

C 64 extern — Der Weg nach draußen (Teil 1)

Die meisten Computer-Anwender schließen nur käufliche Peripherie an ihren Homecomputer an. Oft fehlt das Wissen, wie man sich mit einfachen Mitteln neue Peripherie- und Steuergereäte selbst bauen kann. Dieser Kurs soll sowohl dem C 64- als auch dem VC 20-Besitzer einen Einstieg in diese Materie bieten.

Der C 64 und der VC 20 haben an ihren User- und Control-Ports (Joystickports) viele Anschlüsse, an denen man sehr leicht verschiedene Steuergeräte betreiben kann. Der Schwerpunkt dieses Kurses wird hauptsächlich auf dem C 64 liegen, jedoch sollen auch die VC 20-Besitzer nicht zu kurz kommen. Alle Programmbeispiele sind nämlich auf beiden Rechnern lauffähig. In dieser Folge wollen wir uns den Control-Ports zuwenden und lernen, wie man hier dem Computer Signale zuführen kann. Der Kurs wird vor allem für diejenigen interessant sein, die sich mit solchen »externen« Möglichkeiten noch nie beschäftigt haben.

Wenn Sie die Versuche experimentell nachvollziehen wollen, sollten Sie sich unbedingt je einen Stecker für den Control-Port und für den User-Port besorgen. Diese Stecker sind zwar leider recht teuer, doch für ein sicheres Zugreifen auf die einzelnen Pins des Ports unabdingbar. Sie sind, wie alle für diesen Kurs benötigten Teile, in Elektronik- oder Computerfachgeschäften erhältlich.

Wer Angst um seine Computeranlage hat, sollte vor dem Einschalten alle nicht benötigten Peripheriegeräte (Drucker, Floppy, Datasette etc.) abtrennen. Wenn Sie je-

doch vorsichtig sind und genau nach den gegebenen Anweisungen vorgehen, kann nichts schiefgehen.

Als erstes wollen wir uns den beiden Anschlüssen »POT X« und »POT Y« an den Control-Ports zuwenden. Diese beiden Anschlüsse (Pin 5 und 9, siehe Bild 1) sind dazu da, dem Computer analoge Daten (analog = stufenlos, stetig; in diesem Fall Spannungswerte) mitzuteilen. Die analogen Daten werden in einem sogenannten Analog/Digital-Wandler in digitale Daten, also in eine Zahl zwischen 0 und 255 umgewandelt. Das geschieht ohne unser Zutun automatisch im Computer. Die umgewandelten Daten kann man dann aus Speicherzellen auslesen. Die Zahl in diesen Speicherzellen ist um so kleiner, je größer die Spannung an POT X/Y ist. Deshalb stehen diese Register, wenn nichts angeschlossen ist, auch immer auf 255 (= 0 Volt an POT X/Y).

Paddles im Selbstbau

Der C 64 hat an jedem seiner beiden Control-Ports je ein Anschlußpaar POT X/POT Y. Beim VC 20 gibt es nur ein solches Paar.

Wie sprechen wir diese Funktion des Computers an? Dazu müssen wir dem Computer also analoge Daten zu-

führen. Eine Möglichkeit besteht darin, über ein Potentiometer (= einstellbarer ohmscher Widerstand) verschiedene Spannungswerte an POT X oder POT Y einzustellen. Wenn Sie kein Poti (Potentiometer) zur Hand haben, können Sie in Elektronikläden eins kaufen. Falls Sie ein paar Paddles Ihr Eigen nennen, können Sie natürlich ebensogut dieses anschließen. Ich werde aber im folgenden von Potis sprechen, weil Paddles nur wenig verbreitet sind.

Über den idealen Widerstandswert des Potis gibt es in der Fachliteratur unterschiedliche Angaben. Ich empfehle ein 470-Kilo-Ohm-Poti. Die relativ guten Werte, die man damit erzielt, können durch Parallelschalten eines Festwiderstandes von etwa 1 Mega-Ohm noch verbessert werden. Sie brauchen jedoch vor dem Ausprobieren verschiedener Widerstände keine Angst zu haben, denn ein falscher Widerstand ist völlig ungefährlich: Bei zu kleinem Widerstandswert ist die Zahl 255 im POT X/Y-Register unmöglich, andernfalls wird sie zu früh erreicht, das heißt man muß das Poti nicht ganz aufdrehen, um die Zahl 255 zu erreichen (darunter leidet nur die Fähigkeit, Werte exakt einzustellen).

Jetzt wollen wir unser Poti aber erst einmal an den C 64 beziehungsweise VC 20 anschließen: Von den drei Anschlüssen des Potis verbinden Sie den mittleren mit POT X (Pin 9) und einen der verbleibenden (egal, welchen) mit +5 V (Pin 7) des Control-Ports 1 des C 64 (die VC 20-Leute natürlich mit den Anschlüssen des einzigen vorhandenen Control-Ports). Der dritte Anschluß bleibt frei. Wenn Sie Ihre Arbeit auf eventuelle Kurzschlüsse überprüft haben, schalten Sie Ihren Computer ein. Arbeiten an den Ports sollten grundsätzlich nur bei ausgeschaltetem Gerät durchgeführt werden. Tippen Sie nun das Programm »Männchendemo« (Listing 1) ab. Um das Programm an den VC 20 anzupassen, entfernen Sie eventuell vorhandene RAM-Erweiterungen und geben die Änderungen

mit ein, die am Ende des Listings angegeben sind. Das gilt für alle Programme dieses Kurses.

Starten Sie nun das Programm mit »RUN«. Ein Männchen läuft über den Bildschirm. Wenn Sie am Potentiometer drehen, ändert sich die Laufgeschwindigkeit des Männchens kontinuierlich.

Wie funktioniert das? Sehen Sie sich das Listing zum Programm an. Das Programm besteht aus zwei gleich aufgebauten Teilen



PIN	Belegung
1	JOY0
2	JOY1
3	JOY2
4	JOY3
5	POT Y
6	C 64, Port 2: FIRE C 64, Port 1/VC 20: FIRE/LIGHT PEN
7	+5 V
8	GND
9	POT X

Bild 1. Die Anschlußbelegung der Control-Ports

und einem Unterprogrammteil. Der erste Teil ist für die Bewegung nach rechts, der zweite für die Bewegung nach links zuständig. Aus beiden Teilen wird das Unterprogramm aufgerufen. Das Neue an diesem Programm steckt in der Zeile 590. Hier ist eine Warteschleife programmiert, die J von 1 bis PEEK (C) durchlaufen läßt. C steht für die Adresse, in welcher der Computer die digitale Information über den Spannungswert an POT X ablegt. Ich habe hier und auch in den folgenden Programmen benutzt viele Variablen verwendet, weil dadurch die Adaption an den VC 20 vereinfacht wird. Daher müssen zur Anpassung dieses Programms nur drei Zeilen ausgetauscht werden, weil zum Beispiel die Anfangsadresse des Bildschirm-RAMs oder die Zeilenzahl (in noch folgenden Programmen) als Variable vordefiniert wurden.

Dem POT X-Anschluß ist also beim C 64 die Adresse

```

200 REM MAENNCHENDEMO ! <038>
210 REM ----- <109>
220 REM BY TOBIAS NICOL <190>
225 REM <031>
230 REM ***** INITIALISIERUNG ***** <201>
240 POKE 53280,0 : POKE 53281,0 <112>
250 A = 1545 : B = 1582 : C = 54297 <060>
260 D = 55776 : E = 40 <170>
270 PRINT "{CLR}" <004>
290 REM ***** FARBSPEICHER SETZEN ***** <116>
300 FOR I = D TO D+3*E <104>
310 POKE I,1 <200>
320 NEXT I <150>
350 REM **** BEWEGUNG NACH RECHTS **** <141>
360 FOR I = A TO B <087>
370 GOSUB 550 <172>
380 POKE I-1,32 : POKE I+1+E,32 <110>
390 POKE I-1+E,32 : POKE I-E,32 <015>
400 NEXT I <230>
420 REM ***** BEWEGUNG NACH LINKS ***** <045>
430 FOR I = B TO A STEP -1 <000>
440 GOSUB 550 <242>
450 POKE I+1,32 : POKE I+1+E,32 <164>
460 POKE I-1+E,32 : POKE I-E,32 <085>
470 NEXT I <044>
490 REM *** RUECKSPRUNG ZUM ANFANG *** <060>
500 GOTO 350 <014>
520 REM *** ENDE DES HAUPTPROGRAMMS *** <130>
550 REM ** UNTERPROGRAMM "BEWEGUNG" ** <071>
560 POKE I,160 : POKE I+1,77 <076>
570 POKE I-1,78 : POKE I-E,81 <122>
580 POKE I-1+E,103 : POKE I+1+E,101 <160>
590 FOR J = 1 TO PEEK (C) : NEXT J <025>
600 RETURN <150>
640 REM ***** <014>
650 REM *** AENDERUNGEN FUER VC-20 *** <223>
660 REM ***** <034>
670 REM <224>
680 REM BITTE DIESE ZEILEN ERSETZEN: <027>
690 REM <244>
700 REM 240 POKE 36879,8 <234>
710 REM 250 A=7901 : B=7920 : C=36872 <005>
720 REM 260 D=38620 : E=22 <150>

```

© 64'er

Listing 1. Das Männchendemo (Paddles). Alle Listings bitte mit dem neuen Checksummer aus dieser Ausgabe eingeben.

54297 und beim VC 20 die Adresse 36872 zugeordnet. Für POT Y ist bei beiden Computern jeweils die nächste Speicherzelle auszulesen, also 54298 beziehungsweise 36873. Durch die »extern« variierte Warteschleife wird die Wanderung des Männchens über den Bildschirm unterschiedlich stark verlangsamt, wodurch eine Geschwindigkeitskontrolle realisiert ist.

Probleme mit Port 2

In dem Programm »Tondemo« (Listing 2) werden dann POT X und POT Y gleichzeitig benutzt. Für den richtigen Betrieb des Programms benötigen Sie eigentlich zwei Potentiometer, Sie können jedoch auch eines abwechselnd an POT X und POT Y betreiben. Wenn Sie gerade vor einem C 64 sitzen, dann schließen Sie für dieses Programm beide Potis an den Control-Port 2 an. Ich verwende jetzt bewußt den

zweiten Port, weil man hier noch ein paar Dinge beachten muß, die bei Port 1 nicht notwendig sind.

Zurück zum Programm: Mit dem X-Poti wird die Lautstärke des Tones, mit dem Y-Poti die Frequenz des Tones eingestellt. Die Werte werden in den Zeilen 320 und 330 ausgelesen und für die Tonregister aufbereitet. Bevor wir den Port 2 des C 64 auslesen können, müssen wir diesen allerdings noch dafür vorbereiten. Zunächst schalten wir die Interrupt-routine des C 64 (IRQ) ab. Das geschieht durch POKE 56333,1 und ist notwendig, weil wir sonst schwankende Werte erhalten. Vielleicht ist Ihnen schon aufgefallen, daß die Register, aus denen ausgelesen wird (Zeilen 320/330), dieselben sind wie beim ersten Programm, obwohl wir doch jetzt den Port 2 beschaltet haben. Wenn Sie das bemerkt haben, können Sie sich schon denken, was jetzt noch nötig ist: Wir müs-

```

200 REM ** TONDEMO ** ! <225>
210 REM ----- <232>
220 REM BY TOBIAS NICOL <190>
225 REM <031>
230 REM ***** BILDSCHIRMAUFBAU ***** <094>
240 PRINT "{CLR,SDOWN,7SPACE}TONDEMO" <130>
250 PRINT "{7SPACE}-----" <021>
260 PRINT "{3DOWN,SPACE}POT X = LAUTSTAERK <226>
    E" <150>
270 PRINT "{3DOWN,SPACE}POT Y = TONHOEHE" <150>
290 REM ***** WERTE AUSLESEN ***** <023>
300 POKE 56333,1 <241>
310 POKE 56320,191 <167>
320 L = INT (PEEK (54297) / 17) <126>
330 T = PEEK (54298) <151>
340 POKE 56320,127 <217>
350 POKE 56333,129 <180>
370 REM ***** TON ERZEUGEN ***** <021>
380 POKE 54296,L <059>
390 POKE 54279,200 <082>
400 POKE 54280,T <028>
410 POKE 54285,240 <181>
420 POKE 54283,17 <071>
440 REM ***** WERTE DARSTELLEN ***** <130>
450 PRINT "{HOME,19DOWN}" <120>
460 PRINT "{17SPACE}" <103>
470 PRINT "{UP,SPACE}L = "L,"T = "T <075>
490 REM ***** RUECKSPRUNG ***** <254>
500 GOTO 290 <070>
540 REM ***** <170>
550 REM *** AENDERUNGEN FUER VC-20 *** <123>
560 REM ***** <190>
570 REM <124>
580 REM DIE ZEILEN 300-350 UND 380-420 <168>
590 REM LOESCHEN. <127>
600 REM <154>
610 REM NEU EINGEBEN: <183>
620 REM <174>
630 REM 300 L=INT(PEEK(36872)/17) <041>
640 REM 310 T=INT((PEEK(36873)*.985)/2) <174>
650 REM +128 <140>
660 REM 380 POKE 36878,L <222>
670 REM 390 POKE 36876,T <127>

```

© 64'er

Listing 2. Tondemo (Paddles)

sen dem C 64 mitteilen, welchen Port wir auslesen wollen. Diese Information erhält er über Bit 6 und 7 der Adresse 56320. Das verhält sich so:

	Bit 6	Bit 7
Port 1	1	0
Port 2	0	1

In dieser Adresse müssen wir also eine 127 ablegen (das ergibt sich durch die Kombination der anderen Bits in 56320), wenn wir Port 1 abfragen wollen. Dieser Wert steht hier aber sowieso immer, weshalb wir uns das Einstellen dieses Registers bei Programm 1 sparen konnten (auch das Abschalten des Interrupts ist unnötig, da sich keine Probleme ergeben). Wollen wir Port 2 auslesen, müssen wir vorher 191 nach Adresse 56320 POKE. Erwähnenswert ist außerdem, daß dieser Wert vor jeder Abfrage neu in dieses Register eingeschrieben werden muß, da er vom Betriebssystem immer wieder

mit der 127 von Port 1 überschrieben wird. Um diesen Sachverhalt nachzuprüfen, geben Sie einmal folgendes Miniprogramm ein:

```

10 POKE 56333,1 :
POKE 56320,191
20 PRINT PEEK (56320) :
GOTO 20

```

Nach RUN wird ein paar-mal der Wert 191 und danach nur noch der Wert 127 ausgegeben.

Nach dem Auslesen wird dann in den Zeilen 340/350 der alte Wert wieder in das Register 56320 eingeschrieben und der Interrupt wieder eingeschaltet (POKE 56333,129).

Wenn Sie das Programm für den VC 20 anpassen, fällt Ihnen wahrscheinlich auf, daß sehr viel mehr gelöscht als neu eingefügt werden muß. Das rührt daher, daß beim C 64 sowohl die Tonerzeugung als auch das Auslesen der Port-2-Register komplizierter ist. Bei der Tonerzeugung wird das allerdings durch die weitaus bessere

Qualität der Töne wettgemacht.

Das dritte Programm, »X/Y-Steuerung« (Listing 3), zeigt eine weitere Anwendung der POT X/Y-Register. Ein Objekt (in diesem Fall ein Rechteck) kann durch die zwei Potis, die jetzt wieder an Port 1 betrieben werden müssen, an jede beliebige Position des Bildschirms gebracht werden. Die Schwierigkeit bei dieser Anwendung besteht wieder darin, die Werte, die ausgelesen werden (Zeile 270,280), geeignet aufzubereiten, also in Bildschirmkoordinaten umzurechnen. Das wird in den Zeilen 310 und 320 für die X- und Y-Richtung getan. Für den VC 20 müssen wieder nur die Variablenbelegung und die Datenaufbereitung (wegen des kleineren Bildschirms, aber der gleich großen Werte 0 bis 255) ausgetauscht werden.

Diese drei vorgestellten Programme sind für sich allein betrachtet natürlich wenig sinnvoll, um nicht zu sagen langweilig. Sie sollen jedoch auch nur eine Demonstration dafür sein, was man mit den POT X/Y-Registern machen kann und wie man sie ausliest.

Was man mit den Control-Ports alles machen kann ...

Eine weitere Möglichkeit in der Verwendung von POT X/Y liegt im Anschließen von »Umweltsensoren« wie zum Beispiel Licht- oder Temperaturmessern. So kann man statt des Potis einfach einen LDR (Light Dependent Resistor = lichtempfindlicher Widerstand) anschließen. Die Werte, die man aus den Registern ausliest, hängen dann von der Lichtintensität ab, die auf den LDR fällt. Man kann also durch ein Programm die Helligkeit des Raums überprüfen lassen. Bei einem Programm, das den Benutzer besonders lang vor dem Bildschirm fesselt, kann man damit einen besonderen Gag einbauen: Wenn die Sonne untergeht, und es im Zimmer dunkel wird, meldet sich der Computer mit der Information, daß man langsam das Licht einschalten sollte, um sich nicht die Augen zu verderben (das wäre doch mal etwas ganz Neues!).

```

200 REM X / Y - S T E U E R U N G
205 REM -----
210 REM BY TOBIAS NICOL
215 REM
220 PRINT "{CLR}"
230 A = 1024 : B = 55296 : C = 40
240 D = 54297
260 REM ***** WERTE AUSLESEN *****
270 P1 = PEEK (D)
280 P2 = PEEK (D+1)
300 REM ***** WERTE AUFBEREITEN *****
310 X = INT (P1/6.538)
320 Y = INT (P2/10.625)
340 REM ***** PUNKT VERSCHIEBEN *****
350 POKE A+XA+YA*C,32
360 POKE A+X+Y*C,160
370 POKE B+X+Y*C,1
380 XA = X : YA = Y
400 REM ***** RUECKSPRUNG *****
410 GOTO 260
450 REM *****
460 REM *** AENDERUNGEN FUER VC-20 ***
470 REM *****
480 REM
490 REM FOLGENDE ZEILEN ERSETZEN:
500 REM
510 REM 220 PRINT "{CLR}" : POKE 36879,8
520 REM 230 A=7680 : B=38400 : C=22
530 REM 240 D=36872
540 REM 310 X=INT(P1/12.14)
550 REM 320 Y=INT(P2/11.59)

```

6 64'er

Listing 3. Ein Beispiel zur X/Y-Steuerung (Paddles)

Für den Lichtmesser muß man die auf den POT-Registern ausgelesenen Werte geeignet aufbereiten, wie es schon in der »Tondemo« für die Tonregister geschehen ist. Man eicht den Lichtmesser, wozu man allerdings einen bereits abgeglichenen Lichtmesser benötigt.

Um Temperaturen zu messen, nimmt man statt des lichtempfindlichen einen wärmeempfindlichen Widerstand. Diese Anwendung fällt aber genau wie der Lichtmesser schon in den Bereich der Meßtechnik. Der Phantasie sind in der Verwendung dieser Register keine Grenzen gesetzt.

Wenn Sie ein Paar Paddles besitzen, werden Sie sich vielleicht schon gefragt haben, wie man denn jetzt die beiden Feuerknöpfe an den Paddles abfragt. Diesbezüglich müssen Sie sich aber leider noch bis zur nächsten Folge dieses Kurses gedulden. Zur Zeit fehlen nämlich noch ein paar Grundkenntnisse, die zum Verstehen der Abfrage der Feuerknöpfe wichtig sind.

Zur Verwendung in Programmen eignen sich Paddles besonders dann, wenn eben zum Beispiel Geschwindigkeit oder Tonhöhe eingestellt (Männchen- und Tondemo) oder ein Objekt schnell und präzise zu einer

bestimmten Stelle bewegt werden muß (X/Y-Steuerung), weil durch das Potentiometer theoretisch ein direkter Sprung zu jeder Bildschirmkoordinate möglich ist (durch schnelles Drehen), während man sich beim Joystick meistens mit einer konstanten Geschwindigkeit fortbewegt (Ausnahmen bestätigen die Regel).

Apropos Joystick: Haben Sie sich nicht schon einmal gefragt, wie Sie den Joystick in eigenen Programmen benutzen können? Ich möchte Ihnen jetzt zeigen, wie das funktioniert.

So wird der Joystick abgefragt

Während man an die POT X/Y-Anschlüsse analoge Spannungswerte anlegt, verlangen die Register, die den Joystick überwachen, klare Bedingungen: Hier gibt es nur zwei Zustände, nämlich Strom und keinen Strom beziehungsweise Eins und Null beziehungsweise +5 V (Pin 7) und GND (= Ground: negativer Pol der Computer-versorgungsspannung, Pin 8). Deshalb ist der Commodore-Joystick ein sogenannter Schalter-Joystick.

Etwas anderes als der Schalter-Joystick ist der Potentiometer-Joystick, bei

dem mit dem Steuerhebel Widerstandswerte auf Potentiometerbahnen eingestellt werden. Diese Potentiometer-Joysticks werden an die POT X/Y-Anschlüsse angeschlossen (es handelt sich also praktisch um zwei Paddles, die mit einem gemeinsamen »Steuerhebel« bewegt werden). Doch bleiben wir beim gewohnten Schalter-Joystick.

Durch Bewegen des Joystick-Steuerhebels können vier mögliche Kontakte geschlossen werden, davon maximal zwei gleichzeitig. Das ist bei den Zwischenrichtungen NO, SO, SW und NW der Fall. Der Feuerknopf hat einen eigenen Kontakt.

Mit diesen Kontakten stellt man die Verbindung zwischen GND (Ground, Masse) und den Control-Port-Anschlüssen 1, 2, 3, 4 oder 6 her. Welchen Richtungen diese Anschlüsse entsprechen, können Sie der Tabelle 1 entnehmen. Daß der Feuerknopf an Port 1 auf einen Anschluß namens Light Pen wirkt, lassen wir vorerst außer acht.

Im Gegensatz zu den Paddles gibt es für den Joystick im C 64 zwei Register, in denen die Joystickinformation abgelegt wird, für jeden Port also ein eigenes Register.

Der VC 20 hat zwar auch zwei Joystickregister, jedoch wird hier die Joystickinformation auf eben diese beiden Register aufgeteilt.

Bevor wir den Joystick auslesen können, müssen wir den Computer dafür vorbereiten. In der Vorgehensweise hierfür gibt es wieder erhebliche Unterschiede zwischen den C 64-Ports 1 und 2 und VC 20. Beginnen wir mit dem Port 2 des C 64: Zum Einschalten der Joystickfunktionen müssen wir die Bits 0 bis 4 der Adresse 56322 löschen. Das tun wir eleganterweise ohne die restlichen Bits zu beeinflussen mit POKE 56322, PEEK (56322) AND 224.

Zur Erinnerung: Die 224 ist die Summe der Wertigkeiten der Bits 5 bis 7 ($224 = 128 + 64 + 32 = 2^7 + 2^6 + 2^5$). Das sind genau die, die wir mit unserem POKE nicht verändern wollten. Da die restlichen Bits in 224 gleich Null

sind, werden diese im Ergebnis durch AND gelöscht, was ja unsere Absicht war!

Nachdem wir diesen POKE eingegeben haben, ist unser Computer für die Eingabe auf Joystick umgeschaltet. Sie sollten den genannten POKE übrigens niemals im Direktmodus eingeben, da danach die Tastatur Ihres Computers nicht mehr funktioniert (einschließlich RUN/STOP/RESTORE). Sie können den C 64 also nur noch ausschalten, und alles ist weg.

Beim Port 1 ist im Prinzip dieselbe Operation mit Register 56323, wie bei Port 2 mit Register 56322 nötig. Da in diesem Register jedoch sowieso immer eine Null steht (also alle Bits gelöscht sind), kann man sich die Arbeit sparen.

Beim VC 20 müssen Sie POKE 37139,0 : POKE 37154,127 eingeben, um auf Joystick umzustellen. Auch hier wird die Tastatur teilweise außer Funktion gesetzt. RUN/STOP/RESTORE bleibt Ihnen jedoch erhalten.

Was dieses ganze gePOKE im einzelnen bedeutet und warum es nötig ist, erfahren Sie in der nächsten Folge. Sie sollten jedoch noch wissen, wie Sie die Eingabe wieder auf die Tastatur legen:

C 64, Port 1: keine Eingabe nötig
C 64, Port 2: POKE 56322, PEEK (56322) OR 31
VC 20: POKE 37154,255 : POKE 37139,128

Nachdem die AND-Funktion oben schon erklärt wurde, nun noch kurz das OR: Wir wollen die Bits 0 bis 4 in 56322 setzen und die restlichen nicht verändern. Die 31 ist die Summe der Wertigkeiten der Bits 0 bis 4 ($31 = 00011111$ binär). Die restlichen Bits sind Null und werden daher im Ergebnis durch OR nicht verändert. Die Bits 0 bis 4 werden aber auf jeden Fall gesetzt.

Jetzt sollten wir den Joystick aber einmal abfragen, was durch die Störung der Tastatur jedoch leider im Direktmodus unmöglich ist. Die Information über den Zustand des Joystickhebels steht in folgenden Speicherstellen:

C 64, Port 1 : 56321
C 64, Port 2 : 56320

RICHTUNG	NORDEN	SÜDEN	WESTEN	OSTEN	FEUER
AN-SCHLUSS	JOY 0	JOY 1	JOY 2	JOY 3	FIRE
PIN	1	2	3	4	6
ADRESSE	56321				
BIT NR.	0	1	2	3	4
WERT DES BITS	1	2	4	8	16
ADRESSE	56320				
BIT NR.	0	1	2	3	4
WERT DES BITS	1	2	4	8	16
ADRESSE	37137		37152		37137
BIT NR.	2	3	4	5	7
WERT DES BITS	4	8	16	32	128

Tabelle 1. Alle wissenswerten Daten zur Joystickabfrage auf einen Blick

```

200 REM JOYSTICK DEMO 1
210 REM -----
220 REM BY TOBIAS NICOL
225 REM
230 REM ***** INITIALISIERUNG *****
240 X1=0 : Y1=0 : X2=0 : Y2=0 : F=0
250 POKE 53280,2 : POKE 53281,0 : D=25
260 PRINT "{CLR}" : A=1024 : B=55296 : C=4
    0
280 REM **** EINGABE AUF JOYSTICK ****
290 REM **** AN PORT 1 UMSTELLEN ****
300 POKE 56323,PEEK (56323) AND 224
320 REM **** JOYSTICKPORT AUSLESEN ****
330 J1 = PEEK (56321)
350 REM ***** DATEN AUSWERTEN *****
360 IF (J1 AND 1) = 0 THEN Y2 = Y2-1
370 IF (J1 AND 2) = 0 THEN Y2 = Y2+1
380 IF (J1 AND 4) = 0 THEN X2 = X2-1
390 IF (J1 AND 8) = 0 THEN X2 = X2+1
400 IF (J1 AND 16) = 0 THEN F = 1
420 REM ** KOORDINATEN KONTROLLIEREN **
430 IF X2 < 0 THEN X2 = 0
440 IF Y2 < 0 THEN Y2 = 0
450 IF X2 > C-1 THEN X2 = C-1
460 IF Y2 > D-1 THEN Y2 = D-1
480 REM ***** PUNKT BEWEGEN *****
490 IF F = 0 THEN POKE A+X1+Y1*C,32
500 POKE A+X2+Y2*C,B1
510 POKE B+X2+Y2*C,1
530 REM ** NEUE WERTE & RUECKSPRUNG **
540 X1 = X2 : Y1 = Y2 : F = 0
550 GOTO 320
590 REM *****
600 REM *** AENDERUNGEN FUER VC-20 ***
610 REM *****
620 REM
630 REM AENDERN SIE DIESE ZEILEN:
640 REM
650 REM 250 POKE 36879,10 : D=23
660 REM 260 PRINT "{CLR}" : A=7680 : B=384
    0
670 REM 0 : C=22
680 REM 300 POKE37139,0 : POKE37154,127
690 REM 330 J1 = PEEK (37137)
700 REM 335 J2 = PEEK (37152)
710 REM 390 IF (J2AND128)=0 THEN X2=X2+1
720 REM
730 REM IN DEN ZEILEN 360-380 UND 400
740 REM DIE WERTE MIT DENEN J1 "AND"-
750 REM VERGlichen WIRD DER REIHENFOLGE
760 REM NACH DURCH 4,8,16,32 ERSETZEN.

```

© 64'er

Listing 4. Demo-Programm zur Joystickabfrage

Beim VC 20 steht die Information für Osten in 37152, die für die anderen drei Richtungen und Feuer in 37137. Jedem Kontakt im Joystick ist genau ein Bit in einer dieser Speicherstellen zugeordnet. Welches Bit zu welcher Richtung gehört, zeigt Tabelle 1. Beim C 64 weisen die angegebenen Speicherstellen der beiden Ports die gleiche Bitstruktur auf. Der einzige Unterschied besteht darin, daß in Adresse 56320 das Bit 7 immer gelöscht ist.

Ein Beispiel zur Joystickprogrammierung

Nach soviel Theorie wieder etwas Praxis: Geben Sie zur Übung der Joystickprogrammierung das Programm »Joystickdemo 1« (Listing 4) ein. Mit diesem Programm können Sie per Joystick einen weißen Punkt auf einem schwarzen Bildschirm in alle acht Richtungen bewegen. Ist der Feuerknopf gedrückt, wird der Punkt nicht mehr gelöscht, das heißt er zeichnet.

In Zeile 300 wird die Eingabe auf den Joystick gelegt. Zeile 330 kopiert (bei VC 20 auch Zeile 335) den Inhalt des Joystickregisters in die Variable J1 (und beim VC 20 das zweite Register in J2). Ist der Joystick in Ruhestellung, so ist das zugehörige Bit auf Eins. Bei der Bewegung in eine Richtung wird das entsprechende Bit im Joystickregister gelöscht. Bei den Zwischenrichtungen NO, SO, SW und NW werden jeweils die zwei zugehörigen Bits gelöscht. Der Feuerknopf setzt das »Fire«-Bit auf Null. Ab Zeile 350 wird J1 (und J2) ausgewertet. Wenn die bitweise Überprüfung mit AND Null ergibt, war der Joystick in dieser Richtung bewegt. Mit der Wertekontrolle ab Zeile 420 wird verhindert, daß der Punkt den Bildschirm verläßt und unkontrolliert im RAM herumschreibt. In Zeile 490 wird der alte Punkt eventuell (Abhängigkeit vom Feuerknopf) gelöscht und in 500/510 der neue gesetzt.

In diesem Programm wird der Joystick ausschließlich in Basic abgefragt. Für die hier dargestellte Anwendung ist

```

200 REM JOYSTICKDEMO 2
210 REM -----
220 REM BY TOBIAS NICOL
225 REM
230 REM ** MASCHINENROUTINE EINLESEN **
240 S = 0 : FOR I=842 TO 925
250 READ A : POKE I,A : S = S+A
260 NEXT I
270 IF S=8788 THEN 300
280 PRINT "{CLR,4DOWN}DATA ERROR!" : END
300 REM ***** INITIALISIERUNG *****
310 X1 = 0 : X2 = 0 : Y1 = 0 : Y2 = 0
320 POKE 53281,0 : POKE 53280,1
330 A = 1024 : B = 55296 : C = 40
340 D = 24 : PRINT "{CLR}"
360 REM **** JOYSTICK-WERT NACH J ****
370 SYS 842 : J = PEEK (1023)
380 IF J = 0 OR J = 100 THEN 360
400 REM *** JOYSTICK-WERT AUSWERTEN ***
410 IF J < 100 THEN J1 = J
420 IF J > 100 THEN J1 = J-100
430 ON J1 GOTO 450,360,460,490,470,480
440 ON J1-6 GOTO 520,500,360,360,510
450 Y2=Y2-1 : GOTO 540 : REM NORD
460 X2=X2+1 : GOTO 540 : REM OST
470 Y2=Y2+1 : GOTO 540 : REM SUED
480 X2=X2-1 : GOTO 540 : REM WEST
490 X2=X2+1:Y2=Y2-1: GOTO 540 : REM N/O
500 X2=X2+1:Y2=Y2+1: GOTO 540 : REM S/O
510 X2=X2-1:Y2=Y2+1: GOTO 540 : REM S/W
520 X2=X2-1:Y2=Y2-1: GOTO 540 : REM N/W
540 REM ***** WERTE KONTROLLIEREN *****
550 IF X2 < 0 THEN X2 = 0
560 IF Y2 < 0 THEN Y2 = 0
570 IF X2 > C-1 THEN X2 = C-1
580 IF Y2 > D THEN Y2 = D
600 REM ***** PUNKT BEWEGEN *****
610 IF J < 100 THEN POKE A+X1+Y1*C,32
620 POKE A+X2+Y2*C,160
630 POKE B+X2+Y2*C,1
650 REM ** NEUE WERTE & RUECKSPRUNG **
660 X1 = X2 : Y1 = Y2
670 GOTO 360
700 REM ***** DATAS *****
710 DATA 008, 169, 000, 141, 255, 003
720 DATA 169, 001, 045, 001, 220, 208
730 DATA 008, 169, 001, 109, 255, 003
740 DATA 141, 255, 003, 169, 002, 045
<124>
<228>
<190>
<031>
<132>
<058>
<144>
<090>
<202>
<172>
<017>
<000>
<069>
<006>
<208>
<232>
<244>
<090>
<218>
<217>
<204>
<085>
<146>
<218>
<047>
<075>
<077>
<066>
<209>
<123>
<002>
<001>
<153>
<172>
<179>
<038>
<050>
<013>
<132>
<241>
<084>
<211>
<202>
<234>
<020>
<167>
<084>
<203>
750 DATA 001, 220, 208, 008, 169, 005
760 DATA 109, 255, 003, 141, 255, 003
770 DATA 169, 008, 045, 001, 220, 208
780 DATA 008, 169, 003, 109, 255, 003
790 DATA 141, 255, 003, 169, 004, 045
800 DATA 001, 220, 208, 008, 169, 006
810 DATA 109, 255, 003, 141, 255, 003
820 DATA 169, 016, 045, 001, 220, 208
830 DATA 008, 169, 100, 109, 255, 003
840 DATA 141, 255, 003, 040, 096, 000
880 REM *****
890 REM *** AENDERUNGEN FUER VC-20 ***
900 REM *****
910 REM
920 REM GEBEN SIE DIESE ZEILEN EIN:
930 REM
940 REM 240 S=0 : FOR I=842 TO 943
950 REM 270 IF S=10932 THEN 300
960 REM 320 POKE 36879,9
970 REM 330 A=7680 : B=38400 : C=22
980 REM 340 D=22 : PRINT "{CLR}"
1000 REM BITTE GEBEN SIE STATT DEN ZEI-
1010 REM LEN 700-840 DIESE DATAS EIN:
1020 REM
1030 REM 700 DATA 008,169,127,141,034
1040 REM 710 DATA 145,169,000,141,019
1050 REM 720 DATA 145,141,255,003,169
1060 REM 730 DATA 004,045,017,145,208
1070 REM 740 DATA 008,169,001,109,255
1080 REM 750 DATA 003,141,255,003,169
1090 REM 760 DATA 008,045,017,145,208
1100 REM 770 DATA 008,169,005,109,255
1110 REM 780 DATA 003,141,255,003,169
1120 REM 790 DATA 128,045,032,145,208
1130 REM 800 DATA 008,169,003,109,255
1140 REM 810 DATA 003,141,255,003,169
1150 REM 820 DATA 016,045,017,145,208
1160 REM 830 DATA 008,169,006,109,255
1170 REM 840 DATA 003,141,255,003,169
1180 REM 850 DATA 032,045,017,145,208
1190 REM 860 DATA 008,169,100,109,255
1200 REM 870 DATA 003,141,255,003,169
1210 REM 880 DATA 255,141,034,145,169
1220 REM 890 DATA 128,141,019,145,040
1230 REM 900 DATA 096,000,000,000,000
<238>
<070>
<233>
<200>
<063>
<252>
<122>
<024>
<162>
<159>
<000>
<209>
<020>
<210>
<055>
<230>
<116>
<231>
<217>
<043>
<133>
<178>
<204>
<064>
<174>
<120>
<245>
<159>
<208>
<255>
<045>
<094>
<077>
<079>
<244>
<003>
<025>
<114>
<081>
<055>
<100>
<159>
<037>
<239>
<067>

```

© 64'er

Listing 5. Joystickabfrage als Maschinenroutine

dieses System auch völlig ausreichend. Wenn es aber auf hohe Geschwindigkeit ankommt oder jede der acht Richtungen eine andere Funktion auslösen soll, der Joystick also auf jede Richtung einzeln überprüft werden muß, reicht eine reine Basic-Abfrage oft nicht mehr aus. Ich habe dieses Problem in dem Programm »Joystickdemo 2« (Listing 5) behoben. Nach dem Programmstart wird die in den DATA-Zeilen enthaltene Maschinenroutine bei beiden Computern im Kassettenpuffer abgelegt. Für den VC 20 aktivieren Sie bitte die REM-Zeilen. Da beim C 64 der Port 1 verwendet wird, sind für ihn keine Einstellungen notwendig. Die oben genannten Einstellungen für den VC 20 nimmt dessen Maschinenroutine selbsttätig vor. Daher ist sie auch etwas länger. Außerdem sind natürlich die Abfrageregister und Bitwer-

te angepaßt. Die Routine, die übrigens nur relative Sprünge enthält, also auch in anderen Speicherbereichen abgelegt werden kann, liest bei ihrem Aufruf die oben beschriebenen Joystickregister der Computer aus und legt den ermittelten Wert im letzten Byte des Kassettenpuffers (Adresse 1023 dezimal oder \$03FF hexadezimal) ab. Den Aufbau dieses Wertes zeigt Bild 2. Ist der Feuerknopf gedrückt, wird zu dem errechneten Wert die Zahl 100 addiert. Diese Methode hat den Vorteil, daß nach der Berücksichtigung des Feuerknopf-Wertes, also der addierten 100, (Zeilen 410/420) die Joystick-Information direkt mit ON..GOTO/GOSUB verarbeitet werden kann (Programmzeilen 430 bis 520). Da die Joystick-Werte »2«, »9« und »10« nicht vorkommen, müssen sie bei der ON..GOTO/GOSUB-Anweisung ausgeklammert

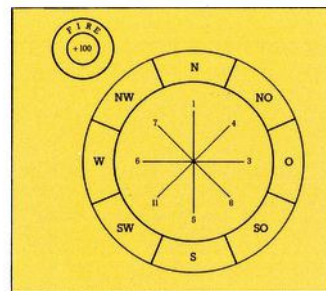


Bild 2. Diese Werte liefern die Maschinenroutinen zur Joystickabfrage (Listing 5)

werden (Sprünge nach Zeile 360 in Zeilen 430/440). Das ON..GOTO ist, nebenbei bemerkt, nur deshalb auf die Zeilen 430/440 aufgeteilt, weil eine Zeile zu voll würde. Bis auf die besprochene Maschinenroutine funktioniert das eben beschriebene Programm genauso wie die »Joystickdemo 1«.

Die Geschwindigkeitsfanatiker unter Ihnen sollten jetzt noch einmal besonders gut aufpassen:

1. Obwohl die zweite Joystickdemo eine Maschinenroutine enthält, ist sie in der Ausführung etwas langsamer als die erste. Das hat einen sehr einfachen Grund: Für die hier gezeigte Anwendung ist die reine Basic-Abfrage zeitlich besser, weil bei den Zwischenrichtungen zwei Bedingungen erfüllt sind und nicht jede Richtung einzeln abgefragt werden muß.

2. Der Cursorpunkt ist schneller, wenn er die Punkte nicht löschen muß, weil dann eine Anweisung entfällt.

Das war's für heute. In der nächsten Folge werden wir die Besprechung des Control-Ports zu Ende bringen und uns anschließend dem User-Port zuwenden. Bis dahin haben Sie Gelegenheit, ausgiebig mit dem Gelernten zu experimentieren.

(Tobias Nicol/ev)