

# PETER'S ASSEMBLERECKE

für ATARI -Computer

Atari-Freunde aufgepaßt: Ab jetzt gibt es hier die Assemblerecke, in der Sie jeden Monat Tips und Tricks für angehende und fortgeschrittenere Assemblerprogrammierer lesen können. Gerade der Atari-Computer mit seiner Vielzahl von Grafik- und Soundregistern bietet dem Assemblerprogrammierer ein unendliches Betätigungsfeld; die Assemblerecke wird dabei mit Tips und Anregungen zur Seite stehen. Aber neben den hervorragenden, für meine Begriffe im Homecomputerbereich noch unübertroffenen grafischen Fähigkeiten, soll auch das komfortable Betriebssystem nicht vergessen werden.

Gleich das heutige Thema befaßt sich mit einem zentralen Punkt des Operating Systems.

Ich bin schon von vielen Leuten gefragt worden, wie man denn in Assembler "etwas auf den Bildschirm" bekommt – bitte schön – Sie können's im Anschluß lesen.

In Zukunft soll der Schwerpunkt nicht nur auf reinen Assemblerprogrammen liegen, sondern auch maschinennahe Sprachen wie "ACTION!" oder "C" werden betrachtet. Natürlich bringen wir auch nützliche Maschinenunterprogramme für BASIC-Programme. Wer Anregungen hat oder ein bestimmtes Thema in der Assemblerecke besprochen haben möchte, der kann sich mit mir in Verbindung setzen. Das wär's für diesmal, jetzt geht's ans Gerät

Peter Finzel

## Textausgabe in Assembler

Leider ist die Ausgabe von Meldungen und Texten in Assembler nicht ganz so einfach wie in BASIC, welches über einen komfortablen "PRINT"-Befehl verfügt. Kein Grund zur Verzweiflung, denn das Betriebssystem Ihres ATARI-Computers läßt Sie nicht im Stich.

Das Interface zum Betriebssystem wird durch das Unterprogramm TXTAUS hergestellt. Alles was Sie diesem Unterprogramm mitteilen müssen ist die Anfangsadresse des ausdruckenden Textes, wobei der niederwertige Teil der Adresse (LSB) im Akku, der höherwertige Teil (MSB) im Y-Register übergeben wird. Sehen Sie sich dazu das Demo-Programm (Zeilen 300-360) an, der Text selbst steht hinter einem .BYTE-Befehl in Anführungszeichen. Bei Eingabe von eigenen Texten sollten Sie niemals das EOL (RETURN)-

Zeichen \$9B am Ende des Textes vergessen.

Tippen Sie das Listing mit Hilfe ihres Assemblers ein, bei MAC/65 und der Editor/Assembler Cartridge das .OPT OBJ in Zeile 90 beachten. Jetzt assemblieren Sie den Quelltext und starten das Maschinenprogramm mit dem Debugger an der Adresse 680 (Hexadezimal). Wenn Sie alles richtig gemacht haben, müßte die Meldung dann am Bildschirm stehen. Falls Ihr Assembler keinen eingebauten Debugger hat, können Sie das Programm auch im DOS starten, z.B. mit DOS II Option M (Run at Address).

Noch ein paar Worte zur Funktion des TXTAUS Unterprogrammes. In den Zeilen 500 und 510 wird die übergebene Adresse in den IOCB Nummer 0 eingetragen. Sie wissen ja sicher, daß der IOCB Nr. 0 immer auf den Screen-Editor geöffnet ist. Die Zeilen 520 und

530 schreiben den Befehl "Put Text Record" in den IOCB, Zeile 540 bis 570 begrenzen die maximale Länge des Ausgabes-  
textes auf 100 Zeichen. Die hier angegebene Länge sollte immer größer als die tatsächliche Länge des Textes sein, sonst wird er nicht vollständig ausgedruckt. Anschließend wird im X-Register eine Null hinterlegt, die dem nachfolgenden CIO-

Aufruf mitteilt, daß der IOCB Nummer 0 zu bearbeiten ist. Sie bekommen von CIO im Y-Register eine Statusmeldung zurück, haben aber im vorliegenden Fall kaum Chancen, einen Fehler zu begehen.

Bleibt noch anzumerken, daß diese Methode der Textausgabe die universellste ist, da sie auf allen alten und neuen ATARIs funktioniert.

```

0000          90      .OPT OBJ
0100          ;*****
0110          ;Textausgabe in Assembler fuer alle ATARI-Computer
0120          ;
0130          ;Peter Finzel 1984
0140          ;*****
0150          ;
0160          ;Einige Adressen des Betriebssystems:
0170          ;
0180          CPTXTR = $09      ; CIO-Befehlscode fuer Put Text Record
0190          ICCDM = $0342    ; IOCB Nr.0 Kommandoregister
0200          ICBADR = $0344    ; Bufferadresse IOCB#0
0210          ICBLN = $0348    ; Laenge dieses Buffers
0220          CIOV = $E456     ; CIO Einsprungsadresse
0230          ;
0240          ;*****
0250          ;Zuerst ein Demo-Programm das Ihnen zeigt,
0260          ;wie TXTAUS benutzt wird:
0270          ;*****
0280          ;
0290          ;
0300          ;
0310          DEMO LDA #TEXT1&255 ; LSB des Zeigers in Akku
0320          LDY #TEXT1/256 ; MSB in Y-Register
0330          JSR TXTAUS ; und ausgeben.
0340          RTS ; das war's!
0350          ;
0360          TEXT1 .BYTE "Diese Zeile wird ausgegeben",$9B

0380          ;*****
0390          ;Das Textausgabe Unterprogramm:
0400          ;
0410          ;So benutzen Sie TXTAUS:
0420          ;-Akku/Y-Register mit der Adresse des Textes laden
0430          ; (Akku mit LSB, Y mit MSB)
0440          ;-JSR TXTAUS ausfuehren
0450          ;*****
0460          ;
0470          ;
0480          ;
0490          ;
0500          TXTAUS STA ICBADR ; Adresse LSB in IOCB#0 eintragen
0510          STY ICBADR+1 ; jetzt ist MSB dran
0520          LDA #CPTXTR ; Befehl: Textausgabe
0530          STA ICCDM ; ebenfalls in IOCB
0540          LDA #100 ; max. Laenge des Textes vorgeben
0550          STA ICBLN ; hier = 100 (willkuerlich gewaehlt)
0560          LDA #0 ; MSB Laenge ist Null
0570          STA ICBLN+1
0580          LDX #0 ; IOCB Nr.0 ist gemeint
0590          JSR CIOV ; Zentrale I/O Routine aufrufen
0600          ; Eventl. Fehlercode ist im Y-Register
0610          RTS
0620          ;
0630          ;
0640          ;
0650          ;
0660          ;
0670          ;
0680          ;
0690          ;
0700          ;
0710          ;
0720          ;
0730          ;
0740          ;
0750          ;
0760          ;
0770          ;
0780          ;
0790          ;
0800          ;
0810          ;
0820          ;
0830          ;
0840          ;
0850          ;
0860          ;
0870          ;
0880          ;
0890          ;
0900          ;
0910          ;
0920          ;
0930          ;
0940          ;
0950          ;
0960          ;
0970          ;
0980          ;
0990          ;
1000         
```

ASSEMBLY ERRORS: 0 31626 BYTES FREE