

Turbo-Tape

In der Assemblerecke möchten wir Ihnen diesmal einen kleinen Knüller präsentieren, nämlich einen Speeder für das sonst so langsame Atari-Kassettenlaufwerk. Sicher waren einige Benutzer dieses Gerätes schon etwas neidisch auf die C64-Besitzer, für die solche Beschleuniger in reichlichem Maße zur Verfügung stehen.

In der Dezember/Januar Ausgabe der CK-Computer Kontakt hatten wir bereits ein kleines Programm abgedruckt, das diesen Zweck erfüllen sollte. Leider hatte es einige Schwierigkeiten mit der Datensicherheit, die unseren Leser Arndt Bär nicht ruhen ließen. Er schickte uns seinen in vielen Punkten verbesserten Kassetten-Speeder, den wir nun vorstellen wollen.

Das Programm "CBAUD" arbeitet mit 1200 Baud anstelle der üblichen 600, was immerhin (fast) eine Verdoppelung der Aufzeichnungs- und Lesegeschwindigkeit mit sich bringt. Versuche ergaben, dass ein etwa 5 KByte langes Basic-Programm im normalen Modus in ca. 100 Sekunden abgespeichert wird (ohne Vorspann gemessen). Schaltet man "CBAUD" zu, so verkürzt sich diese Zeit auf etwa 55 Sekunden!

Sie könnten jetzt einwenden, dass mit einer Verdoppelung der Baudrate auch eine Halbierung der Ladezeit einhergehen müsste. Leider stimmt diese Rechnung nicht ganz, da der Atari nach jeweils einem Block von 128 Byte eine Pause einlegt, die von "CBAUD" nicht verändert wird. Eine Steigerung der Aufzeichnungsgeschwindigkeit um fast das Doppelte ist aber wohl auch nicht zu verachten.

CBAUD in der Praxis

Um Programme schneller laden und speichern zu können, ist nur Listing 1 einzutippen. Speichern Sie es (sicherheitshalber) erst einmal ab und starten es. Nach kurzer Wartezeit können Sie entscheiden, ob das "CBAUD"-Maschinenprogramm auf eine Kassette oder Diskette geschrieben werden soll. Um die erste Möglichkeit zu realisieren, legt man die Kassette ein, spult zurück und drückt dann PLAY und RECORD am Recorder sowie RETURN auf der Tastatur. Nachdem das Programm fertig auf Kassette vorliegt, lässt es sich in Zukunft ganz normal booten, d.h. die Kassette wird eingelegt, die START-Taste betätigt und der Computer eingeschaltet. Alles weitere funktioniert dann wie gehabt.

Wählen Sie dagegen die Diskettenversion, so wird ein File CBAUD.COM erzeugt. Dieses lässt sich ganz normal mit der DOS-Option L laden. Unabhängig davon, ob man sich für Kassette oder Diskette entscheidet, erscheint nach Laden von "CBAUD" ein kleines Menü. Sie haben die Wahl, ob Basic aufgerufen werden soll oder ein Maschinenprogramm von Kassette zu laden ist.. Mit dieser Möglichkeit können Sie Programme laden und starten, die in einem bootfähigen Kassettenformat aufgezeichnet sind. Voraussetzung ist natürlich, dass diese zuvor mit "CBAUD" gespeichert bzw. vom Convert-Programm, zu dem wir später kommen, auf das 1200-BaudFormat übertragen wurden. Die Laderoutine für Maschinenprogramme ist ab Adresse \$9800 abgelegt. Es ist daher problemlos möglich, Programme in den Speicher zu laden, die an der unteren Speichergrenze (\$700) beginnen.

Eine dritte Option erlaubt es, ein Boot-Programm zu laden, es dann aber nicht zu starten, sondern stattdessen Basic (oder DOS, falls vorhanden) aufzurufen. Auf diese Weise könnte man Maschinenprogramme laden und mit geeigneten anderen Programmen dann untersuchen.

Mit Basic

Haben Sie die erste Option gewählt, können die schnellen Kassetten-Operationen in Basic genutzt werden. Dazu sind die Programme einfach mit CSAVE zu speichern und mit CLOAD wieder einzulesen. Es ist auch möglich, nach wie vor mit 600 Baud zu arbeiten, wenn Sie z.B. ein Programm laden wollen, das noch im alten Format gespeichert ist. Ein POKE Befehl hilft weiter:

POKE 54017,253 schaltet auf das langsame Format um. Wieder Zuschaltung des Turbo ist dann möglich mit: POKE 54017,252.

Hier sei noch angemerkt, dass diese Befehle zwischen dem Betriebssystem im ROM und einem zweiten, modifizierten OS im dahinterliegenden RAM hin- und herschalten. Aus diesem Grund läuft das Programm auch nur auf Rechnern ab 64 KByte, denn nur diese Modelle besitzen RAM hinter den OS-ROMs.

So funktioniert es!

Damit ist das Geheimnis des Turbo-Tapes auch fast gelüftet, denn das Programm "CBAUD" macht nichts anderes, als das Betriebssystem des Atari vom ROM ins RAM zu kopieren und dabei einige Befehle der SIO (Serial Input/ Output)-Routine zu ändern. Hier wird aber nicht nur (wie beim Turbo-Tape der Ausgabe 12/1) die Baud-Rate gesteigert, sondern auch noch etwas Aufwand zur Erhöhung der Datensicherheit getrieben.

Kassettenrecorder, besonders Billiggeräte, wie sie leider oft für Computeranwendungen eingesetzt werden, sind mit einem großen Mangel behaftet: Sie halten die Geschwindigkeit des Bandes nicht besonders konstant. Die Entwickler des Atari-Betriebssystems haben dies bereits berücksichtigt und ein Aufzeichnungsverfahren erstellt, das diese Schwankungen auszugleichen versucht. Vor Aufzeichnungen eines Records (= 128 Byte) wird immer zweimal \$AA auf Band geschrieben. Diesen Wert hat man gewählt, weil bei \$AA (binär % 10101010) immer ein Eins-Bit mit einem Null-Bit abwechselt. Die Leseroutine nutzt diesen Sachverhalt und stoppt die Zeit, in der die beiden Bytes eingelesen werden. Diese wird mittels einer im ROM gespeicherten Tabelle in einen Baud-Wert umgerechnet, welcher der tatsächlichen Bandgeschwindigkeit entspricht.

Arndt Bär hat nun herausgefunden, dass diese Korrektur im Betriebssystem recht ineffektiv programmiert wurde, da sie eine Schwankung von 50 Baud mit nur 4 Korrekturwerten unterstützt. Arndt hat Tabelle und Berechnung bei "CBAUD" verbessert und erreicht somit eine viel höhere Datensicherheit.

Hier noch ein paar Tipps des Autors, wie sich Datenverluste vermeiden lassen. Der Recorder sollte immer das erste Gerät in der Peripheriekette sein, da ein langer Übertragungsweg stören kann. Datenkabel und Recorder sind mindestens 50 cm vom Fernseher bzw. Monitor entfernt zu platzieren. Besonders bei 1200 Baud empfiehlt es sich, gute Kassetten zu verwenden, am besten solche, die das Band über einen größeren Radius führen (z.B. BASF). Andere Führungen (scharfe Kanten, Zapfen) können zu Querknicken im Bandmaterial führen.

Kopierprogramm

Das Programm in Listing 2 ermöglicht es, Files zwischen den 600- und 1200-Baud-Formaten zu konvertieren. Dabei spielt es keine Rolle, ob es sich um Maschinenprogramme oder Basic-Files handelt. "Bärcopy 1200" setzt voraus, dass zuvor "CBAUD" geladen wurde. Beim Laden und Speichern eines Programms ist jeweils einzugeben, ob das Format mit 600 oder das mit 1200 Baud benutzt werden soll.

Auf zwei Besonderheiten sei noch hingewiesen. Wenn ein Programm keinen End-of-File Record enthält oder ein Fehler beim Laden auftritt, beendet das Kopierprogramm seine Tätigkeit mit einer Fehlermeldung. Drückt man in diesem Fall ESC, lassen sich die im Speicher befindlichen Daten trotzdem verwenden. Des weiteren benutzt das Programm den ganzen verfügbaren Speicher. Wenn Sie ein DOS geladen hatten, geht dieses verloren. Daher ist es nach der Verwendung von "Bärcopy" neu zu booten (nur für Disk-Benutzer).

Ausnahmsweise haben wir diesmal kein Assembler-Listing abgedruckt. Der Grund liegt ganz einfach darin, dass es zu lang war. Interessierte finden es aber auf der CK-Programmservice-Diskette (Format: Atari-Assembler bzw. MAC/65). Mit dem eigentlich für diese Ausgabe vorgesehenen Thema "Selbstbootende Disketten, Teil II" werden wir uns im nächsten Heft beschäftigen.

Arndt Bär/Peter Finzel (1987)

Info: Das Assembler-Listing zu CBAUD von der Programmservice-Diskette ist in dieser Version der Assemblerecke enthalten!

```

10 REM *****
20 REM *          BASIC-LADER          *
30 REM *          CBAUD 1200/1 V1.0    *
40 REM * (C) 1986 BY ARNDT BAER      *
50 REM *****
60 ? CHR$(125):? :? :? :POKE 756,204:POKE 752,1
70 ? "          BASIC-LADER          "
80 ? "          CBAUD 1200/1 V1.0    "
90 ? " (C) 1986 BY ARNDT BAER      "
100 ? :? :? :? " BITTE EINEN MOMENT GEDULD"
110 GOSUB 210:GOSUB 500
120 POSITION 2,10:? " 1-DISKETTENFILE 'D:CBAUD.COM' ODER"
130 ? " 2-CASSETTENFILE          SCHREIBEN ?"
135 CLOSE #1:OPEN #1,4,0,"K:":GET #1,KEY:CLOSE #1
140 IF KEY<>49 AND KEY<>50 THEN GOTO 135
145 TRAP 190:IF KEY=49 THEN CLOSE #1:OPEN #1,8,0,"D:CBAUD.COM"
147 A=USR(ADR(ML$),16,11,38906,847):GOTO 160
150 CLOSE #1:OPEN #1,8,128,"C:":A=USR(ADR(ML$),16,11,38912,896)
160 IF A>127 THEN GOTO 190
170 CLOSE #1:? :? "FERTIG !!!":END
190 CLOSE #1:? :? CHR$(125);"I/O - FEHLER NR.":PEEK(195):GOTO 120
210 DIM ML$(40)
220 RESTORE 230:FOR I=1 TO 35:READ A:ML$(I,I)=CHR$(A):NEXT I
230 DATA 104,104,104,170,104,104
240 DATA 157,66,3,104,157,69,3
250 DATA 104,157,68,3,104,157,73
260 DATA 3,104,157,72,3,32,86,228
270 DATA 132,212,169,0,133,213,96
280 RETURN
300 DATA 255,255,0,152,60,155,0,7,884
301 DATA 0,152,13,152,169,60,141,2,689
302 DATA 211,24,96,32,90,154,162,0,769
303 DATA 240,2,162,1,169,204,141,244,1163
304 DATA 2,134,203,188,143,153,32,26,881
305 DATA 153,166,203,232,224,161,208,241,1588
306 DATA 169,255,141,80,3,32,36,153,869
307 DATA 201,65,208,31,165,9,41,1,721
308 DATA 133,9,173,118,0,141,1,211,786
309 DATA 165,206,141,14,210,133,16,160,1045
310 DATA 125,32,26,153,169,224,141,244,1114
311 DATA 2,24,96,162,0,142,128,153,707
312 DATA 162,60,142,2,211,201,66,240,1084
313 DATA 7,201,67,208,200,141,128,153,1105
314 DATA 173,118,0,9,2,141,1,211,655
315 DATA 169,0,133,65,169,153,141,85,915
316 DATA 3,169,141,141,84,3,169,3,713
317 DATA 141,82,3,169,4,141,90,3,633
318 DATA 169,128,141,91,3,32,45,153,762
319 DATA 176,128,162,11,189,129,153,157,1105
320 DATA 80,3,202,208,247,32,45,153,970
321 DATA 144,3,76,20,152,24,173,130,722
322 DATA 4,105,128,141,84,3,173,131,769
323 DATA 4,105,0,141,85,3,169,0,507
324 DATA 141,88,3,206,129,4,173,129,873
325 DATA 4,74,141,89,3,110,88,3,512

```

```

326 DATA 32,45,153,144,3,76,20,152,625
327 DATA 173,130,4,133,203,173,131,4,951
328 DATA 133,204,160,127,185,128,4,145,1086
329 DATA 203,24,136,16,247,173,132,4,935
330 DATA 133,2,173,133,4,133,3,169,750
331 DATA 224,141,244,2,32,7,153,32,835
332 DATA 23,153,173,128,153,208,3,108,949
333 DATA 10,0,76,60,152,24,165,203,690
334 DATA 105,6,133,203,165,204,105,0,921
335 DATA 133,204,108,203,0,108,2,0,758
336 DATA 173,7,228,72,173,6,228,72,959
337 DATA 152,96,173,37,228,72,173,36,967
338 DATA 228,72,96,162,16,32,86,228,920
339 DATA 48,2,24,96,132,204,192,136,834
340 DATA 240,248,169,60,141,2,211,162,1233
341 DATA 0,134,203,188,72,154,32,26,809
342 DATA 153,166,203,232,224,18,208,241,1445
343 DATA 169,0,133,213,165,204,133,212, 1229
344 DATA 32,170,217,32,230,216,162,1,1060
345 DATA 134,203,188,128,5,48,9,32,747
346 DATA 26,153,166,203,232,24,144,240,1188
347 DATA 152,41,127,168,32,26,153,160,859
348 DATA 155,32,26,153,56,96,0,255,773
349 DATA 1,7,1,128,4,179,253,128,701
350 DATA 0,4,128,67,58,125,155,4,541
351 DATA 0,0,0,0,0,0,0,0,0
352 DATA 0,0,0,0,7,155,1,67,230
353 DATA 66,65,85,68,32,49,50,48,463
354 DATA 48,47,49,2,155,1,40,99,441
355 DATA 41,32,49,57,56,54,32,98,419
356 DATA 121,32,2,155,1,32,32,65,440
357 DATA 114,110,100,116,32,66,11,114,663
358 DATA 32,2,155,5,3,3,3,3,206
359 DATA 3,3,3,3,3,3,3,3,24
360 DATA 6,155,155,4,0,7,155,1,483
361 DATA 193,2,32,46,46,46,66,65,496
362 DATA 83,73,67,47,68,79,83,32,532
363 DATA 155,5,3,6,155,4,0,7,335
364 DATA 155,1,194,2,32,46,46,46,522
365 DATA 77,76,45,80,82,79,71,82,592
366 DATA 65,77,77,32,76,65,68,69,529
367 DATA 78,155,5,3,6,155,4,0,406
368 DATA 7,155,1,195,2,32,46,46,484
369 DATA 46,194,32,85,78,68,32,193,728
370 DATA 155,5,3,6,155,155,186,2,667
371 DATA 180,2,174,2,166,2,158,2,686
372 DATA 240,2,229,2,220,2,213,2,910
373 DATA 205,2,199,2,190,2,253,155,1008
374 DATA 73,47,79,32,45,32,70,101,479
375 DATA 104,108,101,114,32,78,114,46,697
376 DATA 173,1,211,41,254,133,118,165,1096
377 DATA 16,133,206,169,0,141,14,212,891
378 DATA 141,14,210,133,16,133,203,169,1019
379 DATA 192,133,204,169,208,133,205,32,1276
380 DATA 219,154,169,214,133,204,169,0,1262

```

```
381 DATA 133,203,133,205,32,219,154,169,1248
382 DATA 254,141,1,211,169,230,141,163,1310
383 DATA 235,141,65,253,169,2,141,168,1174
384 DATA 235,141,70,253,169,234,141,3,1246
385 DATA 237,141,4,237,141,5,237,169,1171
386 DATA 118,141,9,237,160,23,185,48,921
387 DATA 154,153,249,237,136,16,247,165,1357
388 DATA 206,133,16,141,14,210,169,64,953
389 DATA 141,14,212,160,63,185,253,154,1182
390 DATA 153,0,206,136,16,247,169,2,929
391 DATA 141,198,2,9,14,141,197,2,704
392 DATA 96,160,0,162,254,177,203,142,1194
393 DATA 1,211,145,203,162,255,142,1,1120
394 DATA 211,230,203,165,203,208,236,24,1480
395 DATA 165,204,105,1,133,204,197,205,1214
396 DATA 208,225,96,0,0,0,0,255,784
397 DATA 255,255,0,21,21,21,21,21,615
398 DATA 21,21,21,112,112,112,112,112,623
399 DATA 112,112,112,255,170,170,170,170,1271
400 DATA 0,0,0,0,0,0,0,15,15
401 DATA 23,21,20,21,20,20,18,10,153
402 DATA 0,0,0,112,176,144,160,160,752
403 DATA 0,0,0,0,0,0,0,224,224
404 DATA 208,176,112,226,2,227,2,6,959
405 DATA 152,224,2,225,2,13,152,21,791
500 RESTORE 300:FOR I=0 TO 105:SUM=0
510 FOR T=0 TO 7:READ A:POKE 38906+I*8+T,A:SUM=SUM+A:NEXT T
520 READ A
525 IF A<>SUM THEN ?:" CHR$(125);" FEHLER IN ZEILE ";300+I:L. 300+I:END
530 NEXT I
540 RETURN
```

```

10 REM *****
20 REM *          COPY 1202/600          *
30 REM *      (c) 1986 by Arndt Baer      *
40 REM *****
50 REM * ZUERST CBAUD 1200/1 LAUFEN *
60 REM *          LASSEN !!!!!!!          *
70 REM *****
80 POKE 65,1:RAM=INT(PEEK(54017)/2)*2:POKE 54017,RAM+1
90 DIM ML$(51):BASTOP=INT((PEEK(144)+256*PEEK(145)+100)/256)
95 MEMLO=INT((BASTOP+2)/2)*2+4:HILEN=(192-MEMLO)*256
100 POKE 106,MEMLO:GRAPHICS 0:POKE 756,204
110 ? :? :? "Wichtig:Um mit 1200 Baud zu kopieren,"
115 ? "          muss zuerst CBAUD 1200/1"
120 ? "          eingeladen werden!"
130 FOR I=1 TO 51:READ A:ML$(I,I)=CHR$(A):NEXT I
140 DATA 104,173,1,211,9,2,141,1,211,104,104,170,104,104
150 DATA 157,66,3,104,157,69,3
160 DATA 104,157,68,3,104,157,73
170 DATA 3,104,157,72,3,32,86,228
180 DATA 132,212,169,0,133,213,173,1,211,41,253,141,1,211,96
190 ? :? :? :GOSUB 540
200 GRAPHICS 0:POKE 756,204
210 ? CHR$(125):? " Baercopy 1200/1 ":? "      (c) 1985 by"
215 ? "      Arndt Baer ":POKE 752,1:? HILEN;" Bytes free."
220 BLOCK=0
230 SETCOLOR 2,0,2:SETCOLOR 1,0,12
240 ? :? "EINLADEN mit ":GOSUB 470
250 POSITION 2,ZE1:? :? "< PLAY und RETURN. >"
260 CLOSE #1:OPEN #1,4,128,"C:"
270 ? :? "Programm laedt ":ZEI=PEEK(84)
280 A=USR(ADR(ML$),16,7,MEMLO*256,HILEN)
290 BLOCK=PEEK(856)+256*PEEK(857)
300 POSITION 20,2:? "Laenge: ";BLOCK;" Bytes"
310 IF A<128 THEN LET A=2
320 CLOSE #1:POSITION 2,ZE1:?
330 IF A<>136 THEN GOSUB 530: IF KEY<>27 THEN GOTO 200
340 SETCOLOR 2,13,4:SETCOLOR 1,0,14
350 ? :? "SPEICHERN mit ":GOSUB 470
360 POSITION 2,ZE1:? :? "< PLAY,RECORD und RETURN. >"
370 CLOSE #1:OPEN #1,8,128,"C:"
380 A=USR(ADR(ML$),16,11,MEMLO*256,BLOCK)
390 CLOSE #1:SOUND 0,0,0,0
400 IF A>127 THEN GOSUB 530:GOTO 420
410 ? :? "O.K.":?
420 SETCOLOR 2,12,2:SETCOLOR 1,0,12
430 ? "1...Noch eine Kopie ":? "2...Neues Original":? "3...Ende"
440 CLOSE #2:OPEN #2,4,0,"K":GET #2,KEY:WAHL=KEY-48
450 ON WAHL GOTO 340,210,460
460 END
470 ? :? "      1..1200 Baud oder":? "      2...600 Baud ? ":ZEI=PEEK(84)
480 CLOSE #2:OPEN #2,4,0,"K:"
490 GET #2,KEY:IF KEY=49 THEN POKE 54017,RAM:GOTO 520
500 IF KEY=50 THEN POKE 54017,RAM+1:GOTO 520
510 GOTO 490
520 POSITION 5,ZE1+KEY-51:? CHR$(27);CHR$(31)
525 RETURN
530 POKE 752,1:? CHR$(125);"I/O - Fehler Nr.":A:? :POKE 54017,RAM+1
540 ? " < beliebige Taste >"
550 CLOSE #2:OPEN # 2,4,0,"K:"
560 GET #2,KEY:RETURN

```

```

0010 ; *****
0020 ; *
0030 ; * CBAUD1200/1 V1.0 *
0040 ; * (c) 3/1985 by Arndt Baer *
0050 ; *
0060 ; *****
0070 *=$9800
0080 LADE
0090 PUTS=$E406 ;ROMROUTINEN:BILDSCHIRMAUSGABE
0100 GETK=$E424 ;DITO:KEYBOARDCODE
0110 ZER1=$CB
0120 ZER2=$CD
0130 BULL=$480 ;BUFFER FUER DEN ERSTEN BLOCK
0140 .BYTE 00,07 ;CASSETTEN-BOOT-PARAMETER
0150 .WORD LADE,ANF
0159 ;
0160 LDA #$3C
0170 STA $D302 ;MOTOR AUS
0180 CLC
0190 RTS
0199 ;
0200 ANF JSR ROMCO
0210 LDX #$00
0220 BEQ WE23
0230 NOCH LDX #$01
0240 WE23 LDA #$CC
0250 STA $2F4 ;ZEICHENSATZ PAGE 204/RAM EIN
0260 LOOP1 STX ZER1 ;MENUAUSGABE
0270 LDY MENU,X
0280 JSR PUSC
0290 LDX ZER1
0300 INX
0310 CPX #LENG
0320 BNE LOOP1
0330 LDA #$FF
0340 STA $350 ;CLOSE IOCB1
0350 LOOP2 JSR GEKE ;AUF TASTENDRUCK WARTEN
0360 CMP #$41 ; "A" GEDRUECKT ?
0370 BNE BOOT ;NEIN,DANN BOOTEN
0380 LDA $09
0390 AND #01 ;CASSETTENBOOT AUS
0400 STA $09 ;BOOTFLAG TILGEN
0410 BARU LDA SAV ;RAM/ROM-ZUSTAND
0420 STA $D301 ;UND
0430 LDA ZER2+1 ;INTERRUPTS
0440 STA $D20E ;WIEDERHERSTELLEN
0450 STA $10
0460 LDY #$7D
0470 JSR PUSC ;BILDSCHIRM LOESCHEN
0480 LDA #$E0 ;NORMALER ZEICHENSATZ
0490 STA $2F4
0500 CLC
0510 RTS ;ZURUECK ZUM BASIC/DOS
0520 BOOT LDX #$00
0530 STX FLAG
0540 LDX #$3C ;MOTOR AUS
0550 STX $D302
0560 CMP #$42 ;"B" GEDRUECKT ?
0570 BEQ BKEY ;JA,DANN NACH BKEY
0580 CMP #$43 ;"C" GEDRUECKT ?
0590 BNE LOOP2 ;NEIN,DANN NEUER TASTENDRUCK
0600 STA FLAG ;FLAG<>0 BEDEUTET OPTION "C"
0610 BKEY LDA SAV ;ROM+BASICROM
0620 ORA #$02
0630 STA $D301 ;ABSCHALTEN
0640 LDA #$00 ;SOUNDR ($41)=0

```



```

0650      STA $41          ;-> KEIN LADETON
0660      LDA #NAME/256    ;IOCB1 LADEN
0670      STA $355
0680      LDA #NAME&$FF    ;ZEIGER AUF FILENAME
0690      STA $354          ;"C:" SETZEN
0700      LDA #$03         ;-> OEFFNEN
0710      STA $352
0720      LDA #$04         ;-> LESEN
0730      STA $35A
0740      LDA #$80         ;-> KURZE PAUSEN (IRG CA. 0.25 SEC)
0750      STA $35B
0760      JSR CIO          ;CIO AUFRUFEN
0770      BCS NOCH         ;FEHLER:ZUM MENU
0780      LDX #$0B
0790 LOOP3  LDA CIDA,X     ;IOCB1 LADEN,
0800      STA $350,X       ;UM EINEN BLOCK
0810      DEX              ;ZU LESEN
0820      BNE LOOP3
0830      JSR CIO          ;CIO AUFRUFEN
0840      BCC EIG
0850      JMP NOCH         ;FEHLER:ZUM MENU
0860 EIG     CLC
0870      LDA BULL+2       ;DIE ERSTEN 6 BYTES
0880      ADC #$80         ;IN LADEADRESSE+128 UMWANDELN
0890      STA $354
0900      LDA BULL+3
0910      ADC #$00
0920      STA $355
0930      LDA #$00         ;UND BLOCKZAHL-1 IN
0940      STA $358         ;ICBLL/H (BUFFERLAENGE)
0950      DEC BULL+1       ;BRINGEN
0960      LDA BULL+1
0970      LSR A
0980      STA $359
0990      ROR $358
1000      JSR CIO          ;REST DES PROGRAMMS LADEN
1010      BCC WSC
1020      JMP NOCH         ;FEHLER:ZUM MENU
1030 WSC     LDA BULL+2     ;DEN ERSTEN BLOCK
1040      STA ZER1         ;VOM BUFFER "BULL"
1050      LDA BULL+3       ;AN DIE LADEADRESSE
1060      STA ZER1+1       ;VERSCHIEBEN
1070      LDY #$7F
1080 LOOP4  LDA BULL,Y     ;
1090      STA (ZER1),Y
1100      CLC
1110      DEY
1120      BPL LOOP4
1130 WEI     LDA BULL+4     ;INITADRESSE IN
1140      STA $02          ;SPEICHERZELLE $02 (CASINI LOW)
1150      LDA BULL+5       ;UND IN
1160      STA $03          ;$03 (CASINI HIGH)
1170      LDA #$E0         ;NORMALER ZEICHENSATZ
1180      STA $2F4
1190      JSR JUMP1        ;INIT=
1200      JSR JUMP2        ;SPRUENGE
1210      LDA FLAG         ;FLAG<>0,D.H. OPTION "C" GEWAHLT ?
1220      BNE BARI         ;DANN INS BASIC/DOS
1230      JMP ($0A)        ;SONST SPRUNG DURCH DOSVEC
1240 BARI     JMP BARU
1250 JUMP1   CLC           ;SPRUNG AN DAS SECHSTE
1260      LDA ZER1         ;UND SIEBTE BYTE DES
1270      ADC #$06         ;CASSETTENFILES
1280      STA ZER1
1290      LDA ZER1+1
1300      ADC #$00

```

```

1310          STA ZER1+1
1320          JMP (ZER1)
1330 JUMP2     JMP ($0002)          ;SPRUNG DURCH CASINI
1340 PUSC     LDA PUT5+1          ;AUSGABE DES
1350          PHA                  ;Y-REGISTERS ALS AT-ASCII-
1360          LDA PUT5            ;ZEICHEN AUF DEM
1370          PHA                  ;BILDSCHIRM
1380          TYA
1390          RTS
1400 GEKE     LDA GETK+1          ;HOLEN DES ASCII-
1410          PHA                  ;KEYBOARDCODES
1420          LDA GETK
1430          PHA
1440          RTS
1450 CIO      LDX #$10
1460          JSR $E456            ;SPRUNG NACH "CIOV"
1470          BMI ERROR           ;STATUS>127 = FEHLER
1480 NOERR     CLC                ;ZURUECK MIT GELOESCHTEM
1490          RTS                  ;CARRY
1500 ERROR    STY ZER1+1          ;FEHLERCODE SICHERN
1510          CPY #$88            ;END-OF-FILE-ERROR ?
1520          BEQ NOERR           ;DANN KEIN FEHLER
1530          LDA #$3C            ;MOTOR AUS
1540          STA $D302
1550          LDX #$00            ;FEHLERMELDUNG AUSGEBEN
1560 ELOOP    STX ZER1
1570          LDY ERRTE,X
1580          JSR PUSC
1590          LDX ZER1
1600          INX
1610          CPX #$12
1620          BNE ELOOP
1630          LDA #$00            ;FEHLERNUMMER IN
1640          STA $D5              ;FLOATINGPOINTREGISTER 0
1650          LDA ZER1+1
1660          STA $D4
1670          JSR $D9AA           ;INTEGER IN FP UMWANDELN
1680          JSR $D8E6           ;FP IN ASCII UMWANDELN
1690          LDX #$01            ;ASCII-FORM (AB 1408
1700 FLOOP    STX ZER1            ;ABGELEGT) AUF DEM
1710          LDY $580,X          ;BILDSCHIRM AUSGEBEN
1720          BMI ENDLO
1730          JSR PUSC
1740          LDX ZER1
1750          INX
1760          CLC
1770          BCC FLOOP
1780 ENDLO    TYA
1790          AND #$7F
1800          TAY
1810          JSR PUSC
1820          LDY #$9B            ;NEUE ZEILE
1830          JSR PUSC
1840          SEC                  ;ZURUECK MIT GESETZTEM CARRY
1850          RTS
1859 ;
1860          SAV=$76
1870 FLAG     .BYTE 00
1880 CIDA     .BYTE 255,1,7,1
1890          .WORD BULL
1900          .BYTE $B3,$FD,128,0,4,128
1910 NAME     .BYTE "C:"
1920 MENU     .BYTE 125,155,"|♥♥♥♥♥♥♥♥♥♥\","155,"|CBAUD 1200/1 |",155
1930          .BYTE "|(c) 1986 by\","155,"| Arndt Bar | ",155,"|",155
1940          .BYTE 155,"|♥\","155,"|H | ...BASIC/DOS ",155,"|"/"
1950          .BYTE 155,"|♥\","155,"|H | ...ML-PROGRAMM LADEN",155,"|"/"

```

```

1960 .BYTE 155,"I\","H | ... UND ",155,"J/",155,155
1970 LENG=*-MENU
1980 BAUD .WORD 698,692,686,678,670,752,741,732,725,717,711,702
1990 ERRTE .BYTE 253,155,"I/O - Fehler Nr. "
1999 ;
2000 ;ROM INS RAM KOPIEREN UND VERAENDERN
2001 ;
2010 ROMCO LDA $D301 ;RAM/ROM-ZUSTAND
2020 AND #$FE ;SICHERN
2030 STA SAV
2040 LDA $10 ;INTERRUPTS SICHERN
2050 STA ZER2+1
2060 LDA #$00 ;..UND AUSSCHALTEN
2070 STA $D40E
2080 STA $D20E
2090 STA $10
2100 STA ZER1 ;ROM $C000-$CFFF INS RAM KOPIEREN
2110 LDA #$C0
2120 STA ZER1+1
2130 LDA #$D0
2140 STA ZER2
2150 JSR UEBER
2160 LDA #$D6 ;DGL. MIT $D600-$FFFF
2170 STA ZER1+1
2180 LDA #$00
2190 STA ZER1
2200 STA ZER2
2210 JSR UEBER
2220 LDA #$FE
2230 STA $D301
2240 LDA #$E6 ;BAUDRATE:LOWBYTE..
2250 STA $EBA3
2260 STA $FD41
2270 LDA #$02 ;..UND HIBYTE INS ROM POKEN
2280 STA $EBA8 ;$2E6 FUER 1200 BAUD STATT
2290 STA $FD46 ;SONST $5CC FUER 600 BAUD
2300 LDA #$EA ;SONSTIGE ROMVERAENDERUNGEN
2310 STA $ED03
2320 STA $ED04
2330 STA $ED05
2340 LDA #$76
2350 STA $ED09
2360 LDY #$17
2370 LIOP LDA BAUD,Y
2380 STA $EDF9,Y
2390 DEY
2400 BPL LIOP
2410 LDA ZER2+1 ;INTERRUPTS WIEDER EINSCHALTEN
2420 STA $10
2430 STA $D20E
2440 LDA #$40
2450 STA $D40E
2460 LDY #$3F ;ZEICHENSATZ VERAENDERN
2470 ZSLO LDA ZEICH,Y
2480 STA $CE00,Y
2490 DEY
2500 BPL ZSLO
2510 LDA #$02 ;BILDSCHIRMFARBE AENDERN
2520 STA $02C6
2530 ORA #$0E
2540 STA $02C5
2550 RTS
2560 UEBER LDY #$00
2570 LIOP LDX #$FE ;UNTERPROGRAMM ZUM
2580 LDA (ZER1),Y ;KOPIEREN VOM ROM
2590 STX $D301 ;INS DARUNTERLIEGENDE

```

```

2600      STA (ZER1),Y      ;RAM
2610      LDX #$FF
2620      STX $D301
2630      INC ZER1
2640      LDA ZER1
2650      BNE LI0B
2660      CLC
2670      LDA ZER1+1
2680      ADC #$01
2690      STA ZER1+1
2700      CMP ZER2
2710      BNE LI0B
2720      RTS
2729 ;
2730 ;DATEN FUER NEUEN ZEICHENSATZ
2731 :
2740 ZEICH      .BYTE 0,0,0,0,255,255,255,0
2750      .BYTE 21,21,21,21,21,21,21,21
2760      .BYTE 112,112,112,112,112,112,112,112
2770      .BYTE 255,170,170,170,170,0,0,0
2780      .BYTE 0,0,0,0,15,23,21,20
2790      .BYTE 21,20,20,18,10,0,0,0
2800      .BYTE 112,176,144,160,160,0,0,0
2810      .BYTE 0,0,0,0,224,208,176,112

```