

- .# -> DECIMALCONVERSION
Umwandlung dez - hex - ASC - binär
- .% -> BINARYCONVERSION
Umwandlung binär - hex - dez - ASC
- ." -> ASCIICONVERSION
Umwandlung ASC - hex - dez - binär
- .+ -> ADDITION
Addiert zwei Hexadressen
- .- -> SUBTRACTION
Subtrahiert zwei Hexadressen
- .& -> CHECKSUM
Bildet Prüfsumme der im angegebenen Speicherbereich sich befindenden Codes
- Bereits an dieser Übersicht können Sie ermes sen, welche Leistungsfähigkeit MIKRO-
MON besitzt. Aber noch nicht genug damit. Mikromon hat zusätzlich noch einige Be-
fehle, um Drucker und Floppylaufwerke zu bedienen.
- .D (d geshiftet) -> PRINTERDISASSEMBLE
Disassemblierung auf Drucker
- .i -> HEADINGANDFORMFEED
Druckt Seitenüberschrift und steuert Seitenvorschub am Drucker
- .j -> PRINTLOADADDRESS
Liest Startadresse einer Programmdatei und gibt diese hexadezimal aus
- .K (k geshiftet) -> KILLMIKROMONADDITIONS
Wie .k, jedoch kann Mikromon wieder mit .g eingeschaltet werden
- .L (l geshiftet) -> LOADFROMDISK
Lädt Programm von Diskette
- .M (m geshiftet) -> PRINTMEMORYDISPLAY
Gibt .m-Kommando auf Drucker aus
- .P -> PRINTERSWITCHING
Schaltet auf Drucker Ausgabe
- .u -> SENDTOPROMPROGRAMMER
Überträgt die Codes eines Speicherbereiches auf Programmiergerät
- .v -> SPECIFYLOADADDRESS
Lädt Programm von Diskette in vorgegebenen Speicherbereich

- .Z (z geshiftet) -> TEXT/GRAPHIKFLIP
Schaltet Zeichensätze um, zugleich keine Dunkelzeilen
- .↑ (Pfeil nach oben) -> CONTROLCHARACTERS
Führt Steuerzeichen aus
- .>
oder
.5 (= Klammeraffe) -> DOSSUPPORTCOMMANDS
Floppy Direktbefehle

Es werden nun die Befehle im einzelnen behandelt. Dazu werden jeweils noch entspre-
chend Beispiele gebracht. Wenn Sie später mit Mikromon umzugehen wissen, werden
Sie dieses Soft-Tool nicht mehr missen wollen.

Programmstart

Nachdem Mikromon geladen wurde, erfolgt der Aufruf mit „SYS 23296“. Nach dem
Start erscheint folgender Bildschirm:

```
=====
mikromon *bam* 8.82
b*
pc  irq  sr  ac  xr  yr  sp
;  5b69 6355 32 00 5b 00 f8
=====
```

Eingaben

Alle Eingaben erfolgen normalerweise, wie bei TIM, hinter einem Punkt. Ausgenommen
sind die Kommandos nach dem Befehl 'p', der die Ausgaben auf Drucker umleitet und
somit den Punkt am Drucker ausgibt.

Ausgabeformat dieser Anleitung

Alle Beispiele in dieser Beschreibung sind im Kleinschreibmodus (Textmodus) ausge-
führt, um die Kommandos geshiftet oder ungeschiftet besser darstellen zu können. So be-
deutet z.B.
.d (ungeshiftet) = Die Disassemblierung erfolgt auf Bildschirm,
während
.D (geschiftet) = Die Disassemblierung auf dem Drucker
bewirkt.

Zudem lassen sich die einzelnen Zeichen leichter unterscheiden.

```

*****
*   -   *
*   -   *
*****
SIMPLEASSEMBLE
EINFACHERASSEMBLER

```

Bei diesem einfachen Assembler handelt es sich um einen „ONE LINE“-Assembler. Das heißt, es wird jeweils nur eine Zeile assembliert und kein größerer Speicherbereich. Somit ist nur 1 mnemotechnischer Befehl mit maximal 3 Bytes (OP-Code u. Adressierung) zugelassen.

Die Eingabe erfolgt zum Beispiel so:

```

=====
. a 0380 lda # $41
=====

```

Wird nun RETURN gedrückt, steht unmittelbar darauf folgender Ausdruck auf dem Bildschirm:

```

=====
. a 0380 a9 41      lda # $41
. a 0382
=====

```

Sie sehen, die Assemblierung erfolgte für den ersten Befehl sofort. Automatisch wird in die nächste Zeile ein ‚a‘ und die Fortsetzungsadresse ausgegeben.

Um aus dem Assemblermodus auszusteigen, brauchen Sie nur bei der Fortsetzungsadresse ein RETURN zu drücken.

Da die Syntax der mnemonischen Befehle eine bestimmte Form haben, ist diese unbedingt einzuhalten. Dies ersieht man auch aus der Disassemblierung. Anhand von Beispielen soll hier die Einhaltung der Assemblerform aufgezeigt werden.

lda # \$41	immediate
lda \$04	zeropage
lda \$04,x	zp, X indiziert
ldx \$04,y	zp, Y indiziert
lda \$0380	absolut
lda \$0380,x	absolut, X indiziert
lda \$0380,y	absolut, Y indiziert
lda (\$01,X)	X indiziert, indirekt
lda (\$01),Y	indirekt, Y indiziert
jmp (\$0380)	indirekt
bne \$0380	relativ (Es kann die direkte Adresse angegeben werden. Assembliert wird allerdings auf den Offset-Wert.)
dex	impliziert

```

*****
*   -   *
*   -   *
*****
BREAKSET
SETZT
UNTERBRECHUNGSPUNKTE

```

Beispiel:

```

=====
. b 1000 00ff
=====

```

Das Beispiel setzt einen Breakpoint bei \$1000, wenn dort \$ff steht. BREAKSET wird dazu benutzt, um bei QUICKTRACE Unterbrechungsmöglichkeiten zu haben. Ein BREAKSET kann z.B. Blanks zählen und bei vorgegebener Anzahl das Programm stoppen.

```

*****
*   -   *
*   -   *
*****
VERGLEICH      COMPAREMEMORY
                SPEICHERBEREICHE

```

Es werden zwei gleich große Speicherbereiche miteinander verglichen. Das heißt, das Programm sieht nach, ob die jeweilig entsprechenden Bytes in den verschiedenen Bereichen gleich sind. Falls beide Bereiche übereinstimmen, wird nichts ausgegeben. Ansonsten werden die Adressen der ungleichen Codes auf dem Bildschirm ausgedruckt.

Bei der Eingabe hinter ‚c‘ wird die Anfangs- und Endadresse eines Speicherbereichs angegeben, während für den zweiten Speicherbereich nur die Anfangsadresse erforderlich ist.

Beispiel:

```

=====
. c 10af 20ff c0af
=====

```

Im Beispiel wird der Bereich \$10af bis \$20ff mit dem Bereich ab \$c0af verglichen.

```

*****
*   -   *
*   -   *
*****
DISASSEMBLIER
DISASSEMBLER

```

Für die Disassemblierung gibt es 2 Möglichkeiten. Entweder es wird die Anfangs- und Endadresse des zu disassemblierenden Bereichs angegeben, oder nur die Anfangsadresse ab der disassembliert werden soll.

Im ersten Fall wird der Speicherbereich in disassembliertem Zustand ausgegeben. Ist dieser Bereich umfangreicher als man mit 25 Zeilen darstellen kann, rollt der Bildschirm

inhalt nach oben. Um nun einzelne Bereiche betrachten zu können, kann das Hochrollen mit Drücken der Taste „↑“ unterbrochen werden. Fortsetzen erfolgt mit der Taste „3“. Da die Ausgabe doch recht schnell rollt, erreicht man durch Drücken des „Pfeilzeichen nach links“ (auf der Tastatur ganz links oben) eine Verlangsamung.

Bei Drücken der STOP-Taste wird die Disassemblierung abgebrochen, aber noch nicht verlassen. Erst nach RETURN kann ein anderes Kommando eingegeben werden.

Falls die RETURN-Taste noch nicht gedrückt wurde, kann nun auf folgende Weise weiter disassembliert werden: Es wird lediglich die Taste „Cursor nach unten“ gedrückt. Wenn der Cursor die unterste Zeile erreicht hat, rollt der Bildschirminhalt nach oben und weitere disassemblierte Zeilen werden unten nachgeschoben. Bei gleichzeitigen Drücken der SHIFT-Taste wandert der Cursor nach oben. Hat er die oberste Bildschirmzeile erreicht, wird der Bildschirminhalt nach unten gerollt und oben disassemblierte Zeilen nachgeschoben. Durch diese Art kann also das SCROLLING nach unten und nach oben durchgeführt werden, was oft eine erhebliche Erleichterung mit sich bringt. Die oberste Bildschirmzeile kann man ganz schnell erreichen, indem die HOME-Taste gedrückt wird.

Die zweite Art, um die Disassemblierung zu erreichen, ist die, wenn hinter „d“ nur die Anfangsadresse angegeben wird. Nach RETURN muß nun solange die Taste „Cursor nach unten“ gedrückt werden, bis die disassemblierten Zeilen am unteren Bildschirmrand erscheinen und bis über den obersten Bildschirmrand gewandert sind. Dann kann wieder normal in beiden Richtungen gescrollt werden.

Als Beispiel soll hier die disassemblierte CHRGET-Routine gezeigt werden.

```
=====
.d 0070 0087
-----
, 0070 e6 77 inc $77
, 0072 d0 02 bne $0076
, 0074 e6 78 inc $78
, 0076 ad 06 02 lda $0206
, 0079 c9 3a cmp #$3a
, 007b b0 0a bcs $0087
, 007d c9 20 cmp #$20
, 007f f0 ef beq $0070
, 0081 38 sec
, 0082 e9 30 sbc #$30
, 0084 38 sec
, 0085 e9 d0 sbc #$d0
, 0087 60 rts
=====
```

← Eingabe

← Ausgabe

```
*****
* - E * EXTMIKROMON
*****
RÜCKKEHR ZU BASIC
```

Bei dem Kommando „e“ wird auf dem Bildschirm „READY“ ausgegeben und der Rechner befindet sich im Basic-Direktmodus. Da noch die Zeiger durch Mikromon ver- stellt sind, kann es notwendig werden, daß man nun noch ein „NEW“ oder „CLR“ gibt.

```
*****
* - F * FILLMEMORY
*****
FÜLLT SPEICHERBEREICH
```

Beispiel:

```
=====
.f 1000 1a00 ff
=====
```

Mit dem Kommando „f“ wird der vorgegebene Speicherbereich mit dem entsprechend angegebenen Code aufgefüllt. Für das Beispiel heißt das, daß nach Ausführung im Speicherbereich von \$1000 bis \$1a00 jede Speicherstelle den Code „ff“ enthält.

```
*****
* - G * GORUN
*****
MASCHINENPROGRAMMSTART
```

Dieses Kommando hat die gleiche Funktion wie die bereits eingebaute Funktion des CBM-Monitors TIM.

```
*****
* - H * HUNTMEMORY
*****
DURCHSUCHE SPEICHER
```

Hinter „h“ wird die Anfangs- und Endadresse des erforderlichen Speicherbereichs eingegeben. Dahinter kommt die zu suchende Angabe. Dabei gibt es zwei Möglichkeiten:
a) Suche nach einem ASCII-String

Beispiel:

```
=====
.h c000 d000 read
=====
```

In diesem Fall wird der zu suchende Begriff mit einem Apostroph eingeleitet.

b) Suche nach Maschinenbefehlen

Beispiel:

```
=====
.h c000 ffff 20 70 00
=====
```

Im zweiten Fall werden nur die entsprechenden Codes angegeben. Im Beispiel wird gesucht, wo die CHRGET-Routine überall im Betriebssystem als Unterprogramm aufgerufen wird.

Die maximale Länge des Suchbegriffs, egal ob es Codes oder Strings sind, ist auf 32 Zeichen festgelegt.

Die Suche kann zu jeder Zeit mit der Stoptaste abgebrochen werden. Wurde der Suchbegriff gefunden, wird dessen Anfangsadresse auf dem Bildschirm ausgegeben. Bei mehreren gleichen Begriffen werden eben sämtliche Anfangsadressen ausgedruckt.

```
*****
_*_*_*_*_*   KILLMEMORY
_*_*_*_*_*
RÜCKKEHR IN DEN CBM-MONITOR (TIM)
```

Beim Start von Mikromon wird der IRQ- und BREAK-Vektor (\$90-\$93) zurückgespeichert. Wird nun der Befehl „.k“ gegeben, werden diese Vektoren wieder hergestellt und das Programm springt in normalen TIM-Monitor.

```
*****
_*_*_*_*_*   LOAD
_*_*_*_*_*
PROGRAMM LADEN
```

Beispiel::

```
=====
.l ,ram-test“,08
=====
```

In diesem Beispiel wird von Diskette die Programmdatei mit Namen „ram-test“ geladen. Steht hinter dem Namen „01“, dann wird das Programm von der Kassette geholt. Allerdings verändert das Laden den IRQ-Vektor. Deshalb sollte man vor dem Laden mit „.x“ in den Basic-Direktmodus zurückkehren und von dort aus mit „SYS 1024“ in den normalen TIM-Monitor einsteigen. Übrigens gilt das gleiche auch für den Monitor-SAVE.

```
*****
_*_*_*_*_*   MEMORYDISPLAY
_*_*_*_*_*
SPEICHERANZEIGE
```

Beispiel:

```
=====
.m 0000 00ff
=====
```

oder

```
=====
.m 0000
=====
```

Beim Kommando „.m“ werden in einer Bildschirmzelle 16 Codes (Hexdump) und deren ASCII-Werte angezeigt.

Wie bei der Disassemblierung gibt es zwei Möglichkeiten. Entweder es wird der Speicherbereich mit Anfangs- und Endadresse angegeben oder nur die Anfangsadresse. Dabei haben die entsprechenden Steuertasten die gleiche Wirkung, wie bei dem Kommando „.d“ (siehe Dissas.). Im oben angeführten Beispiel wurde die Zeropage dargestellt.

Beispiel des Kommandos „.m 00a0 00b0“

```
=====
00a0 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 350123456789012345
00b0 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80ABCDEF GHIJKLMNOI
=====
```

```
*****
_*_*_*_*_*   NEWLOCATER
_*_*_*_*_*
VERANDERT ABSOLUTCODES
```

Dieser sehr interessante Befehl ändert innerhalb eines Speicherbereichs alle 3-Byte-Befehle, indem zu der jeweiligen Absolutadresse ein Offset addiert wird.

Beispiel:

```
=====
.n 7000 7500 6000 1000 2000
                                     (Beispiel 1)
=====
```

```
=====
.n 7500 8000 6000 1000 2000w
                                     (Beispiel 2)
=====
```

Im Beispiel 1 stellen die ersten beiden Adressen die Anfangs- und Endadresse des zu bearbeitenden Bereichs dar. Der dritte Wert (im Beispiel \$6000) ist der Offsetwert. Die


```

*****
* - = *      SAVE
*****
PROGRAMM ABSPEICHERN

```

Beispiel:

```

=====
s,0:prg-name",08,0401,2500
=====

```

Im Beispiel wird ein Speicherbereich mit Namen „prg-name“ auf die Diskette im rechten Floppylaufwerk abgespeichert. Die Endadresse ist allerdings nicht \$2500, sondern \$24ff. Das heißt, es wird nur bis zur Adresse \$2500 abgespeichert, aber nicht einschließlich. Wie schon bei „.1“ (load) erläutert, sollte das Abspeichern vom normalen TIM-Monitor aus erfolgen, da „.s“ den IRQ-Vektor verändert.

```

*****
* - = *      WALKCODE
*****
EINZELSCHRITT MIT R UND D ANZEIGE

```

Wie bei QUICKTRACE wird ein Programm Befehl für Befehl ausgeführt und nach jedem Befehl die Inhalte der Register, sowie der disassemblierte Befehl angezeigt. Die Startadresse kann dabei, wie bei QUICKTRACE, aus dem Programmzähler stammen oder direkt hinter „.w“ angegeben werden.

Beispiel:

```

=====
01 02 03 04 05 0006 8d 80 03 sta $0380
=====

```

Die Hex-Codes sind dabei die Inhalte folgender Register:

```

01 = Statusregister      8d = OP-Code
02 = Akkumulator        80 = unterer Adresteil
03 = X-Register         03 = oberer Adresteil
04 = Y-Register
05 = Stapelzeiger      sta $0380 = Mnemonik
0006 = Programmzähler

```

Da nach RETURN die Ausgabe verhältnismäßig schnell nach oben rollt, haben folgende Tasten für das „.w“-Kommando eine Sonderbedeutung:

Pfeil nach links
Langsamsc(): Es wird weiter gescrollt, jedoch wesentlich langsamer

* / :
Unterbricht WALK CODE

/ 3
Fortsetzung nach der Unterbrechung

RUN/STOP
Beendet die Ausführung des Kommandos „.w“
Sonst ist jede Taste, außer SHIFT und REPEAT, als Steuertaste für Einzelschritt oder Schnellschritt zuständig.

```

*****
* - = *      EXITTODASIC
*****
RÜCKKEHR ZU BASIC

```

Diese Funktion entspricht der TIM-Funktion. Es wird in den BASIC-Warnstart gesprungen. Die Stack-Tiefe wird mit übernommen, so daß gegebenenfalls ein CLR erforderlich ist.

```

*****
* - = *      CHANGECHARAKTerset
*****
ZEICHENSATZ WECHSEL

```

Mit diesem Befehl wird der Zeichengenerator umgeschaltet. Je nachdem, in welchen Zeichensatz sich der Rechner befindet, schaltet „.z“ in den anderen Satz um.

Beispiel:

```

=====
test abc 123
.z
TEST ABC 123
.Z
test abc 123
=====

```


INVERSION
ZAHLENUMWANDLUNGEN

Beispiele: Eingabe Ausgabe

.\$4142		.\$4142	16706	a	b	0100	0001	0100	0010	
.#16706		.#16706	4142	a	b	0100	0001	0100	0010	
.%0100000101000010		.%0100000101000010				4142	16606	a	b	
."a		."a				a	41	65	0100	0001

Je nachdem, welches Zeichen vorne steht, wird die entsprechende Umwandlung vollzogen.

- .\$ = Eingabe hexadezimal
- .# = Eingabe dezimal
- .% = Eingabe binär
- ." = Eingabe des ASCII-Zeichens

ADDITION

Beispiel:

+aaaa 2222 cccc

Nach dem Pluszeichen werden zwei vierstellige Hexzahlen eingegeben. Nach RETURN wird dahinter das Ergebnis hexadezimal angezeigt.

SUBTRACTION

Beispiel:

- cccc 2222 aaaa

Nach dem Minuszeichen werden zwei vierstellige Hexzahlen eingegeben. Nach RETURN wird die zweite Zahl von der ersten abgezogen und dahinter das Ergebnis angezeigt.

CHECKSUM
PRÜFSUMMERBILDEN

Beispiel:

& 0400 0403 00aa

Hinter dem Kommando „.&“ wird ein Speicherbereich (Anfangs- und Endadresse) a gegeben. Nach RETURN werden sämtliche Bytewerte in diesem Bereich addiert und dahinter hexadezimal angezeigt. Man kann daher die Prüfsumme eines Maschinenprogramm bilden.

Bis hierher sind nun die Hauptfunktionen von Mikromon behandelt worden. Zusätzlich beherrscht Mikromon noch einige Befehle, die einen angeschlossenen Drucker a sprechen, oder den Umgang mit einem Floppylaufwerk erleichtern.

PRINTER DISASSEMBLE
DISASSEMBLIERUNG AUF 'DRUCKER (GESHIFTES P)

Der Drucker muß auf die Gerätenummer 4 eingestellt sein. Sofern er eingeschaltet ist erfolgt nach „.D“ die Disassemblierung nicht am Bildschirm, sondern auf dem Drucker.

FORM FEED SET
STEUERT SEITENFORMAT

Wird das Kommando „.i“ eingegeben, legt man damit das Seitenformat auf 57 Zeilen fest. „.i“ arbeitet mit den Kommandos „.D“ und „.M“ zusammen. Das heißt, nachdem Ausdruck von 57 Zeilen auf dem Drucker erfolgt ein Seitenvorschub. Wird hinter „.i“ noch ein String angegeben, so wird dieser auf jeder Druckseite als Überschrift ausgedruckt.

Beispiel:

.i „Seiteneüberschrift“

Mit „.ix“ wird der FORMFEED Modus beendet.


```

*****
_Z_____TEXTFLIP FOR BOZZ&FA 40°S
*****
ZEICHENUMSCHALTER OHNE DUNKELZEILE (SHIFT Z)
    
```

Damit lassen sich die Zeichensätze wie bei „.z“ umschalten. Bei Umschaltung vom Textmodus aus wird der Graphikmodus ohne Dunkelzeilen ausgegeben.

```

*****
-+___CONTROL CHARAKTERS
*****
FUHRT STEUERZEICHEN AUS
    
```

Wird hinter dem „Pfeil nach oben“ das entsprechende ASCII-Zeichen der Steuerodes angegeben, so erfolgt nach RETURN dessen Ausführung. Bei Angabe von „↑g“ ertönt zum Beispiel die Glocke. Oder. ↑ S löscht den Bildschirm.

```

*****
* DOS SUPP O R T *
* * * * *
FLOPPY DIREKTBEFEHLE
    
```

Mikromon enthält auch die bekannten DOS SUPP O R T-Befehle, die eine wesentliche Erleichterung mit dem Umgang der Floppy im Direktmodus mit sich bringt. Die Befehle werden entweder eingeleitet durch Eingabe des „>“ (Größerzeichen) oder durch den „Klammeraffen“ hinter dem Punkt. Danach folgt der entsprechende Floppybefehl. Das DOS SUPP O R T ist auf die Gerätenummer 8 ausgelegt.

Die alleinige Eingabe des Befehls „>“ bewirkt die Ausgabe des Fehlerkanals, entsprechend dem BASIC-4-Befehl „PRINT DS\$“.

➤ \$0 0 . ➤ \$1 0 . ➤ \$

Damit wird das entsprechende Inhaltsverzeichnis der Disketten auf dem Bildschirm ausgegeben.

Weitere Diskettenbefehle

BEFEHLSART:IDOSSUPP O R T:IDOS1	DOS2 BASIC 3	BASIC4
initialisieren einer Diskette	> i	print# 1, „i“ nicht mehr notwendig
beringige Inhaltsverzeichnis.	> v0	print# 1, „v0“ collect d0
lösche Datei	> sl:fnam	pR1, „sl:fnam“ scratchd1, „fnam“

BEFEHLSART:IDOSSUPP O R T:IDOS1	DOS2 BASIC 3	BASIC 4
Diskette dulizieren	> I=0	pR1, „I=0“ backup d0 to d1
Datei kopieren	> c1:zn=0:qn	pR1, „c1:zn=0:qn“ copyd0, „qn“ to d1, „zn“
Datei abhängen	> c1:zn=0:qn, 0:qn2,	nicht vorhanden concatd0, „qn2“ to d1, „zn“
Umbenennen v. Dateinamen	> r0:rn=an	pR1, „r0:rn=an“ rename d0, „an“ to „rn“
Diskette formatieren	> n0:dina:jd	pR1, „n0:dina:jd“ header „dina“, d0:jd

Das DOS SUPP O R T benötigt kein Öffnen des Kommandokanals, da dies bei jedem Befehl automatisch vom Programm durchgeführt wird.

Wir haben uns nun eine Übersicht der Befehle von MIKROMON beschafft. Auf den nächsten Seiten ist das hexadezimale Listing (Hexdump) abgebildet. Dieses ist nun nicht, wie man es von TIM her gewöhnt ist, achtspaltig (Adresse + 8 Byte), sondern 16-spaltig und enthält dahinter noch wirre Zeichenfolgen. Wenn Sie sich die Befehlsanleitung genau angesehen haben, entspricht der Ausdruck genau dem Befehl „.m“, und das ist eben der Speicherauszug wie ihn MIKROMON erzeugt. Eine Zeile bei MIKROMON entspricht zwei Zeilen bei TIM. Das haben Sie dann bei der Eingabe zu berücksichtigen.

Die Eingabe des Programms erfolgt nun auf folgende Weise:

```

Mit „SYS 1024“ wird der CBM-Monitor aufgerufen. Es erscheint der bekannte Registerausdruck. Da diese Programmversion ab 23296 (= $5b00) im Speicher steht, bringt man mit
.m 5b00, 5ba0
.5b00 aa aa aa aa aa aa aa
.5b08 aa aa aa aa aa aa aa
.5b10 aa aa aa aa aa aa aa
.5b18 aa aa aa aa aa aa aa
.....
usw.
    
```

den entsprechenden Speicherauszug auf den Bildschirm. Dann überschreibt man die Codes (hier: aa) mit den entsprechenden Werten des MIKROMON-Listings. Wenn das Ende am Schirm erreicht ist, dann holt man sich mit „.m“ den nächsten Bereich ins Bild und fährt mit der Übertragung fort. Das geht nun solange, bis das gesamte Programm übertragen wurde. Wenn Sie das geschafft haben, dann können Sie es erst einmal aufatmen. Sie haben es hier nämlich mit über 5000 Bytes zu tun. Bevor Sie nun versuchen, MIKROMON in Betrieb zu setzen, sollten Sie zuerst auf alle Fälle gleich vom Monitor aus absichern.

Das sieht dann so aus:
.s ,mikromon",01,5b00,7000
für Kassette
.s ,0:mikromon",08,5b00,7000 (rechtes Laufwerk)

Das Abspeichern ist sehr wichtig. Denn wenn Sie bei der Eingabe einen Fehler gemacht haben und MIKROMON von Basic aus mit „SYS 23296“ einschalten, springt der Rechner im günstigsten Fall in den Monitor. Meist jedoch rührt sich überhaupt nichts mehr und dem Rechner ist nur noch durch Aus- und Einschalten wieder Vernunft beizubringen. Sie können sich dann den Ärger denken, wenn nicht abgespeichert wurde und das Programm noch einmal mühsam eingetippt werden muß.

Nach der Abspeicherung also, geben Sie „.x“ ein. Sie befinden sich dann wieder im Basic-Direktmodus. Geben Sie nun das folgende Basicprogramm ein, das überprüft, ob alles richtig übertragen wurde.

```
10 PRINT CHR$(147) : POKE 59468,12 : PRINT „BITTE WARTEN“
20 IF PEEK(57345) = 127 OR PEEK(57823) = 127 THEN Y = 606295 : GOTO 40
30 Y=607066
40 Z=0:FOR I = 23296 TO 28671:Z=Z+PEEK(I):NEXT
50 PRINT CHR$(147)
60 IF Z <> Y THEN PRINT „FALSCH EINGABE“:GOTO 80
70 PRINT „DUERFTE RICHTIG SEIN“:GOTO 90
80 PRINT:PRINT „DIFFERENZ ZUR PRUEFSUMME = “:Z-Y
90 IF Y = 607066 THEN POKE 59468,14
100 END
```

Nun startet man das Basic-Programm mit „RUN“ und wartet ein paar Sekunden. Erscheint dann am Bildschirm „DUERFTE RICHTIG SEIN“, dann haben Sie unwahrscheinlich Glück gehabt und können MIKROMON mit

SYS 23296

zur Ausführung bringen. In den meisten Fällen jedoch wird sich eine Differenz ergeben. Dann haben Sie irgendwo ein oder mehrere Codes falsch übertragen. In diesem Fall hilft alles nichts, als sich auf die Suche zu machen. Sie lassen sich im Monitor wieder mit „m“ den Hexdump ausgeben und müssen ihn Byte für Byte mit dem MIKROMON-Listing vergleichen und die gefundenen Fehler berichtigen. Wenn Sie Pech haben, können Sie das Spielchen mehrmals betreiben. Dabei werden Sie sehen, daß die Differenz immer geringer wird. Und vergessen Sie nicht jedesmal wieder neu abzuspichern.

Das abgebildete MIKROMON-Listing ist die Version für den CBM 8032. Wenn Sie Besitzer eines CBM 3032 oder 4032 sind, dann müssen Sie nach der Übertragung noch folgende Speicherstelle auf die entsprechenden Werte berichtigen, damit Sie keine Schwierigkeiten mit der Bildschirmausgabe haben.

Table with 2 columns: Address (e.g., 5B94, 5B95) and Value (e.g., 20, D5). Includes values like FDD, DC, A9, 08, 8C, 02, A6, FB, AS, FC, 20.

Die Besitzer der Rechner mit dem 40-Zeichen Bildschirm übertragen das Listing wie die 8032-Anwender. Dann werden die entsprechenden Speicherstellen nach dieser Tabelle abgeändert. Anschließend sollte sofort auf Kassette oder Diskette abgespeichert werden. Danach kann dann der Test mit dem Prüfprogramm erfolgen. Das Prüfprogramm erkennt von ganz alleine, daß dies ein Rechner mit 40-Zeichen Bildschirm oder mit 80-Zeichen ist.

Die MIKROMON-Version für den kleineren Bildschirm stellt ganz automatisch fest, ob es sich um das Betriebssystem für den CBM 3032 oder für den CBM 4032 handelt. Das heißt, daß diese Version für beide Rechner Typen gleichzeitig verwendet werden kann.

Zum Abschluß über MIKROMON wäre noch folgende Bemerkung zu machen: Ist das Programm zum wochenlangen Eintippen nach Feierabend zu mühsam und zu umfangreich, so können Sie es direkt über den Franzis-Verlag beim Autor bestellen. Der Preis liegt bei etwa 150 DM. Bei einer etwaigen Bestellung vergessen Sie bitte nicht, das Speichermedium (Kassette oder Diskette mit Diskettenformat) anzugeben.

In dieser Buchversion liegt MIKROMON im Speicherbereich zwischen \$5B00 und \$7000. Durch die Befehle „NEW LOCATER“ und „TRANSFER“ läßt sich MIKROMON selbst verschieben, so daß auch eine EPROM-Version möglich ist. Allerdings brauchen Sie dann ein paar Utensilien dazu. Benötigt werden batteriegepufferte Softroms und ein Eprommer. Das ist dann aber schon etwas für fortgeschrittenere Anwender.

Table with 2 columns: Address (e.g., 5B00, 5B10) and Value (e.g., 20, 02, 9E, 02). Includes a hex dump and some assembly-like instructions at the bottom.

5C00	20 D5 F0 20 48 F1 4C F7	5B A9 00 85 96 8D FC 03	Ulp HqlwX) 8
5C10	85 0D BD ER 02 56 20 39	44 AE 94 02 D0 10 20 13	1 16 9d .hPKL(
5C20	41 90 08 20 31 5C 20 93	5C 20 AE 68 D0 EB 4C AB	A q " h " q ")
5C30	5B A2 05 20 F1 60 20 13	68 A2 02 20 F1 60 A9 10	1) 9c " q ")
5C40	20 03 69 A9 10 20 B9 63	A2 04 20 F1 60 A0 10 A2	1) 9c " q ")
5C50	00 A1 FB 20 60 5C 20 3B	69 88 D0 F5 60 4C BE 60	1) 9c " q ")
5C60	29 7F C9 20 B0 02 A9 20	4C 09 60 20 06 40 C9 00	1) 9c " q ")
5C70	F0 19 C9 20 D0 03 20 17	6C C9 58 F0 50 20 71 5D	1) 9c " q ")
5C80	BE E8 02 A2 02 00 20 A7	4C 9B 60 A2 04 20 C1 5C	1) 9c " q ")
5C90	4C 9B 60 20 55 69 AE E7	02 F0 31 CE E7 02 D0 2C	1) 9c " q ")
5CA0	AE E8 02 F0 1A A2 06 20	C1 5C A2 14 20 F1 60 8D	1) 9c " q ")
5CB0	A3 02 20 09 60 EB EC E8	02 D0 F4 A2 03 D0 02 A2	1) 9c " q ")
5CC0	09 20 55 69 CA D0 FA A9	39 8D E7 02 60 A9 00 8D	1) 9c " q ")
5CD0	E7 02 8D E8 02 4C 9A F0	20 09 5C 20 CC FF 20 06	1) 9c " q ")
5CE0	40 C9 00 F0 16 C9 24 F0	24 48 20 9E 5D 68 20 09	1) 9c " q ")
5CF0	40 20 06 60 C9 0D D0 E4	4C D0 5B 20 52 69 20 C5	1) 9c " q ")
5D00	5D 20 06 60 C9 0D F0 F0	20 09 60 D0 F4 A2 00 20	1) 9c " q ")
5D10	82 5D 20 8B 5D 20 55 69	20 55 69 A0 03 D0 02 A0	1) 9c " q ")
5D20	02 84 D1 69 08 85 AF 20	06 60 AA A4 96 D0 36 20	1) 9c " q ")
5D30	06 60 A4 96 D0 2F C6 D1	D0 ED 20 36 68 20 52 69	1) 9c " q ")
5D40	20 06 60 F0 05 20 09 60	D0 F6 20 55 69 A9 00 85	1) 9c " q ")
5D50	AF 20 E4 FE F0 C9 D0 05	20 E4 FE F0 FB C9 20 F0	1) 9c " q ")
5D60	F7 C9 03 D0 B4 20 12 5E	20 55 69 4C 93 60 20 17	1) 9c " q ")
5D70	4C C9 22 D0 7B A2 00 20	06 60 C9 0D F0 0C C9 22	1) 9c " q ")
5D80	F0 08 9D A5 02 EB E8 40	90 ED 60 86 D1 A9 A3 85	1) 9c " q ")
5D90	DA A9 02 85 08 20 CC FC	20 F3 50 4C C9 5D A9 08	1) 9c " q ")
5DA0	B5 D4 85 B0 AC 00 C0 C0	D0 08 20 BA F0 A9 6F 20	1) 9c " q ")
5DB0	20 28 F1 4C F7 5B C0 4C	D0 3A 20 D5 F0 A9 6F 20	1) 9c " q ")
5DC0	43 F1 4C F7 5B A9 6F 85	D3 A9 08 85 D4 85 AF AC	1) 9c " q ")
5DD0	00 C0 C0 40 D0 08 20 B6	F0 A5 D3 20 64 F1 4C F7	1) 9c " q ")
5DE0	5B C0 4C D0 08 20 D2 F0	A5 D3 20 93 F1 4C F7 5B	1) 9c " q ")
5DF0	4C BE 60 40 08 85 D4 A9	60 85 D3 AD 00 C0 C9 4C	1) 9c " q ")
5E00	D0 06 20 66 F4 4C F7 5B	C9 4C D0 E4 20 A5 F4 4C	1) 9c " q ")
5E10	F7 5B A9 00 85 AF AD 00	C0 C9 40 D0 03 4C BF F3	1) 9c " q ")
5E20	C9 4C D0 CC 4C CE F3 A9	02 2C 4C E8 08 A9 0E 28	1) 9c " q ")
5E30	F0 02 09 80 20 09 60 4C	93 60 20 09 5C 20 0E 5D	1) 9c " q ")
5E40	20 8B 5D 20 06 60 8D	FB 00 20 09 60 8D FC 00 20	1) 9c " q ")
5E50	12 5E 20 52 69 A9 24 A2	20 20 29 68 20 13 68 4C	1) 9c " q ")
5E60	93 60 20 60 6C 00 6C 3E	5F A0 08 84 D4 A0 4C 8C	1) 9c " q ")
5E70	87 02 A0 44 68 20 4F 68	29 1F C9 9E 4C 34 8E 44	1) 9c " q ")
5E80	BE 60 20 44 68 20 4F 68	29 1F C9 9E 4C 34 8E 44	1) 9c " q ")
5E90	20 2A 6C A5 FD A6 FE 8D	92 02 8E 93 02 20 3C 68	1) 9c " q ")
5EA0	A5 D4 20 E3 5B A9 02 20	09 60 20 52 69 20 13 61	1) 9c " q ")
5EB0	90 0F AE 94 02 D0 0A A1	FB 20 1A 68 20 3B 69 D0	1) 9c " q ")
5EC0	E9 A9 03 20 09 60 20 EF	60 20 CC FF 20 FB 60 4C	1) 9c " q ")
5ED0	7D 6C 20 09 5C 20 01 68	20 6E 5D 86 D1 20 04 5F	1) 9c " q ")
5EE0	20 8D 5D 20 06 60 20 06	60 A9 00 85 AF AD 00 C0	1) 9c " q ")
5EF0	C9 40 D0 06 20 52 F3 4C	01 5F C9 4C D0 81 20 8C	1) 9c " q ")
5F00	F3 4C 12 6A AD 00 C0 C9	4D D0 03 AC 0A FA C9 4C	1) 9c " q ")
5F10	D0 EA 4C 49 FA 50 C4 49	AD 40 0E DA 4A CB CC 5E	1) 9c " q ")
5F20	53 59 BE 5B 89 5B 68 5C	16 5C D8 5C D8 5C 27 5E	1) 9c " q ")
5F30	3A 5E 62 5E 69 5E 77 5E	82 5E D2 5E 69 5B 00 5B	1) 9c " q ")
5F40	41 30 32 31 3B 31 AA A4	AA AA AA AA AA AA AA AA	1) 9c " q ")
5F50	48 C9 C0 D0 06 20 73 5F	4C 62 5F C9 4D 03 20	1) 9c " q ")
5F60	73 5F 68 A2 0C DD 15 5F	F0 06 CA 10 FB 4C BE 60	1) 9c " q ")
5F70	4C 70 5B A9 04 4C E3 5B	AA AA 20 CC FF 4C 9B 60	1) 9c " q ")
5F80	4B C9 0D D0 04 A9 FF 85	5E E6 5E 68 4C D2 FF 60	1) 9c " q ")
5F90	A2 2C 38 5E 20 A4 4C F1	60 AA AA AA AA AA AA AA	1) 9c " q ")
5FA0	09 00 BD B5 5F 20 D2 FF	EB E0 1C D0 F5 A5 90 A6	1) 9c " q ")
5FB0	91 60 AA AA AA 93 0D 0D	12 4D 49 4B 52 4F 4D 4F	1) 9c " q ")
5FC0	4E 92 20 20 2A 42 41 4D	2A 20 20 3B 2E 38 32 0D	1) 9c " q ")
5FD0	0E AA AA AA AA AA AA AA	AA AA AA AA AA AA AA AA	1) 9c " q ")
5FE0	AA AA AA AA AA AA AA AA	AA AA AA AA AA AA AA AA	1) 9c " q ")
5FF0	AA AA AA AA AA AA AA AA	AA AA AA AA AA AA AA AA	1) 9c " q ")
6000	4C 0C 60 4C 6F 60 4C CF	FF 4C 80 5F 7B A5 92 AA	1) 9c " q ")
6010	93 8D E5 02 BE E6 02 AD	F6 6F AE F7 6F 8D E3 06	1) 9c " q ")

6020	BE E4 02 AD F0 6F AE F1	6F 85 92 86 93 A5 90 A6	d -po, qd % &
6030	91 CD E5 D0 05 EC EF 6F	F0 10 8D 9E 02 BE 9F	MncP 1ccp -10-m
6040	02 AD E5 AF AE EF 85 90	86 91 AD EC 4F AE ED	-no, 00
6050	6F E0 80 B0 08 85 34 86	35 85 80 86 31 A9 10 8D	o " 0 4 5 0 1)
6060	84 02 8D 85 02 A9 00 8D	86 02 8D A2 02 5B 00 38) " X B
6070	AD 7B 02 E9 01 8D 7B 02	AD 7A 02 E9 00 8D 7A 02	-a i a -z i z
6080	20 55 69 A2 42 A9 2A 20	29 68 A9 52 D0 23 A9 3F	U i (B) * (h) RE# ?
6090	20 09 60 20 55 69 A9 2E	20 09 60 A9 00 8D 94 02	" ") (U i)
60A0	BD A2 02 AD FF 9A 20 F0	F9 C9 2E F0 A9 20 F0	" ") (U i)
60B0	F5 A2 1D D2 92 6F D0 13	8D 87 02 8A 0A AA BD B0	u " u op *# =0
60C0	6F 85 FB BD B1 6F 85 FC	6C FB 00 CA 10 E5 6C E3	o a=10 01 a j e1c
60D0	02 A2 02 D0 02 A2 00 R4	FB 00 08 B4 FC 02 03 EE	" " P " 48P 48P n
60E0	94 02 D6 FC D6 FB 60 A9	00 8D 8C 02 20 4F 62 AE	vava ")
60F0	09 20 52 69 CA D0 FA 60	A2 02 85 FA 48 BD 91 02	R1JpZ " 5ZH=
6100	95 FA 68 9D 91 02 CA D0	F1 60 AD 92 02 AC 93 02	zh JpZ -
6110	4C 17 61 A5 FD A4 FE 3B	E5 FB 8D 91 02 98 E5 FC	L aXU\$PpEa e0
6120	48 0D 91 02 60 A9 00 F0	02 A9 01 8D 95 02 20 E6	(") P) f
6130	67 20 55 69 20 13 61 20	3C 68 90 1B 20 0A 61 B0	g U i a <h ao
6140	03 4C C5 61 20 7F 61 E6	FD D0 02 E6 FE 20 3B 69	LEA a fUP f i i
6150	AC 94 02 D0 45 F0 E5 20	0A 61 1B AD 91 02 65 FD	" PEPE a - eu
6160	85 FD 98 65 FE 85 FE 20	FB 60 20 7F 61 20 0A 61	u er R x " a
6170	B0 53 20 D1 60 20 D5 60	AC 94 02 D0 1D F0 EB A2	OS 0 " U " P pk"
6180	00 A1 FB AC 95 02 F0 02	B1 FD C1 FD F0 0B 20 13	1a " P uAlp
6190	68 20 52 69 20 AE 68 F0	01 60 4C 93 60 20 01 68	h R i .hp " L " h
61A0	20 0B 68 20 A4 68 20 4F	68 90 17 8D 89 02 AE 94	P a - a : i
61B0	02 D0 12 20 13 61 90 0D	AD 89 02 81 FB 20 3B 69	P i L " L " h h \$
61C0	D0 EC 4C 8E 60 4C 93 60	20 01 68 20 08 68 20 A4	h " \$h i P \$h #
61D0	68 A2 00 A4 68 C9 27 D0	14 20 A4 68 A3 03 02	h " I " P " Pqp
61E0	EB 20 06 60 C9 0D F0 22	E0 20 D0 F1 F0 1C BE 97	wh L # h " I P
61F0	02 20 77 68 90 CC 9D A3	02 EC BE 8E 02 20 55 69	" ch < " P1 U i
6200	09 20 6F 68 90 BC E0 8C	02 EC BE 8E 02 20 55 69	" lau# P Hh1
6210	A2 00 A0 00 B1 FB DD A3	02 D0 0A CB EB EC 88 02	Pr a i j P a
6220	D0 F0 20 BE 61 20 3B 69	AC 94 02 D0 05 20 13 61	o L " 9d " a
6230	B0 DE 4C 93 60 20 3F 64	24 13 61 90 0D A0 2C 20	g " < b " Pn 3cc
6240	E7 60 20 AB 62 20 AE 68	D0 EE 20 B3 65 D0 E3 20	Gi h R i nh ch
6250	47 69 20 13 68 20 52 69	20 0E 4E 48 20 0F 63 68	" c " " P " P -
6260	20 22 63 A2 06 E0 0D 14	AC 8B 02 0E AD 96	ih1a0 " b Pq
6270	02 C9 EB B1 FB B0 1D 20	A1 62 8B D0 F1 0E 96 02	=in -seomp H " b
6280	90 0E 8D E9 6E 20 AD 65	8B EF AE F0 93 20 AD 65	JFR " 7b#P H " b
6290	CA D0 D2 60 20 B7 62 AA	EB D0 01 CB 98 20 A1 62	H " (j J " P
62A0	BA BE 8B 02 20 1A 68 AE	8B 02 60 AD 8B 02 20 B6	b a a " 88a " e a
62B0	62 85 FB 84 FC 60 3B 44	FC AA 10 01 8B 65 FB 90	H " (j J " P
62C0	01 CB 60 8A 4A 9A 00 8A	B0 17 C9 22 F0 13 29 07	J * = n0 JJJJ) P
62D0	09 80 4A AA BD 98 6E B0	04 AA 4A 4A 4A 29 0F D0) * = dn)
62E0	04 80 80 A9 00 AA BD DC	6E 8D 96 02 29 03 8D 8B) * = dn)
62F0	02 9B 29 BF AA 98 A0 03	E0 8A F0 0B 4A 90 0B 4A) * = dn)
6300	4A 09 20 8B D0 FA CB 8B	D0 F2 60 B1 FB 20 A1 62	J PzH Pr 1a 1b
6310	A2 01 20 F1 60 CC 8B 02	CB 90 F0 A2 03 C0 03 90	" q L H p " s
6320	F1 60 AB B9 FA 6E 8D 92	02 B9 36 6F 8D 93 02 49	q " (vvn 960)
6330	00 A0 05 0E 93 02 2E 92	02 2A 8B D0 F6 69 3F 20	* P v1 ?
6340	09 60 CA D0 EA 4C 52 69	20 01 68 A9 03 20 AC 63	* P v1 ?
6350	A0 2C 4C 50 65 8D 05 01	CD FB 6F D0 0B 80 06 01	* P v1 ?
6360	CD F7 4F D0 03 20 D7 68	A5 97 CD 83 02 F0 0A 8D	Myop Whz M P
6370	83 02 A9 10 8D 84 02 D0	24 C9 FF F0 20 AD 84 02) * = dn)
6380	F0 05 CE 84 02 D0 16 CE	85 02 D0 11 A9 02 8D 85	P N P N P
6390	02 A5 9E D0 08 A9 00 85	97 A9 9E 02 85 AB AD F3 6F	% P) (-so
63A0	48 AD F2 6F 48 08 48 48	48 4C 9E 02 8D 89 02 48	H -roH HHHH) H
63B0	20 A4 68 20 19 69 D0 FB	68 48 4C 9E 4C AE 62 20 39	\$h iPhLlLb 9
63C0	64 AE 94 02 D0 0D 20 13	61 90 08 20 D6 63 20 AE	d " P a Vc "
63D0	68 D0 EE 4C 4A 62 20 55	69 A2 2E A9 3A 20 29 68	hPhLb U i " :) h
63E0	20 52 69 20 13 68 A9 10	20 03 69 A9 10 20 89 63	R i h) i) 9c
63F0	A9 12 20 09 60 A0 10 A2	00 A1 FB 29 7E C9 20 80) * = dn)
6400	02 A9 2E 20 09 60 C9 92	F0 04 C9 62 D0 0A A9 14) * = dn)
6410	20 09 60 A9 22 20 09 60	20 3B 69 88 D0 DR A9 92) * = dn)
6420	4C 09 60 20 01 68 A9 10	20 AC 63 20 83 65 20 D6	L " h) " c 3e v
6430	63 A9 3A 8D 6F 02 4C 5C	65 20 01 68 85 FD 86 FE	c) : o Loe h u R

6C90	A0 00 BC 92 02 BC 93 02 20 13 61 90 1D AD 94 02		
6C91	D0 1B AD 00 1B B1 FB 4D 92 02 BD 92 02 98 6D 93		
6C92	02 8D 93 02 20 1B B1 FB 4D 92 02 BD 92 02 98 6D 93		
6C93	AD 92 02 20 1A 68 4C 93 60 AD AD 02 D0 04 A5 9E		
6C94	D0 06 6B AB 68 AA 68 40 AD 6F 02 C9 11 D0 7D A5		
6C95	D8 C9 1B D0 ED A5 CA 85 FD A5 C5 85 FE A9 19 8D		
6C96	9C 02 A0 01 20 BC 6E C9 3A F0 1A C9 2C F0 16 C9		
6C97	24 F0 12 CE 9C 02 F0 CA 3B A5 FD E9 50 85 FD B0		
6D00	E1 C6 FE D0 DD BD 87 02 20 45 6E 8D 87 02		
6D01	C9 3A D0 11 18 A5 FB 69 10 85 FE 90 02 E6 FC 20		
6D02	D6 63 4C 39 6D C9 24 F0 1A 20 0E 6E 20 AB 62 A9		
6D03	00 8D BC 02 40 2C 20 4F 62 A9 00 85 9E 4C 4A 62		
6D04	4C C2 6C 20 3B 69 20 B3 6A 4C 39 6D C9 91 D0 F0		
6D05	A5 D8 D0 EC A5 C4 85 FD A5 C5 85 FE A9 19 8D 9C		
6D06	02 A0 01 20 8C 6E C9 3A F0 1A C9 2C F0 16 C9 24		
6D07	F0 12 CE 9C 02 F0 15 18 A5 FB 69 10 85 FE 90 02		
6D08	E6 FE D0 DD BD 87 02 20 45 6E 8D 87 02 20 45 6E		
6D09	87 02 C9 3A F0 06 C9 24 F0 1D D0 27 20 15 6E 3B		
6D0A	A5 FB E9 10 85 FB 80 02 C6 FC 20 D9 63 A9 00 85		
6D0B	9E 20 40 6E 4C 96 60 20 15 6E 20 D5 60 20 B6 64		
6D0C	4C AD 4D 20 15 6E A5 FB A6 FC 85 FD B6 FE A9 10		
6D0D	BD 9C 02 38 A5 FD ED 9C 02 85 FB A5 FE 90 00 85		
6D0E	FC 20 0E 6E 20 AB 62 20 13 61 F0 07 B0 F3 CE 9C		
6D0F	02 D0 0E 0E EE 8B 02 AD 8B 02 20 B9 63 A2 00 A1 FB		
6E00	BE BC 02 A9 2C 20 AD 69 20 22 62 62 4C AD 6D A2 00		
6E01	A1 FB 4C C3 62 A9 87 85 C8 85 FE A9 00 85 C7 A9		
6E02	50 85 FD A0 80 A2 08 8B B1 C7 91 FD 98 D0 FB C6		
6E03	C8 C6 FE CA D0 F1 A2 4F A9 20 9D 00 80 CA 10 FA		
6E04	A9 13 4C 09 60 C0 50 D0 02 3B 60 20 8C 6E C9 20		
6E05	F0 F3 8B 20 75 6E AA 20 75 6E 85 FB B6 FC A9 FE		
6E06	8D A2 02 85 A7 A5 AA F0 0A A5 A9 A4 C6 91 C4 A9		
6E07	00 85 AA 18 60 20 8C 6E 20 99 68 0A 0A 0A 8D		
6E08	97 02 20 BC 6E 20 8C 6E 0D 97 02 60 B1 FD C8 09		
6E09	7F C9 20 B0 02 09 40 60 40 02 45 03 D0 08 40 09		
6E0A	30 22 45 B3 D0 08 40 09 00 22 44 33 D0 08 40 09		
6E0B	40 02 45 B3 D0 08 40 09 00 22 44 33 D0 08 40 09		
6E0C	11 22 44 33 D0 BC 44 9A 10 22 44 33 D0 08 40 09		
6E0D	10 22 44 33 D0 08 40 09 62 13 78 A9 00 21 81 82		
6E0E	00 00 59 4D 91 92 86 4A 85 9D 2C 29 2C 23 28 24		
6E0F	59 00 58 24 24 00 1C 8A 1C 23 5D 8B 1B A1 9D 8A		
6F00	1D 23 9D 8B 1D A1 00 29 19 AE 69 A8 19 23 24 53		
6F01	1B 23 24 53 19 A1 00 1A 5B 5B A5 69 24 24 AE AE		
6F02	AB AD 29 00 7C 00 15 9C 6D 9C A5 69 29 53 B4 13		
6F03	34 11 A5 69 23 A0 D8 62 5A 48 26 62 94 8B 54 44		
6F04	C8 54 68 44 EB 94 00 B4 08 B4 74 B4 28 6E 74 F4		
6F05	CC 4A 72 F2 A4 8A 00 A4 A2 A2 74 74 74 72 44 68		
6F06	B2 32 B2 00 22 00 1A 1A 26 26 72 72 8B C8 C4 CA		
6F07	26 48 44 44 A2 C8 0D 20 20 20 50 43 20 20 49		
6F08	52 51 20 20 53 52 20 41 43 20 58 52 20 59 52 20		
6F09	53 50 41 42 43 44 46 47 48 4C 4D 4E 51 52 53 54		
6FA0	57 58 2C 3A 3B 24 23 22 28 2D 4F 5A 4B 25 26 45		
6FB0	4D 64 38 67 25 61 35 62 9D 61 B5 66 C8 61 BF 69		
6FC0	BE 63 55 67 89 66 5A 6A BF 69 29 61 C9 66 B5 69		
6FD0	48 63 23 64 93 69 6A 6A 4A 6B BD 68 60 4C 43 63		
6FE0	7B 6A 1F 6C 59 6C E2 68 77 6C 82 69 00 60 55 63		
6FF0	EB 65 B9 6C C6 65 BE 60 BC 68 30 35 32 37 38 31		

2.8 INPUT/OUTPUT (I/O) → Ein/Ausgabe

Wenn man den **eh**-Ehlsatz der CPU 6502 einigermassen kennt, wird man feststellen, daß der Prozessor keinen eigenen Befehl besitzt, der für die Steuerung der Peripheriegeräte zuständig ist. Die Steuerung läuft hier anders ab, nämlich über das sogenannte **INTERFACE**. Ohne Interfacebausteine läuft nichts. Wenn am Computer Peripheriegeräte angeschlossen werden, können diese nicht direkt mit der CPU verbunden werden. Da z.B. die Tastatur oder der Bildschirm meist von anderen Herstellern stammen und nicht zur eigentlichen Bausteinfamilie kompatibel sind, müssen natürlich Anpassungen vorgenommen werden. Zudem hängen am Rechner meist auch noch ein Diskettenlaufwerk und ein Drucker. Die Schnittstellen zwischen Prozessor und den Peripheriegeräten werden durch Interfacebausteine realisiert. Da Signale von beiden Seiten unterschiedlich gesteuert werden, synchronisieren die Interfacebausteine den Signalaustausch.

Die Anpassung beim CBM übernehmen drei Interfaceadapter. Diese sind auf der einen Seite mit der Peripherie und auf der anderen Seite mit der CPU verbunden. Bevor man irgendwelche I/O-Programmierungen betrachtet, sollte man sich die Interfaceadapter im einzelnen betrachten. Die drei Adapter beim CBM bestehen aus 2 PIA-Bausteinen und einem VIA-Baustein.

2.8.1 PIA (MCS 6520)

PIA ist die Abkürzung für „Periphere Interface Adapter“, was eigentlich keiner Übersetzung mehr bedarf. Die nachfolgende Abbildung zeigt ein ganz einfaches Blockschaltbild des PIA. Wie wird nun ein solcher Baustein angesteuert? An sich ganz einfach. Die Interfacebausteine haben Leitungen, die mit den Signalen „CHIP SELECT“ (CS0, CS1 u. CS2 = Bausteinauswahl) angesprochen werden. Das heißt, über diese Leitungen wird eine Basisadresse im Hauptspeicher des Computers festgelegt. Das bezeichnet man oft auch mit „I/O MEMORY MAPPED“. Dadurch können die Interfacechips durch normale Speicherbefehle von der CPU aus gesteuert werden. Es wird z.B. ein entsprechendes Bitmuster mit

LDA #Wert
geladen und anschließend mit
STA PIA-Adresse

in den Baustein abgesetzt. Der PIA hat intern ebenso wie die CPU mehrere Register und die dazugehörigen Verbindungsleitungen (interner BUS). 6 Register können nun von der CPU aus durch Speicherbefehle manipuliert werden. Diese 6 Register sind für die gesamte Steuerung des PIA verantwortlich und tragen folgende Bezeichnung:

1. Steuerregister CRA
2. Datenregisterregister DDRA
3. Interfacepuffer (Tor A)
4. Steuerregister CRA
5. Datenregisterregister DDRB
6. Interfacepuffer (Tor B)