Wird ein weiteres Mal 'XEQ "WDL"' ausgeführt, so unterdrückt
04 AVIEW	'CF 21' den Ausdruck von "SYNFUNKTION?" bei 'AVIEW', es tönen die
05 TONE 9	'TONE 9' und 'TONE 7' (tatsächlich 89 und 87), sodann bleibt das
06 TONE 7	Programm nach 'WNDLNK' auf 'SF 21' stehen, um auf die Eingabe des
07 WNDLNK	Unterprogramms zu warten. Geschieht dies, drängt das neue Unter-
08 SF 21	Programm das bestehende - hier 'TONE 6' aus dem Speicher, weil
09 END	dies nicht mit 'END' sondern mit dem offenen '.END.' abschließt.
01 TONE 6	Anschließend setzt der Rechner das Programm "WDL" mit dem
02 .END.	auf 'WNDLNK' folgenden Befehl - hier 'SF 21' fort und bleibt dann

nach 'END' stehen. Deswegen könnte für die nächste Eingabe auch mit 'R/S' gestartet werden. Sie sollten dennoch immer mit dem Lesestift und den BC 'XEQ "WDL"' starten, und zwar aus folgendem Grund: weil, wie sie inzwischen sicher selbst herausgefunden haben, diese synthetischen Befehle kleine Programme ohne 'LBL' sind, werden sie auch eingelesen, ohne daß vorher "WDL" - genauer 'WNDLNK' - ausgeführt wurde. In diesem Fall steht das Programm dann aber auf dem eingelesenen Befehl und nicht nach dem 'END' von "WDL". Drücken Sie dann 'R/S', wird der synthetische Befehl ausgeführt, was unbeabsichtigt die obengenannten 'schlimmen' Folgen haben kann. Start mit 'XEQ "WDL"' schließt solches aus. Der jeweils zuletzt ausgeführte 'Unterprogramm-Befehl' bleibt als Einzel-

programmzeile ohne 'LBL' nach dem 'END' von "WDL" stehen. Nach Beendigung der Arbeit können Sie alles mit "WDL" XEQ "PCLPS" löschen, sofern Sie ein XF-Modul benutzen. Andernfalls muß mit 'CLP\_\_' gelöscht werden. Um nun auch den zuletzt eingelesenen Befehl mit zu erfassen führen Sie aus: 'GTO "WDL"', Umschalten in den PRGM-Modus, 'BST', löschen Sie dann das 'END' von "WDL" mit der Korrekturtaste, der letzte Befehl gesellt sich zum PRGM "WDL", das nun mit 'CLP "WDL"' gelöscht werden kann.

Mit der Erläuterung des Aufbaus dieser Art synthetischer 'PRGM-Befehle' zur Direktausführung leiten wir eigentlich schon zum nächsten Kapitel über, es handelt sich hierbei ja tatsächlich um eine Art von Programmbarcodes ohne LBL.

## 2. Aufbau



A = 8-Bit Prüfsumme

B = Nybble 1 = 1 PRGM nicht PV- oder = 2 PRGM PV-geschützt  
Nybble 2 = Zeilennummer (MOD 16)

C = Nybble 1 = Anzahl Bytes noch auszuführende Restfunktion  
Nybble 2 = Anzahl Bytes schon ausgeführte Teilfunktion

D1 = Vorsilbe der synthetischen Funktion

D2 = Nachsilbe der synthetischen Funktion

END = die drei Bytes des '.END.'

## 3. Der Typindikator, die SYNFUNCTION und das 'END'

Nur bei diesem Typ kleiner Programmbarcodes wird dem Rechner mitgeteilt: B = 16 es gibt nur eine PRGM-Zeile nicht PV-geschützt. C = 0 es gibt keine unfertigen Funktionen. D1 = Dezimalwert Vorsilbe, z.B. 159 für 'TONE'; D2 = Dezimalwert Nachsilbe, z.B. 26. E = 192 - die Zusammenhänge mit dem 'END' werden im nächsten Kapitel ausführlich behandelt - N = 0 und D = 47.

## 4. Die 8-Bit Prüfsumme

Auch diese Prüfsumme wird nach dem gleichen Verfahren berechnet wie alle anderen Prüfsummen. Es werden zuerst sämtliche Bytedezimalwerte von B bis zum D des 'END' addiert, dann  $256 / \text{INT } 256 \text{ MOD} + -$  Ergebnis = Prüfsumme. Beispiel: 'TONE 26' =  $16 \text{ ENTER } 0 + 159 + 26 + 192 + 0 + 47 + \text{'RCL X' } 256 / \text{INT 'X<>Y' } 256 \text{ MOD} +$  Prüfsumme = 185

## 5. Herstellen von SYNFUNCTION-Barcodes

Es stehen in diesem Fall zwei verschiedene Programme zur Verfügung, eines nur für Rechner und Drucker, ohne Module und eines für Rechner, Plotter, XF- und das Plottermodul. Mit diesem PRGM können dann die Tabellen erstellt werden.

### 5.1 Nur Rechner HP 41 C - Drucker, keine Module PRGM "SSTFBC"

#### Programmbedienung:

-SIZE: 002 ist hier ausreichend

XEQ "SSTFBC"



Zuerst wird rechtsbündig ausgedruckt:  $\text{SinSynTasFld}$  . Dann folgt die Displayfrage:

- "SYNFUNKTION?": Tasten Sie nun die Buchstaben, Zeichen oder Zahlen Ihrer Funktion ein: z.B. TONE 26 - 'R/S'
- "PRE↑POST↑": Geben Sie jetzt den Dezimalwert der Vorsilbe (PREfix) ein, drücken 'ENTER', dann geben Sie dezimal die Nachsilbe (POSTfix) ein - 'R/S' Geben Sie jeweils nichts ein, kehrt das Programm zur Frage zurück. Weitere Prüfungen erfolgen im Interesse der Kürze nicht. HEX-00-Bytes sind unzulässig.

01*LBL "SSTFBC"	21 PROMPT	41 SF 17	61 XEQ 04	81 127
02*LBL 00	22 FC?C 22	42 .	62 ADV	82 X<>Y
03 "α)*"	23 GTO 01	43 ACCOL	63 ADV	83 X<=Y?
04 RCL [	24 ADV	44 ""	64 CF 17	84 GTO 06
05 X<> d	25 STO 01	45 XEQ 04	65 CLST	85 -
06 "SinSynTasFld"	26 X<>Y	46 XEQ 03	66 BEEP	86 1
07 ACA	27 STO 00	47 "8"	67 ADV	87 +
08 ADV	28 +	48 XEQ 04	68 ADV	88 .
09 "SYNFUNKTION?"	29 +	49 0	69 RTN	89 1
10 TONE 7	30 SF 00	50 SF 25	70 GTO 00	90 BLDSPEC
11 STOP	31 XEQ 02	51 ACCOL	71*LBL 03	91 X<>Y
12 ROFF	32*LBL 02	52 RCL 00	72 XEQ 05	92 BLDSPEC
13 FC? 23	33 RCL X	53 XEQ 03	73 ARCL X	93 RTN
14 GTO 00	34 256	54 RCL 01	74 RDH	94*LBL 06
15 ACA	35 ST/ Z	55 XEQ 03	75*LBL 04	95 .
16 ADV	36 MOD	56 "	76 OUTA	96 X<>Y
17 255	37 +	57 XEQ 04	77 CLA	97 BLDSPEC
18*LBL 01	38 INT	58 SF 25	78 RDH	98 END
19 "PRE↑POST↑"	39 FS?C 00	59 ACCOL	79 RTN	204 Bytes
20 TONE 7	40 RTN	60 "/"	80*LBL 05	

Wer einen Plotter sein eigen nennt, kann mit dem nachfolgenden Programm wiederum die Tabellen selbst in DIN A 4 anfertigen und seinen Vorstellungen gemäß abwandeln oder um diverse, bis zu 128 TONE ergänzen.

### 5.2 Rechner HP 41 CV - Plotter, Plottermodul PRGM "SSTPbc"

#### Programmbedienung:

-SIZE: es ist 010 erforderlich.

XEQ "SSTPbc"



- "PEN 1= .3 SWZ": fordert Sie auf, diesen Stift einzusetzen - 'R/S'
- "PEN 2= .4 SWZ": desgleichen - 'R/S'
- "PAPIER?": einlegen oder austauschen - 'R/S'
- "PEN 2= .5 SWZ":
- "PEN 2= .4 SWZ":
- "SYNFUNKTION?": hier geben Sie die Buchstaben-, Zeichen- oder Zahlenfolge der Funktion ein, z.B., RCL M - 'R/S'
- "PRE↑POST↑": geben Sie nun zuerst den Bytedezimalwert der Vorsilbe ein, 'ENTER' und dann den der Nachsilbe - 'R/S'

Nacheinander können so 33 synthetische 'Kurzprogramm'-Funktionen geplottet werden. Danach wird unten rechts auf jeder Seite 'XEQ "WDL"' als Barcode erstellt, damit jeweils mit dem Lesestift gestartet werden kann. Anschließend wird die fertige Seite 'ausgeworfen' und mit der Frage: "PAPIER?" der Beginn einer neuen Seite angefordert. Nach Einlegen des Blattes kann die Arbeit mit 'R/S' fortgesetzt werden. Genau wie bei den Barcodes zum synthetischen Programmieren mit 'SNAP 2' wird auch hier konzentriertes Arbeiten beim Herstellen der Tabellen erlaubt. Weitere Kontrollen zum Abfangen eventueller Eingabefehler sind deswegen im Interesse kurzer Programmlauf- und Arbeitszeit nicht vorgesehen.

01*LBL "SSTPbc"	36 18.03021	71 SF 17	106 MOVE	141 MOVE
02 AUTOID	37 12.331	72 *BARCODES FU+r S"	107 *PRE+POST+	142 *XEQ *WDL**
03 *HP7470A"	38 BCsize	73 *+IMULIERTE"	108 TONE 7	143 LABEL
04 FINDID	39 XEQ 03	74 LABEL	109 PROMPT	144 1
05 SELECT	40*LBL 00	75 *S SYNTHETISCHES"	110 STO 04	145 PEN
06 FIX 0	41 XEQ 04	76 *+ TASTENFE"	111 RDN	146 +
07 SF 28	42 20	77 LABEL	112 STO 03	147 MOVE
08 CF 29	43 STO 09	78 *LD"	113 XEQ 03	148 XEQ 05
09 CLRG	44 CF 00	79 LABEL	114 1.007	149 *F0EWDL"
10 16	45 15	80 CF 17	115 BCREGX	150 BC
11 STO 01	46 STO 11	81 .	116 8	151 6.00781
12 192	47 *PAPIER?"	82 PEN	117 BCKSM	152 STO 08
13 STO 05	48 TONE 7	83 *PEN 2= .4 SWZ"	118 BC	153 .
14 47	49 PROMPT	84 TONE 7	119 FS? 00	154 .
15 STO 07	50 2	85 PROMPT	120 GTO 02	155 PEN
16 6.00781	51 PEN	86 *CSSI.175,.275"	121 ISG 08	156 MOVE
17 STO 08	52 FRAME	87 OUTA	122 GTO 01	157 GTO 00
18 *PEN 1= .3 SWZ"	53 .	88*LBL 01	123 6.00781	158*LBL 03
19 TONE 7	54 13	89 XEQ 04	124 STO 08	159 .
20 PROMPT	55 MOVE	90 2	125 13.5	160 PDIR
21 *PEN 2= .4 SWZ"	56 167	91 PEN	126 ST+ 09	161 LORG
22 TONE 7	57 13	92 RCL 08	127 RCL 09	162 ACOS
23 PROMPT	58 DRAW	93 INT	128 236	163 LDIR
24 PINIT	59 *CS2SI.211,.35"	94 RCL 09	129 X*Y?	164 RTN
25 11	60 OUTA	95 MOVE	130 GTO 01	165*LBL 04
26 263	61 .	96 *SYNFUNKTION?"	131 SF 00	166 *VS19"
27 15	62 PEN	97 TONE 7	132 GTO 01	167 OUTA
28 182	63 *PEN 2= .5 SWZ"	98 AON	133*LBL 02	168 RTN
29 LIMIT	64 TONE 7	99 STOP	134 XEQ 04	169*LBL 05
30 .	65 PROMPT	100 AOFF	135 2	170 *VS"
31 252	66 2	101 LABEL	136 PEN	171 OUTA
32 .	67 PEN	102 XEQ 05	137 ISG 08	172 END
33 PEN	68 6	103 1	138 RCL 08	517 Bytes
34 167	69 9	104 PEN	139 INT	
35 SCALE	70 MOVE	105 +	140 RCL 09	

Ab Seite 208 finden Sie die Tabellen 'BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD'. Daran anschließend folgen die Barcodes aller Programme dieses Kapitels in der Reihenfolge der Besprechung in den einzelnen Abschnitten.

## BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD

"WDL"



GTO "WDL"



BST



<--



PCLPS



PRGM



CLP "WDL"



Mit diesen Barcodes können Sie das Programm "WDL" mit und ohne XF-M löschen

# BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD

ARCL a 155/123



ARCL d 155/126



ARCL N 155/118



ARCL P 155/120



ASTO c 154/125



ASTO M 154/117



ASTO O 154/119



DSE M 151/117



FC? IND M 173/245



FC? IND N 173/246



FC?C IND O 171/247



GTO IND M 174/117



ISG N 150/118



RCL b 144/124



RCL IND c 144/253



RCL e 144/127



RCL N 144/118



ARCL c 155/125



ARCL M 155/117



ARCL O 155/119



ASTO a 154/123



ASTO d 154/126



ASTO N 154/118



ASTO P 154/120



DSE N 151/118



FC?C IND M 171/245



FC?C IND N 171/246



FIX IND M 156/245



ISG M 150/117



RCL a 144/123



RCL c 144/125



RCL d 144/126



RCL M 144/117



XEQ "WDL"



# BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD

RCL O 144/119



RCL Q 144/121



SCI IND d 157/245



SF IND M 168/245



SF IND O 168/247



ST- O 147/119



ST+ d 146/126



ST+ O 146/119



ST/ M 149/117



ST/ O 149/119



STO b 145/124



STO d 145/126



STO IND c 145/253



STO N 145/118



STO P 145/120



STO F 145/122



TONE 3/13 159/13



RCL P 144/120



RCL F 144/122



SCI IND N 157/246



SF IND N 168/246



ST- a 147/123



ST+ a 146/123



ST+ M 146/117



ST\* O 148/119



ST/ N 149/118



STO a 145/123



STO c 145/125



STO e 145/127



STO M 145/117



STO O 145/119



STO Q 145/121



STO IND a 145/251



XEQ "WDL"



# BARCODES FÜR SIMULIERTES SYNTHETISCHES TASTENFELD

TONE 4/14 159/14



TONE 6/26 159/26



TONE 9/29 159/29



TONE 7/57 159/57



TONE 6/66 159/66



TONE 2/72 159/72



TONE 1/81 159/81



TONE 7/87 159/87



TONE 0/100 159/100



VIEW M 152/117



X<>b 206/124



X<>d 206/126



X<>M 206/117



X<>0 206/119



X<>Q 206/121



X<> IND c 206/253



X<> IND N 206/246



TONE 1/21 159/21



TONE 8/28 159/28



TONE 7/47 159/47



TONE 4/64 159/64



TONE 0/70 159/70



TONE 5/75 159/75



TONE 3/83 159/83



TONE 9/89 159/89



TONE 2/112 159/112



X<>a 206/123



X<>c 206/125



X<>e 206/127



X<>N 206/118



X<>P 206/120



X<>t 206/122



X<> IND M 206/245



XEQ "WDL"



Programm: "1BYTBC" - Benoetigte PRGM-Reg: 41

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-4



Reihe 3: Zeilen 4-9



Reihe 4: Zeilen 9-12



Reihe 5: Zeilen 13-21



Reihe 6: Zeilen 21-23



Reihe 7: Zeilen 23-29



Reihe 8: Zeilen 30-37



Reihe 9: Zeilen 37-44



Reihe 10: Zeilen 45-51



Reihe 11: Zeilen 51-57



Reihe 12: Zeilen 58-65



Reihe 13: Zeilen 66-71



Reihe 14: Zeilen 72-78



Reihe 15: Zeilen 78-84



Reihe 16: Zeilen 84-87



Reihe 17: Zeilen 88-95



Programm: "1BYTBC" - Fortsetzung

Reihe 18: Zeilen 95-104



Reihe 19: Zeilen 105-112



Reihe 20: Zeilen 112-120



Reihe 21: Zeilen 120-129



Reihe 22: Zeilen 130-137



g

Programm: "1BYTBc" - Benoetigte PRGM-Reg: 37

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-4



Reihe 3: Zeilen 4-9



Reihe 4: Zeilen 9-12



Reihe 5: Zeilen 13-21



Reihe 6: Zeilen 21-23



Reihe 7: Zeilen 23-29



Reihe 8: Zeilen 30-37



Reihe 9: Zeilen 37-44



Reihe 10: Zeilen 45-51



Reihe 11: Zeilen 51-57



Programm: "1BYTbc" - Fortsetzung

Reihe 12: Zeilen 58-65



Reihe 13: Zeilen 66-71



Reihe 14: Zeilen 72-78



Reihe 15: Zeilen 78-84



Reihe 16: Zeilen 84-87



Reihe 17: Zeilen 88-95



Reihe 18: Zeilen 95-104



Reihe 19: Zeilen 105-112



Reihe 20: Zeilen 112-117



Programm: "1BYTbc" - Benoetigte PRGM-Reg: 32

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-4



Reihe 3: Zeilen 5-10



Reihe 4: Zeilen 10-13



Reihe 5: Zeilen 13-19



Reihe 6: Zeilen 20-24



Reihe 7: Zeilen 24-29



Programm: "1BYTbc" - Fortsetzung

Reihe 8: Zeilen 30-37



Reihe 9: Zeilen 38-45



Reihe 10: Zeilen 45-53



Reihe 11: Zeilen 53-60



Reihe 12: Zeilen 61-68



Reihe 13: Zeilen 68-74



Reihe 14: Zeilen 74-81



Reihe 15: Zeilen 82-86



Reihe 16: Zeilen 86-92



Reihe 17: Zeilen 93-99



Programm: "2BYTBC" - Benoetigte PRGM-Reg: 35

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-12



Reihe 4: Zeilen 13-20



Reihe 5: Zeilen 21-21



Reihe 6: Zeilen 22-30



## Programm: "2BYTBC" - Fortsetzung

Reihe 7: Zeilen 30-37



Reihe 8: Zeilen 38-44



Reihe 9: Zeilen 45-51



Reihe 10: Zeilen 52-60



Reihe 11: Zeilen 61-67



Reihe 12: Zeilen 68-77



Reihe 13: Zeilen 77-83



Reihe 14: Zeilen 83-92



Reihe 15: Zeilen 93-95



Reihe 16: Zeilen 95-101



Reihe 17: Zeilen 102-110



Reihe 18: Zeilen 111-120



Reihe 19: Zeilen 121-127



## Programm: "2BYTBc" - Benoetigte PRGM-Reg: 31

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-12



Programm: "2BYTbc" - Fortsetzung

Reihe 4: Zeilen 13-20



Reihe 5: Zeilen 21-21



Reihe 6: Zeilen 22-30



Reihe 7: Zeilen 30-37



Reihe 8: Zeilen 38-44



Reihe 9: Zeilen 45-51



Reihe 10: Zeilen 52-60



Reihe 11: Zeilen 61-67



Reihe 12: Zeilen 68-77



Reihe 13: Zeilen 77-83



Reihe 14: Zeilen 83-92



Reihe 15: Zeilen 93-95



Reihe 16: Zeilen 95-102



Reihe 17: Zeilen 103-106



Programm: "2BYTbc" - Benoetigte PRGM-Reg: 25

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-9



Programm: "2BYTbc" - Fortsetzung

Reihe 3: Zeilen 9-12



Reihe 4: Zeilen 13-20



Reihe 5: Zeilen 21-21



Reihe 6: Zeilen 22-29



Reihe 7: Zeilen 30-37



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-51



Reihe 10: Zeilen 52-59



Reihe 11: Zeilen 60-69



Reihe 12: Zeilen 69-72



Reihe 13: Zeilen 72-77



Programm: "INDBC" - Benoetigte PRGM-Reg: 16

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 10-17



Reihe 4: Zeilen 17-21



Reihe 5: Zeilen 21-30



Programm: "INDBC" - Fortsetzung

Reihe 6: Zeilen 31-39



Reihe 7: Zeilen 40-49



Reihe 8: Zeilen 49-58



Reihe 9: Zeilen 58-58



Programm: "INDBc" - Benoetigte PRGM-Reg: 12

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 10-18



Reihe 4: Zeilen 18-22



Reihe 5: Zeilen 22-30



Reihe 6: Zeilen 31-39



Reihe 7: Zeilen 39-39



Programm: "INDBc" - Benoetigte PRGM-Reg: 7

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 10-17



Programm: "INDbc" - Fortsetzung

Reihe 4: Zeilen 18-20



Programm: "NPFBC" - Benoetigte PRGM-Reg: 30

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-13



Reihe 4: Zeilen 14-21



Reihe 5: Zeilen 21-22



Reihe 6: Zeilen 22-30



Reihe 7: Zeilen 31-38



Reihe 8: Zeilen 39-47



Reihe 9: Zeilen 48-56



Reihe 10: Zeilen 56-62



Reihe 11: Zeilen 63-71



Reihe 12: Zeilen 71-73



Reihe 13: Zeilen 74-82



Reihe 14: Zeilen 83-91



Reihe 15: Zeilen 92-100



Programm: "NPFBC" - Fortsetzung

Reihe 16: Zeilen 101-109



Reihe 17: Zeilen 109-109



Programm: "NPFBC" - Benoetigte PRGM-Reg: 26

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-13



Reihe 4: Zeilen 14-21



Reihe 5: Zeilen 21-22



Reihe 6: Zeilen 22-30



Reihe 7: Zeilen 31-38



Reihe 8: Zeilen 39-47



Reihe 9: Zeilen 48-56



Reihe 10: Zeilen 56-62



Reihe 11: Zeilen 63-71



Reihe 12: Zeilen 71-73



Reihe 13: Zeilen 74-82



Reihe 14: Zeilen 83-88



Programm: "NPFbc" - Benoetigte PRGM-Reg: 17

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-13



Reihe 4: Zeilen 14-21



Reihe 5: Zeilen 21-22



Reihe 6: Zeilen 22-30



Reihe 7: Zeilen 30-38



Reihe 8: Zeilen 39-42



Reihe 9: Zeilen 42-47



Reihe 10: Zeilen 47-47



Programm: "SNP2BC" - Benoetigte PRGM-Reg: 27

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-9



Reihe 4: Zeilen 9-14



Reihe 5: Zeilen 15-19



Programm: "SNP2BC" - Fortsetzung

Reihe 6: Zeilen 19-25



Reihe 7: Zeilen 26-34



Reihe 8: Zeilen 34-42



Reihe 9: Zeilen 43-48



Reihe 10: Zeilen 49-54



Reihe 11: Zeilen 55-64



Reihe 12: Zeilen 64-72



Reihe 13: Zeilen 73-82



Reihe 14: Zeilen 82-91



Reihe 15: Zeilen 91-91



Programm: "S2Pbc" - Benoetigte PRGM-Reg: 86

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-7



Reihe 3: Zeilen 7-14



Reihe 4: Zeilen 14-16



Reihe 5: Zeilen 16-19



Reihe 6: Zeilen 19-20



Programm: "S2Pbc" - Fortsetzung

Reihe 7: Zeilen 21-25



Reihe 8: Zeilen 25-29



Reihe 9: Zeilen 29-34



Reihe 10: Zeilen 34-36



Reihe 11: Zeilen 36-42



Reihe 12: Zeilen 42-45



Reihe 13: Zeilen 46-53



Reihe 14: Zeilen 53-56



Reihe 15: Zeilen 56-59



Reihe 16: Zeilen 60-60



Reihe 17: Zeilen 60-68



Reihe 18: Zeilen 68-69



Reihe 19: Zeilen 69-70



Reihe 20: Zeilen 70-72



Reihe 21: Zeilen 72-73



Reihe 22: Zeilen 73-76



Reihe 23: Zeilen 76-80



Reihe 24: Zeilen 80-83



Programm: "S2Pbc" - Fortsetzung

Reihe 25: Zeilen 83-84



Reihe 26: Zeilen 85-93



Reihe 27: Zeilen 93-94



Reihe 28: Zeilen 94-102



Reihe 29: Zeilen 103-105



Reihe 30: Zeilen 105-111



Reihe 31: Zeilen 112-118



Reihe 32: Zeilen 118-122



Reihe 33: Zeilen 122-129



Reihe 34: Zeilen 130-135



Reihe 35: Zeilen 136-141



Reihe 36: Zeilen 142-146



Reihe 37: Zeilen 146-152



Reihe 38: Zeilen 153-160



Reihe 39: Zeilen 161-168



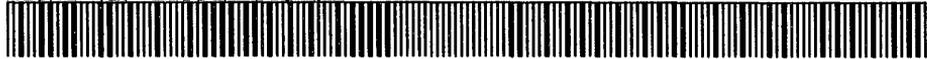
Reihe 40: Zeilen 168-174



Reihe 41: Zeilen 174-181



Reihe 42: Zeilen 181-187



Programm: "S2Pbc" - Fortsetzung

Reihe 43: Zeilen 188-193



Reihe 44: Zeilen 193-200



Reihe 45: Zeilen 201-207



Reihe 46: Zeilen 207-213



Programm: "XROMBC" - Benoetigte PRGM-Reg: 36

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-10



Reihe 4: Zeilen 11-16



Reihe 5: Zeilen 17-18



Reihe 6: Zeilen 19-28



Reihe 7: Zeilen 28-35



Reihe 8: Zeilen 36-44



Reihe 9: Zeilen 44-54



Reihe 10: Zeilen 55-64



Reihe 11: Zeilen 64-72



Reihe 12: Zeilen 73-80



Programm: "XROMBC" - Fortsetzung

Reihe 13: Zeilen 80-86



Reihe 14: Zeilen 87-94



Reihe 15: Zeilen 94-97



Reihe 16: Zeilen 98-106



Reihe 17: Zeilen 106-115



Reihe 18: Zeilen 115-124



Reihe 19: Zeilen 124-132



Programm: "XROMBc" - Benoetigte PRGM-Reg: 31

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-10



Reihe 4: Zeilen 11-16



Reihe 5: Zeilen 17-18



Reihe 6: Zeilen 19-28



Reihe 7: Zeilen 28-35



Reihe 8: Zeilen 36-44



Reihe 9: Zeilen 44-54



Programm: "XROMbc" - Fortsetzung

Reihe 10: Zeilen 55-64



Reihe 11: Zeilen 64-72



Reihe 12: Zeilen 73-80



Reihe 13: Zeilen 80-86



Reihe 14: Zeilen 87-94



Reihe 15: Zeilen 94-97



Reihe 16: Zeilen 98-106



Reihe 17: Zeilen 106-111



Programm: "XROMbc" - Benoetigte PRGM-Reg: 24

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-10



Reihe 4: Zeilen 11-16



Reihe 5: Zeilen 17-18



Reihe 6: Zeilen 19-28



Reihe 7: Zeilen 28-35



Reihe 8: Zeilen 36-44



Programm: "XROMbc" - Fortsetzung

Reihe 9: Zeilen 44-54



Reihe 10: Zeilen 55-63



Reihe 11: Zeilen 63-72



Reihe 12: Zeilen 73-74



Reihe 13: Zeilen 74-79



Programm: "XRMPbc" - Benoetigte PRGM-Reg: 73

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 7-11



Reihe 4: Zeilen 11-11



Reihe 5: Zeilen 12-14



Reihe 6: Zeilen 14-19



Reihe 7: Zeilen 19-24



Reihe 8: Zeilen 24-29



Reihe 9: Zeilen 29-30



Reihe 10: Zeilen 30-34



Reihe 11: Zeilen 35-40



Programm: "XRMPbc" - Fortsetzung

Reihe 12: Zeilen 40-46



Reihe 13: Zeilen 47-51



Reihe 14: Zeilen 52-53



Reihe 15: Zeilen 54-56



Reihe 16: Zeilen 56-62



Reihe 17: Zeilen 63-65



Reihe 18: Zeilen 65-66



Reihe 19: Zeilen 66-69



Reihe 20: Zeilen 69-70



Reihe 21: Zeilen 70-74



Reihe 22: Zeilen 75-78



Reihe 23: Zeilen 78-82



Reihe 24: Zeilen 82-90



Reihe 25: Zeilen 90-91



Reihe 26: Zeilen 92-94



Reihe 27: Zeilen 94-102



Reihe 28: Zeilen 103-103



Reihe 29: Zeilen 103-108



Programm: "XRMPbc" - Fortsetzung

Reihe 30: Zeilen 108-111



Reihe 31: Zeilen 111-118



Reihe 32: Zeilen 119-127



Reihe 33: Zeilen 128-138



Reihe 34: Zeilen 138-146



Reihe 35: Zeilen 147-152



Reihe 36: Zeilen 152-156



Reihe 37: Zeilen 156-163



Reihe 38: Zeilen 164-168



Reihe 39: Zeilen 168-175



Reihe 40: Zeilen 175-175



Programm: "XR-XEQ" - Benoetigte PRGM-Reg: 27

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-10



Reihe 4: Zeilen 11-17



Reihe 5: Zeilen 17-19



Programm: "XR-XEQ" - Fortsetzung

Reihe 6: Zeilen 19-26



Reihe 7: Zeilen 27-34



Reihe 8: Zeilen 34-42



Reihe 9: Zeilen 43-53



Reihe 10: Zeilen 54-62



Reihe 11: Zeilen 62-65



Reihe 12: Zeilen 66-72



Reihe 13: Zeilen 73-79



Reihe 14: Zeilen 79-83



Reihe 15: Zeilen 83-84

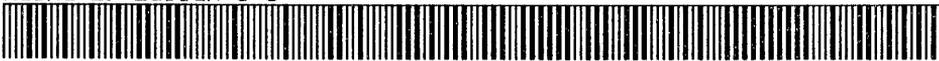


Programm: "XEQBC" - Benoetigte PRGM-Reg: 85

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-20



Reihe 6: Zeilen 21-29



Programm: "XEQBC" - Fortsetzung

Reihe 7: Zeilen 29-37



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-50



Reihe 10: Zeilen 51-57



Reihe 11: Zeilen 57-64



Reihe 12: Zeilen 65-71



Reihe 13: Zeilen 71-77



Reihe 14: Zeilen 78-85



Reihe 15: Zeilen 85-93



Reihe 16: Zeilen 93-102



Reihe 17: Zeilen 102-110



Reihe 18: Zeilen 111-117



Reihe 19: Zeilen 118-118



Reihe 20: Zeilen 119-125



Reihe 21: Zeilen 125-131



Reihe 22: Zeilen 131-138



Reihe 23: Zeilen 139-145



Reihe 24: Zeilen 145-151



# Programm: "XEQBC" - Fortsetzung

Reihe 25: Zeilen 152-158



Reihe 26: Zeilen 158-164



Reihe 27: Zeilen 164-165



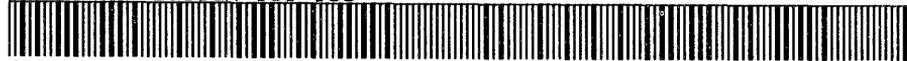
Reihe 28: Zeilen 166-174



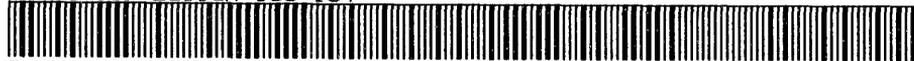
Reihe 29: Zeilen 174-181



Reihe 30: Zeilen 181-188



Reihe 31: Zeilen 188-194



Reihe 32: Zeilen 194-201



Reihe 33: Zeilen 202-209



Reihe 34: Zeilen 209-214



Reihe 35: Zeilen 215-223



Reihe 36: Zeilen 224-231



Reihe 37: Zeilen 232-239



Reihe 38: Zeilen 239-244



Reihe 39: Zeilen 245-251



Reihe 40: Zeilen 251-261



Reihe 41: Zeilen 262-262



Reihe 42: Zeilen 263-270



Programm: "XEQBC" - Fortsetzung

Reihe 43: Zeilen 271-278



Reihe 44: Zeilen 279-288



Reihe 45: Zeilen 289-298



Reihe 46: Zeilen 299-300



Programm: "XEQBc" - Benoetigte PRGM-Reg: 80

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-20



Reihe 6: Zeilen 21-29



Reihe 7: Zeilen 29-37



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-50



Reihe 10: Zeilen 51-57



Reihe 11: Zeilen 57-64



Reihe 12: Zeilen 65-71



Programm: "XEQBC" - Fortsetzung

Reihe 13: Zeilen 71-77



Reihe 14: Zeilen 78-85



Reihe 15: Zeilen 85-93



Reihe 16: Zeilen 93-102



Reihe 17: Zeilen 102-110



Reihe 18: Zeilen 111-117



Reihe 19: Zeilen 118-118



Reihe 20: Zeilen 119-125



Reihe 21: Zeilen 125-131



Reihe 22: Zeilen 131-138



Reihe 23: Zeilen 139-145



Reihe 24: Zeilen 145-151



Reihe 25: Zeilen 152-158



Reihe 26: Zeilen 158-164



Reihe 27: Zeilen 164-165



Reihe 28: Zeilen 166-174



Reihe 29: Zeilen 174-181



Reihe 30: Zeilen 181-188



Programm: "XEQBc" - Fortsetzung

Reihe 31: Zeilen 188-194



Reihe 32: Zeilen 194-201



Reihe 33: Zeilen 202-209



Reihe 34: Zeilen 209-214



Reihe 35: Zeilen 215-223



Reihe 36: Zeilen 224-231



Reihe 37: Zeilen 232-239



Reihe 38: Zeilen 239-244



Reihe 39: Zeilen 245-251



Reihe 40: Zeilen 252-261



Reihe 41: Zeilen 262-262



Reihe 42: Zeilen 263-270



Reihe 43: Zeilen 271-279



Reihe 44: Zeilen 279-279



Programm: "XEQbc" - Benoetigte PRGM-Reg: 71

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Programm: "XEQbc" - Fortsetzung

Reihe 3: Zeilen 9-11



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-20



Reihe 6: Zeilen 21-29



Reihe 7: Zeilen 29-37



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-50



Reihe 10: Zeilen 51-57



Reihe 11: Zeilen 57-64



Reihe 12: Zeilen 65-71



Reihe 13: Zeilen 71-77



Reihe 14: Zeilen 78-85



Reihe 15: Zeilen 85-93



Reihe 16: Zeilen 93-102



Reihe 17: Zeilen 102-110



Reihe 18: Zeilen 111-117



Reihe 19: Zeilen 118-118



Reihe 20: Zeilen 119-125



Programm: "XEQbc" - Fortsetzung

Reihe 21: Zeilen 125-131



Reihe 22: Zeilen 131-138



Reihe 23: Zeilen 139-145



Reihe 24: Zeilen 145-151



Reihe 25: Zeilen 152-158



Reihe 26: Zeilen 158-164



Reihe 27: Zeilen 164-165



Reihe 28: Zeilen 166-174



Reihe 29: Zeilen 174-181



Reihe 30: Zeilen 181-188



Reihe 31: Zeilen 188-194



Reihe 32: Zeilen 194-201



Reihe 33: Zeilen 202-209



Reihe 34: Zeilen 209-214



Reihe 35: Zeilen 214-222



Reihe 36: Zeilen 223-232



Reihe 37: Zeilen 233-235



Reihe 38: Zeilen 235-240



Programm: "CATBC" - Benoetigte PRGM-Reg: 20

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-6



Reihe 3: Zeilen 6-13



Reihe 4: Zeilen 14-19



Reihe 5: Zeilen 19-27



Reihe 6: Zeilen 27-33



Reihe 7: Zeilen 33-38



Reihe 8: Zeilen 39-45



Reihe 9: Zeilen 46-56



Reihe 10: Zeilen 57-65



Reihe 11: Zeilen 66-67



Programm: "CATBc" - Benoetigte PRGM-Reg: 19

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-6



Reihe 3: Zeilen 6-13



Reihe 4: Zeilen 14-19



Programm: "CATBc" - Fortsetzung

Reihe 5: Zeilen 19-27



Reihe 6: Zeilen 27-33



Reihe 7: Zeilen 33-38



Reihe 8: Zeilen 39-45



Reihe 9: Zeilen 46-56



Reihe 10: Zeilen 57-64



Reihe 11: Zeilen 64-64



Programm: "CATbc" - Benoetigte PRGM-Reg: 11

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 5-5



Reihe 3: Zeilen 6-13



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 18-25



Reihe 6: Zeilen 26-31



Programm: "GTOZBC" - Benoetigte PRGM-Reg: 33

Reihe 1: Zeilen 1-3



Programm: "GTOZBC" - Fortsetzung

Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-10



Reihe 4: Zeilen 10-16



Reihe 5: Zeilen 16-22



Reihe 6: Zeilen 23-29



Reihe 7: Zeilen 29-37



Reihe 8: Zeilen 38-45



Reihe 9: Zeilen 46-52



Reihe 10: Zeilen 52-58



Reihe 11: Zeilen 58-68



Reihe 12: Zeilen 69-71



Reihe 13: Zeilen 71-77



Reihe 14: Zeilen 78-86



Reihe 15: Zeilen 87-94



Reihe 16: Zeilen 95-103



Reihe 17: Zeilen 104-113



Reihe 18: Zeilen 114-116



Programm: "GTOZBc" - Benoetigte PRGM-Reg: 29

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-10



Reihe 4: Zeilen 10-16



Reihe 5: Zeilen 16-22



Reihe 6: Zeilen 23-29



Reihe 7: Zeilen 29-37



Reihe 8: Zeilen 38-45



Reihe 9: Zeilen 46-52



Reihe 10: Zeilen 52-58



Reihe 11: Zeilen 58-68



Reihe 12: Zeilen 69-71



Reihe 13: Zeilen 71-77



Reihe 14: Zeilen 78-86



Reihe 15: Zeilen 87-95



Reihe 16: Zeilen 95-95



Programm: "GTOZbc" - Benoetigte PRGM-Reg: 21

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-9



Reihe 3: Zeilen 9-10



Reihe 4: Zeilen 10-16



Reihe 5: Zeilen 16-22



Reihe 6: Zeilen 23-29



Reihe 7: Zeilen 29-36



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 45-54



Reihe 10: Zeilen 54-56



Reihe 11: Zeilen 56-61



Programm: "DELBC" - Benoetigte PRGM-Reg: 31

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-13



Reihe 4: Zeilen 14-18



Programm: "DELBC" - Fortsetzung

Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 42-47



Reihe 9: Zeilen 48-53



Reihe 10: Zeilen 54-64



Reihe 11: Zeilen 64-66



Reihe 12: Zeilen 66-73



Reihe 13: Zeilen 73-82



Reihe 14: Zeilen 82-90



Reihe 15: Zeilen 90-99



Reihe 16: Zeilen 100-109



Reihe 17: Zeilen 110-111



Programm: "DELBC" - Benoetigte PRGM-Reg: 27

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-13



Programm: "DELbc" - Fortsetzung

Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 42-47



Reihe 9: Zeilen 48-53



Reihe 10: Zeilen 54-64



Reihe 11: Zeilen 64-66



Reihe 12: Zeilen 66-73



Reihe 13: Zeilen 73-82



Reihe 14: Zeilen 82-90



Reihe 15: Zeilen 90-90



Programm: "DELbc" - Benoetigte PRGM-Reg: 19

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-13



Reihe 4: Zeilen 14-18



Programm: "DELbc" - Fortsetzung

Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-32



Reihe 7: Zeilen 32-40



Reihe 8: Zeilen 40-49



Reihe 9: Zeilen 50-51



Reihe 10: Zeilen 51-56



Programm: "COPYBC" - Benoetigte PRGM-Reg: 47

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-21



Reihe 5: Zeilen 21-25



Reihe 6: Zeilen 25-31



Reihe 7: Zeilen 32-38



Reihe 8: Zeilen 39-44



Reihe 9: Zeilen 44-51



Reihe 10: Zeilen 51-56



Programm: "COPYBC" - Fortsetzung

Reihe 11: Zeilen 57-64



Reihe 12: Zeilen 65-73



Reihe 13: Zeilen 74-81



Reihe 14: Zeilen 82-88



Reihe 15: Zeilen 88-97



Reihe 16: Zeilen 97-104



Reihe 17: Zeilen 104-110



Reihe 18: Zeilen 111-118



Reihe 19: Zeilen 119-128



Reihe 20: Zeilen 128-137



Reihe 21: Zeilen 137-146



Reihe 22: Zeilen 147-156



Reihe 23: Zeilen 157-158



Reihe 24: Zeilen 158-161



Reihe 25: Zeilen 161-167



Programm: "COPYBc" .- Benoetigte PRGM-Reg: 37

Reihe 1: Zeilen 1-3



# Programm: "COPYBc" - Fortsetzung

Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 19-25



Reihe 6: Zeilen 25-32



Reihe 7: Zeilen 32-39



Reihe 8: Zeilen 39-46



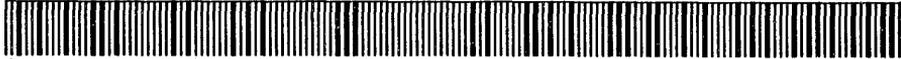
Reihe 9: Zeilen 46-54



Reihe 10: Zeilen 55-62



Reihe 11: Zeilen 63-69



Reihe 12: Zeilen 69-78



Reihe 13: Zeilen 79-85



Reihe 14: Zeilen 85-92



Reihe 15: Zeilen 92-100



Reihe 16: Zeilen 101-110



Reihe 17: Zeilen 110-117



Reihe 18: Zeilen 117-120



Reihe 19: Zeilen 120-123



Programm: "COPYBc" - Fortsetzung

Reihe 20: Zeilen 124-126



Programm: "COPYbc" - Benoetigte PRGM-Reg: 19

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-5



Reihe 3: Zeilen 6-14



Reihe 4: Zeilen 14-17



Reihe 5: Zeilen 18-25



Reihe 6: Zeilen 26-33



Reihe 7: Zeilen 33-43



Reihe 8: Zeilen 43-45



Reihe 9: Zeilen 45-48



Reihe 10: Zeilen 48-54



Reihe 11: Zeilen 54-54



Programm: "CLPBC" - Benoetigte PRGM-Reg: 45

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-8



Programm: "CLPBC" - Fortsetzung

Reihe 3: Zeilen 8-15



Reihe 4: Zeilen 15-21



Reihe 5: Zeilen 21-27



Reihe 6: Zeilen 28-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 42-46



Reihe 9: Zeilen 46-53



Reihe 10: Zeilen 54-59



Reihe 11: Zeilen 59-67



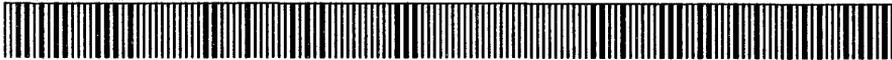
Reihe 12: Zeilen 67-76



Reihe 13: Zeilen 76-84



Reihe 14: Zeilen 84-90



Reihe 15: Zeilen 91-99



Reihe 16: Zeilen 100-106



Reihe 17: Zeilen 107-113



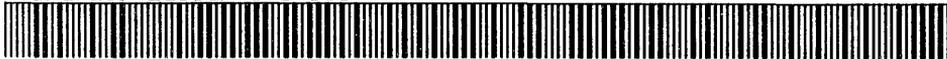
Reihe 18: Zeilen 114-121



Reihe 19: Zeilen 122-130



Reihe 20: Zeilen 131-139



Programm: "CLPBC" - Fortsetzung

Reihe 21: Zeilen 140-149



Reihe 22: Zeilen 150-159



Reihe 23: Zeilen 160-160



Reihe 24: Zeilen 161-165



Programm: "CLPbc" - Benoetigte PRGM-Reg: 35

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-15



Reihe 4: Zeilen 15-21



Reihe 5: Zeilen 21-27



Reihe 6: Zeilen 28-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 42-48



Reihe 9: Zeilen 49-57



Reihe 10: Zeilen 58-65



Reihe 11: Zeilen 66-72



Reihe 12: Zeilen 73-80

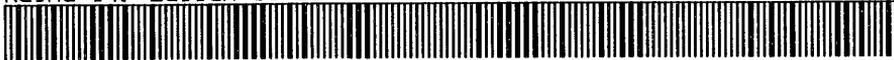


Programm: "CLPBc" - Fortsetzung

Reihe 13: Zeilen 80-86



Reihe 14: Zeilen 86-92



Reihe 15: Zeilen 93-100



Reihe 16: Zeilen 101-110



Reihe 17: Zeilen 111-119



Reihe 18: Zeilen 119-121



Reihe 19: Zeilen 122-124



Programm: "CLPbc" - Benoetigte PRGM-Reg: 17

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-7



Reihe 3: Zeilen 7-14



Reihe 4: Zeilen 15-19



Reihe 5: Zeilen 20-26



Reihe 6: Zeilen 26-33



Reihe 7: Zeilen 34-44



Reihe 8: Zeilen 44-45



Reihe 9: Zeilen 45-49



Programm: "SIZEBC" - Benoetigte PRGM-Reg: 30

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-12



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 27-34



Reihe 7: Zeilen 35-42



Reihe 8: Zeilen 42-47



Reihe 9: Zeilen 48-54



Reihe 10: Zeilen 55-64



Reihe 11: Zeilen 64-67



Reihe 12: Zeilen 68-75



Reihe 13: Zeilen 76-84



Reihe 14: Zeilen 84-92



Reihe 15: Zeilen 93-102



Reihe 16: Zeilen 103-109



Programm: "SIZEbc" - Benoetigte PRGM-Reg: 26

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-12



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 27-34



Reihe 7: Zeilen 35-42



Reihe 8: Zeilen 42-47



Reihe 9: Zeilen 48-54



Reihe 10: Zeilen 55-64



Reihe 11: Zeilen 64-67



Reihe 12: Zeilen 68-75



Reihe 13: Zeilen 76-84



Reihe 14: Zeilen 85-88



Programm: "SIZEbc" - Benoetigte PRGM-Reg: 18

Reihe 1: Zeilen 1-3



Programm: "SIZEbc" - Fortsetzung

Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-12



Reihe 4: Zeilen 12-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 27-34



Reihe 7: Zeilen 34-40



Reihe 8: Zeilen 41-48



Reihe 9: Zeilen 48-52



Reihe 10: Zeilen 53-53

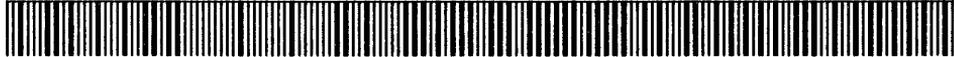


Programm: "ASNBC" - Benötigte PRGM-Reg: 60

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-12



Reihe 4: Zeilen 13-18



Reihe 5: Zeilen 18-23



Reihe 6: Zeilen 23-26



Reihe 7: Zeilen 26-34



Programm: "ASNBC" - Fortsetzung

Reihe 8: Zeilen 35-43



Reihe 9: Zeilen 44-52



Reihe 10: Zeilen 53-62



Reihe 11: Zeilen 62-72



Reihe 12: Zeilen 72-80



Reihe 13: Zeilen 80-86



Reihe 14: Zeilen 87-93



Reihe 15: Zeilen 94-97



Reihe 16: Zeilen 98-104



Reihe 17: Zeilen 104-112



Reihe 18: Zeilen 112-119



Reihe 19: Zeilen 120-128



Reihe 20: Zeilen 129-137



Reihe 21: Zeilen 138-143



Reihe 22: Zeilen 144-153



Reihe 23: Zeilen 153-159



Reihe 24: Zeilen 160-165



Reihe 25: Zeilen 166-173



Programm: "ASNBC" - Fortsetzung

Reihe 26: Zeilen 174-184



Reihe 27: Zeilen 184-193



Reihe 28: Zeilen 193-202



Reihe 29: Zeilen 202-211



Reihe 30: Zeilen 212-215



Reihe 31: Zeilen 216-217



Reihe 32: Zeilen 217-223



Programm: "ASNbc" - Benoetigte PRGM-Reg: 49

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-12



Reihe 4: Zeilen 13-18



Reihe 5: Zeilen 18-23



Reihe 6: Zeilen 23-26



Reihe 7: Zeilen 26-34



Reihe 8: Zeilen 35-43



Reihe 9: Zeilen 44-52



Programm: "ASNbc" - Fortsetzung

Reihe 10: Zeilen 53-62



Reihe 11: Zeilen 62-72



Reihe 12: Zeilen 72-80



Reihe 13: Zeilen 80-86



Reihe 14: Zeilen 87-93



Reihe 15: Zeilen 94-102



Reihe 16: Zeilen 103-110



Reihe 17: Zeilen 110-118



Reihe 18: Zeilen 118-126



Reihe 19: Zeilen 126-134



Reihe 20: Zeilen 135-140



Reihe 21: Zeilen 140-147



Reihe 22: Zeilen 147-156



Reihe 23: Zeilen 157-165



Reihe 24: Zeilen 166-170



Reihe 25: Zeilen 171-175



Reihe 26: Zeilen 175-180



Reihe 27: Zeilen 180-181



Programm: "ASNbc" - Benoetigte PRGM-Reg: 31

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-12



Reihe 4: Zeilen 13-17



Reihe 5: Zeilen 18-24



Reihe 6: Zeilen 24-25



Reihe 7: Zeilen 26-34



Reihe 8: Zeilen 35-44



Reihe 9: Zeilen 45-54



Reihe 10: Zeilen 55-65



Reihe 11: Zeilen 66-74



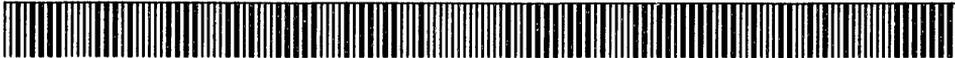
Reihe 12: Zeilen 75-81



Reihe 13: Zeilen 81-90



Reihe 14: Zeilen 91-96



Reihe 15: Zeilen 96-101



Reihe 16: Zeilen 101-105



Reihe 17: Zeilen 106-107



Programm: "LISTBC" - Benoetigte PRGM-Reg: 32

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-33



Reihe 7: Zeilen 33-40



Reihe 8: Zeilen 40-47



Reihe 9: Zeilen 48-52



Reihe 10: Zeilen 53-59



Reihe 11: Zeilen 59-70



Reihe 12: Zeilen 70-71



Reihe 13: Zeilen 71-78



Reihe 14: Zeilen 78-88



Reihe 15: Zeilen 88-96



Reihe 16: Zeilen 97-106



Reihe 17: Zeilen 106-115



Programm: "LISTBC" - Fortsetzung

Reihe 18: Zeilen 115-115



Programm: "LISTBc" - Benoetigte PRGM-Reg: 28

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-33



Reihe 7: Zeilen 33-40



Reihe 8: Zeilen 40-47



Reihe 9: Zeilen 48-52



Reihe 10: Zeilen 53-59



Reihe 11: Zeilen 59-70



Reihe 12: Zeilen 70-71



Reihe 13: Zeilen 71-78



Reihe 14: Zeilen 78-88



Reihe 15: Zeilen 88-94



Programm: "LISTbc" - Benoetigte PRGM-Reg: 19

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 19-24



Reihe 6: Zeilen 25-32



Reihe 7: Zeilen 32-39



Reihe 8: Zeilen 40-49



Reihe 9: Zeilen 49-51



Reihe 10: Zeilen 51-56



Programm: "PRPBC" - Benoetigte PRGM-Reg: 48

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-15



Reihe 4: Zeilen 15-22



Reihe 5: Zeilen 23-26



Programm: "PRPBC" - Fortsetzung

Reihe 6: Zeilen 27-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 41-48



Reihe 9: Zeilen 48-52



Reihe 10: Zeilen 52-58



Reihe 11: Zeilen 58-66



Reihe 12: Zeilen 67-75



Reihe 13: Zeilen 76-83



Reihe 14: Zeilen 83-89



Reihe 15: Zeilen 90-97



Reihe 16: Zeilen 98-106



Reihe 17: Zeilen 107-111



Reihe 18: Zeilen 112-117



Reihe 19: Zeilen 117-125



Reihe 20: Zeilen 126-133



Reihe 21: Zeilen 134-142



Reihe 22: Zeilen 143-151



Reihe 23: Zeilen 152-161



Programm: "PRPBC" - Fortsetzung

Reihe 24: Zeilen 162-171



Reihe 25: Zeilen 172-172



Reihe 26: Zeilen 173-177



Programm: "PRPbc" - Benoetigte PRGM-Reg: 38

Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-15



Reihe 4: Zeilen 15-22



Reihe 5: Zeilen 23-26



Reihe 6: Zeilen 27-33



Reihe 7: Zeilen 33-41



Reihe 8: Zeilen 41-48



Reihe 9: Zeilen 48-56



Reihe 10: Zeilen 57-64



Reihe 11: Zeilen 65-71



Reihe 12: Zeilen 71-79



Reihe 13: Zeilen 80-87



Programm: "PRPbc" - Fortsetzung

Reihe 14: Zeilen 87-92



Reihe 15: Zeilen 92-99



Reihe 16: Zeilen 99-106



Reihe 17: Zeilen 106-116



Reihe 18: Zeilen 117-125



Reihe 19: Zeilen 126-130



Reihe 20: Zeilen 130-135



Reihe 21: Zeilen 135-135



Programm: "PRPbc" - Benoetigte PRGM-Reg: 18

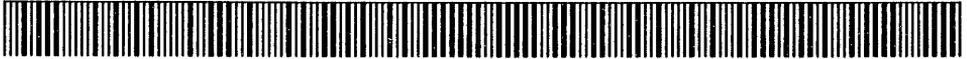
Reihe 1: Zeilen 1-4



Reihe 2: Zeilen 4-9



Reihe 3: Zeilen 9-15



Reihe 4: Zeilen 16-22



Reihe 5: Zeilen 22-27



Reihe 6: Zeilen 28-33



Reihe 7: Zeilen 33-43



Reihe 8: Zeilen 44-49



Programm: "PRPbc" - Fortsetzung

Reihe 9: Zeilen 49-53



Reihe 10: Zeilen 53-54



Programm: "NEWMBC" - Benoetigte PRGM-Reg: 31

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-19



Reihe 5: Zeilen 19-26



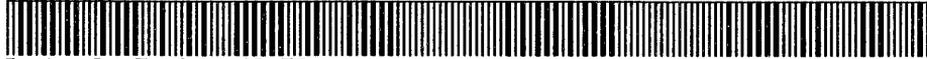
Reihe 6: Zeilen 26-33



Reihe 7: Zeilen 34-42



Reihe 8: Zeilen 42-48



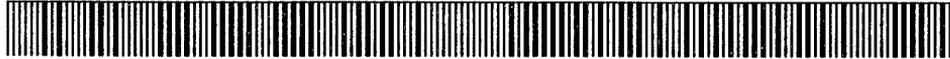
Reihe 9: Zeilen 48-53



Reihe 10: Zeilen 54-62



Reihe 11: Zeilen 63-68



Reihe 12: Zeilen 68-73



Reihe 13: Zeilen 74-82



Reihe 14: Zeilen 82-89



Programm: "NEWMBC" - Fortsetzung

Reihe 15: Zeilen 90-99



Reihe 16: Zeilen 99-108



Reihe 17: Zeilen 109-113



Programm: "NEWMBC" - Benötigte PRGM-Reg: 27

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 26-33



Reihe 7: Zeilen 34-42



Reihe 8: Zeilen 42-48



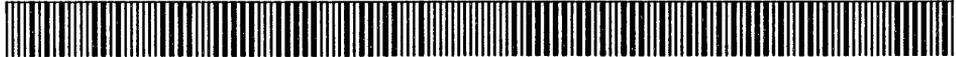
Reihe 9: Zeilen 48-53



Reihe 10: Zeilen 54-62



Reihe 11: Zeilen 63-68



Reihe 12: Zeilen 68-73



Reihe 13: Zeilen 74-82



Programm: "NEWMbc" - Fortsetzung

Reihe 14: Zeilen 82-90



Reihe 15: Zeilen 91-92



Programm: "NEWMbc" - Benoetigte PRGM-Reg: 18

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-19



Reihe 5: Zeilen 19-26



Reihe 6: Zeilen 26-33



Reihe 7: Zeilen 34-40



Reihe 8: Zeilen 41-50



Reihe 9: Zeilen 51-52



Reihe 10: Zeilen 52-56



Programm: "GXLABC" - Benoetigte PRGM-Reg: 58

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-7



# Programm: "GXLABC" - Fortsetzung

Reihe 3: Zeilen 7-12



Reihe 4: Zeilen 13-19



Reihe 5: Zeilen 20-27



Reihe 6: Zeilen 27-32



Reihe 7: Zeilen 32-36



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-48



Reihe 10: Zeilen 48-51



Reihe 11: Zeilen 51-58



Reihe 12: Zeilen 59-64



Reihe 13: Zeilen 65-72



Reihe 14: Zeilen 72-76



Reihe 15: Zeilen 76-83



Reihe 16: Zeilen 83-90



Reihe 17: Zeilen 91-98



Reihe 18: Zeilen 98-106



Reihe 19: Zeilen 107-114



Reihe 20: Zeilen 115-121



Programm: "GXLABC" - Fortsetzung

Reihe 21: Zeilen 121-128



Reihe 22: Zeilen 129-136



Reihe 23: Zeilen 137-142



Reihe 24: Zeilen 143-149



Reihe 25: Zeilen 149-156



Reihe 26: Zeilen 157-167



Reihe 27: Zeilen 167-175



Reihe 28: Zeilen 176-185



Reihe 29: Zeilen 186-194



Reihe 30: Zeilen 195-199



Reihe 31: Zeilen 199-204



Reihe 32: Zeilen 204-204



Programm: "GXLABc" - Benoetigte PRGM-Reg: 48

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-7



Reihe 3: Zeilen 7-12



Reihe 4: Zeilen 13-19



# Programm: "GXLABc" - Fortsetzung

Reihe 5: Zeilen 20-27



Reihe 6: Zeilen 27-32



Reihe 7: Zeilen 32-36



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 44-48



Reihe 10: Zeilen 48-51



Reihe 11: Zeilen 51-58



Reihe 12: Zeilen 59-64



Reihe 13: Zeilen 65-72



Reihe 14: Zeilen 72-80



Reihe 15: Zeilen 81-88



Reihe 16: Zeilen 89-96



Reihe 17: Zeilen 96-102



Reihe 18: Zeilen 103-110



Reihe 19: Zeilen 111-119



Reihe 20: Zeilen 119-124



Reihe 21: Zeilen 124-130



Reihe 22: Zeilen 131-138



Programm: "GXLABc" - Fortsetzung

Reihe 23: Zeilen 139-149



Reihe 24: Zeilen 150-159



Reihe 25: Zeilen 159-160



Reihe 26: Zeilen 160-164



Programm: "GXLABc" - Benoetigte PRGM-Reg: 30

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-7



Reihe 3: Zeilen 7-12



Reihe 4: Zeilen 13-20



Reihe 5: Zeilen 21-27



Reihe 6: Zeilen 28-32



Reihe 7: Zeilen 32-37



Reihe 8: Zeilen 37-44



Reihe 9: Zeilen 45-48



Reihe 10: Zeilen 48-51



Reihe 11: Zeilen 52-58



Reihe 12: Zeilen 58-63



Programm: "GXLABc" - Fortsetzung

Reihe 13: Zeilen 63-71



Reihe 14: Zeilen 72-81



Reihe 15: Zeilen 82-83



Reihe 16: Zeilen 83-88



Programm: "Wunsch" - Benoetigte PRGM-Reg: 21

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-8



Reihe 3: Zeilen 9-18



Reihe 4: Zeilen 19-28



Reihe 5: Zeilen 28-31



Reihe 6: Zeilen 32-35



Reihe 7: Zeilen 35-38



Reihe 8: Zeilen 39-41



Reihe 9: Zeilen 41-47



Reihe 10: Zeilen 48-55



Reihe 11: Zeilen 55-60



Programm: "WDL" - Benötigte PRGM-Reg: 5

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-9



Programm: "SSTFBC" - Benötigte PRGM-Reg: 30

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 6-9



Reihe 4: Zeilen 9-10



Reihe 5: Zeilen 11-18



Reihe 6: Zeilen 19-21



Reihe 7: Zeilen 22-31



Reihe 8: Zeilen 31-38



Reihe 9: Zeilen 39-45



Reihe 10: Zeilen 46-51



Reihe 11: Zeilen 52-57



Reihe 12: Zeilen 58-64



Programm: "SSTFBC" - Fortsetzung

Reihe 13: Zeilen 65-73



Reihe 14: Zeilen 74-83



Reihe 15: Zeilen 84-92



Reihe 16: Zeilen 93-98



Programm: "SSTPbc" - Benoetigte PRGM-Reg: 74

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-6



Reihe 3: Zeilen 7-14



Reihe 4: Zeilen 14-18



Reihe 5: Zeilen 18-19



Reihe 6: Zeilen 20-21



Reihe 7: Zeilen 21-26



Reihe 8: Zeilen 27-31



Reihe 9: Zeilen 31-36



Reihe 10: Zeilen 36-38



Reihe 11: Zeilen 38-44



Reihe 12: Zeilen 45-48



# Programm: "SSTPbc" - Fortsetzung

Reihe 13: Zeilen 49-56



Reihe 14: Zeilen 56-59



Reihe 15: Zeilen 59-62



Reihe 16: Zeilen 63-63



Reihe 17: Zeilen 63-71



Reihe 18: Zeilen 71-72



Reihe 19: Zeilen 72-73



Reihe 20: Zeilen 73-75



Reihe 21: Zeilen 75-76



Reihe 22: Zeilen 76-80



Reihe 23: Zeilen 80-83



Reihe 24: Zeilen 83-86



Reihe 25: Zeilen 86-89



Reihe 26: Zeilen 89-96



Reihe 27: Zeilen 96-98



Reihe 28: Zeilen 99-106



Reihe 29: Zeilen 107-109



Reihe 30: Zeilen 110-115



Programm: "SSTPbc" - Fortsetzung

Reihe 31: Zeilen 116-122



Reihe 32: Zeilen 123-126



Reihe 33: Zeilen 126-133



Reihe 34: Zeilen 134-141



Reihe 35: Zeilen 142-144



Reihe 36: Zeilen 145-149



Reihe 37: Zeilen 149-153



Reihe 38: Zeilen 154-161



Reihe 39: Zeilen 161-167



Reihe 40: Zeilen 168-172



## Kapitel 5

### PROGRAMM-BARCODES UND BARCODEANALYSE

#### I. Barcodes von Programmen

Programmbarcodes sind wohl die bisher am meisten benutzten und auch verbreiteten Barcodes des HP 41. Kein anderer Datenträger ist so zur Massenverbreitung geeignet. Bevor wir den Aufbau dieses Barcodetyps besprechen, müssen vorher noch einige allgemeine Punkte behandelt werden.

#### 1. Globale Marken, Tastenzuweisungen, Autostart und das 'END'

Grundsätzlich können Programmbarcodes, wie im Abschnitt simuliertes synthetisches Tastenfeld praktiziert, ohne Marke angefertigt und eingelesen werden. Solche Programmsequenzen entziehen sich jedoch dem Zugriff, sobald mit 'GTO..' einmal ein 'END' gesetzt wurde. Sie sind dann nur wieder aufzufinden, indem der 'CAT 1' durchgeblättert und mit 'R/S' an richtiger Stelle gestopt wird. Mindestens eine globale Marke ist deshalb in jedem Programm vonnöten, die gleichzeitig den Programm-Namen darstellt, der normalerweise 7 Zeichen lang sein darf. Bei der Arbeit mit Barcodes muß das globale LBL auf 6 Zeichen begrenzt werden, weil der Programm-Name in Datenregister oder im STACK abgespeichert wird. Der Aufbau globaler Marken ist von 'Wickes' und 'Jarett' erschöpfend behandelt. Es sind 4 'LBL'-Bytes: 1 = zwischen 192 und 205, 2 = Null, 3 = Text 1 bis 7 (hier abweichend vom üblichen Textaufbau 242 bis 248) und 4 = Null, sofern keine Tastenzuweisung getätigt ist, sonst trägt 4 den Bitcode des Tastenzuweisungsregisters; allgemein hat eine globale Marke die Form: 192 0 241+n 0; n ist dann die Anzahl der Zeichen des Namens. Alle folgenden Programme für die Anfertigung von PRGM-Barcodes benutzen Funktionen, die PRGM-Namenzuweisungen auf Tasten mit aufzeichnen. Dabei muß sowohl bei der Herstellung der Barcodes, als auch beim Einlesen der 'USER'-Modus eingeschaltet sein. Ist keine Tastenbelegung im Programm mit aufzeichnet, so bleibt jede nachträgliche Zuweisung über das Tastenfeld oder mit 'ASN'-Barcodes selbstverständlich möglich.

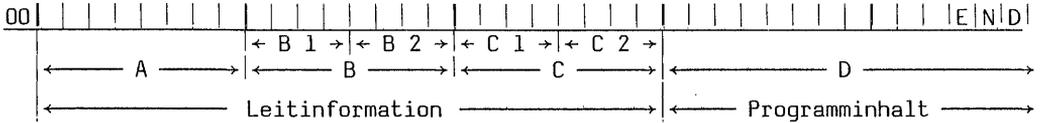
Flag 11 für Autostart kann in Programmbarcodes nicht verschlüsselt werden. Programme, die mit Lesestift eingelesen wurden, müssen mit 'XEQ' gestartet werden. Es gibt dennoch eine Möglichkeit des Sofortstarts, die Sie auch bereits kennen, nämlich die Funktion 'WNDLNK'. Geben Sie diese Lesestiftfunktion vor dem Einlesen eines Programms ein, startet das PRGM sofort nach 'WORKING'. Allerdings hat dies einen kleinen Schönheitsfehler: 'WNDLNK' setzt immer erst ein 'END', welches dann zwischen dem vorstehenden und dem gerade eingelesenen Programm zusätzlich eingefangen ist.

Das Ende eines Programms ist entweder ein abschließendes 'END' oder, falls das Programm als letztes im Speicher steht, das ständige '.END.'. Es besteht aus drei Bytes, die Informationen für den Rechner enthalten, welche variabel sind und sich je nach Zustand des Programms ändern. Das erste Byte zwischen 192-205 stellt einen Code dar, welcher die Anzahl der Nybbles definiert, welche im letzten PRGM-Register vor dem 'END' vorhanden sind. Null Nybbles = 192 - 12 Nybbles = 204. Das zweite Byte bezeichnet die Registerzahl, welche sich zwischen dem Register mit der globalen Marke und dem 'END'-Register befinden. Das dritte Byte enthält Informationen über den Zustand des Programms. Grundlegend ist die Zahl 9, welche anzeigt, das PRGM ist gepackt und kompiliert, das 'END' ist ein 'END'. Die weiteren Alternativen werden jeweils durch 9 + die Summe aus den folgenden Ziffern gebildet: 2 = das PRGM ist unkompiliert - Rücksprungadressen müssen neu berechnet werden - 4 = das PRGM ist nicht gepackt, es befinden sich Nullbytes im Speicher - 32 = es gibt kein 'END' sondern das PRGM-ENDe ist das ständige '.END.'. Das dritte END-Byte eines nicht kompilierten, ungepackten und '.END.'-Programms hat demnach den Wert  $9+2+4+32=47$ . Diesen Wert

kann man ohne weiteres einsetzen. Ein universelles 'END' sind die drei Bytes 192 0 47, die bei den simulierten synthetischen PRGM-Befehlen verwendet sind.

## 2. Aufbau

Der Aufbau dieser Barcodes ist in Kapitel 4 Abschnitt f.1 bereits zum Teil erläutert. Grundsätzlich verändert sich nichts, es wird jetzt lediglich vervollständigt:



A = 8-Bit Prüfsumme mit Übertrag aus vorhergehender Zeile

B1 = 1 PRGM nicht PV- = 2 PRGM PV-geschützt

B2 = Zeilennummer (MOD 16)

C1 = Anzahl Bytes noch auszuführende Restfunktion der vorigen Reihe

C2 = Anzahl Bytes schon ausgeführte Teilfunktion der laufenden Reihe

D = 2 bis 13 Bytes Programminformation

END = die Schlußbytes, die unabhängig von der Art des 'END' immer gesetzt werden müssen. Universell sind die Bytes 192 0 47.

### 2.1 Der Typindikator und die übrigen Leitinformationen

Es wird dem Rechner mitgeteilt B1 = 1 (Dez 16) oder = 2 (Dez 32) für Programmbarcodes - 1 = nicht 'PRIVATE' - 2 = 'PRIVATE'. B2 = Zeilennummer MOD 16  
Zählbeginn bei 0 = Reihe 1. C1: Wird eine Funktion am Ende der vorhergehenden Reihe abgebrochen, so enthält dies Nybble die Anzahl der zu dieser Funktion noch gehörenden Bytes in der folgenden Reihe. Z.B.: 'GTO "XYZ"' = 29 243 88 89 90; in der vorhergehenden Reihe konnten noch 29 und 243 dargestellt werden, und die nächste Reihe beginnt mit 88 89 90, so enthält C1 in dieser Reihe die Zahl 3 (Dez 48). C2: In diesem Nybble ist in der laufenden Reihe die Anzahl der bereits ausgeführten Bytes einer Funktion enthalten. In unserem Beispiel sind 29 und 243 bereits ausgeführt, somit enthält C2 die Zahl 2 (Dez 2).

### 2.2 'Abgebrochene' Funktionen und Länge einer Reihe

Ein Beispiel:

	HEX	28	10	02	C6	00	F2	00	4B	1D	F3
Reihe 1:	PRGM	A	B1B2	C1C2	L	B	L	α	K	GTO	α
	DEZ	40	16	2	198	0	242	0	75	29	243
	HEX	50	11	31	58	59	5A	77	68	69	90
Reihe 2:	PRGM	A	B1B2	C1C2	X	Y	Z	CLX	INT	FRC	RCL
	DEZ	80	17	49	88	89	90	119	104	105	144
	HEX	76	12	10	10	C2	02	2F			
Reihe 3:	PRGM	A	B1B2	C1C2	16	E	N	D			
	DEZ	118	18	16	16	194	2	47			

Der Programminhalt ist unsinnig und dient nur der Erläuterung. Reihe 1: Die alphanumerische Marke 'LBL "K"' besteht aus den Bytes 198 0 242 0 75. Genaugenommen ist das LBL durch Byte 192 = HEX C0 definiert. Der Rechner ermittelt jedoch



### 2.3 Die Zeilennummer und die 8-Bit Prüfsumme

Zeilennummer und Prüfsumme sind miteinander verwoben. Innerhalb der jeweiligen Reihe ist die Zeilen- oder Reihen-Nummer in B2 dargestellt. Obwohl die 1. Reihe üblicherweise mit 'Reihe 1' bezeichnet wird, hat B2 in dieser Reihe den Wert Null. Ist das Programm nicht PV geschützt, so wird der Indikator B1 = 1 und der Wert des Byte B für Reihe 1 ist HEX 10 - DEZ 16. Bei PV-Schutz ist der Indikator B1 = 2 und Reihe 1 ist HEX 20 - DEZ 32. Für die folgenden Reihen wird in B2 MOD 16 weitergezählt. Nun läßt sich mit 4 Bit maximal bis 15 zählen, dann beginnt die Zählung wieder bei Null. Ein direkter Überlauf kann hier nun logischerweise nicht erfolgen, weil der Indikator dann nicht mehr stimmen würde. Um trotzdem eine richtige Aufeinanderfolge der Zeilen zu gewährleisten, wird jeweils die Prüfsumme der vorhergehenden Reihe in die Prüfsumme der laufenden Reihe übertragen. Das Verfahren bleibt ansonsten wie bekannt. Berechnen wir die Prüfsummen der drei Barcodereihen des unsinnigen Programms aus Beispiel 2.2.

Reihe 1:  $16 + 2 + 198 + 0 + 242 + 0 + 75 + 29 + 243$   
=  $805 : 256 = 3 \text{ Rest } 37 - \text{Prüfsumme } 40$

Reihe 2:  $17 + 49 + 88 + 89 + 90 + 119 + 104 + 105 + 144$   
=  $805 + \text{Prfs Reihe 1} = 845 : 256 = 3 \text{ Rest } 77 - \text{Ps } 80$

Reihe 3:  $18 + 16 + 16 + 194 + 2 + 47$   
=  $293 + \text{Prfs Reihe 2} = 373 : 256 = 1 \text{ Rest } 117 - \text{Ps } 118$

### 3. Herstellen von Programmbarcodes auf dem Thermodrucker

Es gibt kein Programm ohne Module. Grundsätzlich wäre es zwar möglich auch ein solches Programm zu schreiben, doch abgesehen von der beträchtlichen Länge und Laufzeit würde dies unbequeme Vorbereitungen erfordern:

- Alle anderen Programme müßten gelöscht und ein ganzbestimmter SIZE eingestellt werden.
- Das zu bearbeitende Programm müßte als erstes im Speicher stehen.
- Es müßte mit 'Dummy-Bytes' so aufgefüllt werden, daß die Programm-Byteanzahl glatt durch 7 teilbar wäre. Dies deswegen, weil sonst das 'END' des zu bearbeitenden Programms und das LBL des Barcode-Herstellungsprogramms im gleichen Register liegen könnten. Dies wäre bei Versetzen der 'Spanischen Wand' (siehe Wickes) unverträglich. Ferner wären die 'Dummy-Bytes' auch im barcodegedruckten Programm enthalten.
- Es müßte die genaue Anzahl der Programmregister über das Tastenfeld mit 'CAT 1'  $7 \text{ MOD} = 0$  als Ausgangswert für die Registerumwandlung ermittelt werden.
- Es gäbe noch eine Reihe weiterer Probleme, die so ein System sehr empfindlich machen. Deswegen ist es einfacher, wenigstens mit dem XF-Modul zu arbeiten.

#### 3.1 Rechner HP 41 CV - Drucker, XF-Modul PRGM "PRGMBc"

##### Programmbedienung:

- SIZE: das Programm stellt 'SIZE 015' ein.

XEQ "PRGMBc"



**Warnung vorweg:** Die Zeilen 10, 12, 14, 53, 55 und 57 dürfen keinesfalls mit 'SST' ausgeführt werden. Das Programm sollte bis zur Frage: "PRIVATE? J/N" auch nicht gestopt werden. Dies führt zu 'MEMORY LOST'!

- "XEQ PACK R/S": Sie werden aufgefordert 'PACK' auszuführen, um evtl. HEX 00 Bytes aus dem Programm zu entfernen.
- "XM LOESCHEN?": Stellt die Frage, ob das X-Memory gelöscht werden darf. Falls nein müssen die Files erst abgespeichert werden, falls ja 'R/S' drücken.

- "PRGM-NAME?": Geben Sie hier den Namen des als Barcodes zu druckenden Programms ein - 'R/S'
- "nn Reihen?": Rechtsbündig wird der Programm-Name und darauf die Anzahl (nn) Reihen ausgedruckt; im Display mit Fragezeichen. Ist die Anzahl akzeptabel = 'R/S' - ist sie zu lang, kann abgebrochen werden.
- "PRIVATE? J/N": Fragt, ob das PRGM PV-geschützt angefertigt werden soll = J oder nicht = N.
- Fehlermeldungen: Die Programmlänge ist auf 60 Register = 420 Bytes begrenzt. Dies ist vermutlich mehr als Papiervorrat und Akku im jeweiligen Zustand gerade 'verkräften'. Möchten Sie dies weiter einschränken, können Sie Zeile 36 verändern. Wird die festgesetzte Registerzahl überschritten, erscheint der Hinweis "PRGM zu lang" rechtsbündig auf dem Druckerstreifen, und das Programm hält an. Mit 'R/S' wird auch hier wieder neu gestartet. Soll ein PV-geschütztes Programm nicht PV-geschützt angefertigt werden - Verweigerungsmeldung: "PRIVATE" und Programmstop. 'R/S' startet neu. Ein PV-geschütztes Programm kann allerdings 'PV' dupliziert werden.

01*LBL "PRGMbc"	42 CLA	83 GTO a	124 DSE 00	165 X=Y?
02*LBL a	43 ARCL X	84*LBL 01	125 GTO 02	166 GTO 01
03 *XEQ PACK R/S"	44 "f Reihen"	85 FS? 03	126 7	167 RDN
04 TONE 7	45 ACA	86 GTO 00	127 STO 00	168 206
05 PROMPT	46 ADV	87 CF 12	128 FC?C 00	169 X=Y?
06 *XM LOESCHEN?"	47 "f ??"	88 ""	129 GTO 00	170 GTO 01
07 TONE 7	48 TONE 7	89 ARCL 10	130 RCL 14	171 RDN
08 PROMPT	49 PROMPT	90 "f"	131 -	172 239
09 64	50 "++++"	91 ACA	132 RCL IND X	173 X<Y
10 X< c	51 XTOA	92 ADV	133 RCL 14	174 X=Y?
11 .	52 190	93 ISG 09	134 +	175 GTO 04
12 STO IND c	53 X< c	94 "+"	135 239	176 X<Y
13 X<Y	54 RCL I	95 "R "	136 -	177 -
14 STO c	55 STO IND c	96 ARCL 09	137 X0?	178 GTO 06
15 SIZE?	56 X<Y	97 FS? 43	138 GTO 07	179*LBL 01
16 15	57 STO c	98 "fP"	139 STO 14	180 2
17 X<Y?	58 CLX	99 SF 12	140*LBL 00	181 GTO 06
18 PSIZE	59 RCL Z	100 ACA	141 RCL 14	182*LBL 03
19 "x+Bd"	60 "PRIVATE? J/N"	101 ADV	142 7	183 29
20 RCL I	61 TONE 7	102 ADV	143 X<Y?	184 X=Y?
21 STOF LAG	62 AON	103*LBL 00	144 GTO 08	185 GTO 00
22 CLRG	63 STOP	104 FS? 03	145 X<Y	186 X<Y
23*LBL b	64 AOFF	105 XEQ 14	146 STO 13	187 30
24 "PRGM-NAME?"	65 74	106 7	147 -	188 X=Y?
25 TONE 7	66 ATOX	107 STO 00	148 STO 00	189 GTO 00
26 AON	67 X=Y?	108 STO 08	149*LBL 03	190 1
27 STOP	68 RAD	109 CLX	150 191	191 GTO 06
28 AOFF	69 FS? 43	110 STO 01	151 RCL IND 00	192*LBL 00
29 FC?C 23	70 CF 03	111 STO 02	152 X<Y?	193 RCL 00
30 GTO b	71 GTO 01	112 STO 03	153 GTO 00	194 1
31 ASTO 10	72*LBL 00	113 STO 04	154 143	195 -
32 SF 25	73 CLST	114 STO 05	155 X<Y	196 X=0?
33 SAVPEP	74 CLA	115 STO 06	156 X<Y?	197 GTO 00
34 FC?C 25	75 "PRGM zu lang"	116 STO 07	157 GTO 03	198 RCL IND X
35 GTO 15	76 ACA	117 CLA	158 GTO 01	199 238
36 60	77 ADV	118 GETX	159*LBL 00	200 -
37 FLSIZE	78 CF 21	119 STO I	160 205	201 GTO 06
38 X<Y?	79 AVIEW	120*LBL 02	161 X<Y	202*LBL 04
39 GTO 00	80 TONE 0	121 ALENG	162 X=Y?	203 3
40 ACA	81 SF 21	122 ATOX	163 GTO 05	204 GTO 06
41 ADV	82 RTN	123 STO IND Y	164 207	205*LBL 05

206 RCL 00	255 XEQ 12	304 XEQ 00	353*LBL 04	402 X>Y?
207 2	256 FS? 02	305*LBL 00	354 CLA	403 XEQ 00
208 -	257 GTO 04	306 RCL X	355 ARCL 10	404 .
209 .	258 STO 13	307 256	356 PURFL	405 CLA
210 X<>Y	259 XEQ 13	308 ST/ Z	357 CLST	406 ARCL 10
211 X=Y?	260 GTO 10	309 MOD	358 DEG	407 SEEKPTA
212 GTO 00	261*LBL 00	310 +	359 CLA	408 RTN
213 RCL IND X	262 XEQ 12	311 INT	360 FS?C 02	409*LBL 00
214 237	263 FS? 02	312 FS?C 06	361 GTO 00	410 "a+b"++
215 -	264 GTO 04	313 RTN	362 CF 12	411 X<> [
216 X>0?	265 XEQ 13	314 RCL 00	363 BEEP	412 X<> d
217 GTO 06	266 GTO 10	315 130	364 ADV	413 RTN
218 GTO 07	267*LBL 00	316 +	365 ADV	414*LBL 13
219*LBL 00	268 STO 12	317 XTOA	366 ADV	415 .
220 SF 00	269 STO 13	318 X<>Y	367 STOP	416 STO 12
221 RCL 00	270 -	319 XTOA	368 RTN	417 +
222 STO 12	271*LBL 09	320 X<> 11	369 GTO a	418 7
223 7	272 STO 14	321 RCL T	370*LBL 00	419 X<>Y
224 STO 00	273 RCL 00	322 XTOA	371 CF 05	420 -
225 RCL Z	274*LBL 10	323 SF 17	372 *PRIVATE*	421 E3
226 CHS	275 FS? 03	324 .	373 ACA	422 /
227 GTO 09	276 GTO 01	325 ACCOL	374 ADV	423 7
228*LBL 06	277 FS?C 05	326 OUTA	375 CF 21	424 +
229 RCL 00	278 GTO 01	327 CLA	376 AVIEW	425 STO 00
230 X<>Y	279 RCL IND X	328 RCL Z	377 TONE 0	426 RTN
231 -	280 ST+ 11	329 SF 25	378 SF 21	427*LBL 14
232 X>0?	281 RDN	330 X=0?	379 RTN	428 FS?C 04
233 GTO 00	282 DSE X	331 ACCOL	380 GTO a	429 RTN
234 RCL 00	283 GTO 10	332 XTOA	381*LBL 12	430 RCL Z
235 STO 12	284 .	333 OUTA	382 2	431 2
236 X<>Y	285 RCL 09	334 CLA	383 -	432 -
237 X=0?	286 1	335 RCL 00	384 FS?C 03	433 CLA
238 STO 12	287 -	336*LBL 11	385 GTO 00	434 ARCL 10
239 CHS	288 16	337 RCL IND X	386 47	435 SEEKPTA
240 7	289 MOD	338 SF 25	387 STO IND Y	436 SF 04
241 STO 00	290 FS? 43	339 X=0?	388 X<>Y	437 RTN
242 X<>Y	291 16	340 ACCOL	389 0	438*LBL 15
243 GTO 09	292 LASTX	341 XTOA	390 X<>Y	439 CLST
244*LBL 00	293 +	342 OUTA	391 -	440 CLA
245 STO 00	294 +	343 CLA	392 STO 00	441 *PGM NONEXIST*
246 GTO 03	295 ST+ 11	344 RDN	393 RTN	442 ACA
247*LBL 07	296 RCL 13	345 DSE X	394*LBL 00	443 ADV
248 SF 01	297 16	346 GTO 11	395 "a+b"++	444 CF 21
249 7	298 *	347 SF 25	396 X<> [	445 AVIEW
250 RCL 00	299 RCL 12	348 ADV	397 X<> d	446 TONE 0
251 X*Y?	300 +	349 ADV	398 RCL IND [	447 SF 21
252 GTO 00	301 ST+ 11	350 CF 25	399 1/X	448 RTN
253 RCL 12	302 X<> 11	351 FC?C 01	400 "*B<B"	449 GTO a
254 +	303 SF 06	352 GTO 01	401 RCL [	450 END 820 Bytes

### 3.2 Rechner HP 41 CV - Drucker, XF- und Plottermodul PRGM "PRGMbc"

#### Programmbedienung:

- SIZE: das Programm stellt nötigenfalls 'SIZE 001' ein

XEQ "PRGMbc"



- "PRGM-NAME?": Geben Sie den Namen des als Barcodes gewünschten Programms ein.
- "P-LAENGENTEST": Eine Laufschrift zeigt an, die Länge des Programms wird getestet. Das kann bei diesem Verfahren bis zur Fehlermeldung: "PRGM zu lang" bis zu 1 1/2 Minuten beanspruchen. Die Länge ist in Zeile 11 auf 36x13 Bytes begrenzt = 468 Bytes. Möchten Sie diesen Wert reduzieren, so müssen Sie 36 vermindern; 1 minus entspricht 13 Bytes.
- "nn Reihen?": Rechtsbündig wird der PRGM-Name und darauf die Anzahl (nn) Reihen ausgedruckt - im Display mit zwei Fragezeichen. Ist die Anzahl ok - 'R/S' drücken, andernfalls kann abgebrochen werden.
- "PRIVATE? J/N": Fragt, ob das PRGM PV-geschützt gedruckt werden soll = J oder nicht = N - dann 'R/S' drücken.
- Fehlermeldungen: Ist ein Programm zu lang, wird rechtsbündig der Hinweis ausgedruckt: "PRGM zu lang". - 'R/S' startet neu. Existiert ein Programm nicht im Speicher oder ist es PV-geschützt wird, ebenfalls rechtsbündig, gedruckt: "NONEXI or PV". Auch hier kann mit 'R/S' neu gestartet werden. Das Duplizieren PV-geschützter Programme ist mit "PRGMbc" nicht möglich.

01*LBL "PRGMbc"	28 GTO 04	55 ARCL X	82 ABS	109 CLST
02*LBL 00	29 RDN	56 "+ Reihen"	83 ARCL X	110 CLA
03 SIZE?	30 CF 21	57 ACA	84 FS? 43	111 "NONEXI or PV"
04 1	31 "LAENGETESTP-"	58 ADV	85 "+P"	112 ACA
05 X*Y?	32 AVIEW	59 "+ ??"	86 FS? 43	113 ADV
06 PSIZE	33 RCLFLAG	60 TONE 7	87 CHS	114 CF 21
07 "##+Fa"	34 .	61 PROMPT	88 SF 12	115 AVIEW
08 RCL I	35 STO d	62 ADV	89 ACA	116 TONE 0
09 STOFAG	36 RDN	63 RCL Z	90 ADV	117 SF 21
10 PINIT	37 STOFAG	64 "PRIVATE? N/J"	91 CF 12	118 RTN
11 36	38 RDN	65 TONE 7	92 BCP	119 GTO 00
12 STO 00	39 1	66 AON	93 Rt	120*LBL 05
13*LBL 01	40*LBL 02	67 STOP	94 CHS	121 SF 21
14 "PRGM-NAME?"	41 DSE 00	68 AOFF	95 BCO	122 CLST
15 TONE 7	42 GTO 00	69 74	96 RDN	123 CLA
16 AON	43 GTO 05	70 ATOX	97 X*0?	124 "PRGM zu lang"
17 STOP	44*LBL 00	71 X=Y?	98 GTO 03	125 ACA
18 AOFF	45 BCP	72 RAD	99 DEG	126 ADV
19 FC?C 23	46 X*0?	73 RCL Z	100 CLA	127 CF 21
20 GTO 01	47 GTO 02	74 1.09	101 CLST	128 AVIEW
21 ACA	48 SF 21	75*LBL 03	102 BEEP	129 TONE 5
22 ADV	49 LASTX	76 ""	103 ADV	130 SF 21
23 1	50 .5625	77 ARCL Y	104 ADV	131 RTN
24 ASTO Y	51 /	78 "+""	105 ADV	132 GTO 00
25 SF 25	52 .4	79 ACA	106 RTN	133 END
26 BCP	53 +	80 ADV	107 GTO 00	292 Bytes
27 FC?C 25	54 CLA	81 "R "	108*LBL 04	

#### 4. Herstellen von Programm-Barcodes auf dem ThinkJet-Drucker

Die grundsätzlichen Ausführungen in den Kapiteln 1 und 2 zur Herstellung von Barcodes mit dem ThinkJet-Drucker gelten auch hier. Es gibt für eine Programmlösung zwei Probleme: 1. die Programmlänge und 2. die Übertragungsgeschwindigkeit der Bausteinbytes. Für fortgeschrittene Synthetiker wäre es eine reizvolle Aufgabe das vorliegende Programm, welches den Lösungsweg auch dem Neuling deutlich macht, so umzuformen, daß die Laufzeit und PRGM-Länge erheblich verkürzt werden. Die Bausteinbytes könnten im X-Memory gespeichert werden. Absolute Sprungadressen könnten in Datenregistern stehen, die dann bei Verwendung von 'STO b', die Übertragungsgeschwindigkeit bei Verzicht auf 'XEQ IND' stark erhöhen. Die 'Auslagerung' der Bausteinbytes in den XM würde das Programm im Hauptspeicher verkürzen. Damit die absoluten Registeradressen immer stimmen,

wäre es sicherlich nötig bei stets gleichem 'SIZE' ein solches Programm auch ständig an der genau gleichen Stelle im Programmspeicher zu haben. Aber dies wäre ebenfalls lösbar. Eine Anregung für den zu beschreitenden Weg findet man beim Studium des Programms "MC" (Morsezeichen senden) im Buch von K. Jarrett 'Synthetisches Programmieren auf dem HP-41 - leicht gemacht' Heldermaun-Verlag. Eine entsprechende Programmlösung wäre sehr willkommen.

4.1 Rechner HP 41 CV - ThinkJet-Drucker, XF- und Plottermodul PRGM "PRTJbc"

Programmbedienung:

- SIZE: es wird erforderlichenfalls 'SIZE 019' vom PRGM eingestellt.

XEQ "PRTJbc"



- "PRGM-NAME?": Geben Sie den Namen des als BC zu druckenden PRGM ein - 'R/S'.
- "PRIVATE? J/N": Soll das PRGM PV-geschützt werden "J" - sonst "N" eingeben, dann 'R/S' drücken. Sofort wird die Kopfzeile gedruckt mit der Anzahl der benötigten Register; anschließend 'Reihe 1: Zeilen 1-n'. Ist eine Seite PRGM-BC fertiggestellt werden Sie gefragt:
- "PAPIER?": Legen Sie neues Papier ein und drücken dann 'R/S'. Sind alle Bar-codes gefertigt ertönt 'BEEP'. Für ein zweites Programm können sie mit 'R/S' neu starten.

01*LBL "PRTJbc"	36 EMDIR	71 1.14	106 RDN	141 SF 01
02 " ! PRGM-Barcodes "	37 SAVEP	72 STO 17	107 "Reihe "	142*LBL 04
03 " ! auf ThinkJet- "	38 ACA	73 22	108 LASTX	143 RCL IND 00
04 " ! Drucker 2225B "	39 EMDIR	74 STO 18	109 INT	144 CLA
05*LBL a	40 PURFL	75*LBL b	110 ARCL X	145 RCL X
06 "HP2225B"	41 ***	76 "BC"	111 RDN	146 16
07 FINDID	42 ACA	77 CLFL	112 "t: Zeilen "	147 ST/ Z
08 SELECT	43 -	78 RCL 16	113 ARCL Z	148 MOD
09 SIZE?	44 "E&k0S"	79 RCL 17	114 "t--"	149 X<>Y
10 19	45 OUTA	80 RCL 18	115 RCL Z	150 INT
11 X<>Y?	46 " Bentiste "	81 CLRG	116 FRC	151 15
12 PSIZE	47 OUTA	82 STO 18	117 E3	152 +
13 "E++Fa"	48 "Register:"	83 RDN	118 *	153 XEQ IND X
14 RCL I	49 ACA	84 STO 17	119 ARCL X	154 X<>Y
15 STOFAG	50 2	85 DSE 18	120 PRA	155 15
16 PINIT	51 -	86 GTO 01	121 RDN	156 +
17 SF 25	52 ACX	87 22	122 RDN	157 XEQ IND X
18 "BC"	53 "PRIVATE? N/J"	88 STO 18	123 RCL 17	158 ALENG
19 23	54 TONE 7	89 "PAPIER?"	124 FS? 43	159 ST+ 15
20 CRFLAS	55 AON	90 TONE 3	125 CHS	160 APPCHR
21 SF 17	56 STOP	91 TONE 3	126 BCP	161 DSE 00
22 "E&k1SE&dDE&l0L"	57 AOFF	92 PSE	127 ABS	162 GTO 04
23 OUTA	58 74	93 TONE 5	128 STO 17	163 FC? 01
24 "PROGRAMM:"	59 ATOX	94 RDN	129 X=0?	164 "4p"
25 ACA	60 X=Y?	95 PROMPT	130 SF 00	165 FS? 01
26 "E&k3S"	61 RAD	96*LBL 01	131*LBL 02	166 "t-t"
27 OUTA	62 " - "	97 RDN	132 ALENG	167 ALENG
28 " - "	63 X=Y?	98 2	133 X=0?	168 ST+ 15
29 ACA	64 "t nicht"	99 STO 15	134 GTO 03	169 APPCHR
30 "PRGM-NAME?"	65 "t PV-geschzt"	100 RDN	135 ATOX	170 SF 17
31 TONE 7	66 OUTA	101 STO 16	136 STO IND Y	171 "E&d0E+r1280S"
32 AON	67 PRBUF	102 RCL 17	137 GTO 02	172 OUTA
33 STOP	68 ADY	103 BCP	138*LBL 03	173 "E+rA"
34 AOFF	69 "E&d0E&k2S"	104 ALENG	139 "pp"	174 OUTA
35 ASTO 16	70 OUTA	105 STO 00	140 APPREC	175 1.024

176 STO 00	218 "BC"	260 RTN	302 "↑↑↑↑"	344 RTN
177*LBL 05	219 PURFL	261*LBL 00	303 SF 01	345*LBL 27
178 .	220 PCLBUF	262 "↑↑↑↑"	304 RTN	346 FC? 01
179 SEEKPT	221 CLA	263 CF 01	305*LBL 23	347 GTO 00
180 SF 17	222 BEEP	264 RTN	306 FC? 01	348 "↑↑pp"
181 "E*b"	223 RTN	265*LBL 19	307 GTO 00	349 SF 01
182 ARCL 15	224 GTO a	266 FC? 01	308 "↑↑↑↑"	350 RTN
183 "↑W"	225*LBL 15	267 GTO 00	309 CF 01	351*LBL 00
184 OUTA	226 FC? 01	268 "↑pt↑↑"	310 RTN	352 "↑↑↑↑"
185 CF 17	227 GTO 00	269 CF 01	311*LBL 00	353 CF 01
186*LBL 06	228 "↑pppp"	270 RTN	312 "↑↑ppp"	354 RTN
187 CLA	229 SF 01	271*LBL 00	313 SF 01	355*LBL 28
188 SF 25	230 RTN	272 "↑↑ppp"	314 RTN	356 FC? 01
189 ARCLREC	231*LBL 00	273 SF 01	315*LBL 24	357 GTO 00
190 FC? 25	232 "↑↑↑↑"	274 RTN	316 FC? 01	358 "↑↑pt↑"
191 GTO 07	233 CF 01	275*LBL 20	317 GTO 00	359 CF 01
192 OUTA	234 RTN	276 FC? 01	318 "↑↑↑↑"	360 RTN
193 GTO 06	235*LBL 16	277 GTO 00	319 SF 01	361*LBL 00
194*LBL 07	236 FC? 01	278 "↑pt↑↑"	320 RTN	362 "↑↑↑↑"
195 ISG 00	237 GTO 00	279 SF 01	321*LBL 00	363 SF 01
196 GTO 05	238 "↑pppt"	280 RTN	322 "↑↑ppp"	364 RTN
197 4	239 CF 01	281*LBL 00	323 CF 01	365*LBL 29
198 STO 00	240 RTN	282 "↑↑pt↑"	324 RTN	366 FC? 01
199*LBL 08	241*LBL 00	283 CF 01	325*LBL 25	367 GTO 00
200 SF 17	242 "↑↑↑↑"	284 RTN	326 FC? 01	368 "↑↑↑↑"
201 "E*blW"	243 SF 01	285*LBL 21	327 GTO 00	369 CF 01
202 OUTA	244 RTN	286 FC? 01	328 "↑↑↑↑"	370 RTN
203 CF 17	245*LBL 17	287 GTO 00	329 SF 01	371*LBL 00
204 "	246 FC? 01	288 "↑pt↑p"	330 RTN	372 "↑↑↑p"
205 ACA	247 GTO 00	289 SF 01	331*LBL 00	373 SF 01
206 DSE 00	248 "↑pppt↑"	290 RTN	332 "↑↑pt↑"	374 RTN
207 GTO 08	249 CF 01	291*LBL 00	333 CF 01	375*LBL 30
208 SF 17	250 RTN	292 "↑↑↑↑"	334 RTN	376 FC? 01
209 "E*rB"	251*LBL 00	293 CF 01	335*LBL 26	377 GTO 00
210 OUTA	252 "↑↑↑↑p"	294 RTN	336 FC? 01	378 "↑↑↑↑"
211 CF 17	253 SF 01	295*LBL 22	337 GTO 00	379 SF 01
212 FC? 00	254 RTN	296 FC? 01	338 "↑↑↑↑"	380 RTN
213 GTO b	255*LBL 18	297 GTO 00	339 CF 01	381*LBL 00
214 .	256 FC? 01	298 "↑pt↑↑"	340 RTN	382 "↑↑↑↑"
215 X(>F	257 GTO 00	299 CF 01	341*LBL 00	383 CF 01
216 DEG	258 "↑pppt↑"	300 RTN	342 "↑↑pt↑"	384 END
217 CLST	259 SF 01	301*LBL 00	343 SF 01	1040 Bytes

## 5. Herstellen von Programmbarcodes auf dem Plotter 7470A

Mit dem folgenden Programm "PGMPbc", den in Kapitel 1 erwähnten Tuschestiften der Firma STAEDTLER, einem speziellen Plotterpapier, dem XF- und dem Plotter-Modul sind alle Tabellen und Programme für dieses Buch hergestellt. In keinem Fall wurde ein PRGM mit PV-Schutz gefertigt. Die Programmbeschreibung ist etwas länger, weil dies PRGM komfortable Bedienung bietet.

### 5.1 Rechner HP 41 CV - Plotter, XF- und Plottermodul PRGM "PGMPbc"

Programmbedienung: -SIZE: es wird mindestens 005 eingestellt. Vor Programmstart erst mit 'CAT 1' die Registerzahl des als BC zu plottenden Programms ermitteln.

XEQ "PGMPbc"



- "PEN 1= .3 SWZ": Fordert auf, diesen Stift einzusetzen - 'R/S'
- "FORTSZG? J/N": Sie haben die Möglichkeit nach Fertigstellung einer Seite Plotter und Rechner auszuschalten, um die Arbeit zu einem späteren Zeitpunkt fortzusetzen. Starten Sie gerade einen neuen Barcodeplot "N" - bei Fortsetzung "J" eingeben - 'R/S'
- "PAPIER?": Erforderlichenfalls Papier einlegen - 'R/S'
- "PEN 2= .5 SWZ": diesen Stift einsetzen - 'R/S'
- "PRGM-NAME?": Geben Sie den Namen des als Barcodes zu plottenden Programms ein und drücken dann 'R/S'
- "PEN 2= .3 SWZ": Stift .5 gegen .3 auswechseln - 'R/S'
- "PGM-REGZAHL?": Geben Sie nun die vor Start mit 'CAT 1' ermittelte Anzahl der Programmregister ein - 'R/S'
- "PRIVATE? N/J": Geben Sie ein, ob das PRGM PV-geschützt "J" oder nicht "N" ausgeführt werden soll - 'R/S' - PV-Schutz-PRGMe duplizieren nicht möglich. Nach diesen Eingaben beginnt das Plotten. Ist Ihr Programm auf der in Arbeit befindlichen Seite abgeschlossen werden Sie gefragt:
- "MEHR PR? J/N": Geben Sie "N" 'R/S' ein, wird der Plot beendet; im Display erscheint: "PLOT ENDE", Sie hören 'BEEP' und das Papier wird 'ausgeworfen'. Geben Sie "J" ein - 'R/S' wird aufgefordert:
- "PEN 2= .5 SWZ": Stiftwechsel - 'R/S' und der weitere Verlauf setzt sich fort ab "PRGM-NAME?". Ist dann die Seite fertiggestellt werden Sie wieder gefragt:
- "PAPIER?": Setzen Sie die Arbeit fort, nehmen Sie die fertige Seite heraus und legen einen neuen Bogen ein - 'R/S' - und der Plot wird fortgesetzt bis das ganze Programm fertiggestellt ist und dann gefragt wird: "MEHR PR? J/N". Sollte die laufende Seite zufällig keinen neuen Programmanfang mehr aufnehmen können wird stattdessen erst nach "PAPIER?" gefragt. Legen Sie einen neuen Bogen ein - 'R/S' - und es geht weiter. Möchten Sie den Plot hier unterbrechen, können Sie dies tun: einfach nach Herausnahme der fertigen Seite und bei längerer Pause der Stifte Plotter und Rechner ausschalten. Zur Fortsetzung der Arbeit Plotter und Rechner wieder einschalten und mit 'XEQ "PGMPbc" neu starten. Die Fragen beginnen entsprechend dem Erststart mit "PEN 1= .3 SWZ", dann folgt wieder
- "FORTSZG? J/N": Hier geben Sie nun "J" ein - 'R/S'
- "PAPIER?": ...einlegen und die Arbeit wird fortgesetzt.

Natürlich dürfen die Registerinhalte und der Status des Rechners zwischenzeitlich nicht verändert werden. Falls dies wegen anderer Arbeiten erforderlich wird, müssen diese Daten aufgezeichnet und vor Plotfortsetzung eingelesen sein. Als Fehlermeldungen gibt es wie beim Programm "PRGMbc" "NONEXI or PV" und, falls aus irgendeinem Grund eine falsche Zeilennummer vorliegt: "ZEILNR. ERROR"

01*LBL "PGMPbc"	19 "PEN 1= .3 SWZ"	37*LBL 00	55 FIX 0	73 ATOX
02 " ! PRGM-Barcodes"	20 TONE 7	38 CF 23	56 CF 22	74 X=Y?
03 " ! auf Plotter"	21 PROMPT	39 "PRGM-NAME?"	57 "PGM-REGZAHL?"	75 RAD
04 " ! HP7470A"	22 CLX	40 TONE 7	58 TONE 7	76 " P"
05 AUTOIO	23 LORG	41 AON	59 PROMPT	77 FS? 43
06 "HP7470A"	24 ACOS	42 STOP	60 FC?C 22	78 LABEL
07 FINDID	25 LDIR	43 AOFF	61 GTO 01	79 CF 17
08 SELECT	26 "FORTSZG? J/N"	44 FC?C 23	62 "Benoetigte "	80 SF 04
09 STO 04	27 AON	45 GTO 00	63 "H-PRGM-Reg: "	81 RCL 03
10 SIZE?	28 TONE 7	46 ASTO 00	64 ARCL X	82 RCL 01
11 5	29 STOP	47 ASTO X	65 LABEL	83 MOVE
12 XXY?	30 AOFF	48 .	66 DEG	84 XEQ 13
13 PSIZE	31 ATOX	49 SF 25	67 "PRIVATE? N/J"	85 RCL 00
14 .	32 74	50 BCP	68 TONE 7	86 1
15 X<>F	33 X=Y?	51 FC?C 25	69 AON	87 PEN
16 PINIT	34 GTO F	52 GTO 11	70 STOP	88 FS? 43
17 CLST	35 XEQ 06	53 XEQ 10	71 AOFF	89 CHS
18 PEN	36 XEQ 07	54*LBL 01	72 74	90*LBL 02

91 STO L	135 RCL 03	179 1	223 TONE 9	267 AVIEW
92 RDN	136 RCL 01	180 PEN	224 TONE 7	268 RTN
93 WHERE	137 MOVE	181 RCL 00	225 TONE 0	269 GTO 00
94 STO 01	138 CLX	182 RCL 02	226 PROMPT	270*LBL 12
95 X<>Y	139 2	183 GTO 02	227 RTN	271 .
96 STO 03	140 SF 25	184*LBL 04	228*LBL 09	272 PDIR
97 R↑	141 PEN	185 CF 23	229 3.61	273 LORG
98 XEQ 12	142 R↑	186 *MEHR PR? J/N*	230 ST+ 01	274 LASTX
99 CLX	143 R↑	187 TONE 8	231 6.665	275 X<>Y
100 ENTER↑	144 RCL 04	188 AON	232 STO 03	276 ACOS
101 .4425	145 LISTEN	189 STOP	233 RCL 01	277 LDIR
102 I MOVE	146 LABEL	190 AOFF	234 MOVE	278 RTN
103 R↑	147 .	191 FC?C 23	235 4.2	279*LBL 13
104 R↑	148 PEN	192 GTO 04	236 CSIZE	280 FS?C 04
105 STO L	149 R↑	193 ATOX	237 .	281 GTO 03
106 *OE*	150 R↑	194 74	238 PEN	282 .
107 OUTA	151 X=0?	195 X*Y?	239 *PEN 2= .5 SWZ*	283 PEN
108 INA	152 GTO 04	196 GTO 14	240 TONE 7	284 *PEN 2= .3 SWZ*
109 CLA	153 STO 02	197 RCL 01	241 PROMPT	285 TONE 7
110 BCP	154 RDN	198 112	242 SF 17	286 PROMPT
111 R↑	155 STO 00	199 X<Y?	243 2	287*LBL 03
112 X=0?	156 RCL 03	200 GTO 05	244 PEN	288 FS?C 03
113 GTO 15	157 RCL 01	201 13.61	245 *Programm: "	289 RTN
114 RDN	158 6.915	202 ST+ 01	246 RCL 04	290 2.5
115 SF 25	159 +	203 XEQ 09	247 LISTEN	291 CSIZE
116 0C	160 STO 01	204 GTO 00	248 LABEL	292 RTN
117 FIX 0	161 MOVE	205*LBL 05	249 RTN	293*LBL 14
118 CF 29	162 133	206 XEQ 06	250*LBL 10	294 *PLOT ENDE*
119 *Reihe "	163 X<Y?	207 XEQ 07	251 ***	295 BEEP
120 LASTX	164 GTO 03	208 GTO 00	252 ARCL 00	296 AVIEW
121 FS? 43	165*LBL F	209*LBL 06	253 *"- "	297 .
122 CHS	166 XEQ 07	210 SF 01	254 LABEL	298 .
123 INT	167 XEQ 10	211 13.9425	255 3	299 PEN
124 ARCL X	168 *Fortsetzung"	212 STO 01	256 CSIZE	300 MOVE
125 RDN	169 FS? 43	213 RTN	257 3.61	301 RTN
126 *"- Zeilen "	170 *"- P"	214*LBL 07	258 ST+ 01	302 GTO 14
127 ARCL Z	171 LABEL	215 XEQ 08	259 SF 03	303*LBL 15
128 *"-	172 CF 17	216 XEQ 09	260 XEQ 13	304 *ZEILNR. ERROR*
129 RCL Z	173 SF 04	217 RTN	261 2	305 TONE 0
130 FRC	174 XEQ 13	218*LBL 08	262 PEN	306 PROMPT
131 E3	175 RCL 03	219 7.495	263 RTN	307 GTO 15
132 *	176 RCL 01	220 FC?C 01	264*LBL 11	308 END
133 ARCL X	177 MOVE	221 STO 01	265 *NONEXI OR PV*	772 Bytes
134 RDN	178*LBL 03	222 *PAPIER?*	266 TONE 0	

### Einige Bemerkungen zum PRGM-Längentest und zu weiteren Programmlösungen

Für die Feststellung der Länge eines als Barcodes herzustellenden Programms gibt es verschiedene Möglichkeiten:

- Ohne Module kann man 'CAT 1' aufrufen. Steht der Drucker auf 'TRACE' wird der Inhalt des Programmspeichers ausgegeben wobei neben dem 'END' eines jeden Programms die Anzahl Bytes gedruckt wird. Diese Anzahl geteilt durch sieben und aufgerundet in jedem Fall auch unter 0.5 ergibt die Anzahl der Register.
- Die Befehlsfolge "PRGM-NAME" - 'SAVEP' - 'FLSIZE' ergibt bei Einsatz des XF-Moduls die Anzahl der Register.
- Nach 'SAVEP' steht der Record-Pointer auf der letzten Recordzeile. Die Befehle 'RCLPT' minus 3 ergeben die Anzahl Byte und 7 / INT 1 + ergibt die Anzahl

der Register.

- Soll der 'EMDIR' nicht benutzt werden, so kann auch die Funktion 'BCP' des Plottermoduls zum Längentest eingesetzt werden, wie dies im Programm "PRGMbc" verwirklicht ist.
- Schließlich ergibt die Befehlsfolge 'EMDIR' - 'SAVEP' - 'EMDIR' - 2 minus, ebenfalls die Programm-Registerzahl.
- Die Funktion 'PLNG' aus dem CCD-Modul ermittelt die Anzahl Bytes eines Programms und mit der Funktion 'PPLNG' wird die Byteanzahl programmgesteuert ins X-Register geschrieben.

Die Herstellung von Programm-Barcodes auf dem Thermodrucker ohne das Plottermodul hat HP-41-Programmierer immer wieder beschäftigt. Die erste Lösung ging von Einzelbyte-Eingabe nach Bytetabelle aus, ein mühseliges Unterfangen. In Ausgabe 10/11 1982 veröffentlichte W. Maschke in 'PRISMA', der Clubzeitschrift des CCD, eine Lösung, die ein Programm im erweiterten Speicher in Daten umwandelt und von dort mit 'GETX' 'STO M' 1-Register = 7-byteweise ins Alpharegister überträgt. Auch das vorliegende Programm "PRGMbc" verwirklicht diese Möglichkeit, allerdings sehr viel komfortabler und in der Nachvollziehbarkeit der Aufbauvorgänge für eine Barcodereihe transparenter - eine Konzession an unsere HP 41 Anfänger. In Prisma 3/4 stellte G.Kruse ein Programm für den ThinkJet-Drucker vor, welches die Möglichkeiten des CCD-Moduls nutzt und kein XF-Modul benötigt. Die Bausteinbytes in diesem Programm, das auch im Handbuch zum CCD-Modul enthalten ist, stammen aus diesem Buch. Das PRGM ist wegen des geringen Speicherbedarfs eine interessante Lösung. Allerdings werden für eine Barcodereihe mit gesamt 13 Bytes immer noch 3 1/2 Minuten Fertigungszeit gebraucht. Die Anregung ist, evtl. mit einer 'STO b'-Lösung, diese Zeit zu verkürzen.

## II. Barcodeanalyse

Gelegentlich mag es vorkommen, eine 'unbekannte' Barcodereihe soll analysiert und identifiziert werden. Zu diesem Zweck stellt der Lesestift die Funktionen 'WNDSCN' und 'WNDTST' bereit. Die Funktion 'WNDSCN' liefert Dezimalwerte, die in den Datenregistern 01 bis nn abgelegt werden. Die Anzahl nn Bytes der Reihe steht im X-Register, der STACK wird angehoben. Das XROM-Programm "WNDTST" basiert ebenfalls auf 'WNDSCN', wandelt jedoch die Dezimalwerte in Binärzahlen um. Das nachfolgende Programm "BCANLY" behält diese beiden Möglichkeiten der Dezimal- und Binärausgabe auf dem Drucker bei. Es muß nur einmal eingelesen werden, und Sie können sich entscheiden, ob Sie nur Dez- oder auch Bin-Ausgabe erhalten möchten.

### 1. Nur Rechner HP 41 C - Drucker, keine Module Analyse-PRGM "BCANLY"

#### Programmbedienung:

-SIZE: das PRGM erfordert je nach Länge der BC-Reihe 'SIZE nnn+2' - maximal 017

XEQ "BCANLY"



- "Barcodeanalyse: Dez/Bin": Sie erhalten zuerst den Ausdruck einer Überschrift
- "NUR DEZ? J/N": Geben Sie "J" ein, wird nur die Dezimalbyte-Folge ausgedruckt Geben Sie "N" ein oder drücken ohne Eingabe gleich 'R/S' wird beides gedruckt
- "BEZ. BCREIHE?": Hier tasten Sie die Buchstaben Ihrer Bezeichnung der Analysereihe ein, z.B. die Buchstaben WNDSCN - 'R/S'; sie wird dann linksbündig doppeltbreit wiederholt. Drücken Sie ohne Eingabe 'R/S' erfolgt nur Vorschub.
- "BC-EINLESEN-": Fordert Sie auf, Ihre Testbarcodes einzulesen. Danach erfolgt sofort das Analyseergebnis entsprechend Ihrer Dez- oder Bin-Entscheidung. Ein Beispiel mit Ergebnisausdruck folgt auf Seite 290. Falls Sie HEX-Angabe benötigten ist die Umrechnung sehr einfach. Ausgehend vom Dez-Wert, z.B. der Zahl 30:  $16 / = 1,875$  (1)  $FRC = 0,875$   $16 \times = 14$ . HEX-Zahl also 1E. Wie Sie wissen, werden die Dez-Zahlen 10-15 hexadezimal als A bis F geschrieben.



Bedienen Sie nun das PRGM bis zur Aufforderung, Sie mögen Ihre "BC-EINLESEN-".

Lesen Sie dann die hier stehenden Barcodes ein. Sie erhalten den rechts nebenstehenden Analyseausdruck, sofern Sie bei "NUR DEZ? J/N" sofort 'R/S' gedrückt, oder "N" eingeben haben. Für eine weitere Analyse brauchen Sie dann nur 'R/S' zu drücken, und das Programm startet neu. Es können so alle vom als HP-41-Barcodes vom Lesestift verstandenen BC-Bytes-Folgen analysiert werden. Auch als Barcodereihe bei normaler Abtastung abgelehnte Strichcodierungen werden gelesen, und es wird ihre Zusammensetzung ausgegeben. Prüfen Sie zum Beispiel die nachfolgende Barcodereihe bei nur Dez und ohne Eingabe einer Bezeichnung. Rechts sehen Sie das Resultat der Prüfung. So können alle HP-41-Barcodes identifiziert werden.

Barcode-Analyse: Dez/Bin  
WNDSCN  
2 Bytes  
B01= 30  
B02= 197

B01=00011110  
B02=11000101

Barcode-Analyse: Dez/Bin  
5 Bytes  
B01= 1  
B02= 2  
B03= 3  
B04= 4  
B05= 5



01*LBL "BCANLY"	23 "BEZ. BCREIHE?"	45 STO 00	67*LBL 02	89 GTO 04
02*LBL 00	24 TONE 7	46*LBL 01	68 RCL 00	90 ACA
03 CF 12	25 AON	47 XEQ 05	69 RCL	91 ADV
04 "Barcode-Analyse"	26 STOP	48 FMT	70 1	92 ISG 00
05 "t: Dez/Bin"	27 AOFF	49 ARCL IND 00	71 +	93 GTO 03
06 PRA	28 FS? 23	50 ACA	72 STO 00	94 RTN
07 FIX 0	29 PRA	51 ADV	73*LBL 03	95 GTO 00
08 CF 00	30 CF 21	52 ISG 00	74 XEQ 05	96*LBL 05
09 "J"	31 "BC-EINLESEN-"	53 GTO 01	75 8	97 "B"
10 ASTO X	32 AVIEW	54 ADV	76 RCL IND 00	98 ACA
11 ENTER↑	33 TONE 7	55 FC? 00	77 120	99 CLA
12 "NUR DEZ? J/N"	34 SF 21	56 XEQ 02	78 /	100 RCL 00
13 AON	35 WNDSCN	57 CLX	79 CLA	101 INT
14 TONE 7	36 CLA	58 CLA	80*LBL 04	102 10
15 STOP	37 ARCL X	59 ADV	81 ENTER↑	103 X/Y?
16 AOFF	38 "t Bytes"	60 ADV	82 FRC	104 "0"
17 ASTO X	39 ACA	61 ADV	83 -	105 ARCL 00
18 X=Y?	40 ADV	62 ADV	84 ARCL X	106 "t="
19 SF 00	41 E3	63 ADV	85 X<> L	107 ACA
20 CLST	42 /	64 CF 12	86 2	108 CLA
21 SF 12	43 1	65 RTN	87 *	109 END
22 CF 23	44 +	66 GTO 00	88 DSE Y	233 Bytes

Die Umformung der Dezimalzahlen in Binärdarstellung ist dem XROM-Programm "WNDTST" entnommen. Seite 291 stellt noch die Typindikatormöglichkeiten vor.

Eine kurze Betrachtung zum Nutzungsgrad der Typindikator-Möglichkeiten

Die in den Kapiteln 2 bis 5 beschriebenen Typindikatoren nutzen nur die Hälfte der Möglichkeiten. Es stehen insgesamt 0 bis 15 = 16 Variationen zur Verfügung. Belegt bzw. nicht benutzt zeigt die folgende Aufstellung:

Typindikatoren

0	00000000	00	nicht benutzt
1	00010000	16	Programm-Barcodes nicht 'PV'
2	00100000	32	Programm-Barcodes mit 'PV'-Schutz
3	00110000	48	nicht benutzt
4	01000000	64	XEQ-Barcodes zur Direktausführung
5	01010000	80	nicht benutzt
6	01100000	96	Zahlen-Barcodes
7	01110000	112	Alpha-Replace-Barcodes
8	10000000	128	Alpha-Append-Barcodes
9	10010000	144	Barcodes für sequentielle Zahleneingabe
10	10100000	160	Barcodes für sequentielle Alphaeingabe
11	10110000	176	BC für sequentielle Alpha-Append-Eingabe - nicht offiziell
12	11000000	192	nicht benutzt
13	11010000	208	nicht benutzt
14	11100000	224	nicht benutzt
15	11110000	240	nicht benutzt

Programm: "PRGMbc" - Benoetigte PRGM-Reg: 118

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 3-4



Reihe 3: Zeilen 4-6



Reihe 4: Zeilen 6-13



Reihe 5: Zeilen 14-19



Reihe 6: Zeilen 19-24



Reihe 7: Zeilen 24-27



Reihe 8: Zeilen 28-34



Reihe 9: Zeilen 34-40



Reihe 10: Zeilen 41-44



Reihe 11: Zeilen 45-50



Reihe 12: Zeilen 50-54



Reihe 13: Zeilen 54-60



Reihe 14: Zeilen 60-63



Reihe 15: Zeilen 64-71



Reihe 16: Zeilen 72-75



Reihe 17: Zeilen 75-81



Programm: "PRGMBc" - Fortsetzung

Reihe 18: Zeilen 82-88



Reihe 19: Zeilen 89-95



Reihe 20: Zeilen 95-100



Reihe 21: Zeilen 101-110



Reihe 22: Zeilen 111-121



Reihe 23: Zeilen 121-128



Reihe 24: Zeilen 129-137



Reihe 25: Zeilen 138-148



Reihe 26: Zeilen 149-155



Reihe 27: Zeilen 156-163



Reihe 28: Zeilen 164-170



Reihe 29: Zeilen 171-179



Reihe 30: Zeilen 180-188



Reihe 31: Zeilen 189-198



Reihe 32: Zeilen 198-206



Reihe 33: Zeilen 207-215



Reihe 34: Zeilen 216-225



Reihe 35: Zeilen 225-235



# Programm: "PRGMBC" - Fortsetzung

Reihe 36: Zeilen 236-246



Reihe 37: Zeilen 247-255



Reihe 38: Zeilen 256-262



Reihe 39: Zeilen 262-269



Reihe 40: Zeilen 270-278



Reihe 41: Zeilen 279-287



Reihe 42: Zeilen 288-296



Reihe 43: Zeilen 297-304



Reihe 44: Zeilen 304-312



Reihe 45: Zeilen 312-320



Reihe 46: Zeilen 320-327



Reihe 47: Zeilen 328-335



Reihe 48: Zeilen 336-343



Reihe 49: Zeilen 344-351



Reihe 50: Zeilen 352-360



Reihe 51: Zeilen 361-369



Reihe 52: Zeilen 370-373



Reihe 53: Zeilen 374-381



Programm: "PRGMbc" - Fortsetzung

Reihe 54: Zeilen 382-390



Reihe 55: Zeilen 391-396



Reihe 56: Zeilen 396-400



Reihe 57: Zeilen 400-407



Reihe 58: Zeilen 408-412



Reihe 59: Zeilen 412-423



Reihe 60: Zeilen 424-434



Reihe 61: Zeilen 434-441



Reihe 62: Zeilen 441-443



Reihe 63: Zeilen 444-450



Reihe 64: Zeilen 450-450



Programm: "PRGMbc" - Benoetigte PRGM-Reg: 42

Reihe 1: Zeilen 1-3



Reihe 2: Zeilen 4-8



Reihe 3: Zeilen 8-14



Reihe 4: Zeilen 14-19



Reihe 5: Zeilen 19-26



# Programm: "PRGMbc" - Fortsetzung

Reihe 6: Zeilen 27-31



Reihe 7: Zeilen 31-35



Reihe 8: Zeilen 36-44



Reihe 9: Zeilen 45-50



Reihe 10: Zeilen 51-56



Reihe 11: Zeilen 56-60



Reihe 12: Zeilen 61-64



Reihe 13: Zeilen 64-70



Reihe 14: Zeilen 71-77



Reihe 15: Zeilen 78-84



Reihe 16: Zeilen 84-91



Reihe 17: Zeilen 91-100



Reihe 18: Zeilen 101-111



Reihe 19: Zeilen 111-112



Reihe 20: Zeilen 113-121



Reihe 21: Zeilen 121-124



Reihe 22: Zeilen 124-130



Reihe 23: Zeilen 131-133



PROGRAMM = "PRIJbc" Benötigte Register: 149 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-2



Reihe 3: Zeilen 2-3



Reihe 4: Zeilen 3-4



Reihe 5: Zeilen 4-4



Reihe 6: Zeilen 4-6



Reihe 7: Zeilen 6-11



Reihe 8: Zeilen 12-14



Reihe 9: Zeilen 14-19



Reihe 10: Zeilen 19-22



Reihe 11: Zeilen 22-23



Reihe 12: Zeilen 23-24



Reihe 13: Zeilen 25-28



Reihe 14: Zeilen 28-30



Reihe 15: Zeilen 30-35



Reihe 16: Zeilen 36-41



Reihe 17: Zeilen 41-45



Reihe 18: Zeilen 45-46



Reihe 19: Zeilen 46-48



Reihe 20: Zeilen 48-53



Reihe 21: Zeilen 53-53



Reihe 22: Zeilen 53-60



Reihe 23: Zeilen 61-64



Reihe 24: Zeilen 64-65



Reihe 25: Zeilen 65-68



Reihe 26: Zeilen 69-70



Reihe 27: Zeilen 70-74



Reihe 28: Zeilen 75-79



Reihe 29: Zeilen 80-86



Reihe 30: Zeilen 86-89



Reihe 31: Zeilen 89-95



Reihe 32: Zeilen 96-103



Reihe 33: Zeilen 104-107



Reihe 34: Zeilen 108-112



Reihe 35: Zeilen 112-115



Reihe 36: Zeilen 115-122



Reihe 37: Zeilen 123-129



Reihe 38: Zeilen 130-136



Reihe 39: Zeilen 136-141



Reihe 40: Zeilen 142-148



Reihe 41: Zeilen 149-156



Reihe 42: Zeilen 157-162



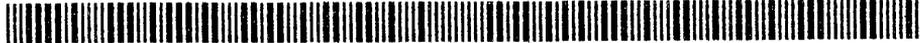
Reihe 43: Zeilen 162-166



Reihe 44: Zeilen 166-171



Reihe 45: Zeilen 171-171



Reihe 46: Zeilen 171-175



Reihe 47: Zeilen 175-180



Reihe 48: Zeilen 181-184



Reihe 49: Zeilen 185-191



Reihe 50: Zeilen 191-197



Reihe 51: Zeilen 198-202



Reihe 52: Zeilen 202-207



Reihe 53: Zeilen 208-211



Reihe 54: Zeilen 212-218



Reihe 55: Zeilen 218-224



Reihe 56: Zeilen 224-228



Reihe 57: Zeilen 228-232



Reihe 58: Zeilen 232-238



Reihe 59: Zeilen 238-242



Reihe 60: Zeilen 242-246



Reihe 61: Zeilen 246-249



Reihe 62: Zeilen 250-253



Reihe 63: Zeilen 254-258



Reihe 64: Zeilen 258-262



Reihe 65: Zeilen 262-267



Reihe 66: Zeilen 267-271



Reihe 67: Zeilen 272-275



Reihe 68: Zeilen 275-278



Reihe 69: Zeilen 278-282



Reihe 70: Zeilen 282-288



Reihe 71: Zeilen 288-292



Reihe 72: Zeilen 292-295



Reihe 73: Zeilen 296-298



Reihe 74: Zeilen 299-302



Reihe 75: Zeilen 302-308



Reihe 76: Zeilen 308-312



Reihe 77: Zeilen 312-316



Reihe 78: Zeilen 316-319



Reihe 79: Zeilen 319-323



Reihe 80: Zeilen 323-328



Reihe 81: Zeilen 328-332



Reihe 82: Zeilen 332-336



Reihe 83: Zeilen 337-339



Reihe 84: Zeilen 340-343



Reihe 85: Zeilen 343-348



Reihe 86: Zeilen 348-352



Reihe 87: Zeilen 352-356



Reihe 88: Zeilen 357-359



Reihe 89: Zeilen 360-363



Reihe 90: Zeilen 363-368



Reihe 91: Zeilen 368-372



Reihe 92: Zeilen 372-376



Reihe 93: Zeilen 376-378



Reihe 94: Zeilen 379-382



Reihe 95: Zeilen 382-384



PROGRAMM = "PGMPbc" Benötigte Register: 111 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-2



Reihe 3: Zeilen 2-3



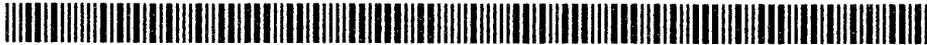
Reihe 4: Zeilen 3-4



Reihe 5: Zeilen 4-6



Reihe 6: Zeilen 6-10



Reihe 7: Zeilen 10-17



Reihe 8: Zeilen 18-19



Reihe 9: Zeilen 19-23



Reihe 10: Zeilen 24-26



Reihe 11: Zeilen 26-31



Reihe 12: Zeilen 31-36



Reihe 13: Zeilen 36-39



Reihe 14: Zeilen 39-44



Reihe 15: Zeilen 44-50



Reihe 16: Zeilen 50-55



Reihe 17: Zeilen 56-57



Reihe 18: Zeilen 57-61



Reihe 19: Zeilen 62-62



Reihe 20: Zeilen 62-63



Reihe 21: Zeilen 63-67



Reihe 22: Zeilen 67-68



Reihe 23: Zeilen 69-76



Reihe 24: Zeilen 76-82



Reihe 25: Zeilen 83-88



Reihe 26: Zeilen 89-97



Reihe 27: Zeilen 98-102



Reihe 28: Zeilen 102-108



Reihe 29: Zeilen 108-115



Reihe 30: Zeilen 115-119



Reihe 31: Zeilen 119-125



Reihe 32: Zeilen 126-126



Reihe 33: Zeilen 127-132



Reihe 34: Zeilen 133-140



Reihe 35: Zeilen 141-148



Reihe 36: Zeilen 148-157



Reihe 37: Zeilen 158-162



Reihe 38: Zeilen 162-167



Reihe 39: Zeilen 167-168



Reihe 40: Zeilen 168-172



Reihe 41: Zeilen 172-178



Reihe 42: Zeilen 179-186



Reihe 43: Zeilen 186-186



Reihe 44: Zeilen 186-193



Reihe 45: Zeilen 193-199



Reihe 46: Zeilen 200-203



Reihe 47: Zeilen 203-208



Reihe 48: Zeilen 208-211



Reihe 49: Zeilen 212-218



Reihe 50: Zeilen 219-222



Reihe 51: Zeilen 222-225



Reihe 52: Zeilen 226-231



Reihe 53: Zeilen 231-236



Reihe 54: Zeilen 236-239



Reihe 55: Zeilen 239-242



Reihe 56: Zeilen 242-245



Reihe 57: Zeilen 245-250



Reihe 58: Zeilen 251-254



Reihe 59: Zeilen 254-259



Reihe 60: Zeilen 259-265



Reihe 61: Zeilen 265-265



Reihe 62: Zeilen 266-273



Reihe 63: Zeilen 273-281



Reihe 64: Zeilen 281-284



Reihe 65: Zeilen 284-287



Reihe 66: Zeilen 288-294



Reihe 67: Zeilen 294-296



Reihe 68: Zeilen 297-303



Reihe 69: Zeilen 303-304



Reihe 70: Zeilen 304-308



Reihe 71: Zeilen 308-308



PROGRAMM = "BCANLY" Benötigte Register: 35 - nicht PV-geschützt

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 3-4



Reihe 3: Zeilen 4-5



Reihe 4: Zeilen 5-7



Reihe 5: Zeilen 8-12



Reihe 6: Zeilen 12-14



Reihe 7: Zeilen 14-21



Reihe 8: Zeilen 22-23



Reihe 9: Zeilen 23-28



Reihe 10: Zeilen 28-31



Reihe 11: Zeilen 31-34



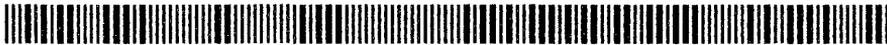
Reihe 12: Zeilen 34-38



Reihe 13: Zeilen 38-44



Reihe 14: Zeilen 45-50



Reihe 15: Zeilen 51-56



Reihe 16: Zeilen 57-66



Reihe 17: Zeilen 66-74



Reihe 18: Zeilen 75-82



Reihe 19: Zeilen 83-89



Reihe 20: Zeilen 90-96



Programm: "BCANLY" - Fortsetzung

Reihe 21: Zeilen 97-104



Reihe 22: Zeilen 104-109



Reihe 23: Zeilen 109-109



Diese Programm-Barcodes sind auf Dem ThinkJet-Drucker mit dem Programm "PRTJbc" angefertigt. Die gesamte Fertigungszeit einschließlich Papierwechsel betrug 1 1/2 Stunden. Daran ist zu erkennen, daß schnellere Programmlösungen sehr gefragt werden müssen. Auf dem Plotter beträgt die Fertigungszeit nur 30 Minuten.

## Kapitel 6

### DER 'EAN' UND ANDERE BARCODES

#### I. Barcodes mit vier Strichen

Diese Barcodetypen verwenden vier verschiedene Strich- und dementsprechende Zwischenraum-Breiten für die Darstellung der Information. Dadurch wird eine hohe Dichte erreicht. Solche Barcodes sind natürlich gegenüber dem Schwarzcode wesentlich komplizierter aufgebaut. Dies hat allerdings den Vorteil großer Lesestabilität, weil die Wahrscheinlichkeit zufällige, nicht zum Code gehörende Marken oder Zwischenräume zu dekodieren, verringert wird. Die Anforderungen an Auflösung und Genauigkeit des Lesestiftes sind dabei ebenfalls höher.

#### a. Der Europäische Artikel-Nummern-Barcode 'EAN'

Der 'EAN'-Barcode ist inzwischen sicherlich jedermann bekannt, weil dieser Code in unterschiedlicher Ausbildung auf fast allen Verpackungen zu finden ist. Dieser Barcode stellt nur Zahlen dar, welche ihrerseits eine Länderkennziffer, eine bundeseinheitliche Betriebsnummer, die Artikelnummer des Herstellers und eine Prüfziffer verschlüsseln. Einzelheiten über den präzisen Aufbau, der drucktechnischen Ausführung, des Trägerfeldes usw. sind in der DIN 66236 niedergelegt. Wer sich mit diesem Code eingehend beschäftigen möchte, findet sehr gründliche Informationen in den beiden Schriften 'EAN Die Europäische Artikelnummerierung in der Bundesrepublik Deutschland', die von der Centrale für Coorganisation Gesellschaft zur Rationalisierung des Informationsaustausches zwischen Handel und Industrie mbH, Postfach 190209, Spichernstr. 55, 5 Köln 1, Telefon (0221) 5749-02, herausgegeben wurden. Weitere Informationen über Funktionssysteme mit Strichcodes erhält man von der Firma INTERMEC GmbH, Langgasse 22, 6392 Neu-Anspach 1, Telefon (06081) 7091.

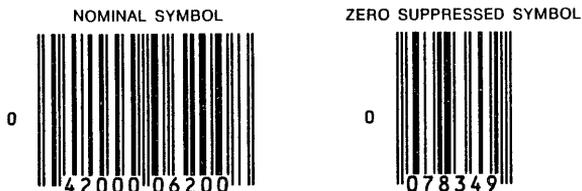
Der 'EAN'-Code



#### b. Der 'UPC' Universal Product Code

Aus diesem amerikanischen Produktcode wurde der 'EAN' entwickelt. Wegen der notwendigen Kompatibilität folgt die Prüfsummenberechnung des 'EAN' dem amerikanischen Verfahren. Dieser Strichcode ist prinzipiell dem 'EAN' ähnlich.

Der 'UPC'-Code



Ansonsten hat dieser Code in Europa keine Verbreitung. Beide Codes stellen in der Normalausführung 13 Zahlen dar. Hieran ist zu erkennen, welche enorme Informationsdichte dieser Code im Vergleich zum System des HP-41-Barcode hat.

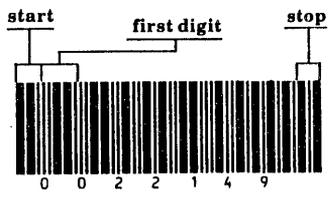
## II. Barcodes mit zwei Strichbreiten

Der in den Kapiteln 1 bis 5 behandelte Barcodetyp zum HP 41 ist ein Schwarzcode mit zwei Balkenbreiten, der die Zwischenräume gleich groß hält und diese nicht für Informationen nutzt. Dadurch ist dieser Barcodetyp etwas aufwendig, wie wir bereits feststellten. Andererseits basiert dieser BC auf dem Binärzahlensystem, und es kann deswegen jedes ASCII-Zeichen hieraus ohne besondere Hilfsmittel abgeleitet werden. Andere Typen legen demgegenüber spezifische Zeichensätze zugrunde. An zwei Codes soll dies kurz vorgestellt werden.

### a. Andere Barcodes

In der Schrift 'BARCODE SYMBOLOGY' Some Observations on Theory and Practice by David C. Allais, February 16, 1982, zu beziehen von T & K Strichcode-Service GmbH., Stapenhorststr. 426, 4800 Bielefeld, werden 33 teils sehr unterschiedliche bis ähnliche BC-Typen aufgeführt, die alle eigene Zeichensätze benutzen, und die für unterschiedliche Anwendungen im Einsatz sind. Die Zeichensätze verwenden bis zu neun 'Bit' und sind sehr verschieden strukturiert. Nachstehend einige Barcodebeispiele. Betrachten Sie die 'Code 39' / 'Code 93' unten links.

Beispiele für unterschiedliche andere Barcodes

 91000168 <sup>39</sup> <b>CODE 11</b>	 *APT-038414* <b>CODE 39</b>	0015532  <b>Codabar</b>	 200101911254 <b>UPC Symbol</b>
 0 0 2 2 1 4 9 <b>Two of Five</b>	6491032456789098765432100000378300975  B032 8 8 3 0 DM 3783.00 <b>Nixdorf Code</b>		
4 1 5 8 9 6 3 6 4 1  <b>Plessey Code</b>	 .515-64-6238. <b>Ames Code</b>	*SR 0005*  <b>CODE 39</b>	
CODE 39-Length 2.47 Inches  * 1 G 1 A Z 3 7 A 4 C R 4 2 3 4 5 6 *	 <b>Two of Five Symbol</b>		
CODE 93-Length 1.71 Inches  ◻1G1AZ37A4CR423456◻	 start                      first digit                      stop 0 0 2 2 1 4 9		

Interleaved  
Two of Five

Als Beispiele für zwei Zeichensätze stellen wir Teilsätze der Codes 3 aus 9 = 'Code 39' und 'Code 93' gegenüber. Beide benutzen 9 'Bit'. Wir betrachten den Buchstaben "A" - diese Sätze kennen den vollständigen ASCII-Zeichensatz.

Code 39					Code 93		
Zchn.	Muster	Bars	Spaces		Zchn.	Muster	'Bits'
1		10001	0100		0		100010100
2		01001	0100		1		101001000
3		11000	0100		2		101000100
4		00101	0100		3		101000010
5		10100	0100		4		100101000
6		01100	0100		5		100100100
7		00011	0100		6		100100010
8		10010	0100		7		101010000
9		01010	0100		8		100010010
0		00110	0100		9		100001010
A		10001	0010		A		110101000
B		01001	0010		B		110100100
C		11000	0010		C		110100010
D		00101	0010		D		110010100
E		10100	0010		E		110010010
F		01100	0010		F		110001010
G		00011	0010		G		101101000
H		10010	0010		H		101100100
I		01010	0010		I		101100010
J		00110	0010		J		100110100
K		10001	0001		K		100011010
L		01001	0001		L		101011000
					M		101001100
					N		101000110

Beim Code 39 sind alle Module so aufgebaut, daß jeweils nur 3 der 9 'Bit' den Wert 1 besitzen. Dabei haben in einem Teil des Gesamtsatzes, der aus 5 'Bit' besteht, die Einsen einen breiten und die Nullen einen schmalen Balkenwert. In einem weiteren Teil, der aus 4 'Bit' besteht, repräsentieren die Einsen einen breiten und die Nullen einen schmalen Zwischenraum. Die 'Bit' sind verschachtelt. Der gesamte Zeichensatz des Code 39 besteht aus insgesamt 44 Symbolen: Zahlen 0-9, Großbuchstaben A-Z sowie den Zeichen -, ., leer, \*, \$, /, + und %. Das Zeichen \* ist hier nicht das Multiplikationssymbol, sondern das Start/Stop-Zeichen. Die letztgenannten 4 Symbole \$ bis % sind Kontrollzeichen, mit deren Hilfe der vollständige 128-ASCII-Zeichensatz benutzt werden kann. Steht zum Beispiel das Zeichen '+' vor "A", so ist "+A" der Buchstabe "a".

Der 'Code 93', der übrigens genau wie 'Code 39' von INTERMEC Corp. entwickelt wurde, zeichnet sich durch eine weitere Verdichtung der Information aus. Ferner ist dieser Code zum 3 aus 9 kompatibel. Auch hier ist die Grundlage ein spezieller Zeichensatz, der oben teils zu sehen ist und bei dem die Einsen einen Balken und die Nullen einen Leerraum darstellen. Dieser Zeichensatz besteht aus 48 Symbolen, welche mit entsprechenden Kontrollzeichen genauso alle 128 ASCII-Zeichen darstellen. Betrachten Sie noch einmal die BC-Beispiele auf Seite 309 unten links. Im Original belegt 'Code 39' insgesamt rund 65 mm und 'Code 93' 45 mm Länge der Reihe. Eine so dichte Darstellung ist mit unserem HP 41 BC nicht erreichbar. Nicht nur, weil in der Beispielreihe der Codes 39/93 mit dem Start/Stop-Symbol 19 Zeichen dargestellt sind, sondern auch, weil diese Form der drucktechnischen Darstellung auf dem Plotter bei der geringen Strichbreite mit Tuschestiften sehr störungsanfällig ist. Andere brauchbare Barcode-Drucke so geringer Strichbreite sind dem Autor von keiner Fachseite bekanntgeworden.

Der Barcodesleser zum HP 75 D versteht nur für das Einlesen von Daten die nachfolgenden Barcodes:

- Code 11 (USD-8)
- 3-of-9-Code (USD-3)
- Interleaved 2-of-5-Code (USD-1)
- Industrial 2-of-5-Code
- 2-of-7 Code (Codabar) (USD-4)
- Universal Product Code (UPC A of E)
- European Article Number (EAN 8 or 13)

Für den HP 71 B existiert zur Zeit (Oktober 1985) noch kein Lesestift. Eine englische Firma entwickelt einen BC-Leser für diesen Rechner, der jedoch genau wie der Leser zum HP 75 nur Daten einlesen können soll. Ein Prototyp wurde auf der Hannover-Messe 1985 vorgestellt; danach ist ein geschlossenes Arbeitssystem wie für den HP 41 verfügbar, hier offenbar und leider nicht geplant. -Abwarten-

### III. Zukünftige Weiterentwicklungen

Offenbar tendiert die weitere Entwicklung zu immer komplexeren und engeren Barcodes, die mit Laserscannern zigtausendmal in der Sekunde abgetastet und so auch bei beschädigtem Barcode trotzdem anhand der 'gesunden' Teile richtig eingelesen werden. Die Informationsdichte wird dabei immer weiter komprimiert. Die Anforderungen an den Lesestift oder Scanner steigen. Die Lesesicherheit steht allenthalben im Vordergrund. Schon heute ist es an Registrierkassen möglich, ziemlich unabhängig von der Haltung der BC auf der Ware, beim Kassieren mit Scannern, eine sehr schnelle und sichere Einlesung der 'EAN' zu praktizieren.

### IV. Schlußwort

Ich hoffe, Sie alle, die mein Buch gelesen haben, konnten Nutzen daraus gewinnen, Anregungen und Erleichterungen für die eigene Arbeit finden; vielleicht konnten Sie auch diesen oder jenen Gedanken aufgreifen und Einzelprobleme noch besser lösen. Bitte teilen Sie dies dann unter Berücksichtigung meines Vorwortes mit. Jede bessere Lösung nehme ich dankbar auf. Die Computerei ist nun einmal dynamisch.

Hier gehen Sie allein weiter. Ich hoffe Ihnen mit Barcodes bequeme, informationsreiche und zeitersparende Arbeitsmöglichkeiten gegeben zu haben. Alles Gute!

**Anhang: 'STO b' - Programm "PTbc" für HP 41 CX, PM und ThinkJet-Drucker**

Mit diesem Programm stelle ich zum Schluß noch eine 'STO b' - Version zur Herstellung von Programmbarcodes auf dem ThinkJet-Drucker und mit dem HP 41CX vor. Wer keinen CX besitzt kann auch andere HP 41-Rechner mit entsprechenden Modulen einsetzen; es muß dann lediglich die Zeile 80, der Befehl 'CLRGX' durch die Zeilen 80 bis 85 aus nebenstehenden Listingauszug ersetzt werden, um die Register 16.029 zu löschen. Dies Programm benutzt die Idee von Clifford Stern aus seinem Morseprogramm "MC", veröffentlicht in "Synthetisches Programmieren auf dem HP-41-leicht gemacht" - siehe Literaturverzeichnis. Durch die Programmzeigermanipulation mit 'STO b' wird eine bedeutende Erhöhung der Übertragungsgeschwindigkeit erzielt. Allerdings erfordert das Speichern der Adressen die Register 0-15. Für die Aufbereitung der Informationsbytes werden die Register 16-29 und für sonstige Daten die Speicher 30-34 belegt; das gesamte Programm mit 921 Bytes 'verbraucht' also bereits 167 Register. Dafür wird zur Herstellung einer Zeile Barcodes fast eine Minute eingespart. Bei einem Programm mit 84 Zeilen wie "PTbc" rund 1 Stunde 20 Minuten. Trotzdem das Programm durch Übernahme einiger guter Routinen aus "PBC" von Gerhard Kruse verkürzt wurde, liegt eine Schwachstelle der Übertragungsgeschwindigkeit in den Zeilen 170-187. Insbesondere die Funktion 'GETREC' verbraucht Zeit. Dennoch bin ich überzeugt, es werden Lösungen gefunden, die schneller sind.

Kurze Programmbeschreibung: Um Normalisierungsprobleme (siehe Wickes) zu vermeiden, wird in den Zeilen 17-25 das ASCII-Zeichen 29 nach 'b' gebracht. Das 'XEQ c' in Zeile 27 setzt die erste Rücksprungadresse lediglich zur Füllung von 'b' und das 'XEQ IND 33' in Zeile 133 die zweite, welche später benutzt wird. Die Befehle 'RCL b' ab Zeile 204 bringen den Registerinhalt als speicherbare Alphakette nach 'X' - Zeiger- und Rücksprungadresse sind 'gesichert'. Beim ersten Antreffen eines 'RTN' in den Marken der Bausteinbytes erfolgt Rücksprung auf die 'XEQ IND 33' folgende Zeile 134 'LBL M', und das zweite 'RTN' Zeile 140 bewirkt Rücksprung auf Zeile 28, wo die in 'X' stehende Zeiger-/Sprungadresse abgespeichert wird. Sind alle Adressen gesammelt, setzt das PRGM mit Zeile 31 fort. Ab 'LBL M' Zeile 134 werden die Informationsbytes mit 'RCL IND 30' aufgerufen, in Nybbles zerlegt und in den Zeilen 149/150 bzw. 155/156 als 'STO b'-Adresse benutzt. Praktisch verzögerungsfrei springt das Programm auf die jeweilige, dem Befehl 'RCL b' folgende Zeile. Die Bausteinbytes werden im Alpharegister gesammelt und bei Antreffen eines 'RTN' springt das Programm auf 'LBL M' zurück. Dieser Vorgang kann mit 'SST' nachvollzogen werden. Das 'STOP' in Zeile 202 darf nicht durch 'RTN' ersetzt werden, weil dies Sprung auf 28 bewirkt.

01*LBL "PTbc"	15 .015	29 ISG 30	43 TONE 7	57 ACX
02 *HP2225B"	16 STO 30	30 GTO a	44 AON	58 "PV? N/J"
03 FINDID	17 RCL b	31 PINIT	45 STOP	59 TONE 7
04 SELECT	18 FC?C 18	32 SF 25	46 AOFF	60 AON
05 SIZE?	19 GTO a	33 "-"	47 ASTO 31	61 STOP
06 35	20 CLA	34 23	48 ACA	62 AOFF
07 X>Y?	21 X<> I	35 CRFLAS	49 "E&k2S"	63 74
08 PSIZE	22 ASTO X	36 "E&k0S"	50 ACA	64 ATOX
09*LBL A	23 "-"	37 ACA	51 " Bentiste "	65 X=Y?
10 "dy**"	24 ARCL X	38 *PROGRAMM: "	52 OUTA	66 RAD
11 RCL I	25 ASTO b	39 ACA	53 "Res: "	67 X=Y?
12 STO d	26*LBL a	40 "E&k1S"	54 ACA	68 "t - PV"
13 .016	27 XEQ c	41 ACA	55 TONE 7	69 ACA
14 STO 33	28 STO IND 30	42 *NAME? "	56 PROMPT	70 PRBUF

71 ADV	130 SF 01	189*LBL ↑	240 "↑↑↑↑"	307 GTO 00
72 1.14	131 GTO [	190 "E*b1H*"	249 RTN	308 "↑↑↑↑"
73 STO 32	132*LBL c	191 ACA	250*LBL 00	309 SF 01
74 22	133 XEQ IND 33	192 DSE X	251 "↑↑↑↑"	310 RTN
75 STO 34	134*LBL [	193 GTO ↑	252 SF 01	311*LBL 00
76*LBL B	135 FS?C 03	194 "E*rB"	253 RTN	312 "↑↑↑↑"
77 "z"	136 GTO ]	195 ACA	254*LBL 05	313 RTN
78 CLFL	137 FS? 02	196 FC? 00	255 RCL b	314*LBL 11
79 16.029	138 GTO \	197 GTO B	256 FC?C 01	315 RCL b
80 CLRGX	139 ISG 33	198 "z"	257 GTO 00	316 FC?C 01
81 DSE 34	140 RTN	199 PURFL	258 "↑↑↑↑"	317 GTO 00
82 GTO 01	141 RCL IND 30	200 PCLBUF	259 SF 01	318 "↑↑↑↑"
83 22	142 RCL X	201 BEEP	260 RTN	319 RTN
84 STO 34	143 16	202 STOP	261*LBL 00	320*LBL 00
85 "PAP?"	144 ST/ Z	203 GTO A	262 "↑↑↑↑"	321 "↑↑↑↑"
86 TONE 7	145 MOD	204*LBL 00	263 RTN	322 SF 01
87 PROMPT	146 X<>Y	205 RCL b	264*LBL 06	323 RTN
88*LBL 01	147 INT	206 FC?C 01	265 RCL b	324*LBL 12
89 RCL 31	148 SF 02	207 GTO 00	266 FC?C 01	325 RCL b
90 RCL 32	149 RCL IND X	208 "↑↑↑↑"	267 GTO 00	326 FC?C 01
91 BCP	150 STO b	209 SF 01	268 "↑↑↑↑"	327 GTO 00
92 "Reihe "	151*LBL \	210 RTN	269 SF 01	328 "↑↑↑↑"
93 LASTX	152 FC?C 02	211*LBL 00	270 RTN	329 SF 01
94 INT	153 GTO ]	212 "↑↑↑↑"	271*LBL 00	330 RTN
95 ARCL X	154 SF 03	213 RTN	272 "↑↑↑↑"	331*LBL 00
96 RDN	155 RCL IND Z	214*LBL 01	273 RTN	332 "↑↑↑↑"
97 "t: Zeilen "	156 STO b	215 RCL b	274*LBL 07	333 RTN
98 ARCL Z	157*LBL ]	216 FC?C 01	275 RCL b	334*LBL 13
99 "t--"	158 APPCHR	217 GTO 00	276 FC?C 01	335 RCL b
100 RCL Z	159 CLA	218 "↑↑↑↑"	277 GTO 00	336 FC?C 01
101 FRC	160 DSE 30	219 RTN	278 "↑↑↑↑"	337 GTO 00
102 E3	161 GTO [	220*LBL 00	279 RTN	338 "↑↑↑↑"
103 *	162 FC? 01	221 "↑↑↑↑"	280*LBL 00	339 RTN
104 ARCL X	163 "↑↑"	222 SF 01	281 "↑↑↑↑"	340*LBL 00
105 PRA	164 FS? 01	223 RTN	282 SF 01	341 "↑↑↑↑"
106 R↑	165 "↑↑"	224*LBL 02	283 RTN	342 SF 01
107 RCL 32	166 APPCHR	225 RCL b	284*LBL 00	343 RTN
108 FS? 43	167 "E*r12005"	226 FC?C 01	285 RCL b	344*LBL 14
109 CHS	168 ACA	227 GTO 00	286 FC?C 01	345 RCL b
110 BCP	169 24	228 "↑↑↑↑"	287 GTO 00	346 FC?C 01
111 ABS	170*LBL d	229 RTN	288 "↑↑↑↑"	347 GTO 00
112 STO 32	171 RCLPT	230*LBL 00	289 RTN	348 "↑↑↑↑"
113 X=0?	172 E3	231 "↑↑↑↑"	290*LBL 00	349 RTN
114 SF 00	173 *	232 SF 01	291 "↑↑↑↑"	350*LBL 00
115 15.015	174 "E*b"	233 RTN	292 SF 01	351 "↑↑↑↑"
116 ALENG	175 ARCL X	234*LBL 03	293 RTN	352 SF 01
117 +	176 "H"	235 RCL b	294*LBL 09	353 RTN
118 STO 30	177 ACA	236 FC?C 01	295 RCL b	354*LBL 15
119*LBL b	178 CLX	237 GTO 00	296 FC?C 01	355 RCL b
120 ALENG	179 SEEKPT	238 "↑↑↑↑"	297 GTO 00	356 FC?C 01
121 X=0?	180*LBL e	239 SF 01	298 "↑↑↑↑"	357 GTO 00
122 GTO 00	181 GETREC	240 RTN	299 SF 01	358 "↑↑↑↑"
123 15	182 OUTA	241*LBL 00	300 RTN	359 SF 01
124 +	183 FS? 17	242 "↑↑↑↑"	301*LBL 00	360 RTN
125 ATOX	184 GTO e	243 RTN	302 "↑↑↑↑"	361*LBL 00
126 STO IND Y	185 X<>Y	244*LBL 04	303 RTN	362 "↑↑↑↑"
127 GTO b	186 DSE X	245 RCL b	304*LBL 10	363 END
128*LBL 00	187 GTO d	246 FC?C 01	305 RCL b	921 Bytes
129 "pp"	188 4	247 GTO 00	306 FC?C 01	

PROGRAMM: P T b c Benötigte Reg: 132

Reihe 1: Zeilen 1-2



Reihe 2: Zeilen 2-5



Reihe 3: Zeilen 6-10



Reihe 4: Zeilen 10-14



Reihe 5: Zeilen 14-18



Reihe 6: Zeilen 19-24



Reihe 7: Zeilen 24-29



Reihe 8: Zeilen 29-34



Reihe 9: Zeilen 34-37



Reihe 10: Zeilen 38-38



Reihe 11: Zeilen 39-42



Reihe 12: Zeilen 42-47



Reihe 13: Zeilen 47-50



Reihe 14: Zeilen 51-51



Reihe 15: Zeilen 51-54



Reihe 16: Zeilen 55-58



Reihe 17: Zeilen 58-64



Reihe 18: Zeilen 65-69



Reihe 19: Zeilen 69-74



Reihe 20: Zeilen 74-79



Reihe 21: Zeilen 79-83



Reihe 22: Zeilen 83-87



Reihe 23: Zeilen 88-92



Reihe 24: Zeilen 92-97



Reihe 25: Zeilen 97-99



Reihe 26: Zeilen 99-105



Reihe 27: Zeilen 105-112



Reihe 28: Zeilen 112-116



Reihe 29: Zeilen 116-122



Reihe 30: Zeilen 123-128



Reihe 31: Zeilen 129-133



Reihe 32: Zeilen 133-138



Reihe 33: Zeilen 138-143



Reihe 34: Zeilen 144-150



Reihe 35: Zeilen 151-155



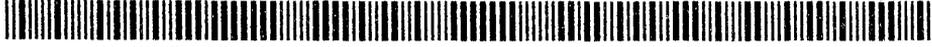
Reihe 36: Zeilen 156-161



Reihe 37: Zeilen 161-165



Reihe 38: Zeilen 165-167



Reihe 39: Zeilen 167-172



Reihe 40: Zeilen 172-176



Reihe 41: Zeilen 177-182



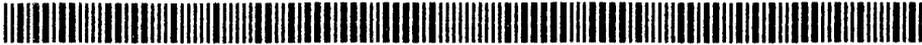
Reihe 42: Zeilen 183-187



Reihe 43: Zeilen 188-191



Reihe 44: Zeilen 191-194



Reihe 45: Zeilen 195-199



Reihe 46: Zeilen 200-206



Reihe 47: Zeilen 206-209



Reihe 48: Zeilen 210-215



Reihe 49: Zeilen 215-218



Reihe 50: Zeilen 219-222



Reihe 51: Zeilen 223-228



Reihe 52: Zeilen 228-231



Reihe 53: Zeilen 231-237



Reihe 54: Zeilen 238-241



Reihe 55: Zeilen 242-245



Reihe 56: Zeilen 246-249



Reihe 57: Zeilen 250-253



Reihe 58: Zeilen 254-258



Reihe 59: Zeilen 258-262



Reihe 60: Zeilen 262-267



Reihe 61: Zeilen 268-271



Reihe 62: Zeilen 272-275



Reihe 63: Zeilen 276-278



Reihe 64: Zeilen 279-282



Reihe 65: Zeilen 282-288



Reihe 66: Zeilen 288-291



Reihe 67: Zeilen 291-297



Reihe 68: Zeilen 297-300



Reihe 69: Zeilen 301-305



Reihe 70: Zeilen 305-308



Reihe 71: Zeilen 308-312



Reihe 72: Zeilen 312-318



Reihe 73: Zeilen 318-321



Reihe 74: Zeilen 321-326



Reihe 75: Zeilen 326-329



Reihe 76: Zeilen 329-333



Reihe 77: Zeilen 334-338



Reihe 78: Zeilen 338-341



Reihe 79: Zeilen 341-347



Reihe 80: Zeilen 347-351



Reihe 81: Zeilen 351-354



Reihe 82: Zeilen 354-358



Reihe 83: Zeilen 358-362



Reihe 84: Zeilen 362-363



## Stichwortverzeichnis

"8-BIT-"	15	"ESCAPE"	15
Abtastrichtung	8	ESCAPE-Grafik-Modus	63
'ACCOL'	15,55	ESCAPE-Modus	12
"ACCOL" (Prgm)	16	Farbdarstellung	3
Acht-Bit-ESCAPE-Grafik-Mode-Mix	64	Faserstifte	17
Acht-Bit-Modus	13,60	Flagstatus	25
Acht-Bit-System	5	Flagsteuerung	51
'ACSPEC'	55	Folgenummer	112
Äußerer Aufbau von Barcodes	3	Fotokopien	16
Akkuspannung	17	"FS"	53
Alpha-Einzelzeichen	21	Füllernybble	96
Andere Barcodes	1	'STOFLAG'	54
Arbeitsgeschwindigkeit	17	Globale Marken	278
ASCII-Erkennungscode	58	Grafik	60
'ASN'	189	'Graphics Language'	17
"AT"	48	'GTO "ALPHA"'	199
Ausdruck von PRGM-Listings	196	'GTO___'-Zeilennummer	179
Ausdruck von PRGM-Zeilen	193	Halbbyte	5
Autostart	2,278	"HHLDEZ"	62
"AZ"	105	IL-Modul	19
Baustein-Bytefolgen	11	Informationsbyte	11,14
'BCREGX'	58,112,113	Informationsdichte	2,3,8
Besondere Funktionen	ab 140	Informationsmengen	12
Binärzahlensystem	4	Initialisieren eines Mediums	197
Bit	4	Instruktionsbytes	12
Bit Dezimalwert	5	Kleinbuchstaben	21
Black-Box	49,144	Kontrast	16
'BLDSPEC'	55	Kurzformexponenten	103
Byte	5	'LBL "ALPHA"'	199
Byteschnapper	145,146	"LD"	113
Catalog-Funktionen	178	Leitinformation	8,9,12
CCD-Modul	21	'LIST'	193
'CLP'	185	'Load Bytes' - Programm	62
Codezahlen für Typindikatoren	9	Löschen von Programmen	185
Codierungsschema	4	Löschen von Programmzeilen	181
'COPY'	183	Massenvervielfältigung	2
Datenerfassung		Mehrbyte-Barcodereihen	9,18,202
Dezimaler Bytewert	4	Nachsilbe	8
Dialogeingabe	18,21	Netzgerät	17
Direktausführung	199	'NEW'	197
Druckerbuffer	55,60	Nicht programmierbare Funktionen	138
Drucker-Mode-Schalter	18	Nicht tastbare Zeichen	2
Drucker-Modi	15,60	Null-Bytes	14
Druckerpapier	16	Nybble	5,96
Druckersonderzeichen	2,55	Nybbleziffern	96
Druckerumbauten	17	Plotter	19
Druckvorlagen	2	Plotter-Modul	17,19,25,280
Dualzahlensystem	4	Plotter-Papier	17
EAN	1	Port-Extender	20
'eGØBEEP'	141,142	Programmlänge feststellen	199,288
Einbytefunktionen	9,18,134	Programm-Listing	196
Einsatzbereiche von Barcodes	1		
'END'	278		

Programm-Name	278	Synthetische Texte	2,18,48,112
Programmzeilen drucken	193	Start/Stop - Signale	8
Programmgesteuertes Einlesen	2	Statusregister	51
Programme löschen	185	Stellenwert	4
Programmzeilen löschen	181	"T"	146
'PRP'	196	Tastenbelegungen	2,168,189,192,202,278
Prüfsumme	8,9	Tastenregister	167
Prüfsumme Berechnungsroutinen	10	Textbyte	8
PV-Schutz	3	Textzeilen	42,106
"Q"	144	Thermodrucker	19
Q-Bote	143	TIME-Modul	19
Quard-Modul	19	Toleranzen	21
'RCLFLAG'	54	Tusche	17
Register d	53,54	Tuschestifte	17
Registerzuweisung 'SIZE'	186	Typindikatoren	291
Reihen-Folgenummer	105	Überlauf (Nybble-Zähler)	9,10,106
'R/S' - Barcodes	62,132	Vorsilbenbyte	8
Rundung	98	X/F-Modul	19
Schwarzcode	2	'XEQ "ALPHA"'	199
Sequentielle Daten	105	'XEQ' - Funktionen	172
Simuliertes synthet. Tastenfeld	203	XM-Modul	19
'SIZE'	186	XROM-XEQ	173
'SNAP 2'	147	XROM-Funktionen	133,140,153
Spaltendruckzahlen	55	"XROMKA"	168
Spaltenmodus (Drucker)	12,13,14,49	Zähler	106
Speicher-Erweiterungs-Modul	19	Zahlen	96
Spiegelprüfsumme	9,10,130	Zeichengeschwindigkeit	17,25
Synthetische Befehle	2,3,147	Zeichenstifte	17
Synthetische Prgmierung	19,21,48,54	Zeilenummer	180
Synthetisches Tastenfeld	3		

### Literaturhinweise

- 'Synthetische Programmierung auf dem HP-41C/CV'  
W.C.Wickes - Deutsch von H. Dalkowski 1983  
2. erweiterte Auflage, Helder mann Verlag Berlin,  
Herderstraße 6-7, D-1000 Berlin 41
- 'Tricks, Tips und Routinen für Taschenrechner der Serie HP-41'  
J.S.Dearing - Deutsche Ausgabe von H. Dalkowski 1984  
Helder mann Verlag Berlin,  
Herderstraße 6-7, D-1000 Berlin 41
- 'Synthetisches Programmieren auf den HP-41 - leicht gemacht'  
K.Jarett - Deutsche Ausgabe von H. Dalkowski 1985  
Helder mann Verlag Berlin,  
Herderstraße 6-7, D-1000 Berlin 41
- 'BAR CODE SYMBOLOGY'  
Some Observations on Theory and Practice  
By David C. Allais - February 16, 1982  
über T & K Strichcode-Service GmbH.,  
Stapenhorststraße 42b, 4800 Bielefeld 1
- 'EAN Die Europäische Artikelnummerierung in der Bundesrepublik  
Deutschland' - 5. überarbeitete Auflage  
Herausgeber: Centrale für Coorganisation  
Gesellschaft zur Rationalisierung des Informationsaustausches  
zwischen Handel und Industrie mbH.,  
Postfach 190209, Spichernstraße 55, 5000 Köln 1
- 'Bar Code-Leser Bedienungshandbuch' - HP 82153A
- 'Plottermodul Benutzerhandbuch' - HP 82184A
- Einige Ausgaben von 'PRISMA' - Clubzeitschrift des CCD  
Herausgeber: CCD - Computerclub Deutschland e.V.  
Postfach 2129, Limburger Straße 15, 6242 Kronberg 2
- 'Anwenderhandbuch HP-41C/CV'  
Karl-Heinz Gosmann  
Verlag: Friedr. Vieweg & Sohn  
Braunschweig/Wiesbaden



## COMPUTERBÜCHER IM HELDERMANN VERLAG

### **Albers, K.: HP-41 Barcodes mit dem HP-IL-System.**

Der Barcodelesestift ist ein preiswertes, dabei jedoch sehr effizientes Zubehörtel für das Einlesen von Daten oder zum Programmieren. Barcodes (BCs) sind die kostengünstigste Methode der externen Datenspeicherung und deren Massenverbreitung. Über HP-41 BCs, ihre oft verblüffend ergiebige Anwendung, Herstellung und ihren Aufbau ist wenig dokumentiert. Mit diesem Buch schließt der Autor die Lücke.

Das Buch behandelt auf rund 300 Seiten ausführlich Vor- und Nachteile von BCs und Lesestift, gibt eine Übersicht über alle BC-Typen des 41-er Systems und erläutert den grundsätzlichen Aufbau und das Kodierungsschema. Die Herstellung von BCs auf dem IL-Thermodrucker, dem ThinkJet-Drucker und dem Plotter 7470A mit vielen nützlichen Hinweisen für einwandfreie Ergebnisse, sowie die dafür notwendigen oder wünschenswerten Geräte werden eingehend beschrieben. 2-Byte-BCs der ASCII-Zeichen 0 – 127, 3-Byte-Ausführung 0 – 255, Alpha- und synthetische Textzeilen-BCs zum Programmieren, überraschend einfache und schnelle Verfahren für 'load bytes' ohne Hilfsprogramm und BCs von einfachen oder längeren 'BLDSPEC'-Sonderzeichen geben rationelle Arbeitshilfen. Die Drucker-Modi 8-Bit/ESCAPE werden genau erklärt. BCs von Zahlen, trickreiches Programmieren großer Zahlen, Kurzformexponenten, numerische und Alphafolgedaten für sequentielle Registereingabe werden ausführlich beschrieben. BCs von 1- und 2-Byte-Tastenfeldfunktionen und von nicht programmierbaren Rechnerfunktionen, INDirekt, besondere Funktionen und deren BCs ( $\uparrow$ --,  $\mu$ --, eGØBEEP--, --, \$T + N IA--, sowie der Q-Bote) werden erschöpfend behandelt. Aus einer speziellen Art BCs bewirkt 'SNAP 2', der BC-Byteschnapper die verblüffend problemlose Lesestifteingabe jedes synthetischen Befehls an beliebiger Programmstelle ohne Tastenbelegungen. XROM-Funktionen-BCs, Mehrbyte-BCs von Tastenfeldfunktionen mit und ohne Argument, alle nicht programmierbaren Rechner- und XROM-Funktionen mit Argument (SIZE, DEL, ASN, PRP usw.) sowie BCs von LBL/GTO/XEQ-"Alpha" und für ein simuliertes synthetisches Tastenfeld für Sofortausführung jeder Art synthetischer Befehle ohne Tastenzuweisung erweitern die Arbeitsmöglichkeiten ganz erheblich. Der Aufbau von Programm-BCs und deren Herstellung auf dem Thermodrucker ohne das Plottermodul, auf dem ThinkJet-Drucker und dem Plotter wird in allen Einzelheiten erklärt. Eine große Zahl von Tabellen der Funktions- und Alphazeichen 0 – 127, 0 – 255 (replace und append), für das synthetische Programmieren, für ein simuliertes, synthetisches Tastenfeld sowie der XROM-Befehl aller zur Zeit verfügbaren Systemmodule sind willkommene Arbeitserleichterungen auch für den Anwender, der keinen Drucker besitzt. Es werden eine Menge praktischer Anwendungen gezeigt für das synthetische Programmieren, zur Datenmasseneinlesung für Schriftfahnen oder Querdruck, zur Herstellung beliebiger Textzeilen und zum Umgang mit speziellen LBL, GTO bzw. XEQ-Befehlen.

Jedes Kapitel enthält komfortable Programme zur BC-Herstellung oder für Umrechnungen in Form von Listing und BCs, insgesamt über 90 Programme! Nach Möglichkeit sind alle Programme dreifach vorhanden: 1. zur Herstellung nur mit Rechner und Thermodrucker ohne Systemmodule; 2. zusätzlich mit dem XF-Modul und 3. außerdem mit dem Plottermodul. Durch 1. sind die Möglichkeiten der BC-Herstellung weitestgehend auch dem Benutzer erschlossen, der nicht über Zusatzmodule verfügt. In einem Anhang werden nicht HP-BCs wie

der Europäische Artikelnummern-BC und einige anderen Typen vorgestellt. Weiterhin wird ein Ausblick auf die zukünftige Entwicklung gegeben.

Wer die Möglichkeiten seines Lesestiftes mit dem HP-41 System optimal nutzen und selbst BCs anfertigen will, braucht dieses leicht verständlich geschriebene Buch. Es wird ihm bald zur unentbehrlichen Arbeitshilfe werden.

(1986, ISBN 3 – 88538 – 804 – 9)

#### **Dearing, J.S.: Tricks, Tips und Routinen für Taschenrechner der Serie HP-41**

Dieses Buch enthält über 350 Routinen und Tips für den HP-41. Von elementaren Tricks und pfiffigen Abkürzungen bis hin zu komplizierten synthetischen Programmen aus dem PPC-Modul und umfangreichen Druck-Routinen ist alles vertreten, was die große Gemeinde der HP-41-Benutzer im Laufe von Jahren herausgefunden und zusammengetragen hat. Dem Autor ist die herkulische Leistung zu verdanken, die Fülle dieser Ergebnisse gesammelt, gesichtet und geordnet zu haben. Der Übersetzer, Heinz Dalkowski, hat in Zusammenarbeit mit dem Autor das Original erweitert und in einem Nachwort die Funktionen des neuesten Rechners aus der Serie HP-41, des HP-41CX, beschrieben. Ein umfangreiches Stichwortverzeichnis von über 1000 Einträgen erschließt die Sammlung lückenlos und läßt zielsicher auffinden, worüber man sich zu informieren wünscht. Man spart so manche Programmierstunde und kommt jedem merkwürdigen Verhalten des HP-41 auf die Spur.

(1984, ISBN 3 – 88538 – 801 – 4)

#### **Horn, J.: HP-71 Basic – leicht gemacht**

Der HP-71 bietet bei erstaunlich geringen Abmessungen Anwendungsmöglichkeiten in einer Vielfalt, wie sie bisher bei Rechnern dieser Größe unbekannt war. Es ist klar, daß zu einem so komfortablen Rechner auch entsprechend umfangreiche Handbücher gehören. Deren Lektüre stellt aber, nicht nur für Computer-Neulinge, eine teils harte, teils zu gehaltvolle Kost dar, deren Genuß durchaus zu Schwierigkeiten führen kann. In diesem Vergleich spielt nun das Buch "HP-71 Basic – leicht gemacht" die Rolle des Kräuter-Likörs, der Völlegefühl und Verstimmungen beseitigt und damit weiteren Appetit auf das ganze Menue macht.

Der betont lockere Stil des Originaltextes wurde auch in der deutschen Bearbeitung voll beibehalten, weil der Umgang mit einem Rechner vorwiegend Freude und möglichst wenig Streß erzeugen soll.

(1986, ISBN 3-88538-807-3)

#### **Jarett, K.: Erweiterte Funktionen des HP-41 – leicht gemacht**

Das Buch beschreibt die Eigenschaften des erweiterten Speichers und der X-Funktionen, mit denen der HP-41C/CV ausgerüstet werden kann und die im HP-41CX fest eingebaut sind. Da das Bedienungshandbuch zum Umgang mit X-Modulen nur knappe Hinweise gibt, war ein Buch nötig, das die Fähigkeiten der X-Funktionen und des HP-41CX vollständig beschreibt. Der Autor, ein führender Experte des HP-41 Systems und Pionier der synthetischen Programmierung, hat dieses Buch in seinem unnachahmlichen Stil – einfach, klar und doch präzise – geschrieben. Die vorliegende deutsche Ausgabe von Heinz Dalkowski wurde darüber hinaus in vielerlei Hinsicht inhaltlich ergänzt. Nach der Lektüre kann der Leser die X-Funktionen wirkungsvoll einsetzen und, wenn er Kenntnisse in synthetischer Programmierung besitzt, die Kraft dieser Kunst mit der der X-Funktionen verbinden. Insbesondere bekommt er über 30 ausgereifte Programme, die von den führenden Experten auf dem Gebiet

stammen, an die Hand, darunter einen umfangreichen Text-Editor für den HP-41C/CV, ein Adressenverzeichnis-Programm, eine Simulation des HP-16, mathematische Programme, Programme zum Übertragen von Textdateien auf Magnetkarten, sowie – in Verbindung mit synthetischer Programmierung – Programme zum Reparieren fehlerhaften Verhaltens einiger speziellen Vorgänge beim Einsatz von X-Funktionen (Betriebssystemfehler). Sämtliche Programme sind als BCs abgedruckt, insgesamt 4181 Bytes! Wer aus seinem Kraftpaket herausholen will, was drin steckt, benötigt dieses Buch.  
(1986, ISBN 3 – 88538 – 803 – 0)

**Jarett, K.: Synthetisches Programmieren auf dem HP-41 – leicht gemacht**

Der Autor wendet sich an HP-41 Benutzer, denen die gründliche aber anspruchsvolle Darstellung der synthetischen Programmierung durch W. C. Wickes ("Synthetische Programmierung auf dem HP-41C/CV", Heldermann Verlag) Schwierigkeiten bereitet. Es gelingt ihm, in ausführlicher Weise einen Zugang zu diesem Gebiet zu bereiten, der zugleich abwechslungsreich und spannend ist. Andererseits berücksichtigt er neueste Erweiterungen des HP-41 durch den Hersteller und bringt Programme, welche die Funktionen aus dem X-Modul und dem Time-Modul beinhalten, wodurch viele synthetische Programme – z. Bsp. das Tastenzuweisungsprogramm – wesentlich kürzer und schneller werden. Für Kundige ist dieses Buch somit eine spielend lesbare Ergänzung des Buches von Wickes, für Anfänger ist es die ideale Einführung.

Die Übersetzung besorgte in gewohnt fachmännischer Weise Heinz Dalkowski.

Diesem Buch liegt eine HP-41 Quick Reference Card bei.

(1985, ISBN 3 – 88538 – 802 – 2)

**Meschede, W.: Plotten und Drucken auf dem HP-41 Thermodrucker**

Dieses Buch enthält 18 Programme zum Plotten auf den HP-41 Thermodruckern und zusätzlich 224 Zeichen in drei Darstellungsformen und alle benötigten Zahlencodes für BLDSPEC und die synthetische Programmierung der Zeichen. Jedem Programm ist ein Programm-Ablaufplan und eine ausführliche Bedienungsanleitung beigegeben, damit sowohl reine Programm-Anwender, als auch Selbst-Programmierer voll auf ihre Kosten kommen. Durch diese Programme stellen selbst logarithmische Skalierung, die Darstellung mehrerer Funktionen in einem Schaubild, hochauflösendes Plotten, sowie Histogramme (Balkendiagramme) keine Probleme mehr dar und durch kleine Änderungen ist die Anpassung an ganz spezielle Anforderungen leicht möglich. Hat man ein Programm längere Zeit nicht mehr benutzt, ermöglichen die Kurzanleitungen im Anhang ein schnelles Rekapitulieren der sehr einfachen und komfortablen Bedienung. Zum Schluß werden dann in Kapitel 7 alle Wünsche nach ganz speziellen Zeichen für die selbst programmierte Ausgabe – einschließlich Querschrift – erfüllt.

Alle Programme sind als BCs abgedruckt, insgesamt 6755 Bytes! Wer graphische Ausgaben oder mehr als nur die einfachen 127 Zeichen benötigt, kann an diesem Buch nicht vorbeigehen.

(1985, ISBN 3 – 88538 – 805 – 7)

**Stroinski, W.: Zusammenfassung der Bedienungshandbücher und Programmieranleitungen für das I/O-ROM, IB- und IL-Interface der Rechner HP-83/85 und HP-86/87**

Handbücher in deutscher Sprache sind meist nur für den Computer und die wichtigsten

Peripherie-Geräte (Drucker, Monitor, Massenspeicher) erhältlich, für die "seltene" Peripherie, durch deren Anschluß der Computer erst seine volle Wirksamkeit erlangt, sind die Beschreibungen häufig nur in englischer Sprache lieferbar. Unabhängig von der Qualität der vorhandenen Sprachkenntnisse ist das Verstehen dieser neuen Materie sicherlich einfacher, wenn die Beschreibungen in deutscher Sprache vorliegen.

Für die Hewlett-Packard-Rechner der 80er Serie (HP-83/85 bzw. HP-86/87) wird mit diesem Buch über das I/O-ROM und die Interfaces für HP-IB (IEEE 488 bzw. IEC 625) und HP-IL (Interface-Loop) dieser Mangel behoben. Es enthält die Übersetzungen der nachfolgend genannten HP-Druckschriften in korrigierter Form:

I/O-ROM, Owner's Manual, 00087 – 90121, Jan. 83

HP-IB Interface, Owner's Manual, 82937 – 90017, Jan. 82

HP-IL Interface, Owner's Manual, 82938 – 90001, Jan. 82.

Der Vorläufer dieser Handbücher (I/O Programming Guide, 00085 – 90142) wurde ebenfalls berücksichtigt, wenn die dort gegebenen Erläuterungen umfangreicher waren, als in den neueren Beschreibungen. Auch für die GPIO-, BCD- und Serial Interfaces sind im Syntax-Anhang vollständige Angaben über die Auswirkungen der einzelnen Anweisungen zu finden.

(1986, ISBN 3 – 88538 – 806 – 5)

### **Wickes, W. C.: Synthetische Programmierung auf dem HP-41C/CV, 2. erweiterte Auflage**

Die englische Originalausgabe dieses Buches ist in den U.S.A. ein Bestseller unter der Literatur über Kleinrechner geworden und auch in Deutschland wurden über 8000 Exemplare verkauft. Die deutsche Ausgabe von Heinz Dalkowski enthält gegenüber dem Original zahlreiche Verbesserungen, Verfeinerungen und Ergänzungen und wird zu Recht die "Bibel der synthetischen Programmierung" genannt.

Der Autor führt den Leser in leicht verständlicher Weise "durch" den HP-41C/CV, entdeckt ihm alle seine verborgenen Fähigkeiten und geht so inhaltlich weit über das hinaus, was das "Handbuch" bietet. Der Leser lernt die Synthetische Programmierung kennen, durch die der Rechner zu unglaublichen Taten veranlaßt werden kann: Erzeugung neuer Zeichen in der Anzeige; Verwendung des Alpha-Registers als arithmetisches Daten-Register; vollständige Benutzerkontrolle über alle Flags (einschließlich der normalerweise unzugänglichen Systemflags); Zugriff auf sämtliche Informationen über den Zustand des Rechners; schnelle Alphabetisierung von Alpha-Daten; Erzeugung neuer Töne; Verwandlung von Programmzeilen in Daten und umgekehrt; programmierter Zugriff auf beliebige Zeilen in ROM's; Herstellung programmierender Programme.

Synthetische Programmierung ist nicht nur für Hobby-Anwender, sondern in gleicher Weise für professionelle Benutzer von Interesse, wenn es darum geht, Speicherplatz zu sparen oder die Bearbeitungsgeschwindigkeit zu erhöhen.

Die zweite Auflage des Buches ist um inzwischen bekannt gewordene Fortschritte der synthetischen Programmierung erweitert worden: Der Byte-Schnapper, Programme zur automatischen Erzeugung synthetischer Programmzeilen, die Funktion eGØBEEP, Programme zur Herstellung vollständiger hexadezimaler Speicherauszüge, synthetischer Vorstoß ins X-Memory.

Obwohl dieses Buch manche Schwierigkeit enthält, was für den einen oder anderen Leser die vorausgehende Lektüre des Buches von K. Jarett, "Synthetisches Programmieren auf dem HP-41 – leicht gemacht" (ebenfalls im Helder mann Verlag) empfehlenswert macht, ist dieses Buch unerläßlich für jeden HP-41C/CV/CX Benutzer, der die Fähigkeiten und Möglichkeiten des Rechners voll ausschöpfen will.

(1983, ISBN 3 – 88538 – 800 – 6)

## **DIE HP-PALETTE DES HELDERMANN VERLAGES**

Albers, K.: HP-41 Barcodes mit dem HP-IL-System. 320 Seiten, 44.00 DM, ISBN 3-88538-804-9 (1986)

Dalkowski, H., Fegert, S.: Eine Programmsammlung für den HP-41. ISBN 3-88538-809-X (1987)

Dearing, J. S.: Tricks, Tips und Routinen für Taschenrechner der Serie HP-41. Deutsche Ausgabe von H. Dalkowski. 220 Seiten, 36.00 DM, ISBN 3-88538-801-4 (1984).

Emery, K.: HP-41 Mikrocode-Programmierung für Anfänger. ISBN 3-88538-808-1 (1987)

Horn, J.: HP-71 Basic — leicht gemacht. Deutsche Ausgabe von W. Stroinski. Ca. 200 Seiten, ca. 40.00 DM, ISBN 3-88538-806-5 (1986)

Jarett, K.: Erweiterte Funktionen des HP-41 — leicht gemacht. Deutsche Ausgabe von H. Dalkowski. 240 Seiten, 44.00 DM, ISBN 3-88538-803-0 (1986).

Jarett, K.: Synthetisches Programmieren auf dem HP-41 — leicht gemacht. Deutsche Ausgabe von H. Dalkowski. 170 Seiten, 40.00 DM, ISBN 3-88538-802-2 (1985). Dem Buch liegt eine Quick Reference Card bei.

Meschede, W.: Plotten und Drucken auf dem HP-41 Thermodrucker. 176 Seiten, 36.00 DM, ISBN 3-88538-805-7 (1985).

Stroinski, W. (Herausgeber): Zusammenfassung der Bedienungshandbücher und Programmieranleitungen für das I/O-ROM, IB- und IL-Interface der Rechner HP-83/85 und HP-86/87. Ca. 200 Seiten, ca. 58.00 DM, ISBN 3-88538-806-5 (1986).

Wickes, W. C.: Synthetische Programmierung auf dem HP-41C/CV. Deutsche Ausgabe von H. Dalkowski. 165 Seiten, 36.00 DM, ISBN 3-88538-800-6 (1983).

HP-41 Kombinierte Hex/Dezimale Byte Tabelle, 7 × 11.5 cm Plastikkarte, 6.00 DM.

HP-41 Quick Reference Card, 7 × 15 cm Plastikkarte, 8.00 DM. (Diese Karte liegt dem Buch "Synthetisches Programmieren auf dem HP-41 — leicht gemacht" von K. Jarett kostenlos bei.)

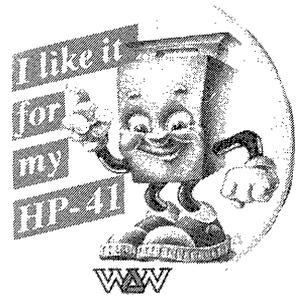
Alle Produkte sind zu den oben angegebenen Preisen ohne Versandkosten direkt vom Verlag erhältlich, die Plastikkarten nur auf diese Weise. Bitte richten Sie Ihre Bestellung an

**Heldermann Verlag Berlin**  
**Nassauische Str. 26**  
**D-1000 Berlin 31**

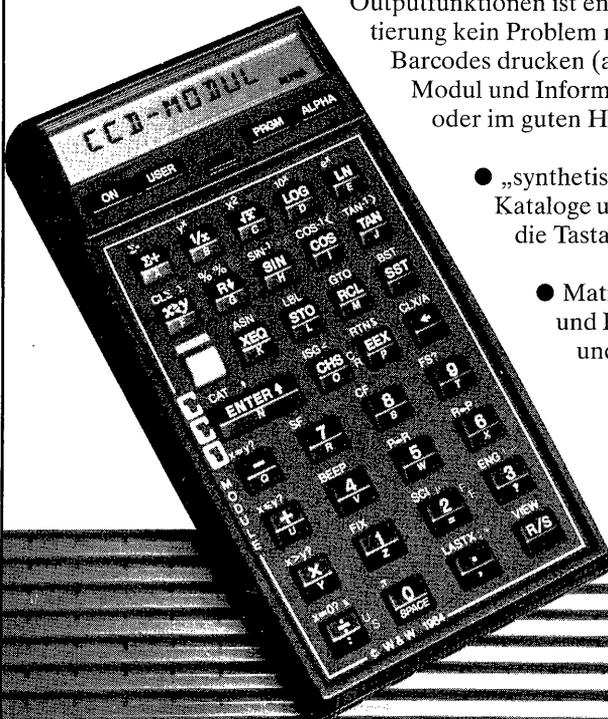
**Jetzt gemeinsam**

**HP-41 und**

**CCD-Modul**



Das CCD-Modul – ein Kraftpaket für den HP-41. In dieser intelligenten Erweiterung stecken 8 kByte (über 100 neue Funktionen!) pure Energie, mit der das Arbeiten auf dem Rechner noch mehr Spaß macht. Bisher schwer durchführbare Operationen lassen sich durch die vielen interessanten Funktionen jetzt spielend programmieren. Besondere Stärken sind z. B. die Betriebssystemerweiterungen, die nun eine direkte Eingabe von „synthetischen“ Funktionen und Kleinbuchstaben ermöglichen. Tastenzuordnungen, Zeitalarme und Matrizen (max. 24x25) können auch im Extended-Memory abgespeichert und sekundenschnell ausgetauscht werden. Mit den Input-/Outputfunktionen ist endlich auch die Druckerformatierung kein Problem mehr, ja es lassen sich sogar Barcodes drucken (auch mit dem ThinkJet!). Das Modul und Informationsmaterial ist direkt bei uns oder im guten HP-Fachhandel erhältlich.



- „synthetische“ Programmierung, neue Kataloge und Kleinbuchstaben direkt über die Tastatur
- Matrixfunktionen, Binärfunktionen und Funktionen für formatierte Ein- und Ausgaben
- PEEK- und POKE-Funktionen für fortgeschrittene Programmierer



Im Ahlemaar 20  
Postfach 800133  
5060 Bergisch Gladbach 2  
☎ (02202) 8 50 68

Korrektur zu Seite 50 des Buches: "HP-41 Barcodes  
mit dem HP-IL-System" von K.Albers

Programm "SYNTBC": zwischen den Zeilen '70 GTO 01'  
und 'LBL 04' sind die folgenden 3 Befehle einzufügen:  
'RCL 00', '1', '-'. Nach 'LBL 04' ist einzufügen:'CLA'.

Programm "SYNTBc": zwischen den Zeilen '57 GTO 00' und  
'LBL 01' sind ebenfalls die obigen 3 Befehle einzufügen.  
Nach 'LBL 01' ist einzufügen: 'CLA'.

Korrektur zu Seite 63:

Steuerzeile 05 muß richtig lauten: 253,27,42,98,56,51,71.



Der Barcodelesestift ist ein preiswertes und doch effizientes Zubehorteil für das Einlesen von Daten oder zum Programmieren. Barcodes (BCs) sind die kostengünstigste Methode der externen Datenspeicherung und deren Massenverbreitung. Über HP-41 BCs, ihre oft verbluffend ergiebige Anwendung, Herstellung und ihren Aufbau ist wenig dokumentiert. Mit diesem Buch schließt der Autor die Lücke.

Das Buch behandelt ausführlich Vor- und Nachteile von BCs und Lesestift, gibt eine Übersicht über alle BC-Typen des 41-er Systems und erläutert den grundsätzlichen Aufbau und das Kodierungsschema. Die Herstellung von BCs auf dem IL-Thermodrucker, dem ThinkJet-Drucker und dem Plotter 7470A werden eingehend beschrieben. Großer Wert wird auf die Verwendung von Barcodes zum Programmieren gelegt. Eine Vielzahl praktischer Anwendungen für das synthetische Programmieren, zur Datenmasseneinlesung für Schriftfahnen oder Querdruck, zur Herstellung beliebiger Textzeilen und zum Umgang mit speziellen LBL, GTO bzw. XEQ-Befehlen werden vorgestellt.

Das Buch enthält über 90 Programme, meist dreifach vorhanden: 1. für Rechner und Thermodrucker ohne Systemmodule; 2. für das XF-Modul und 3. für das Plottermodul.

In einem Anhang werden nicht HP-BCs wie der Europäische Artikelnummern-BC und einige andere Typen vorgestellt. Weiterhin wird ein Ausblick auf die zukünftige Entwicklung gegeben.

Wer die Möglichkeiten seines Lesestiftes mit dem HP-41 System optimal nutzen und selbst BCs anfertigen will, braucht dieses leicht verständlich geschriebene Buch. Es wird ihm bald zur unentbehrlichen Arbeitshilfe werden.