

Neues vom Hypra-Basic

Zehn anwenderbezogene Befehle erwarten Sie diesmal. Sie unterstützen die Menüprogrammierung und das Diskettenhandling.

Auch in dieser Ausgabe sollen Hypra-Basic, dem Listing des Monats aus der Ausgabe 4/86, neue Befehle angehängt werden. Alle Module halten sich diesmal an die Philosophie des Hypra-Basic, in sich geschlossen und frei verschiebbar zu sein. Bei Modul 30 ist zu beachten, daß es sich in den Interrupt hängt. Vor dem Aufrufen einiger Befehle muß der Interrupt ausgeschaltet werden. Es ist übrigens aus dem Sonderheft Tips & Tricks 2/86 entnommen.

Listing 1.
 Modulnummer: 21
 Anzahl der Befehle: 1
 1) SET 49152-49173,0
 Funktion: Setzt den Cursor auf eine bestimmte Position und führt einen Printbefehl aus;
 Syntax: SET Spalte, Zeile, Variablenliste;
 Beispiel: SET 10,12,a\$;"Text";b
 Der Cursor wird auf Spalte 12 und Zeile 10 positioniert und die angehängte Liste gePRINTet.

Ab der nächsten Ausgabe erwarten Sie die ersten Module unserer Leser — darunter könnten auch Sie sich befinden. Denn die besten der uns eingesandten Module werden natürlich veröffentlicht. Dazu werden wir in der nächsten Ausgabe das Erstellen und Einbauen eines eigenen Befehls ausführlich erklären, um Anfängern die Arbeit mit Hypra-Basic zu erleichtern.

(R. Aretz/og)

programm : 21erw .ass c000 c018

```
c000 : 20 9e b7 86 d3 20 fd ae c2
c008 : 20 9e b7 86 d6 20 10 e5 b0
c010 : 20 fd ae 4c a4 aa 00 c0 85
```

Listing 1

programm : 22erw .ass c000 c017

```
c000 : 4c 0b c0 20 79 00 f0 0c 79
c008 : 20 fd ae 20 9e b7 20 ff ff
c010 : e9 18 90 ef 60 00 c0 c0 b2
```

Listing 2

programm : 23erw .ass c000 c055

```
c000 : 20 8a ad 20 f7 b7 a0 00 94
c008 : 20 73 00 f0 06 99 3c 03 24
c010 : c8 d0 f5 a2 00 bd 4d c0 b7
c018 : 99 3c 03 c8 e8 e0 06 d0 f8
c020 : f4 a5 01 85 9b 78 a9 35 66
c028 : 85 01 a0 00 b1 14 85 02 2c
c030 : a5 9b 85 01 58 a5 7b 48 56
c038 : a5 7a 48 a9 3c 85 7a a9 8f
c040 : 03 85 7b 20 a5 a9 68 85 3d
c048 : 7a 68 85 7b 60 b2 c2 28 be
c050 : 32 29 00 00 c0 00 00 00 23
```

Listing 3

programm : 24erw .ass c000 c07f

```
c000 : 20 9e b7 8e 3c 03 20 fd 87
c008 : ae 20 8a ad 20 f7 b7 20 ff
c010 : fd ae 20 9e b7 8e 3d 03 2b
c018 : 20 fd ae 20 9e b7 8e 3e 45
c020 : 03 a5 14 8d 00 d4 a5 15 14
c028 : 8d 01 d4 a9 00 85 c6 a9 3b
c030 : 0f 8d 18 d4 a9 19 8d 05 4a
c038 : d4 a9 44 8d 06 d4 a9 32 b6
c040 : 8d 02 d4 a9 46 8d 03 d4 bf
c048 : ad 3e 03 8d 04 d4 ae 3d a3
c050 : 03 20 72 c0 a9 00 8d 04 f1
c058 : d4 ae 3d 03 20 72 c0 ad 27
c060 : 3c 03 d0 06 a5 c6 f0 c7 f7
c068 : d0 05 ce 3c 03 d0 c0 60 70
c070 : a2 0a a0 00 ea ea c8 d0 0a
c078 : fb ca d0 f6 60 00 c0 00 f4
```

Listing 4

programm : 25erw .ass c000 c05b

```
c000 : 20 9e b7 8a 86 9b 20 fd 70
c008 : ae 20 9e b7 20 70 e5 20 c2
c010 : fd ae 20 9e b7 86 9c 20 a3
c018 : fd ae 20 8a ad 20 f7 b7 f1
c020 : a5 01 85 02 78 a9 35 85 9c
c028 : 01 a0 00 84 9e a4 9b b1 eb
c030 : d1 85 d7 29 3f 06 d7 24 ab
c038 : d7 10 02 09 80 90 04 a6 a3
c040 : d4 d0 04 70 02 09 40 a4 3e
c048 : 9e 91 14 e6 9b e6 9e c6 8a
c050 : 9c d0 da a5 02 85 01 58 c1
c058 : 60 00 c0 03 20 72 c0 ad 3d
```

Listing 5

programm : 26erw .ass c000 c04d

```
c000 : a5 d3 48 a5 d6 48 20 9e c3
c008 : b7 86 d3 20 fd ae 20 9e 0e
c010 : b7 86 d6 20 10 e5 a9 0f b9
c018 : a2 08 a0 0f a8 20 ba ff 3f
c020 : a9 00 20 bd ff 20 c0 ff 8d
c028 : a2 0f 20 c6 ff 20 cf ff 73
c030 : 20 d2 ff a5 90 c9 40 d0 68
c038 : f4 a9 0f 20 c3 ff 20 cc 1f
c040 : ff 68 85 d6 68 85 d3 20 f2
c048 : 10 e5 60 00 c0 e6 9e c6 ae
```

Listing 6

programm : 27erw .ass c000 c0ac

```
c000 : a9 01 85 02 20 9e ad 20 b9
c008 : 82 b7 a6 22 a4 23 20 bd b3
c010 : ff a9 00 85 f7 85 f9 a9 7b
c018 : 04 85 f8 a9 d8 85 fa a5 43
c020 : 02 f0 47 20 fd ae 20 9e 83
c028 : b7 a9 01 a0 01 20 ba ff 04
c030 : 20 c0 ff a2 01 20 c9 ff 3d
c038 : a0 00 b1 f7 20 d2 ff b1 40
c040 : f9 20 d2 ff e6 f7 e6 f9 bc
c048 : d0 04 e6 f8 e6 fa a5 f8 c2
c050 : c9 07 d0 e4 a5 f7 c9 e8 80
c058 : d0 de 20 cc ff a9 01 20 ca
c060 : c3 ff 60 a9 00 85 02 4c 3d
c068 : 04 c0 20 fd ae 20 9e b7 6a
c070 : a9 01 a0 00 20 ba ff 20 da
c078 : c0 ff a2 01 20 c6 ff 00 7a
c080 : 00 20 e4 ff 91 f7 20 e4 ec
c088 : ff 91 f9 e6 f7 e6 f9 d0 eb
c090 : 04 e6 f8 e6 fa a5 f8 c9 76
c098 : 07 d0 e4 a5 f7 c9 e8 d0 08
c0a0 : de 20 c4 ff a9 01 20 c3 c6
c0a8 : ff 60 00 c0 00 00 00 00 f0
```

Listing 7

programm : 28erw .ass c000 c077

```
c000 : 20 9e b7 86 02 20 fd ae a4
c008 : 20 9e b7 8a d0 02 a2 ff 5e
c010 : 86 f7 a2 00 20 73 00 f0 ba
c018 : 06 9d 3c 03 e8 d0 f5 a9 9d
c020 : b2 9d 3c 03 e8 a9 22 9d b0
c028 : 3c 03 e8 86 f8 a6 02 20 fe
c030 : c6 ff 20 cf ff c9 0d f0 5c
c038 : 0c a6 f8 9d 3c 03 e8 86 16
c040 : f8 c6 f7 d0 ed 20 cc ff c6
c048 : a6 f8 a0 00 b9 70 c0 9d f0
c050 : 3c 03 e8 c8 c0 05 d0 f4 c2
c058 : a5 7b 48 a5 7a 48 a9 3c 8b
c060 : 85 7a a9 03 85 7b 20 a5 ed
c068 : a9 68 85 7a 68 85 7b 60 57
c070 : 22 aa 22 22 00 00 c0 20 f8
```

Listing 8

programm : 29erw .ass c000 c00d

```
c000 : 20 8a ad 20 f7 b7 84 37 92
c008 : 85 38 4c 44 a6 00 c9 7f d6
```

Listing 9

programm : 30erw .ass c000 c09e

```
c000 : 4c 82 c0 78 ad 00 dc 29 6d
c008 : 01 d0 03 ce 01 d0 ad 00 59
c010 : dc 29 02 d0 03 ee 01 d0 69
c018 : ad 00 dc 29 04 d0 03 ce 92
c020 : 00 d0 ad 00 dc 29 08 d0 cd
c028 : 03 ee 00 d0 ad 10 d0 c9 ef
c030 : 01 d0 1c ad 00 d0 c9 40 84
c038 : d0 03 ce 00 d0 ad 00 d0 5a
c040 : c9 00 d0 08 ce 00 d0 a9 c2
c048 : 00 8d 10 d0 4c 68 c0 ad 93
c050 : 00 d0 c9 16 d0 03 ee 00 ce
c058 : d0 ad 00 d0 c9 ff d0 08 09
c060 : a9 01 8d 00 d0 8d 10 d0 49
c068 : ad 01 d0 c9 e6 d0 03 ce a2
c070 : 01 d0 ad 01 d0 c9 31 d0 27
c078 : 03 ee 01 d0 4c 31 ea 4c df
c080 : 03 c0 78 ad 80 c0 8d 14 23
c088 : 03 ad 81 c0 8d 15 03 58 18
c090 : 60 78 a9 31 8d 14 03 a9 96
c098 : ea 8d 15 03 58 60 00 00 77
```

Listing 10

Listing 2.

Modulnummer: 22
 Anzahl der Befehle: 1
 1) CLEARZ 49152-49172,0
 Funktion: Löscht angegebene Bildschirm-Zeilen;
 Syntax: CLEARZ nr (,nr,nr...)
 Beispiel: CLEARZ 14,15
 Die Bildschirmzeilen mit den Nummern 14 und 15 werden gelöscht. Die oberste Zeile hat hier die Nummer 0.

Listing 3.

Modulnummer: 23
 Anzahl der Befehle: 1
 1) RAM 49152-49228,6
 Funktion: Liest Wert aus dem RAM und weist diesen einer Variablen zu;
 Syntax: RAM adresse,variable
 Beispiel: RAM 40965,a
 In diesem Beispiel wird der Inhalt der Adresse 40965 (ähnlich PEEK) in die Variable a geschrieben. Sie haben somit die Möglichkeit auf das »versteckte« RAM direkt zuzugreifen, da sich dieser Befehl ausschließlich auf den RAM-Bereich bezieht.

Listing 4.

Modulnummer: 24
 Anzahl der Befehle: 1
 1) BEEP 49152-49263,13
 Funktion: Erzeugt einen Signalton.
 Syntax: BEEP n,frequenz,tondauer,wellenform
 Beispiel: BEEP 10,8000,50,33
 n = wie oft soll der Ton gespielt werden.
 n = 0 Ton wird so lange wiederholt, bis eine Taste gedrückt wird. Die gedrückte Taste kann mit dem GET-Befehl erkannt werden.
 8000 = Frequenzparameter laut Handbuch.
 50 = Tondauer: Je größer desto länger
 Wellenform: 33 = Sägezahn
 17 = Dreieck
 65 = Rechteck
 129 = Rauschen

Listing 5.

Modulnummer: 25
 Anzahl der Befehle: 1
 1) ZEILBS 49152-49240,0
 Funktion: Zeile vom Bildschirm holen, in ASCII-Code wandeln und im Speicher ablegen.
 Syntax: ZEILBS spalte,zeile,länge,adresse
 Beispiel: ZEILBS 5,5,10,41955
 Der Text in Zeile 5, ab Spalte 5, bis Spalte 15 wird nach ASCII gewandelt und in Speicher geschrieben. Auch hier kann wieder der gesamte RAM-Bereich genutzt werden.

Listing 6.

Modulnummer: 26
 Anzahl der Befehle: 1
 1) ERCHAN 49152-49226,0
 Funktion: Liest den Fehlerkanal der Floppy und zeigt die Meldung auf dem Bildschirm an.
 Syntax: ERCHAN spalte,zeile
 Beispiel: ERCHAN 0,10
 Der Fehlerkanal der Floppy wird gelesen und der String ab Spalte 0, Zeile 10 auf dem Bildschirm ausgegeben.

Listing 7.

Modulnummer: 27
 Anzahl der Befehle: 2
 1) SCRNSV 49152-49250,0
 Funktion: Speichert den aktuellen Bildschirminhalt inklusive Farbe auf Kassette oder Diskette ab.
 Syntax: SCRNSV »name«,gerätenummer
 Beispiel: SCRNSV »bild«,8
 Speichert Bildschirm auf Diskette ab.
 2) SCRNLD 49251-49321,0
 Funktion: Gegenstück zu (1). Laden eines mit SCRNSY gespeicherten Bildschirms.
 Syntax: SCRNLD »name«,geräteummer
 Beispiel: SCRNLD »bild«,8
 Lädt einen Bildschirminhalt von Diskette.

Listing 8.

Modulnummer: 28
 Anzahl der Befehle: 1
 1) DISKIN 49152-49263,5
 Funktion: Holt bestimmte Anzahl von Bytes aus einem File, das vorher geöffnet werden muß, in eine Stringvariable. Der Befehl ist schneller als die entsprechende Basic-Schleife.
 Syntax: DISKIN lfnr,anz,var\$
 Beispiel: DISKIN 1,20,a\$
 Es werden 20 Byte aus dem File mit der logischen Filenummer 1 geholt. In Basic müßte hier eine Schleife 20mal durchlaufen werden, in der bei jedem Durchlauf ein GET #-Statement ausgeführt werden müßte. Diese 20 Bytes kommen in die Variable a\$. Es können maximal 150 Byte geholt werden. Die Eingabe stoppt automatisch, wenn ein chr\$(13) vom Laufwerk gesendet wird.

Listing 9.

Modulnummer: 29
 Anzahl der Befehle: 1
 1) EOB 49152-49164,0
 Funktion: Basic-Ende heruntersetzen (End of Basic).
 Syntax: EOB wert
 Beispiel: EOB 32768
 Der Bereich ab 32768 (\$ 8000) wird vor Überschreiben durch Basic geschützt. Dadurch wird der Basic-Speicher begrenzt. Der Parameter muß zwischen 2051 und 40960 liegen. Der Wert wird nicht überprüft.

Listing 10.

Modulnummer: 30
 Anzahl der Befehle: 2
 1) JSEIN 49152-49296,0
 Funktion: Sprite 1 wird interruptgesteuert über einen Joystick an Port 2.
 Syntax: JSEIN
 Beispiel: JSEIN
 Sprite 1 wird nun durch den Joystick am Port 2 kontrolliert. Seine zugehörigen Parameter, wie Farbe, Aussehen, Priorität etc. bleiben unbeeinflusst.
 2) JSAUS 49297-49309,0
 Funktion: Sprite 1 wird vom Joystick wieder abgekoppelt, der Interrupt normalisiert.
 Syntax: JSAUS
 Beispiel: JSAUS
 Der Joystick beeinflusst das Sprite nicht mehr, der IRQ steht wieder auf \$ EA31.