



Der Autor des
»Mathematical Basic«
stellt sich vor

Ich bin am 9.7.1963 geboren und zur Zeit Praktikant in einem Softwarehaus in Bad Kreuznach.

Für die Entwicklung des Programms ist hauptsächlich der Ärger über das permanente Fehlen der gebräuchlichsten naturwissenschaftlichen Funktionen in jedem Basic-Interpreter von Commodore verantwortlich.

Um mehrere Funktionen über den USR-Vektor zu implementieren, müßte man vor jedem Aufruf auch lästigerweise die jeweilige Adresse in diesen Vektor schreiben. Das ist sehr umständlich; also müssen mehrere USR-Funktionen her. Wenn nun aber schon eine Routine für das Selektieren der Adressen geschrieben wird, so kann man auch gleich das Ganze etwas komfortabler gestalten und eine Interpreter-Erweiterung basteln. Das waren meine anfänglichen Überlegungen und heraus kam »Mathematical Basic«. Das Programm wurde so gehalten, daß es jederzeit weitere Befehle und Funktionen aufnehmen kann. 6502-Freunde mit Kenntnissen über das VC 20-Betriebssystem werden sich schnell mit »Mathematical Basic« zurechtfinden und auch eigene Änderungen durchführen können.

(Wolfgang W. Wirth)

(ev)

MATHEMATICAL-BASIC:

DAS SUPER-BASIC FÜR DEN VC 20

Einen echten Knüller für alle Besitzer eines VC 20 mit 8 KByte Erweiterung haben wir zum »Listing des Monats« gewählt. Mathematical Basic stellt über 50 neue Befehle und Funktionen zur Verfügung und setzt damit einen Standard, an dem andere Toolkits künftig gemessen werden müssen.

Das Programm ist vollständig in Assembler geschrieben und liegt zum Eintippen als Basic-Lader vor. Nach dem Starten des Laders wird das eigentliche Maschinenprogramm ab Adresse 29211, also am oberen Ende von Speicherblock 3 abgelegt und gleichzeitig vor dem Überschreiben durch Basic-Programme oder Variable geschützt. Bei einem voll ausgebauten VC 20 (+ 24 KByte) kann das Programm ohne weitere Veränderungen benutzt werden. Aber auch Besitzer einer 8-KByte-Erweiterung können diese Basic-Erweiterung nutzen. Allerdings muß dann diese 8-KByte-Erweiterung hardwaremäßig auf den Adreßbereich von \$6000 bis \$7FFF, also auf Speicherblock 3 umgeschaltet wer-

den. Schlagen Sie bitte die erforderliche Vorgehensweise im Handbuch für die 8-KByte-Erweiterung nach. Es ist die gleiche Umstellung, die auch nötig wird, wenn Sie das 8-KByte-Modul zusammen mit einer 16-KByte-Erweiterung als 24 KByte RAM nutzen wollen.

Haben Sie allerdings keine zusätzliche 16-KByte-Erweiterung, dann steht Ihnen für das Arbeiten mit Mathematical-Basic nur der Grundversionsspeicher zur Verfügung. Leider ist es nicht möglich, nur das 16-KByte-Modul zu verwenden, da dieses immer auf Speicherblock 1 und 2 eingestellt ist. Doch nun zum Programm selbst: Mathematical-Basic erweitert den

Wortschatz des Basic-Interpreters um zusätzliche Befehle und Funktionen. Insbesondere wurde die Handhabung der trigonometrischen Funktionen verbessert (man kann zwischen Altgrad, Neugrad und Bogenmaß wählen). Daneben wurden Spezialbefehle für Diskettenbetrieb wie CATALOG, DLOAD, DSAVE etc. eingebaut. Strukturierte Programmierung wird durch DO...UNTIL-Schleifen unterstützt. Viele weitere Befehle und Funktionen vereinfachen die Programmierung zum Teil ganz erheblich.

Auf jeden Fall braucht »Mathematical-Basic« den Vergleich auch mit kommerziellen Softwareprodukten dieser Art nicht zu scheuen.

```

2820 data16,d4,68,8d,0f,d4,68,8d
2830 data0e,d4,ac,ee,07,c8,8c,12
2840 datad4,68,8d,01,d0,68,8d,00
2850 datad0,a9,09,8d,15,d0,ba,e4
2860 data02,f0,3d,68,8d,08,d4,68
2870 data8d,07,d4,ac,ee,07,c8,8c
2880 data0b,d4,68,8d,03,d0,68,8d
2890 data02,d0,a9,0b,8d,15,d0,ba
2900 datae4,02,f0,1c,68,8d,01,d4
2910 data68,8d,00,d4,ac,ee,07,c8
2920 data8c,04,d4,68,8d,05,d0,68
2930 data8d,f0,3d,a9,0f,8d,15,d0
2940 data4c,00,33,a2,a6,86,fc,a2
2950 data30,86,fd,ad,ee,07,29,04
2960 dataf0,09,a9,10,8d,ee,07,85
2970 data02,10,2f,ad,ee,07,29,70
2980 datad0,0d,a9,d8,85,fc,a9,14
2990 data8d,ee,07,a0,0a,10,2b,ad
3000 dataee,07,0a,8d,ee,07,85,02
3010 datac9,40,f0,04,a9,00,f0,02
3020 dataa9,0a,a2,27,9d,58,da,ca
3030 data10,fa,a5,fc,18,69,0a,85
3040 datafc,a5,02,0a,85,02,90,f2
3050 dataa0,0a,b1,fc,99,97,06,88
3060 datad0,f8,4c,bd,32,a2,0d,bd
3070 datae8,06,29,7f,9d,e8,06,ca
3080 data10,f5,ae,ef,07,e0,0f,d0
3090 data03,20,ab,35,e0,7f,d0,05
3100 data20,ab,35,a2,ff,8a,18,69
3110 data10,8d,ef,07,8d,18,d4,0a
3120 data85,02,a2,06,86,fd,a2,e8
3130 data86,fc,a2,03,a5,02,0a,85
3140 data02,90,0b,a0,03,b1,fc,09
3150 data80,91,fc,88,10,f7,a5,fc
3160 data18,69,05,85,fc,ca,d0,e4
3170 data4c,bd,32,ad,f2,07,49,07
3180 data8d,f2,07,8d,17,d4,a0,6e
3190 datab9,f8,da,49,0e,99,f8,da
3200 data88,10,f5,60,4d,41,52,54
3210 data49,4e,20,41,48,4c,42,4f
3220 data52,4e,20,33,34,31,38,20
3230 data55,53,4c,41,52
4000 ende

```

ready. Listing »Synthesizer« (Schluß)

```

10 *save-routine für c-64 synthesizer
20 *bitte im direktmodus eingeben,
30 *also ohne zeilennummern
40 *
100 poke 781,geraetenummer (1 bzw 8)
110 sys 65466
120 nm$="synthesizer-obj"
130 poke183,len(nm$)
140 poke781,681and255
150 poke782,681/256
160 forc=1to15:poke680+c,asc(mid$(nm$,c)
):next
170 poke250,12032and255
180 poke251,12032/256 *startadresse
190 poke780,250
200 poke781,13790and255
210 poke782,13790/256 *endadresse
220 sys65496 *save

```

ready.

Listing »Save-Routine« für »Synthesizer«

Mathematical Basic

Fortsetzung von Seite 50

Eine gute Nachricht für alle Freunde des VC 20: Unser »Listing des Monats« macht mit über 50 neuen Befehlen das Programmieren zum Vergnügen.

Leider gibt es aber auch eine schlechte Nachricht. Sie müssen gut und gerne 150 DATA-Zeilen eintippen. Lassen Sie sich aber davon nicht entmutigen. Der Aufwand lohnt sich ganz bestimmt. Das Programm hilft Ihnen bei der wohl unvermeidlichen Suche nach Tippfehlern durch das blockweise Bilden von Prüfsummen.

⋮: Befehle des Mathematical Basic

erlaubt eine REM-Anweisung innerhalb einer Zeile ohne sofortige Programmfortsetzung ab folgender Zeile.

BEEP h, l:

gibt einen Ton variabler Höhe (h) und Länge (l) aus. h ist im Bereich von 0 bis 126 und l von 0 bis 255 definiert. Beträgt l=0, so wird kein Ton ausgegeben.

CATALOG:

bringt das Inhaltsverzeichnis der Diskette auf den Bildschirm. Dieser Befehl ist nur für das Gerät mit der Nummer 8 definiert. Wird während der Ausgabe die STOP-Taste betätigt, so wird die Ausgabe sofort abgebrochen. Wird die Leertaste gedrückt, so wird die Ausgabe nur unterbrochen und kann mit einer weiteren Betätigung dieser Taste fortgesetzt werden.

COLOR

c,h,r: setzt die Farben für Cursor (c), Hintergrund (h) und Rahmen (r). r und c erstrecken sich von 0 bis 7 und h von 0 bis 15. Die einzelnen Farben bezüglich der Nummerncodes können dem VC 20-Handbuch entnommen werden.

CLS:

löscht den Bildschirmspeicher.

DEFUSRn TO x:

definiert einen der neun möglichen USR-Vektoren. Die einzelnen USR-Funktionen werden durch das Zeichen n unterschieden. n kann dabei A, B, C, D, E, F, G oder H sein. Die neunte USR-Funktion ist die, welche über den Vektor in den Adressen 1 und 2 angesprochen wird. Soll dieser Vektor definiert werden, so muß n weggelassen werden. Die Variable x steht für den Vektor selbst, also der Adresse, ab der die USR-Funktion starten soll. Beispiel: DEFUSRA TO 30000. Die USR-Funktion USRA erhält die Einsprungadresse 30000.

DEGREE:

stellt die Routinen für trigonometrische Funktionen auf Normalgrad (0 bis 360) ein.

DELETE a-b:

löscht Zeilen eines Basic-Programms von Zeile a bis Zeile b. Für a und b sind nur Konstanten erlaubt.

DIRECTORY:

siehe CATALOG

DLOAD pn\$:

lädt ein Programm mit dem Namen pn\$ vom Floppy-Laufwerk mit der Nummer 8. Außer dem Namen sind keine weiteren Parameter erlaubt.

DO:

erlaubt zusammen mit UNTIL eine Schleife. Der zwischen diesen beiden Schlüsselwörtern liegende Programmteil wird so oft wiederholt, bis der Ausdruck, der UNTIL folgt, nicht mehr auf logisch 0 ist. Dazu ein Beispiel: DO GET a\$: UNTIL a\$="" ". Das Programm verläßt die Schleife nicht eher, bis die Leertaste gedrückt wird. DO-UNTIL-Schleifen können acht Ebenen

tief geschachtelt werden. Eine neunte DO-Anweisung würde die Meldung »out of memory« zur Folge haben. Sollte UNTIL einmal ohne einen vorangegangenen DO-Befehl aufgerufen werden, so gibt der VC 20 »until without do« aus.

DSAVE pn\$:

sichert ein Programm mit dem Namen pn\$ auf dem Floppy-Laufwerk mit der Nummer 8. Außer dem Namen sind keine weiteren Parameter zulässig. Sollte dem Befehl DSAVE, DLOAD oder DVERIFY kein Parameter folgen, so wird automatisch »*« als Parameter gesetzt.

DVERIFY pn\$:

vergleicht das Programm mit dem Namen pn\$ im Speicher mit dem gleichen auf der Diskette. Als Parameter ist nur der Name erlaubt.

EXECUTE a\$:

wandelt den String a\$ in Interpretercode und führt die darin enthaltenen Anweisungen aus. Beispiel: a\$="a=22":EXECUTE a\$. Der Variablen a wird der Wert 22 zugewiesen. Der Befehl darf nicht im Direkt-Modus gegeben werden und der Befehlsstring (a\$) darf eine Länge von 88 Zeichen nicht überschreiten.

GRAD:

stellt die trigonometrischen Routinen auf Neugrad (0 bis 400).

IF:

Die IF..THEN-Anweisung ist in ihren Aufbaumöglichkeiten erweitert worden. Es ist nicht mehr zwingend erforderlich, den THEN-Dummy zu schreiben. Beispiel: IF A=4 THEN PRINT B\$ läßt sich auch als IF A=4 PRINT B\$ eingeben.

INITIALIZE:

initialisiert die Floppy mit der Nummer 8.

LOCATE z,s:

setzt den Cursor in Zeile z und Spalte s. z liegt im Bereich von 1 bis 23 und s von 1 bis 22. Folgende Variationsmöglichkeiten sind gegeben: LOCATE z definiert nur eine neue Zeilenposition. LOCATE ,s setzt den Cursor innerhalb einer Zeile nur an eine neue Spaltenposition.

LPRINT:

verhält sich wie PRINT. Die Zeichen werden aber nicht auf den Bildschirm, sondern auf den Drucker (Nummer 4) ausgegeben. LPRINT und PPRINT sollten nur bei angeschlossenen Geräten benutzt werden, da sonst der IEC-Bus blockiert wird.

RADIAN:

stellt die trigonometrischen Routinen auf Bogenmaß ein.

RENUMBER z,s:

reorganisiert die Zeilennummern eines Basic-Programms. z ist dabei die Startzeile und s die Schrittweite. z und s dürfen nur Konstanten sein. Sollten dem Befehl keine Parameter folgen, so gilt z = 100 und s = 10.

RESTORE TO z:

positioniert den DATA-Pointer auf die Zeile z. Das Wort TO kann dabei wegfallen. RESTORE allein setzt den Pointer wie gewohnt auf den Programmstart.

RETURN TO z:

ermöglicht es, aus einem Unterprogramm in eine bestimmte Zeile z zurückzukehren.

RUN "name":

hat die gleiche Funktion wie DLOAD. Zusätzlich wird aber noch der String RUN +CHR\$(13) in den Tastaturpuffer geschrieben. Das hat zur Folge, daß das gerade geladene Programm sofort gestartet wird.

PPRINT:

verhält sich wie PRINT. Die Zeichen werden aber, anstatt auf den Bildschirm, auf den Plotter (Nummer 6) ausgegeben.

QUIT:

führt ein Total-Reset aus. Mathematical Basic wird dabei gelöscht.

UNITL a: siehe DO.

Die Funktionen des Mathematical Basic

! a:

ist ein Äquivalent zur CHR\$(a)-Funktion. Hierbei sind aber keine Klammern nötig. a ist nur als Konstante erlaubt.

&:

ist das »Hexadezimal-Vorzeichen« für Hex-Konstanten. Beispiel: &a02b ist das Äquivalent für 41003. Die Anzahl der Hex-Ziffern ist beliebig, es wird maximal eine 16-Bit-Zahl errechnet.

←:

entspricht dem Ausdruck CHR\$(13) und kann auch genauso gehandhabt werden.

[a]:

hat die gleiche Wirkung wie die ABS-Funktion. Unterscheidet sich aber durch die bessere Selbstdokumentation.

ACS(a):

berechnet abhängig von der jeweiligen Einstellung durch RADIAN, DEGREE oder GRAD den Arcus-Cosinus von a. Die hier angesprochene Abhängigkeit gilt für alle trigonometrischen Funktionen.

ACT(a):

bestimmt den Arcus-Cotangens von a.

ASN(a):

ergibt den Arcus-Sinus von a.

CHR\$(z,l):

ist eine Variante von CHR\$(z). z entspricht dabei dem Zeichen des normalerweise nur 1 Byte langen Strings. Mit l läßt sich nun aber zusätzlich auch die Länge variieren. l ist dabei von 0 bis 255 definiert.

COT(a):

berechnet den Cotangens von a.

CRSCOL:

holt die momentane Spaltenposition des Cursors.

CRSLIN:

holt die momentane Zeilenposition des Cursors.

CVF(a\$):

wandelt den fünf Zeichen langen String a\$ in eine Fließkommazahl um.

CVI(a\$):

wandelt den zwei Zeichen langen String a\$ in eine Integerzahl um.

DEC(a\$):

berechnet aus dem 4-Byte-String a\$, der sich aus Hex-Ziffern aufbaut, eine dezimale 16-Bit-Zahl.

DIV(a,b):

ist gleichwertig mit dem Ausdruck INT(a/b).

FRC(a):

holt die Nachkommazahl von a.

FUNCTION(a\$):

wandelt den maximal 88 Zeichen langen String a\$ in Interpretercode und berechnet ihn. Beispiel: y=FUNCTION("5+2"). y wird der Wert 7 zugewiesen. Diese Funktion darf nicht im Direktmodus stehen.

HEX\$(a):

wandelt die 16-Bit-Zahl a in einen 4-Byte-Hex. String.

INSTR(a\$,b\$):

testet, an welcher Stelle a\$ sich in b\$ befindet. Ist a\$ nicht in b\$ enthalten, so ist das Ergebnis 0, ansonsten entspricht es der Position von a\$ in b\$. a\$ und b\$ müssen mindestens 1 Zeichen lang sein und a\$ darf nicht länger als b\$ sein. INSTR darf nicht im Direktmodus stehen.

LGD (a):

berechnet den dekadischen Logarithmus von a.

LGU(a,b):
bestimmt den Logarithmus von a zur Basis b.

MKF\$(a):
wandelt die Zahl a in einen fünf Zeichen langen String.

MKI\$(a):
wandelt die Integerzahl a in einen zwei Zeichen langen String.

MOD(a,b):
bestimmt den ganzzahligen Rest aus der Division von a durch b.

RANDOM:
entspricht dem Ausdruck RND(1).

RANDOM(a,b):
entspricht dem Ausdruck RND(1)*b+a.

TIMES:
ist eine modifizierte Form von TI\$. Der Unterschied liegt darin, daß bei TIMES noch zwei Trennzeichen eingefügt werden.

USRn:
entspricht im Prinzip der USR(a)-Funktion. Unterschiede liegen darin, daß n (was der Kennung A, B, C, D, E, F, G oder H entspricht) folgen muß, aber keine »Klammer-auf-Klammer-zu«-Sequenz. Diese Routine (Aufruf durch JSR\$cef1) muß zusätzlich an USR-Routinen, die für die Standard-USR-Funktion ausgelegt sind, angehängt werden. Der Vorteil bei den USRn-Funktionen liegt nun darin, daß man den USR-String mit mehreren Parametern übergeben kann. Die dazu nötigen Routinen sind:

cefa	Test auf Klammer auf und nächstes Zeichen holen.
cd9e	Ausdruck holen und String- und Integer-Flag setzen.
cd8d	Test auf numerischen Ausdruck.
cd8f	Test auf String.
cef7	Test auf Klammer zu und nächstes Zeichen holen.
cefd	Test auf Komma und nächstes Zeichen holen.

Damit sind alle neuen Befehle und Funktionen vorgestellt. Bleibt nur noch, Ihnen viel Spaß beim Programmieren mit »Mathematical Basic« zu wünschen. (Wolfgang W. Wirth/ev)

```

100 rem-----
101 rem" * * *      M A T H E M A T I C
    AL  B A S I C  V1.02      * * * "
102 rem-----
103 rem" createt in october 1984 by W.Wi
rth,Th. Heuss Ring 20,6556 Woellstein "
104 rem-----
105 rem
106 rem
107 rem
108 clr:if peek(55)+peek(56)*256>29211 t
hen poke 55,27:poke56,114:clr
109 poke 36879,14:print chr$(147);chr$(1
3);chr$(14);chr$(5);spc(17);chr$(34)
110 dim cs(14)
111 rem
112 print"   Ladeprogramm fur"
113 print
114 print" ";chr$(18);" Math. Basic V1.
02 "
115 print
116 print" Das Programm wird"
117 print
118 print" jetzt abgelegt."
119 print
120 print
121 rem
122 rem----- hilfs

```

```

routine ablegen -----
123 rem
124 poke 0,76
125 poke 1,12
126 poke 2,2
127 rem
128 for i=512 to 543
129 : read j
130 : cs=cs+j
131 : poke i,j
132 next
133 rem
134 if cs=2635 then 148
135 rem
136 print
137 print
138 print" Datafehler im"
139 print
140 print" Hilfsrout.-Block !":end
141 rem
142 rem----- hilfs
routinenblock -----
143 rem
144 data136,177,34,201,58,144,2,233,8,23
3
145 data47,96,32,130,215,32,,2,133,102,3
2,
146 data2,10,10,10,10,5,102,76,148,215
147 rem
148 rem----- haupt
programm ablegen -----
149 rem
150 for i=0 to 3556
151 : print i;chr$(145)
152 : read j$:l=usr(j$)
153 : cs(i/240)=cs(i/240)+1:poke 29211+
i,1
154 next
155 rem
156 rem----- fehle
rerkennung -----
157 rem
158 for i=0 to 14
159 : read cs:if cs(i)=cs then 163
160 : print" Fehler im Block von"
161 : print" Zeile";171+i*10;"bis";180+
i*10:print
162 : print:ef=1
163 next
164 rem
165 if ef then end
166 sys 30414
167 rem
168 rem----- mathe
matical basic v1.02 -----
169 rem
170 data48,d2,48,d2,48,d2,48,d2,48,d2,48
,d2,48,d2,48,d2,20,f1,ce,20,82,d7,c9,04
171 datad0,62,88,b1,22,a2,0f,dd,73,73,f0
,05,ca,10,f8,30,53,8a,ea,ea,ea,ea,a2,04
172 data6a,66,62,66,63,ca,d0,f8,88,10,e0
,a2,90,38,4c,49,dc,20,f1,ce,a5,14,48,a5
173 data15,48,20,f7,d7,a9,04,20,7d,d4,a8
,a5,14,20,81,72,a5,15,20,81,72,68,85,15
174 data68,85,14,4c,fb,d6,48,29,0f,20,8c
,72,68,4a,4a,4a,4a,aa,bd,73,73,88,91,62
175 data60,4c,58,d6,4c,48,d2,20,a6,d3,20
,fa,ce,20,9e,cd,20,a3,d6,c9,00,f0,ed,48

```

Listing »Mathematical Basic«

176 data8a,48,98,48,20,fd,ce,20,9e,cd,20
 ,a3,d6,85,60,aa,f0,da,20,f7,ce,68,85,65
 177 data68,85,64,68,85,66,a5,60,c5,66,90
 ,c5,a9,00,85,0d,a2,01,a0,ff,c8,c4,66,d0
 178 data06,8a,4c,94,d7,a0,00,b1,22,d1,64
 ,f0,ef,e6,22,d0,02,e6,23,e8,c6,60,a5,60
 179 datac5,66,b0,e9,4c,f7,d8,20,73,00,20
 ,8a,cd,4c,f7,d7,4c,08,cf,20,f1,ce,6c,01
 180 data00,c9,28,f0,f6,38,e9,41,c9,08,b0
 ,ec,0a,69,1b,8d,21,73,20,73,00,6c,00,72
 181 data4c,b3,d3,c9,b7,d0,f9,20,73,00,c9
 ,a4,d0,08,20,fa,72,84,01,85,02,60,38,e9
 182 data41,c9,08,b0,c3,0a,48,20,73,00,c9
 ,a4,d0,ba,20,fa,72,68,aa,a5,14,9d,1b,72
 183 dataa5,15,9d,1c,72,60,08,c9,22,f0,03
 ,4c,20,77,a2,04,78,86,c6,bd,f8,ed,9d,76
 184 data02,ca,d0,f7,58,4c,58,7a,30,31,32
 ,33,34,35,36,37,38,39,41,42,43,44,45,46
 185 dataa9,00,85,62,85,63,20,73,00,a0,0f
 ,d9,73,73,f0,06,88,10,f8,4c,56,72,98,2a
 186 data2a,2a,2a,a2,04,2a,26,63,26,62,ca
 ,d0,f8,f0,df,00,00,ee,0a,0a,ee,0a,0a,ee
 187 data0a,0a,ee,0a,0a,ee,0a,0a,ee,0a,0a
 ,ee,0a,0a,00,c9,26,f0,bd,c9,5b,d0,06,20
 188 data73,00,4c,57,7e,c9,21,d0,08,48,48
 ,20,73,00,4c,5c,7d,c9,5f,f0,03,4c,92,c8
 189 data48,48,20,73,00,4c,75,7d,20,fa,ce
 ,20,9e,d7,a8,8a,48,c0,2c,d0,05,20,f1,d7
 190 data8a,ac,a9,01,20,7d,d4,20,f7,ce,68
 ,a0,00,f0,03,91,62,c8,c4,61,90,f9,4c,fb
 191 datad6,a2,06,ac,a2,04,a0,60,20,50,fe
 ,20,14,f3,20,9d,ca,ea,ea,4c,b7,cb,a2,04
 192 databd,4e,76,9d,a0,02,ca,d0,f7,60,20
 ,79,00,f0,13,a0,00,20,4e,74,f0,0c,c9,2c
 193 datad0,6f,20,73,00,20,4e,74,d0,67,60
 ,b0,64,20,6b,c9,a5,14,99,a1,02,c8,ea,ea
 194 dataea,a5,15,99,a1,02,c8,4c,79,00,20
 ,79,00,f0,4a,08,a0,00,20,b7,74,48,20,13
 195 datac6,68,28,f0,07,c9,ab,d0,38,20,73
 ,00,08,a0,02,20,b7,74,d0,2d,28,d0,04,c6
 196 data15,c6,af,a5,5f,85,7a,a5,60,85,7b
 ,20,13,c6,a0,01,90,0a,b1,5f,aa,88,b1,5f
 197 data85,5f,86,60,a5,ae,c5,ac,a5,af,e5
 ,ad,ea,ea,ea,b0,99,4c,08,cf,20,6b,c9,a6
 198 data14,96,ac,a6,15,96,ad,4c,79,00,20
 ,29,74,20,35,74,20,b3,75,20,c0,75,f0,0d
 199 data20,0a,76,b0,05,20,1a,76,f0,f1,4c
 ,48,d2,20,b3,75,20,c0,75,d0,2c,20,b3,75
 200 data20,c0,75,f0,12,a0,02,b9,ab,00,91
 ,7a,88,d0,f8,20,0a,76,20,1a,76,f0,e9,20
 201 datac4,75,a5,7a,85,2d,a5,7b,20,57,c6
 ,20,33,c5,4c,74,c4,20,c0,75,a9,80,85,0f
 202 dataa9,80,2c,a9,40,45,0f,85,0f,20,73
 ,00,a8,f0,b9,c9,22,f0,ed,a6,0f,d0,f2,c9
 203 data8f,f0,e8,a2,08,dd,55,76,f0,05,ca
 ,d0,f8,f0,e2,20,cf,75,20,73,00,a2,03,dd
 204 data52,76,f0,f3,ca,d0,f8,20,79,00,b0
 ,d0,20,6b,c9,20,b3,75,20,c0,75,f0,13,20
 205 datac0,75,c5,15,d0,04,c4,14,f0,0e,20
 ,0a,76,20,1d,76,f0,e8,a0,f9,a9,ff,d0,04
 206 dataa4,ad,a5,ac,20,23,76,20,fe,75,86
 ,ae,20,73,00,e6,ae,a1,ae,f0,17,90,05,a0
 207 dataff,20,d9,75,a1,ae,81,7a,20,c4,75
 ,c9,3a,b0,e8,20,98,e3,10,e3,20,79,00,b0
 208 data9c,a0,01,20,d9,75,f0,f4,a2,02,bd

,a0,02,95,ab,ca,d0,f8,4c,8e,c6,20,c4,75
 209 dataa8,e6,7a,d0,02,e6,7b,a2,00,a1,7a
 ,60,a2,02,b5,79,95,59,ca,d0,f9,60,20,cf
 210 data75,a9,03,85,c2,b1,7a,d0,02,e6,c2
 ,c8,85,c1,b1,7a,48,a5,c1,81,7a,f0,04,a9
 211 data04,85,c2,20,c4,75,68,c6,c2,d0,e9
 ,a2,02,b5,59,95,79,86,af,ca,d0,f7,60,a2
 212 datafe,18,b5,ae,7d,a5,01,95,ae,e8,d0
 ,f6,c9,fa,60,20,c0,75,20,c4,75,d0,fb,60
 213 data84,62,85,63,a2,90,38,20,49,dc,4c
 ,dd,dd,20,65,74,a0,00,b1,5f,91,7a,e6,5f
 214 datad0,02,e6,60,20,c4,75,a5,5f,c5,2d
 ,a5,60,e5,2e,90,e9,4c,05,75,64,00,0a,00
 215 dataab,a4,2c,9b,8a,a7,89,8d,cb,8c,8e
 ,00,00,00,00,00,00,a5,90,f0,05,68,68,4c
 216 datade,f6,20,e1,ff,f0,f6,20,f9,f1,c9
 ,20,d0,05,20,f9,f1,f0,fb,4c,19,ef,a0,00
 217 dataa2,08,84,90,20,52,fe,a9,01,a2,60
 ,a0,c3,20,49,fe,20,9d,f4,a9,08,20,b4,ff
 218 dataa9,00,20,96,ff,20,64,76,20,64,76
 ,20,64,76,20,64,76,20,64,76,85,63,20,64
 219 data76,85,62,20,d1,dd,20,64,76,aa,f0
 ,06,20,42,e7,4c,b9,76,20,d7,ca,20,d7,ca
 220 data4c,a6,76,a2,ff,78,9a,d8,20,8d,fd
 ,a2,1b,a0,72,cc,84,02,90,07,d0,08,ec,83
 221 data02,b0,03,20,7b,fe,20,a9,7e,20,a4
 ,e3,a5,2b,a4,2c,20,08,c4,a9,98,a0,7d,20
 222 data0f,e4,4c,81,e3,a2,08,a0,0f,20,52
 ,fe,a9,01,a2,72,a0,c3,20,49,fe,4c,9d,f4
 223 data20,26,77,4c,60,c6,20,26,77,4c,e9
 ,76,00,20,26,77,4c,72,c8,48,a9,00,8d,5e
 224 data77,8d,69,79,68,60,ad,5e,77,f0,4a
 ,20,9e,cd,a5,61,f0,04,ce,5e,77,60,ae,5e
 225 data77,bd,98,77,85,39,bd,a0,77,85,3a
 ,bd,a8,77,85,7a,bd,b0,77,85,7b,d0,20,4c
 226 data35,c4,a2,00,e0,08,b0,f7,a5,39,9d
 ,99,77,a5,3a,9d,a1,77,a5,7a,9d,a9,77,a5
 227 data7b,9d,b1,77,ee,5e,77,20,79,00,4c
 ,fd,7e,a9,89,85,22,a9,77,4c,45,c4,55,4e
 228 data54,49,4c,20,57,49,54,48,4f,55,54
 ,20,44,cf,bf,ff,ff,bf,ff,ff,ff,ff,ff,ff
 229 dataff,ff,ff,ff,ff,ff,ff,ff,ff,ff,ff
 ,ff,ff,ff,ff,ff,ff,ff,ff,ff,ff,4c,e3
 230 datac8,4c,1d,c8,f0,fb,20,fa,77,20,79
 ,00,20,6b,c9,20,13,c6,90,ea,a5,5f,a4,60
 231 datae9,01,4c,24,c8,4c,eb,c8,4c,e0,c8
 ,a9,ff,85,4a,20,8a,c3,9a,c9,8d,d0,f1,20
 232 data79,00,f0,e9,28,28,28,28,28,28,fa
 ,77,4c,a0,c8,c9,a4,d0,4b,4c,73,00,20,fa
 233 datace,20,8a,cd,20,ea,d9,a2,24,a0,78
 ,20,d4,db,20,fd,ce,20,8a,cd,20,ea,d9,a9
 234 data24,a0,78,20,0f,db,4c,f7,ce,81,b1
 ,72,17,f8,7f,5e,5b,d8,a9,20,f1,ce,20,ea
 235 datad9,a9,29,a0,78,4c,28,da,4c,bb,e0
 ,c9,28,d0,f9,20,73,00,20,8a,cd,a2,60,a0
 236 data7c,20,d4,db,20,fd,ce,20,8a,cd,a2
 ,70,a0,78,20,d4,db,20,f7,ce,20,bb,e0,a9
 237 data70,a0,78,20,28,da,a9,60,a0,7c,4c
 ,67,d8,00,00,00,00,00,20,9e,d7,e0,08,b0
 238 data1f,8e,86,02,20,f1,d7,e0,10,b0,15
 ,8a,38,2a,0a,0a,0a,48,20,f1,d7,e0,08,b0
 239 data07,68,05,65,8d,0f,90,60,4c,48,d2
 ,f0,0b,20,9e,d7,e0,7f,b0,f4,8a,09,80,ae
 240 dataa9,fa,48,20,79,00,f0,04,20,f1,d7
 ,ad,a2,05,68,e0,01,90,22,86,fe,8d,0c,90

```

241 dataa9,0f,a8,0d,0e,90,8d,0e,90,a6,fe
,66,61,c6,60,d0,fa,ca,d0,f7,ce,0e,90,88
242 datad0,ef,8c,0c,90,60,ad,16,7f,4a,29
,03,a8,88,8c,69,79,60,a2,5b,a0,7c,20,d4
243 datadb,46,66,a9,bc,a0,d9,20,5b,dc,d0
,0e,a9,dd,a0,e2,20,a2,db,a9,5b,a0,7c,4c
244 data28,da,a9,20,a0,7c,20,a2,db,20,b4
,df,a9,5b,a0,7c,20,28,da,20,49,d8,20,49
245 datad8,20,71,df,a9,5b,a0,7c,20,0f,db
,4c,0b,e3,20,ed,78,a9,dd,a0,e2,4c,50,d8
246 data20,0b,e3,4c,34,79,a9,dd,a0,e2,20
,50,d8,4c,b1,e2,7b,0e,fa,35,12,7b,00,ad
247 datafc,90,86,65,2e,e0,d4,86,7e,a5,dd
,5e,4b,50,55,5a,20,f1,ce,a9,5f,a0,00,f0
248 data0c,88,8d,71,79,b9,5f,79,a0,79,4c
,28,da,60,20,63,79,4c,68,e2,20,63,79,4c
249 data61,e2,20,63,79,4c,b1,e2,20,63,79
,4c,41,79,20,f1,ce,20,ed,78,a9,61,4c,68
250 data79,20,f1,ce,20,31,79,4c,97,79,20
,f1,ce,20,0b,e3,4c,97,79,20,f1,ce,20,3b
251 data79,4c,97,79,20,f1,ce,20,58,dc,a2
,2e,a0,7a,20,d4,db,20,cc,dc,a9,2e,a0,7a
252 data4c,50,d8,20,fa,ce,20,8a,cd,20,fd
,ce,a2,33,a0,7a,20,d4,db,20,8a,cd,a2,38
253 dataa0,7a,20,d4,db,a9,bc,a0,d9,20,0f
,db,a9,33,a0,7a,20,28,da,20,ba,79,a9,38
254 dataa0,7a,20,28,da,8f,00,00,79,b3,dc
,4c,f7,ce,20,fa,ce,20,8a,cd,a2,29,a0,7a
255 data20,d4,db,20,fd,ce,20,8a,cd,a9,29
,a0,7a,20,0f,db,20,cc,dc,4c,f7,ce,00,00
256 data00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,4c,54,e2,a2,08,a0
257 data00,20,50,fe,20,79,00,d0,f1,a9,01
,a2,38,a0,e4,4c,49,fe,a9,01,2c,a9,00,85
258 data0a,20,40,7a,4c,6c,e1,4c,08,cf,4c
,48,d2,4c,58,d6,4c,ab,d3,20,f1,ce,38,b0
259 data04,20,9e,cd,18,08,a6,3a,e8,f0,ed
,20,82,d7,f0,e2,c9,59,b0,e1,8a,f0,02,b1
260 data22,99,00,02,88,10,f8,28,a5,7a,48
,a5,7b,48,08,a9,00,a0,02,85,7a,84,7b,20
261 data79,c5,28,b0,12,20,73,00,20,fd,7e
,20,79,00,c9,3a,f0,f3,a8,f0,08,d0,a8,20
262 data73,00,20,8a,cd,68,85,7b,68,85,7a
,60,20,f1,ce,20,8d,cd,20,bf,d1,a5,64,48
263 dataa5,65,48,a9,02,20,7d,d4,68,a0,01
,91,62,4c,f6,d6,20,f1,ce,20,8d,cd,a2,ff
264 data20,cc,db,a9,05,20,7d,d4,a0,04,b9
,ff,00,91,62,88,10,f8,4c,fb,d6,41,53,ce
265 data41,43,d3,43,4f,d4,41,43,d4,4c,47
,c4,4c,47,d5,43,56,c9,43,56,c6,4d,4b,49
266 dataa4,4d,4b,46,a4,43,52,53,4c,49,ce
,43,52,53,43,4f,cc,54,49,4d,45,a4,49,4e
267 data53,54,d2,48,45,58,a4,44,45,c3,46
,55,4e,43,54,49,4f,ce,4d,4f,c4,44,49,d6
268 data46,52,c3,52,41,4e,44,4f,cd,52,41
,44,49,41,ce,44,45,47,52,45,c5,47,52,41
269 datac4,51,55,49,d4,a7,4c,4f,43,41,54
,c5,42,45,45,d0,43,4f,4c,4f,d2,43,4c,d3
270 data45,58,45,43,55,54,c5,44,4c,4f,41
,c4,44,53,41,56,c5,44,56,45,52,49,46,d9
271 data43,41,54,41,4c,4f,c7,44,49,52,45
,43,54,4f,52,d9,49,4e,49,54,49,41,4c,49
272 data5a,c5,44,cf,55,4e,54,49,cc,50,50
,52,49,4e,d4,4c,50,52,49,4e,d4,52,45,4e
273 data55,4d,42,45,d2,44,45,4c,45,54,c5

```

```

,00,00,00,00,00,00,00,00,00,00,00,00,00
274 data00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,00,00,00,00,00,00
275 data00,00,00,00,00,00,00,00,00,00,00,00
,00,00,31,c8,42,c7,1e,cd,f8,c8,a5,cb,bf
276 datacb,81,d0,06,cc,a5,c9,a0,c8,59,73
,91,7e,bf,77,83,c8,de,77,3b,c9,2f,c8,4b
277 datac9,2d,d8,65,e1,53,e1,62,e1,26,73
,24,d8,80,ca,a0,ca,57,c8,9c,c6,13,77,86
278 dataca,27,e1,bb,e1,c4,e1,7b,cb,19,77
,a4,d3,ad,a4,d6,c8,4c,a2,d3,20,a6,db,84
279 data0d,60,20,40,7a,4c,56,e1,00,00,00
,00,00,08,cf,08,cf,00,00,00,00,4d,7e,4d
280 data7e,4d,7e,0c,73,4d,7e,4d,7e,4d,7e
,4d,7e,4d,7e,4d,7e,7f,79,79,79,85,79,a5
281 data79,4d,7e,4d,7e,4d,7e,4d,7e,4d,7e
,eb,73,4d,7e,4d,7e,4d,7e,08,cf,91,79,9c
282 data79,8b,79,ae,79,2e,78,01,78,7a,7d
,8b,7d,c7,7a,e3,7a,49,7c,46,7c,00,7d,9a
283 data72,5c,72,2b,72,6e,7a,ce,79,09,7a
,b7,79,3e,78,e1,78,e1,78,e1,78,22,fd,f8
284 datac8,3c,7d,9e,78,75,78,5f,e5,74,7a
,58,7a,55,7c,55,7a,81,76,81,76,00,77,5d
285 data77,31,77,14,74,17,74,c5,74,30,76
,08,cf,08,cf,08,cf,08,cf,08,cf,08,cf,08
286 datacf,08,cf,08,cf,20,84,cf,84,5e,88
,84,71,a0,06,84,5d,a0,24,20,68,de,a9,00
287 data8d,07,01,ad,04,01,8d,06,01,ad,03
,01,8d,05,01,ad,02,01,8d,03,01,ad,01,01
288 data8d,02,01,a9,3a,8d,01,01,8d,04,01
,4c,6d,d4,4c,48,d2,c9,2c,f0,0f,20,9e,d7
289 dataca,e0,17,b0,f1,86,d6,20,79,00,f0
,0a,20,f1,d7,ca,e0,16,b0,e2,86,d3,4c,87
290 datae5,a5,14,48,a5,15,48,20,79,00,20
,6b,c9,a5,15,d0,cd,a6,14,68,85,15,68,85
291 data14,ad,a2,0d,4c,ef,d6,20,f1,ce,20
,a3,d6,c9,02,d0,b5,86,64,84,65,4c,61,cf
292 data20,f1,ce,20,a3,d6,c9,05,d0,a4,4c
,4f,7c,93,0e,05,b0,c0,c0,c0,c0,c0,c0,c0
293 datac0,c0,c0,c0,c0,c0,c0,c0,c0,c0,c0
,c0,c0,ae,dd,9c,20,20,4f,50,45,52,41,54
294 data49,4e,47,20,53,59,53,54,45,4d,20
,20,05,dd,dd,20,cd,c1,d4,c8,2e,20,c2,c1
295 datad3,c9,c3,20,20,d6,31,2e,30,32,20
,dd,dd,9c,42,59,20,d7,2e,d7,49,52,54,48
296 data20,30,36,37,30,33,2f,38,36,38,05
,dd,ad,c0,c0,c0,c0,c0,c0,c0,c0,c0,c0,c0
297 datac0,c0,c0,c0,c0,c0,c0,c0,c0,0d,0d
,0d,20,42,41,53,49,43,20,4d,45,4d,4f,52
298 data59,3a,11,0d,20,00,3a,c4,83,c4,1b
,7f,d6,7f,f4,7e,d4,7e,bf,ea,85,7e,68,7e
299 data0a,f4,4a,f3,c7,f2,09,f3,f3,f3,0e
,f2,7a,f2,70,f7,f5,f1,ef,f3,d2,fe,49,f5
300 data85,f6,68,68,ad,16,7f,48,aa,4c,ad
,cf,20,8a,cd,46,66,a9,5d,4c,ff,ce,00,4c
301 data56,ff,4c,de,fe,48,8a,48,98,48,ad
,1d,91,10,f0,2d,1e,91,aa,29,02,f0,eb,2c
302 data11,91,20,34,f7,20,e1,ff,d0,dd,20
,a9,7e,6c,02,c0,4c,3b,c9,4c,7a,77,20,9e
303 datacd,a5,61,f0,f3,20,79,00,c9,a7,d0

```

Listing »Mathematical Basic« (Fortsetzung)

```

,ef,20,73,00,b0,ea,4c,a0,c8,00,00,7B,a2
304 data2d,a0,7e,20,56,fd,a2,0b,bd,21,7e
,9d,00,03,ca,10,f7,20,f9,fd,20,18,e5,a9
305 data0e,8d,0f,90,a9,01,8d,86,02,ea,60
,4c,f3,dc,4c,c2,73,a9,00,85,0d,20,73,00
306 data90,f1,c9,e1,b0,f0,c9,b4,90,ec,20
,0e,7f,4c,8d,cd,4c,08,cf,4c,a5,c9,4c,12
307 datac8,20,73,00,20,fd,7e,4c,ae,c7,f0
,ce,aa,10,ec,c9,cb,f0,eb,c9,a3,90,04,c9
308 datae1,90,dd,0a,8d,16,7f,20,73,00,6c
,c8,7c,00,00,00,a6,7a,a0,04,84,0f,bd,00
309 data02,10,07,c9,ff,f0,3e,e8,d0,f4,c9
,20,f0,37,85,08,c9,22,f0,56,24,0f,70,2d
310 datac9,3f,d0,04,a9,99,d0,25,c9,30,90
,04,c9,3c,90,1d,84,71,a0,00,84,0b,88,B6
311 data7a,ca,c8,e8,bd,00,02,38,f9,9e,c0
,f0,f5,c9,80,d0,30,05,0b,a4,71,e8,c8,99
312 datafb,01,b9,fb,01,f0,59,38,e9,3a,f0
,04,c9,49,d0,02,85,0f,38,e9,55,d0,9f,85
313 data08,bd,00,02,f0,df,c5,08,f0,db,c8
,99,fb,01,e8,d0,f0,a6,7a,e6,0b,c8,b9,9d
314 datac0,10,fa,b9,9e,c0,d0,b4,a0,ff,ca
,c8,e8,bd,00,02,38,f9,00,7b,f0,f5,c9,80
315 datad0,02,f0,ad,a6,7a,e6,0b,c8,b9,ff
,7a,10,fa,b9,00,7b,d0,e2,bd,00,02,10,9b
316 data4c,09,c6,4c,ef,c6,b1,5f,4c,1a,c7
,c9,cc,90,f7,c9,ff,f0,f3,24,0f,30,ef,84
317 data49,38,e9,cb,aa,a0,ff,ca,f0,08,c8
,b9,00,7b,10,fa,30,f5,c8,b9,00,7b,30,d3
318 data20,47,cb,d0,f5
319 rem
320 rem----- pruef
summen -----
321 rem
322 data 31231
323 data 25675
324 data 28397
325 data 31427
326 data 30997
327 data 33043
328 data 30573
329 data 30423
330 data 28240
331 data 26807
332 data 21966
333 data 26340
334 data 31124
335 data 29634
336 data 27953

```

ready.

Listing »Mathematical Basic« (Schluß)

Ohne gutes Werkzeug geht es nicht:

SMON Teil 2

Der Maschinensprache-Monitor SMON wird immer leistungsfähiger. Dieser 2. Teil erweitert ihn um wichtige Ausgabe-Routinen, läßt das Verschieben eines Programms mit und ohne Adreßumrechnung zu und kann Zahlen vom Dezimal- in das Binärsystem und umgekehrt umrechnen.

Wir hoffen, daß wir Ihnen in der letzten Ausgabe nicht zuviel zugemutet haben, und daß sich Ihre wunden Finger inzwischen wieder erholen konnten. Bestimmt haben Sie im vergangenen Monat schon eifrig mit dem neuen Monitor gearbeitet und sind inzwischen mit den bisherigen Befehlen vertraut. Denn nun folgt der zweite Teil und mit diesem natürlich wieder einige neue Befehle, die es zu erklären gilt.

Und das bieten wir Ihnen heute:

I/O-SET, LOAD, SAVE, PRINTER-SET, die verschiedenen Zahlenumrechnungen (HEX-DEZ-BIN-ADD-SUB), OCCUPY, CONVERT, VERSCHIEBEN und WRITE.

I/O-SET

I 01 legt die Device-Nummer für LOAD und SAVE auf 1 (Kassette). Jedes Laden und Abspeichern erfolgt jetzt auf das angegebene Gerät. Die voreingestellte Device-Nummer ist 8 (für die Floppy also: I 08). Wenn Sie nur mit der Floppy arbeiten, brauchen Sie diesen Befehl also nicht.

LOAD

L "Name" lädt ein Programm vom angegebenen Gerät (wie oben beschrieben) an die Originaladresse in den Speicher. Die Basic-Zeiger bleiben bei diesem Ladevorgang unbeeinflusst, das heißt, sie werden nicht verändert.

Beispiel: Unser Monitor soll an seiner Originaladresse (\$C000) im Speicher stehen. Also brauchen Sie ihn nur mit »L "SMON"« zu laden, damit er dort erscheint. Wenn Sie einmal ein Programm an eine andere als die Originaladresse laden wollen, dann bietet Ihnen SMON dazu folgende Möglichkeit: 'L "Name" ADRESSE lädt ein Programm an die angegebene Adresse. Nehmen Sie doch bitte noch einmal unser letztes Test-Programm und geben es mit dem Assembler ab Adresse \$4000 ein. Speichern Sie es mit »S "SUPERTEST" 4000 4023« ab und laden es dann

1. an die Originaladresse (L "SUPERTEST") und