

Schildkröte à

Im Buch „Mikrocomputer selbstgebaut...“ ist die Beschreibung des Z-80-Grundprogramms etwas zu kurz gekommen, obwohl es viele Befehle enthält, die sich in eigenen Programmen in der „Zeichensprache“ bestens verwenden lassen.

An dieser Stelle folgt nun eine ausführliche Darstellung der Befehle, nach der viele von Ihnen schon am Telefon gefragt haben. Jeder Befehl wird mit einem kleinen Beispiel erläutert. Die Beispiele sind so abgefaßt, daß sie sich mit dem Menü „ÄNDERN“ ab Adresse 8800 eingeben und ausprobieren lassen. Sie werden staunen, was sich mit der Zeichensprache im Grundprogramm alles anfangen läßt. Doch genug der Vorreden, fangen wir gleich an mit:

SCHREITE

(Schildkrötensprache)
Der Befehl SCHREITE läßt die Schildkröte um die im HL-Registerpaar angegebene Anzahl weiterlaufen. Dabei schreitet sie in der aktuellen Blickrichtung weiter. Die Zahl kann auch negativ sein, dann erfolgt die Angabe im Zweierkomplement. Das Grundprogramm kann aber die Umrechnung direkt vornehmen, wenn man z. B. -#100 eingibt.
Beispiel:

```
21#100.W ;dezimale Zahl
           ;Hundert
CD SCHREITE ;Befehl Schreite
           ;ausführen
C9 ;Programmende
```

DREHE

(Schildkrötensprache)
Die Schildkröte wird um

einen im HL-Registerpaar angegebenen Winkel gedreht. Der Winkel ist dabei in Grad angegeben. Eine positive Zahl dreht die Schildkröte entgegen dem Uhrzeigersinn, also mathematisch positiv.
Beispiel:

```
21-#90.W ;um 90 Grad im
           ;Uhrzeigersinn
CD DREHE ;drehen
C9
```

HEBE

(Schildkrötensprache)
Nach Aufruf des Befehls hinterläßt die Schildkröte keine Schreibspur mehr.
Beispiel:

```
21#10.W ;10 Schritte
CD SCHREITE ;mit Schreibspur
CD HEBE ;dann ohne
21#10.W ;Schreibspur
CD SCHREITE ;neue Position
           ;anfahen
C9
```

SENKE

(Schildkrötensprache)
Wenn man die Schreibspur mit Hebe aufgehoben hat, wird sie durch diesen Befehl wieder eingeschaltet.
Beispiel:

```
21#10.W ;erst mal schreiten
CD SCHREITE
CD HEBE ;dann Stift hoch
21#10.W ;ohne Schreibspur
CD SCHREITE ;schreiten
CD SENKE ;dann wieder
           ;einschalten
21#10.W ;und wieder Linie
           ;zeichnen
CD SCHREITE
C9 ;Programmende
```

SCHLEIFE

ENDSCHLEIFE

Damit kann man Schleifenkonstruktionen realisieren. Dazu wird die Befehlssequenz, die man mehrfach ausgeführt haben möchte, zwischen SCHLEIFE und END-

SCHLEIFE genommen. In das Registerpaar HL läßt man zuvor die Anzahl der Schleifendurchläufe. Man kann mit diesen beiden Befehlen auch Schleifen mehrfach ineinander schachteln.
Beispiel:

```
21#72.W ;äußere Schleife
CD SCHLEIFE ;72mal durchlaufen
21#36.W ;innere Schleife
CD SCHLEIFE ;36mal durchlaufen
21#10.W ;Kreis annähern
CD SCHREITE ;
21#10.W ;
CD DREHE ;
CD END- ;Ende der inneren
SCHLEIFE ;Schleife
21#5.W ;wird 72mal
           ;ausgeführt
CD DREHE ;
CD END- ;Ende der äußeren
SCHLEIFE ;Schleife
C9 ;Programmende
```

SET (Schildkrötensprache)

Damit kann man die Schildkröte auf eine absolute Position setzen. Im Registerpaar HL wird dazu die X-Koordinate übergeben (Bereich 0 bis 511) und im Registerpaar DE die Y-Koordinate (auch 0 bis 511, jedoch ist nur jeder zweite Punkt auf dem Bildschirm unterscheidbar) und im Registerpaar BC steht der Winkel, in den die Schildkröte blickt. Dabei zeigt die Schildkröte bei einem Winkel von 90 Grad nach oben.
Beispiel:

```
21#256.W ;HL:=256,
           ;X-Koordinate
11#300.W ;DE:=300,
           ;Y-Koordinate
01#180.W ;BC:=180, Winkel
CD SET
C9
```

Die Schildkröte zeigt dann nach links.

la I/O-Karte

MOVETO

Damit kann man die Koordinatenregister des Graphikprozessors setzen. HL ist dabei die X-Koordinate (0 bis 511 sichtbar) und DE die Y-Koordinate (0 bis 255 sichtbar). Hier wird im Gegensatz zur Schildkrötengraphik die physikalische Adressierung gewählt, das heißt, ein Bildpunkt entspricht dem Unterschied 1. Bei der Schildkrötengraphik wurde, um symmetrische Bilder zu erhalten, die Y-Koordinate bei der Umrechnung durch 2 dividiert. Beispiel:

```
21#256.W ;X:=256
11#128.W ;Y:=128
CD MOVETO ;Positionieren
CD WAIT ;warten bis GDP
;bereit
3E 80 ;Befehl für
;Einzelpunkt
D3 70 ;an Graphik-
;prozessor ausgeben
C9 ;Programmende
```

Das kleine Programm setzt einen Punkt auf dem Bildschirm in die Bildmitte.

DRAWTO

Von der letzten Position, die im Koordinatenregister des GDPs (z.B. durch MOVETO) gespeichert ist, wird eine Linie zur neuen Position gezeichnet. Die neue Position wird durch HL und DE bestimmt. HL enthält dabei die X-Koordinate (0 bis 511) und DE die Y-Koordinate (0 bis 255).

Beispiel:

```
21 0.W ;X:=0
11 0.W ;Y:=0
CD MOVETO ;erst mal
;Positionieren
21#511.W ;X:=511
11#255.W ;Y:=255
CD DRAWTO ;Linie zeichnen
C9 ;Programmende
```

SCHR16TEL (Schildkrötengraphik)

Dieser Befehl arbeitet wie der SCHREITE-Befehl, jedoch wird im HL-Registerpaar $\frac{1}{16}$ Schrittweite angegeben. Dies ist möglich, da das Grundprogramm intern alle Koordinaten für die Schildkrötengraphik mit 16facher Genauigkeit speichert. Dieser Befehl ist z.B. für die Realisierung unterschiedlicher Kreisdurchmesser nötig. Man achte bei der Anwendung des Befehls darauf, daß sich bei ungünstiger Verwendung arithmetische Fehler häufen können. Man sollte daher versuchen, zunächst mit dem SCHREITE-Befehl auszukommen und nur im Notfall den SCHR16TEL-Befehl anzuwenden.

Beispiel:

```
21#36.W ;Kreis als 36-Eck
CD SCHLEIFE ;
21#50.W ;SCHRITTWEITE
;laden
CD SCHR16TEL
21#10.W ;Gesamtwinkel muß
;360 Grad sein
CD DREHE ;für Kreis
CD END-
SCHLEIFE
C9 ;Programmende
```

Formel zur Errechnung der Schrittweite für den SCHR16TEL-Befehl, für das obige Programm aus gegebenem Radius:

SCHRITTWEITE
:= $(R \star 1000) / 358$

WRITE

Damit kann man einen Text auf dem Bildschirm ausgeben. Der Text wird zusammen mit weiteren Parametern im Speicher wie folgt abgelegt:

```
x.W ;X-Koordinate
;des Textanfangs
```

```
y.W ;Y-Koordinate
;des Textanfangs
höhe.W ;Textgröße als Code :
;breitey*16+breitex
schräge.w ;Schriftrichtung,
;normalerweise 0
;(4, 8, 12 möglich)
„HALLO“ ;Text in ASCII
0 ;Ende des Textes
```

Im Registerpaar HL wird vor Aufruf des Befehls WRITE die Adresse des Speicherblocks mitgeteilt, auf der sich der Text befindet.

Beispiel:

```
21 8900.W ;Adresse
;des Textblocks
CD WRITE ;Befehl ausführen
C9 ;Programmende
...
8900: ;Startadresse
;für Textblock
#10.W ;X-Koordinate
#100.W ;Y-Koordinate
33.B ;Schriftgröße
;y,x = 3
0.B ;Schriftrichtung
;Standard
„Hallo“ ;Text
0 ;Endekennung
```

Der Text wird normalerweise hell auf dunklem Grund ausgegeben. Das Programm WRITE richtet sich aber nach dem Zustand des PEN/ERASER-Bits im Graphikprozessor. Dieses Bit kann man auch gezielt stellen.

Beispiel:

```
CD WAIT ;warten bis
;GDP bereit
3E 01 ;ERASER auswählen
D3 70 ;an GDP-Port
C9 ;Ende
;ERASERSELEKT
CD WAIT ;warten
3E 00 ;PEN auswählen
D3 70 ;an GDP-Port
C9 ;Ende PENSELEKT
```

Wenn man den ERASER

ELO-SOFTWARE

auswählt, kann man übrigens gezeichnete Linien auch wieder weglöschen. Dabei darf aber nicht die Schildkröte eingeblendet sein, denn für die Einblendung der Schildkröte wird der GDP wieder auf PEN geschaltet.

READ

Damit kann ein Text in den Speicher eingelesen werden. Dazu wird auch wieder ein Parameterblock verwendet, der ähnlich wie beim WRITE-Befehl aussieht.

x.W	;X-Koordinate ;des Textanfangs
y.W	;Y-Koordinate ;des Textanfangs
höhe.W	;Textgröße als Code : ;breitey*16+breitex
0.B	;nur eine Schrift- ;richtung erlaubt
max.B	;maximale Anzahl ;der einzulesenden ;Zeichen
aktuell.B	;dort steht nachher ;die tatsächliche ;Anzahl der Zeichen
----	;Speicherbereich ;freihalten, dort ;kommt der Text hin

Beim Aufruf des READ-Befehls steht im Registerpaar HL die Adresse des Parameterblocks. Mit einer „1“ im C-Register kann man eine Umrahmung des Textfensters verlangen. Wenn der Wert 0 im Register C steht, wird keine Umrahmung ausgegeben. Die Breite und Höhe des Rahmens wird durch die Werte „max“ und „höhe“ automatisch berechnet:

21 8900.W	;Adresse des ;Parameterblocks
0E 01	;mit Rahmen
CD READ	;einlesen
C9	;Programmende
...	
8900:	;Dort Rest eingeben
#10.W	;X-Koordinate
#20.W	;Y-Koordinate
22.B	;Schriftgröße y,x = 2
0.B	;immer 0
6.B	;max-Zahl
0.B	;wird erst später ;besetzt
0 0 0 0 0 0	;Platz für Text

Nach dem Start erscheint ein umrahmtes Textfenster auf dem Bildschirm. Nun kann man einen beliebigen Text eingeben, der maximal sechs Zeichen lang sein kann. Mit der Taste CR wird die Eingabe beendet, oder wenn man mehr als sechs Zeichen eingibt. Im Parameterblock wird danach die Anzahl der Zeichen gültig sein und der Text im restlichen Block abgelegt sein. Der Text wird außerdem durch eine 0 beendet, die hinter dem Text steht, so daß man immer ein Byte mehr im Speicher reservieren muß, als die „max“-Zahl angibt. Das zum Abbruch führende Zeichen ist zusätzlich im Register A enthalten, z. B. der Code für CR, wenn man ein Carriage Return eingegeben hat. Jedes Steuerzeichen führt zum Abbruch, außer Delete und Backspace, mit dem man die Texte im Fenster korrigieren kann.

CI

Eingabe eines Zeichens von der Tastatur im ASCII-Code. Das Zeichen wird im Register A im ASCII-Code abgelegt.

Beispiel:

TERMINAL:=\$;Name des ;Programms
CD CI	;Zeichen einlesen
CD WAIT	;warten bis ;GDP fertig
D3 70	;Ausgabe als ;Kommando
C3 TERMINAL	;wieder zurück

Wenn man einen Text eingibt, so erscheint er auf dem Bildschirm, jedoch ist bei diesem einfachen Programm noch kein Zeilenvorschub möglich.

CLR

Alle vier Bildseiten der GDP-Baugruppe werden gelöscht. Die Löschung erfolgt in ca. 80 Millisekunden.

Beispiel:

CD CLR	;löschen
C9	;Ende

CSTS

Damit kann man prüfen, ob eine Taste auf der Tastatur gedrückt wurde. Wenn ja, so erscheint im Register A der Wert FF, sonst der Wert 0.

Beispiel:

DAUER:=\$;Programmname
21#1.W	;einen Schritt gehen
CD SCHREITE	
21#1.W	;um Kreis zu
CD DREHE	;zeichnen
CD CSTS	;solange, bis Taste ;gedrückt wird
B7	;(OR A, Flags ;bestimmen)
CA DAUER	;=0, dann nochmals
C9	;sonst beenden

RI

Wie CI, jedoch wird ein Zeichen von der CAS-Baugruppe gelesen. Das Zeichen erscheint auch im Register A.

Beispiel:

WARTE:=\$	
CD RI	;Zeichen lesen
FE FF	;warten bis FF ;auftaucht
C2 WARTE	;nein, dann zurück
C9	;Ende gefunden

PO

Zeichen an die CAS-Baugruppe ausgeben. Dabei steht das Zeichen im Register C. Wenn man selbst einmal Daten auf dem Kassettenrecorder mit einem eigenen Programm abspeichern will, so muß man darauf achten, daß das erste ausgegebene Zeichen ungültig ist. Daher sollte man, auch um einen definierten Start zu haben, zunächst ein paar Synchronisationsbytes ausgeben.

Beispiel:

SYNCAUS:=\$	
21#10.W	;erst mal ;10 Stück FF
CD SCHLEIFE	
0E FF	;Zeichen laden
CD PO	;aufzeichnen
CD END- SCHLEIFE	;und dann
0E 00	;z. B. Erkennungsbyte
CD PO	;danach kann man ;Daten aufzeichnen
C9	;hier aber Ende

CLPG

Wenn man eine unsichtba-

ELO-PRAXIS

re Seite löschen will, so muß man diesen Befehl anwenden. Er ist langsamer als der CLR-Befehl, kann jedoch, ohne Bildstörungen zu verursachen, eine beliebige Seite löschen. Die Löschseite wird im Seitenregister (Port 60H) angegeben.

Beispiel:

```
CD WAIT      ;erst muß GDP
              ;bereit sein
3E 40        ;Seite 1 schreiben,
              ;0 lesen
D3 60        ;und
CD CLPG      ;diese Seite löschen
C9
```

Den Code für das Seitenregister kann man wie folgt berechnen:

$\text{schreibseite} * 64 + \text{lese-}$
 $\text{seite} * 16$

Der Wertebereich ist:
0,1,2,3

WAIT

Diesen Befehl hatten wir schon oft verwendet. Nach Aufruf wartet das Programm so lange, bis der Graphikprozessor seinen letzten Befehl abgearbeitet hat. Wenn man einmal ein WAIT vergißt, so kann es zu Störungen kommen. Die Schildkrötenbefehle, wie auch MOVETO, DRAWTO, WRITE und READ, warten automatisch, so daß man WAIT eigentlich nur dann braucht, wenn man direkt auf die Register des GDPs zugreifen will.

Beispiel:

```
CD WAIT
3E 41        ;Buchstabe "A"
              ;ausgeben
D3 70        ;mit aktuellen
              ;Parametern
C9
```

RAM

Ein Symbol, das nicht als Befehl verwendet werden kann. Damit wird die Adresse 8800 angesprochen. Man kann den NAMEN RAM z.B. eingeben, wenn man im Menü AENDERN die erste Speicherzelle verwenden will.

Hier noch ein Hinweis. Der Bereich 8000 bis 81CA ist vom Grundprogramm re-

serviert und sollte nicht verwendet werden. Der Bereich 81CA bis 8FFF ist eigentlich frei, jedoch werden, beginnend bei 81CA, die Symbole abgelegt, die man durch ":= " definiert hat, und daher sollte man erst bei 8800 beginnen,

Programme abzulegen. Die Adresse 8800 hat sich als guter Kompromiß herausgestellt. Wenn man nicht sehr viele Symbole verwendet, so kann man auch bei 8400 oder gar bei 8200 beginnen.

Rolf-Dieter Klein

ELO-PRAXIS

Erweiterung für NDR-Klein-Computer

Freudenknüppel an NDR-Computer

Jeder Heimcomputer, der auf sich hält, hat Anschlüsse für die Joysticks (auf deutsch: Freudenknüppel). Was den anderen recht ist, soll dem NDR-Computer billig sein.

Wir haben den Commodore-Joystick verwendet, der ja dieselben Anschlüsse besitzt wie der von Atari und vielen anderen Rechnern. Es gibt hier eine reiche Auswahl in allen Preisklassen. **Bild 1** zeigt, wie der Joystick mit der IOE-Karte verbunden wird. Mit der gezeigten Verbindung ist es zum Beispiel möglich, für das Mondlandespiel aus der Sendung (siehe auch Z-80-Aufbauprogramme) anstelle der im Film gezeigten Zusatzstatatur den Joystick zu verwenden. Der Joystick wird an den Eingang I/O 0 angeschlossen, an den Eingang I/O 1 einen zweiten Joystick nach gleichem Muster anzuschließen. **Bild 2** zeigt, wie die Verbindung zwischen der IOE-Karte und dem 9poligen Cannonstecker (aus dem Fachhandel) verdrahtet wird. Viel Spaß beim ersten selbstprogrammierten Videospiele.

Rolf-Dieter Klein
Jürgen Plate

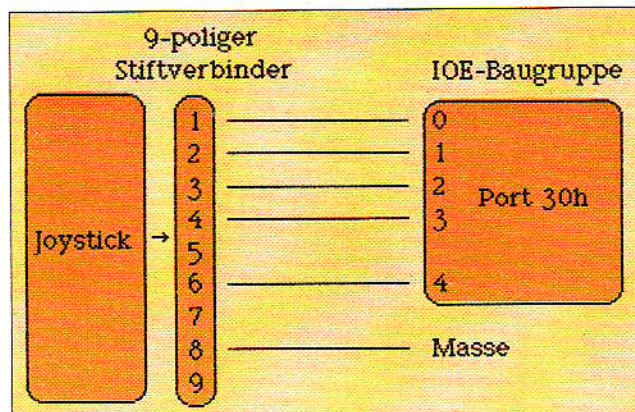


Bild 1: So werden Joystick und IOE-Karte verbunden

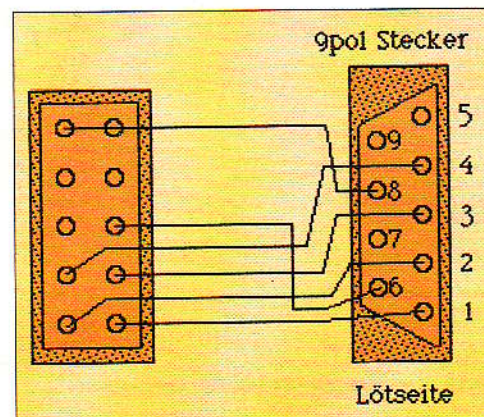


Bild 2: Anschluß des Cannon-Steckers