

Bei der Verwendung professioneller Software (wie zum Beispiel Spielen oder Zeichenprogrammen), fällt es einem schon gar nicht mehr auf. Der Gebrauch des Joysticks wird hier fast als Selbstverständlichkeit empfunden. Will man es dann selbst einmal versuchen, kommt man mit dem Commodore-Handbuch nicht weit. Deshalb soll hier ein Minimalprogramm zur Abfrage des Joysticks weiterhelfen. Diese zehn hier zu erarbeitenden Basic-Zeilen können vielleicht einmal die Basis eines komfortablen Menüs bilden.

Zunächst einmal müssen wir wissen, was wir mit der Abfrage bezwecken wollen. Um uns auf das Wesentliche konzentrieren zu können, bewegen wir einfach ein invertiertes

Verfügung, die ebenfalls mit PEEK gelesen werden kann.

```
20 ME=PEEK(BILD):POKE BILD,
160: POKE BILD+FARBE,
PEEK(646)
```

Jetzt kommt aber endlich der Joystick dran. Die Übergabe erfolgt in der Variablen JS:

```
30 JS=PEEK(PO):BEWEG=0
```

BEWEG wird nachher die Summe der Bewegungen, die natürlich vor der Auswertung von JS auf Null gesetzt werden muß.

Vom Byte des Joystick-Ports interessieren uns genau die ersten fünf Bits, die über die logische Verknüpfung AND ausgefiltert werden. Ein vom Joystick aktiviertes Bit hat den Wert Null, dem Ruhezustand entspricht die logische Eins (Vorsicht, Pferdefuß). Um auf dem Bildschirm

```
unter unserem »Cursor« wieder her.
80 IF (JS AND 16)=0 THEN POKE
BILD,ME:END
```

Um nicht aus dem Bildschirm »herauszufahren« (dahinter liegt schließlich unser Basic-Programm), müssen wir testen, ob die alte Position zuzüglich der Summe der Bewegungen noch im Bereich von 1024 bis 2023 liegt.

```
90 IF BILD+BEWEG<1024 OR
BILD+BEWEG>2023 THEN
BEWEG=0
```

Jetzt brauchen wir nur noch unseren »Cursor« bewegen. Dazu POKEn wir zunächst das vorhin gemerkte Zeichen auf die aktuelle Position. Dann zählen wir die errechneten Bewegungen zur Bildschirmadresse dazu. Alles was bleibt, ist ein Rücksprung zu Zeile 20, um unseren

Der Trick mit dem Joystick

**Wissen Sie eigentlich, wie man den Joystick abfragt?
Wenn nicht, dann sollten Sie unbedingt weiterlesen.**

Leerzeichen über den Bildschirm, gesteuert durch den Joystick an Port 2. Die Abfrage selbst geschieht durch den Basic-Befehl PEEK, also das Auslesen einer Speicherzelle. Im Falle des Port 2 ist das die Adresse 56320. Für Port 1 wäre es 56321. Um uns auf dem Bildschirm orientieren zu können, benötigen wir dessen Anfangsadresse (1024, das ist links oben) und Endadresse (2023, rechts unten), sowie den dazugehörigen Farbspeicher (pro Bildpunkt 54272 addieren). Unser Programm fängt also so an:

```
10 PO=56320:BILD=1024:FARBE=
54272
```

Als nächstes müssen wir unser Zeichen auf den Bildschirm bringen – und zwar ohne ein eventuell bereits vorhandenes Zeichen zu verlieren. Und für den Fall, daß kein altes Zeichen vorhanden war, ist noch die aktuelle Farbe zu setzen. Das zu merkende Zeichen lesen wir mit PEEK aus, und heben es in der Variable ME auf. Dann schreiben wir unser Zeichen (es hat den Code 160) mit POKE auf den Bildschirm. Die aktuelle Farbe stellt das Betriebssystem in der Adresse 646 zu unserer

eine Bewegung nach oben darzustellen, subtrahiert man von der Position des Zeichens den Wert 40. Eine Zeile nach oben entspricht dem ersten Bit der Variablen JS. In Basic heißt das:

```
40 IF (JS AND 1)=0 THEN BEWEG=
BEWEG-40
```

Mit den anderen Richtungen wird es genauso gehandhabt. Abwärts entspricht dem zweiten Bit:

```
50 IF (JS AND 2)=0 THEN BEWEG=
BEWEG+40
```

Das dritte Bit ist für Bewegung nach links zuständig, und entspricht auf dem Bildschirm der Subtraktion von Eins:

```
60 IF (JS AND 4)=0 THEN BEWEG=
BEWEG-1
```

und nach rechts:

```
70 IF (JS AND 8)=0 THEN BEWEG=
BEWEG+1
```

Auf diese Weise sind auch die möglichen Diagonalbewegungen erfaßt, zum Beispiel entspricht die Bewegung nach links unten dem Wert +39.

Die letzte Abfrage gilt dem Feuerknopf. Bei dessen Betätigung wollen wir das Programm beenden. Dabei stellen wir natürlich das Zeichen

Cursor wieder korrekt auf den Bildschirm zu bringen.

```
100 POKE BILD,ME:BILD=BILD+
BEWEG:GOTO20
```

Will man den Joystick an Port 1 anbringen, ist lediglich Zeile 10 zu ändern. Die Adresse ist dann 56321. Also:

```
10 PO=56321...
```

Um zwei Joysticks gleichzeitig abzufragen, sind folgende Zeilen umzuschreiben:

```
10 P1=56321:P2=56320:BI=1024...
```

```
30 J1=PEEK(P1):J2=PEEK(P2):
```

```
JS=NOT(NOT J1 OR NOT J2):
```

```
BEWEG=0
```

Die Verknüpfung entsteht durch die negative Logik der Ports.

Da ein Joystick an Port 1 Zeichen in den Tastaturpuffer schreibt, ist es sinnvoll, beim Verlassen des Programms diesen zu löschen; am besten, indem man in Zeile 80 vor den END-Befehl ein »POKE 198,0« setzt.

Damit wäre unser Minimal-Programm beendet, obwohl es jetzt eigentlich erst richtig losgeht. Wir hoffen, daß Sie die Joystick-Abfrage in Ihren eigenen Programmen ausnutzen können. Dann haben Sie das Lernziel erreicht. (og)