

PETER'S ASSEMBLERECKE

Turbo-Tape

In der Assemblerecke möchten wir Ihnen diesmal einen kleinen Knüller präsentieren, nämlich einen Speeder für das sonst so langsame Atari-Cassettenlaufwerk. Sicher waren einige Benutzer dieses Gerätes schon etwas neidisch auf die C-64-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 Cassetten-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 Lese-geschwindigkeit 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 (sicherheits-halber) erst einmal ab und starten es. Nach kurzer Wartezeit können Sie entscheiden, ob das "CBAUD"-Maschinenprogramm auf eine Cassette oder Diskette geschrieben werden soll. Um die erste Möglichkeit zu realisieren, legt man die Cassette ein, spult zurück und drückt dann PLAY und RECORD am Recorder sowie RETURN auf der Tastatur. Nachdem das Programm fertig auf Cassette vorliegt, läßt es sich in Zukunft ganz normal booten, d.h., die Cassette 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 Cassette oder Diskette entscheidet, erscheint nach Laden von "CBAUD" ein kleines Menü. Sie haben die Wahl, ob Basic aufgerufen werden soll oder ein Maschinenprogramm von Cassette zu laden ist. Mit dieser Möglichkeit können Sie Programme laden und starten, die in einem bootfähigen Cassettenformat aufgezeichnet sind. Voraussetzung ist natürlich, daß diese zuvor mit "CBAUD" gespeichert bzw. vom Convert-Programm, zu dem wir später kommen, auf das 1200-Baud-Format 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) aufzuru-

fen. 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 Cassettenoperationen 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. Wiederschaltung des Turbos 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's!

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.

Cassettenrecorder, 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, der 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 Tips 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 plazieren. Besonders bei 1200 Baud empfiehlt es sich, gute Cassetten 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 Assemblerlisting 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

Listing 1

```

10 REM *****
20 REM *          BASIC-LADER          *
30 REM *          CBAUD 1200/1  V1.0    *
40 REM * (C) 1986 BY ARNDT BAER      *
50 REM *****
60 ? CHR$(125):? :? :? :POKE 756,204:P
OKE 752,1
70 ? "          BASIC-LADER"
80 ? "          CBAUD 1200/1  V1.0"
90 ? "          (C) 1987 BY ARNDT BAER"
100 ? :? :? :? "      Bitte einen Moment
Geduld"
110 GOSUB 210:GOSUB 500
120 POSITION 2,10:? " 1-Diskettenfile
'D:CBAUD.COM' oder"
130 ? " 2-Cassettenfile  schreiben ?"
135 CLOSE #1:OPEN #1,4,0,"K:":GET #1,K
EY:CLOSE #1
140 IF KEY<>49 AND KEY<>50 THEN GOTO 1
35
145 TRAP 190:IF KEY=49 THEN CLOSE #1:O
PEN #1,8,0,"D:CBAUD.COM":A=USR(ADR(ML$
),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 - FEH
LER NR.":PEEK(195):GOTO 120
210 DIM ML$(40)
220 RESTORE 230:FOR I=1 TO 35:READ A:M
L$(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 240,2,162,1,169,204,141,244,1
163

```

```

304 DATA 2,134,203,188,143,153,32,26,8
81
305 DATA 153,166,203,232,224,161,208,2
41,1588
306 DATA 169,255,141,80,3,32,36,153,86
9
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 162,60,142,2,211,201,66,240,1
084
313 DATA 7,201,67,208,200,141,128,153,
1105
314 DATA 173,118,0,9,2,141,1,211,655
315 DATA 169,0,133,65,169,153,141,85,9
15
316 DATA 3,169,141,141,84,3,169,3,713
317 DATA 141,82,3,169,4,141,90,3,633
318 DATA 169,128,141,91,3,32,45,153,76
2
319 DATA 176,128,162,11,189,129,153,15
7,1105
320 DATA 80,3,202,208,247,32,45,153,97
0
321 DATA 144,3,76,20,152,24,173,130,72
2
322 DATA 4,105,128,141,84,3,173,131,76
9
323 DATA 4,105,0,141,85,3,169,0,507
324 DATA 141,88,3,206,129,4,173,129,87
3
325 DATA 4,74,141,89,3,110,88,3,512
326 DATA 32,45,153,144,3,76,20,152,625
327 DATA 173,130,4,133,203,173,131,4,9
51
328 DATA 133,204,160,127,185,128,4,145
,1086
329 DATA 203,24,136,16,247,173,132,4,9
35
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 105,6,133,203,165,204,105,0,9
21
335 DATA 133,204,108,203,0,108,2,0,758
336 DATA 173,7,228,72,173,6,228,72,959
337 DATA 152,96,173,37,228,72,173,36,9
67
338 DATA 228,72,96,162,16,32,86,228,92
0
339 DATA 48,2,24,96,132,204,192,136,83
4

```

```

340 DATA 240,248,169,60,141,2,211,162,
1233
341 DATA 0,134,203,188,72,154,32,26,80
9
342 DATA 153,166,203,232,224,18,208,24
1,1445
343 DATA 169,0,133,213,165,204,133,212
,1229
344 DATA 32,170,217,32,230,216,162,1,1
060
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 205,2,199,2,190,2,253,155,100
8
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 16,133,206,169,0,141,14,212,8
91
378 DATA 141,14,210,133,16,133,203,169
,1019
379 DATA 192,133,204,169,208,133,205,3
2,1276
380 DATA 219,154,169,214,133,204,169,0
,1262
381 DATA 133,203,133,205,32,219,154,16
9,1248
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 237,141,4,237,141,5,237,169,1
171
386 DATA 118,141,9,237,160,23,185,48,9
21
387 DATA 154,153,249,237,136,16,247,16
5,1357
388 DATA 206,133,16,141,14,210,169,64,
953
389 DATA 141,14,212,160,63,185,253,154
,1182
390 DATA 153,0,206,136,16,247,169,2,92
9
391 DATA 141,198,2,9,14,141,197,2,704
392 DATA 96,160,0,162,254,177,203,142,
1194
393 DATA 1,211,145,203,162,255,142,1,1
120
394 DATA 211,230,203,165,203,208,236,2
4,1480
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 112,112,112,255,170,170,170,1
70,1271
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:IF A<>SUM THEN ? :? CHR$(12
5);"      Datenfehler in Zeile ";300+I:L
IST 300+I:END
530 NEXT I
540 RETURN

```

Listing 2

```

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$(51):BASTOP=INT((PEEK(144)+2
56*PEEK(145)+100)/256):MEMLO=INT((BAST
OP+2)/2)*2+4:HILEN=(192-MEMLO)*256
100 POKE 106,MEMLO:GRAPHICS 0:POKE 756
,204
110 ? :? :? "Wichtig:Um mit 1200 Baud
zu kopieren,":? "           muss zuers
t CBAUD 1200/1"
120 ? "           eingeladen werden
!"
130 FOR I=1 TO 51:READ A:ML$(I,I)=CHR$(
A):NEXT I
140 DATA 104,173,1,211,9,2,141,1,211,1
04,104,170,104,104
150 DATA 157,66,3,104,157,69,3
160 DATA 104,157,68,3,104,157,73
170 DATA 3,104,157,72,3,32,86,228
180 DATA 132,212,169,0,133,213,173,1,2
11,41,253,141,1,211,96
190 ? :? :? :GOSUB 540
200 GRAPHICS 0:POKE 756,204
210 ? CHR$(125):? " Baercopy 1200/1 ";
? " (c) 1985 by":? " Arndt Baer ":P
OKE 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 RE
TURN. >"
260 CLOSE #1:OPEN #1,4,128,"C:"
270 ? :? "Programm laedt ":ZE1=PEEK(84
)
280 A=USR(ADR(ML$),16,7,MEMLO*256,HILE
N)
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<>2
7 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$),16,11,MEMLO*256,BLO
CK)
390 CLOSE #1:SOUND 0,0,0,0
400 IF A>127 THEN GOSUB 530:GOTO 420
410 ? :? "O.K.":?
420 SETCOLOR 2,12,2:SETCOLOR 1,0,12
430 ? "1...Noch eine Kopie ":? "2...Ne
ues Original":? "3...Ende"
440 CLOSE #2:OPEN #2,4,0,"K:":GET #2,K
EY:WAHL=KEY-48
450 ON WAHL GOTO 340,210,460
460 END
470 ? :? " 1..1200 Baud oder":? "
2...600 Baud ?":ZE1=PEEK(84)
480 CLOSE #2:OPEN #2,4,0,"K:"
490 GET #2,KEY:IF KEY=49 THEN POKE 540
17, RAM:GOTO 520
500 IF KEY=50 THEN POKE 54017, RAM+1:GO
TO 520
510 GOTO 490
520 POSITION 5,ZE1+KEY-51:? CHR$(27);C
HR$(31)
525 RETURN
530 POKE 752,1:? CHR$(125);"I/O - Fehl
er Nr.":A:? :POKE 54017, RAM+1
540 ? " < beliebige Taste >"
550 CLOSE #2:OPEN #2,4,0,"K:"
560 GET #2,KEY:RETURN

```