

Schnelle Linien

Jedes mal, wenn es interessant werde, so beschwerte sich W. Arnold in einem Leserbrief, komme anstelle einer eigenen Lösung eine Routine aus dem Betriebssystem zur Anwendung. Die Assemblerecke solle schließlich darlegen, so schrieb er weiter, wie man ein bestimmtes Problem selbst lösen kann, ohne gleich die Hilfe des Betriebssystems zu benötigen.

Recht hat er! Der Wunsch unserer Leser ist uns selbstverständlich Befehl. Sollten Sie Anregungen für künftige Themen haben, die Sie gern in der Assemblerecke behandelt sehen würden, so schreiben Sie uns. W. Arnold teilte uns noch mit, er sei besonders an einer möglichst schnellen DRAWTO-Routine interessiert. In Ausgabe 1/1986 war ein ähnliches Thema zu finden; allerdings wurden dort die missliebigen ROM-Routinen verwendet. Nehmen wir nun einmal die Sache selbst in die Hand. Was dabei herauskommt, zeigt Listing 1.

Lassen Sie sich durch seine Länge nicht erschrecken. Wie bei fast allen Assembler-Programmen wird der wesentliche Teil (die Linien- Routine) durch viel Beiwerk wie Variablen Definitionen, Demo und Utility-Unterprogramme ein wenig in den Hintergrund gedrängt.

Den harten Kern bildet die Unteroutine DRAWTO. Sie übernimmt die Berechnung der Verbindungspunkte zwischen einem gegebenen Anfangs- und Endpunkt. Hierzu wird ein sehr schneller Algorithmus verwendet, den Paul Chabot in der amerikanischen Zeitschrift ANTIC (Ausgabe 6/85) veröffentlichte. Dort war das Programm zwar in Action! codiert, aber die Umsetzung in Assembler macht die Routine noch schneller.

Die DRAWTO-Routine erledigt ihre Aufgabe durch eine Fallunterscheidung in vier Quadranten. Sie errechnet den Abstand von Anfangs- zu Endpunkt sowohl für die X- als auch die Y-Koordinate. Der längere Weg dient als Zähler für die Menge der zu bestimmenden Punkte. Während der längere Abstand dann einfach linear durchgezählt wird, nimmt eine geschickte Näherung die Korrektur des kürzeren vor.

Auf diese Art erspart man sich die umständlichen und zeitraubenden Multiplikationen bzw. Divisionen, die bei einer direkten Anwendung der Geradengleichung $Y=mx+t$ zwangsläufig durchzuführen wären. Aus diesem Grunde arbeitet diese DRAWTO-Routine auch verhältnismäßig schnell. Wenn Sie Spaß daran haben, können Sie das Demo des Assembler-Programms auch versuchsweise einmal in Basic codieren, was keine großen Schwierigkeiten bereitet. Hier sind deutlich weniger als 10 Zeilen erforderlich, aber der Unterschied in der Ablaufgeschwindigkeit ist einfach enorm.

Gerechterweise muss man aber sagen, dass dies nicht an einer schlechten Programmierung des Betriebssystems liegt, sondern daran, dass die Routinen im OS sehr allgemein gehalten sind. Während die in Listing 1 vorgestellten Routinen nur mit bestimmten Grafikauflösungen funktionieren, ist die im OS für alle Grafikmodi ausgelegt. Das kostet natürlich Rechenzeit.

Hinzu kommt, dass auch der beste Algorithmus zur Berechnung der Linie nichts nützt, wenn die Ausgabe von einzelnen Grafikpunkten nicht ebenfalls optimiert ist. Deshalb wird im Programmbeispiel eine tabellenorientierte PLOT-Funktion verwendet, ebenfalls eine Methode, die sich aufgrund des Speicherplatzbedarfs für ein Betriebssystem in einem 16K-ROM von vornherein verbietet.

Sie sehen jetzt schon, wo der Trennungsstrich zu ziehen ist. Die Benutzung von OS-Routinen ist überall dort zu empfehlen, wo es nicht auf Geschwindigkeit, sondern auf Vielseitigkeit ankommt. Eigene Routinen sind dagegen für spezielle Anwendungen (wie z.B. Grafikprogramme, Spiele) unentbehrlich.

Noch ein paar Anmerkungen zum Listing: Sie können es mit ATMAS II assemblieren und im Monitor mit G A800 starten. Diesmal wurden (zur Verringerung der Schreibarbeit) zwei Makros, ADD und SUB, eingeführt. Sie erleichtern die Rechenoperationen mit 16-BitZahlen. Zur Anwendung kommt die zweifarbige Grafikstufe 6, die durch einen CIO Aufruf eingeschaltet wird (Unterprogramm GRAPHICS).

Wollen Sie andere Grafikstufen benutzen, so müssen die Routinen PLOT und PLOT AB entsprechend modifiziert werden (Bytes pro Zeile usw.). Das PLOT-Unterprogramm arbeitet übrigens im EXOR-Modus; daher ist es möglich, eine Linie einfach durch nochmaliges Zeichnen zu entfernen. Wenn Sie den normalen Zeichenmodus verwenden wollen, muss der EOR-Befehl der PLOT-Routine in eine ORA-Anweisung abgewandelt werden.

P. Finzel

```

*****
*      SCHNELLE LINIENROUTINE
*
*  ASSEMBLER:  ATMA5-II
*
*  PETER FINZEL                      1987
*****
*
*  IOCB-STRUKTUR, CIO-BEFEHLE...
*

ICCOM      EQU  $342
ICBAL      EQU  $344
ICBAH      EQU  $345
ICAX1      EQU  $34A
ICAX2      EQU  $34B
CIOV       EQU  $E456          CIO-VEKTOR
COPEN      EQU  3
CCLSE      EQU  12

*
*  BETRIEBSSYSTEM-VARIABLE
*

SAUMSC     EQU  $58    BILDSCHIRM-ADRESSE

*
*  KONSTANTE FUER GRAPHICS 6
*

YMAX       EQU  96    AUFLOESUNG VERT.
XMAX       EQU  160   AUFLOESUNG HOR.
ZLAENGE    EQU  20    BYTES PRO ZEILE

*
*  ZEROPAGE-VARIABLE
*

XNEU       EQU  $E0    (BYTE) ENDPUNKT X
YNEU       EQU  $E1    (BYTE) ENDPUNKT Y
XALT       EQU  $E2    (BYTE) STARTPUNKT X
YALT       EQU  $E3    (BYTE) STARTPUNKT Y
ZAEHLER    EQU  $E4    (BYTE) PIXELZAEHLER
DELTAX     EQU  $E5    (BYTE) ABSTAND X
DELTAY     EQU  $E6    (BYTE) ABSTAND Y
XFLAG      EQU  $E7    (BYTE) MERKER LINKS/RECHTS
YFLAG      EQU  $E8    (BYTE) MERKER OBEN/UNTEN
HILFA      EQU  $E9    (WORD) HILFSREGISTER
HILFB      EQU  $EB    (WORD) FUER NAEHERUNG
HILFT      EQU  $ED    (WORD)
ZEIGER     EQU  $EF    (WORD) VEKTOR F. PLOT

*
*  ZWEI MAKROS FUER 16-BIT ZAHLEN...
*

ADD        MACRO P1,P2
            CLC
            LDA P2
            ADC P1
            STA P2
            LDA P2+1
            ADC P1+1
            STA P2+1
            MEND

```

```

SUB      MACRO P1,P2,P3
          SEC
          LDA P1
          SBC P2
          STA P3
          LDA P1+1
          SBC #0
          STA P3+1
          MEND

*
          ORG $A800          IM RES. BEREICH

*****
* DEMO-PROGRAMM 'HIGHLIGHTS'
*****
*
* GRAPHICS 6+16
*
          LDA #6+16          GRAPHICS 7
          JSR GRAPHICS       GANZEN SCREEN
          JSR PLOTAB         TABELLEN...
DEMO     LDA #0              LINIENSTART
          STA XSTART
DEMO2    LDA #160            LINIENENDE
          STA XENDE

DEMO1    LDA XSTART          ANFANGSPUNKT
          STA XALT           SETZEN (X & Y)
          LDA #0
          STA YALT
          LDX XENDE          ENDPUNKT SETZEN
          LDY #95

          JSR DRAWTO         LINIE ZEICHNEN

          SEC               NEUE KOORDINATEN
          LDA XENDE         BERECHNEN
          SBC #20
          STA XENDE
          BNE DEMO1
          CLC
          LDA XSTART        STARTPUNKT
          ADC #39
          STA XSTART
          CMP #160
          BCC DEMO2
          JMP DEMO          ENDLOSES DEMO

*
* VARIABLEN DEMOPROGRAMM
*
XSTART   DFB 0
XENDE    DFB 0

```

```

*
*****
* SCHNELLE BERECHNUNG VON LINIEN
*
* PARAMETER: XALT,YALT: STARTPUNKT
*             <X>,<Y>  : ENDPUNKT
*****
*

DRAWTO    STX XNEU          ENDPUNKT MERKEN
          STY YNEU
          LDX XALT          ANFANGSPUNKT
          LDY YALT          ZEICHNEN
          JSR PLOT
          LDA #0            FLAGS RUECKSETZEN
          STA XFLAG
          STA YFLAG
          LDX XNEU          ENDPUNKT IN REGISTER
          LDY YNEU          (EINFACHER)

          CPX XALT          ANFANG = ENDE?
          BNE DR1
          CPY YALT
          BNE DR1
          RTS              JA FERTIG!==>

DR1        CPX XALT          NEUER PUNKT IST
          BCC DRX           LINKS VON ALTEN P.->
          INC XFLAG         RECHTS!
          TXA               DELTA AUSRECHNEN
          SBC XALT
          JMP DR2

DRX        DEC XFLAG        LINKS!
          SEC
          LDA XALT          DELTA HER.
          SBC XNEU

DR2        STA DELTAX        ABSTAND MERKEN
          CPY YALT          NEUER PUNKT IST
          BCC DRY           OBERHALB ALTEN P.
          INC YFLAG         UNTERHALB
          TYA
          SBC YALT          DELTA BERECHNEN
          JMP DR3

DRY        DEC YFLAG        OBERHALB
          SEC
          LDA YALT          ABSTAND (DELTA)
          SBC YNEU

DR3        STA DELTAY        UND MERKEN
          LDA XALT          NEUER P.= ALTER P.
          STA XNEU
          LDA YALT
          STA YNEU
          LDA DELTAX        WELCHES DELTA IST
          CMP DELTAY        GROESSER?
          BCC DRYSTEP       DY IST GROESSER->

```

```

LDA DELTAY          DX IST GROESSER!
ASL
STA HILFA           NAEHERUNG FUER
LDA #0              SCHRITTWEITE
ROL
STA HILFA+1

SUB HILFA,DELTAX,HILFT
SUB HILFT,DELTAX,HILFB
LDA DELTAX          ZAEHLER FUER
STA ZAEHLER         PIXELS EINRICHTEN

DXSCHL  CLC
LDA XNEU            X WEITERZAEHLEN
ADC XFLAG
STA XNEU
LDA HILFT+1         SCHRITT NACH Y
BPL DRX5            ERFODERLICH? JA->
ADD HILFA,HILFT
JMP DRX4            KEIN SCHRITT

DRX5     CLC
ADD HILFB,HILFT
CLC
LDA YNEU            SCHRITT NACH Y
ADC YFLAG           AUSFUEHREN
STA YNEU

DRX4     LDX XNEU     PIXEL PLOTTEN
LDY YNEU
JSR PLOT
DEC ZAEHLER         GANZES DELTAX
BNE DXSCHL          ABGEFAHREN? NEIN->
JMP DREND           FERTIG !==>

DRYSTEP  LDA DELTAX   DELTA Y WAR GROESSER
ASL
STA HILFA           NAEHERUNG FUER
LDA #0              SCHRITT IN X-RICHT.
ROL
STA HILFA+1

SUB HILFA,DELTAY,HILFT
SUB HILFT,DELTAY,HILFB

LDA DELTAY          ZAEHLER FUER ABSTAND
STA ZAEHLER         EINRICHTEN

DYSCHL   CLC
LDA YNEU            Y WEITERZAEHLEN
ADC YFLAG
STA YNEU
LDA HILFT+1         SCHRITT NACH X
BPL DRY5            NOETIG? JA ->
ADD HILFA,HILFT
JMP DRY4            KEINE X-KORREKTUR

DRY5     ADD HILFB,HILFT

```

```

        CLC
        LDA XNEU          X-KORREKTUR AUS-
        ADC XFLAG          FUEHREN
        STA XNEU
DRY4    LDX XNEU          PIXEL AUSGEBEN
        LDY YNEU
        JSR PLOT
        DEC ZAEHLER        ALLE PUNKTE?
        BNE DYSCHL        NEIN -->

DREND   LDX XNEU          ENDE DER LINIE
        LDY YNEU          KANN ANFANG EINER
        STX XALT          NEUEN SEIN.
        STY YALT
        RTS              FERTIG !

*****
* GRAPHICS-UNTERPROGRAMM
*
* AUFRUF: JSR GRAPHICS
* PARAMETER:
* <A> 0 BIS 15 (XL/XE)
*      0 BIS 11 (400/800)
*****

GRAPHICS PHA              GRAPHIK-STUFE MERKEN
        LDX #$60          IOCB NR. 6
        LDA #CCL5E        SCREEN-IOCB ZUERST
        STA ICCOM,X        SCHLIESSEN
        JSR CIOV
        PLA              GRAPHIK-STUFE
        STA ICAX2,X        ZURUECKHOLEN
        AND #$F0          UND PASSENDE
        EOR #$10          BIT-KOMBINATION
        ORA #$0C          FUER HNDLER
        STA ICAX1,X        HERSTELLEN
        LDA #COPEN        JETZT DEN BEFEHL
        STA ICCOM,X        ZUM OEFFEN DES SCREENS
        LDA #SDEVICE       ZEIGER AUF DEVICE-
        STA ICBAL,X        BEZEICHNUNG
        LDA #SDEVICE/256
        STA ICBAN,X
        JSR CIOV
        RTS

SDEVICE ASC "S:" DISPLAY-HANDLER

*****
* HI-SPEED PLOT FUER EINFARB-MODI
* AUFRUF: JSR PLOT
*
* PARAMETER:
* <X>,<Y> JE NACH GRAPHIKSTUFE
*      X,Y WERDEN ZERSTOERT!
*****

PLOT    CPY #YMAX          GRENZEN
        BCS PLOTEND        PRUEFEN
        CPX #XMAX
        BCS PLOTEND
        LDA ADRLO,Y        BILDSCHIRM-
        STA ZEIGER          ADRESSE
        LDA ADRHI,Y        IN ZEROPAGE

```

```

        STA ZEIGER+1
        TXA
        LSR                ;GETEILT
        LSR                ;DURCH 8
        LSR
        TAY                ;INDEX F. X-POS
        TXA                ;X-POSITION
        AND #7
        TAX
        LDA PIXTAB,X        WELCHES PIXEL
        EOR (ZEIGER),Y      UND PIXEL MANIPULIEREN
        STA (ZEIGER),Y      ZURUECK IN GRAPHIK
PLOTEND RTS

```

```

*****
* ERZEUGT ADRESSTABELLEN FUER PLOT
*
* MUSS VOR DER ERSTEN VERWENDUNG VON
* PLOT UND NACH DEM GRAPHICS-BEFEHL
* STEHEN!
*****

```

```

PLOTAB  LDA SAUMSC          ANFANGSADRESSE
        STA ZEIGER          DES VIDEO-RAM5
        LDA SAUMSC+1
        STA ZEIGER+1
        LDY #0              INDEX AUF 0
NXTADR  LDA ZEIGER          ADRESSTABELLEN
        STA ADRLO,Y         AUFBAUEN
        LDA ZEIGER+1        MSB-TABELLE
        STA ADRHI,Y
        CLC
        LDA ZEIGER          ADRESSE DES
        ADC #ZLAENGE        NAECHSTEN ZEILEN
        STA ZEIGER          ANFANGS BERECHNEN
        LDA ZEIGER+1
        ADC #0
        STA ZEIGER+1
        INY
        CPY #YMAX           SCHON F. ALLE ZEILEN?
        BNE NXTADR          NEIN - >
        RTS

```

```

*
* AB HIER STEHEN DIE TABELLEN
*

```

```

PIXTAB  DFB 128,64,32,16,8,4,2,1
ADRLO   ORG *+YMAX          PLATZ FUER
ADRHI   ORG *+YMAX          TABELLEN

```