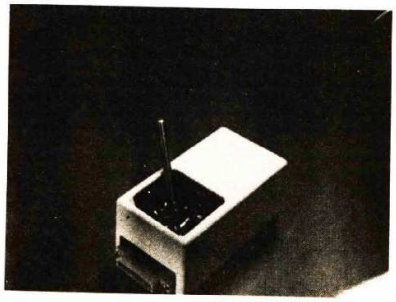
Schrottknüppel

Schrottknüppel
Ein "Joystick" ist ein Steuerknüppel, der es gestattet, die Koordinaten einer Ebene in einen Rechner einzugeben. Dazu muß die analoge Stellung des Knüppels auf der X- und der Y-Achse in eine digitale Darstellung umgesetzt werden. Eine Möglichkeit zur Realisierung wäre die, für beide Achsen je ein Potentiometer und je einen Analog/Digital-Umsetzer zu verwenden.

Die Potentiometer müssen dabei über eine entsprechende Mechanik so verbunden sein, daß sie sich mit einem auf beide Achsen wirkenden Knüppel gleichzeitig einstellen lassen.
Aus der Fernsteuertechnik sind derartig aufwendige und entsprechend teure (ca. DM 50.--) Konstruktionen bekannt (meist wird für jeden der vier Quadranten je ein Potentiometer verwendet).

Noch kostspieliger wird die Geschichte durch die leider nicht zu umgehenden A/D-Umsetzer. Computerfreunde, die bereits über Umsetzer verfügen und noch zwei Kanäle frei haben, sollten trotzdem weiterlesen, bevor sie sich ein Quadropot anschaffen und ihre letzten beiden Kanäle verbraten. Es geht nämlich auch billiger, und zwar mit Bauteilen, die sich in (fast) jeder Bastelkiste finden.

Hardware
Zwei Potentiometer (500K lin.) werden in ihrer Achse so miteinander verbunden, daß ihre Achsen einen rechten Winkel bilden (Bild 1). Beim Abflachen der Achsen ist darauf zu achten, daß die Potentiometer nur mit ihrer Achse in den Schraubstock eingespannt werden. Ein Pot wird auf einen Winkel montiert, während am zweiten der Knüppel (z. B. Messingrohr 6 mm) befestigt wird. Beide sind mechanisch so zu justieren, daß sich der Anschlag in einer Ecke befindet (z. B. in der entfernteren, linken Ecke) und von dort aus ein Knüppelausschlag von 90 Grad in X- und Y-Richtung möglich ist.



Die Digitalisierung geschieht durch zwei Monoflops, deren Ausgangsimpulsdauer proportional dem Widerstandswert der X- bzw. Y-Potentiometer ist (Bild 2). Beide Impulszeiten zusammen sind somit ein Abbild der Stellung des Knüppels in der Ebene.

Software
Die Auswertung beider Zeiten und ihre Umsetzung in den Dualcode leistet ein kleines Unterprogramm (s. Listing), das entweder direkt oder zyklisch, etwa durch einen Timer-Interrupt (IRQ) angesprungen wird.

Im letzteren Fall ist vorher der IRQ-Vektor zu setzen und in Zeile 570 oder 'RTS' durch 'RTI' zu ersetzen.
Die Messchleife des Programms (Zeile 250... 400) dauert 49 us und wird nach dem Triggern der Monoflops (Zeile 230) so lange durchlaufen, bis beide Monoflops wieder in ihren stabilen Zustand zurückgekippt sind. Bei jedem Durchlauf der Schleife werden zwei den Potentiometern zugeordnete Speicherplätze inkrementiert, sofern das zugehörige Monoflop noch gefeuert ist. Andernfalls wird eine entsprechende Zeit verschwendet, um auf eine konstante Durchlaufzeit von 39 us zu kommen.

Um beim Hochzählen der Speicherplätze einen Bereich von 00 ... 7F zu erreichen, müssen die

Impulszeiten der Monoflops also 128 * 39 us = 5 ms betragen. Da für den Baustein 74LS123 ein minimaler Widerstand von ca. 5K vorgeschrieben ist, muß die daraus resultierende AN-Zeit der Monoflops bei Nullstellung der Potentiometer berücksichtigt und kompensiert werden. Dies wird im Programm durch Vorbelegen der Speicherplätze mit einem negativen Wert erreicht (Zeile 170 ... 190). Um eine saubere Nullstellung zu erzielen, wird dieser Wert etwas negativer als erforderlich gewählt und in Zeile 420 ff. (wie auch der Wert für Vollausschlag) korrigiert.

Erfahrungen

Gehobenen Linearitätsansprüchen genügt dieses Verfahren natürlich nicht. Es hat sich jedoch gezeigt, daß die Anordnung aufgrund der optischen Rückkopplung über den Bildschirm fehlerfrei funktioniert.

Die Anwendungen für diesen 1-Chip-Joystick sind vielfältig und nicht nur auf Bildschirmspiele beschränkt. Beispiele für nicht triviale Anwendungen zu finden, sei dem Leser überlassen. Ist das Ding erst einmal installiert, so wird sich bald die Notwendigkeit für einen weiteren "Freuden-Knüppel" ergeben. Und ob der Schrott nun in der Kiste vergammelt oder nicht ....

Anmerkung für KIM-Benutzer:

Das Programm ist zwar verschiebbar, da es nur relative Sprungadressen aufweist. Die absolut zu adressierenden Speicherplätze für "HOR" und "VERT" sind jedoch in einen verfügbaren Speicherbereich zu lesen.

```
MCSASM
LINE #  LOC   CODE       LINE

 0010  0200
 0020  0200              ;   ********** JOY-STICK fuer 6502 **********
 0030  0200              ;   * digitalisiert zwei Potentiometerwerte *
 0040  0200              ;   *          R.SCHOENE, 18.07.79           *
 0050  0200              ;   ****************************************
 0060  0200
 0070  0200              PA      =$F600          ;6532-PORT
 0080  0200              PAD     =PA+1           ;DIRECTION REGISTER
 0090  0200              HOR     =$F7E0          ;WERT FUER X-ACHSE
 0100  0200              VERT    =HOR+1          ;WERT FUER Y-ACHSE
 0110  0200
 0120  0200                      *=$F700
 0130  F700
 0140  F700  48          JOY     PHA             ;ACCU UND
 0150  F701  8A                  TXA             ;X-REG.
 0160  F702  48                  PHA             ;RETTEN.
 0170  F703  A9 F8               LDA #$F8        ;ZERO OFFSET
 0180  F705  8D E0 F7            STA HOR         ;FUER BEIDE
 0190  F708  8D E1 F7            STA VERT        ;MONOFLOPS.
 0200  F70B  A9 01               LDA #1          ;BIT 0
 0210  F70D  8D 00 F6            STA PA          ;='HI'
 0220  F710  8D 01 F6            STA PAD         ;UND AUSGANG.
 0230  F713  CE 00 F6            DEC PA          ;TRIGGER ='LO'.
 0240  F716
 0250  F716  EA          LOOP    NOP             ;VERZOEG.       2us
 0260  F717  2C 00 F6    LOOP1   BIT PA          ;HOR.MF AN?     4    4us
 0270  F71A  50 05               BVC KEINX       ;NEIN.          3    2
 0280  F71C  EE E0 F7            INC HOR         ;JA, +1              6
 0290  F71F  70 06               BVS NVX         ;KEINE VERZ.         3
 0300  F721  AD E0 F7    KEINX   LDA HOR         ;VERZOEG.       4
 0310  F724  8D E0 F7            STA HOR         ;               4
 0320  F727  2C 00 F6    NVX     BIT PA          ;VERT.MF AN?    4    4
 0330  F72A  10 05               BPL KEINY       ;NEIN           3    2
 0340  F72C  EE E1 F7            INC VERT        ;JA, +1              6
 0350  F72F  30 06               BMI NVY         ;KEINE VERZ.         3
 0360  F731  AD E1 F7    KEINY   LDA VERT        ;VERZOEG.       4
 0370  F734  8D E1 F7            STA VERT        ;               4
```

```
0380  F737  2C 00 F6    NVY     BIT PA          ;BEIDE AUS?   4   4
0390  F73A  30 DA              BMI LOOP        ;NEIN, VERZ.  3   2
0400  F73C  70 D9              BVS LOOP1       ;NEIN.            3
0410  F73E
0420  F73E  A2 01              LDX £1          ;HOR UND VERT
0430  F740  BD E0 F7    KORR    LDA HOR,X       ;KORRIGIEREN.
0440  F743  C9 F8              CMP £$F8        ;MINIMUM?
0450  F745  90 02              BCC OK1         ;NEIN.
0460  F747  A9 00              LDA £0          ;JA.
0470  F749  C9 80        OK1   CMP £128        ;MAXIMUM?
0480  F74B  90 02              BCC OK2         ;NEIN.
0490  F74D  A9 7F              LDA £127        ;JA.
0500  F74F  9D E0 F7    OK2     STA HOR,X       ;POTENTIOMETER-
0510  F752  CA                  DEX             ;STELLUNG ALS
0520  F753  10 EB               BPL KORR        ;DUALCODE 00...7F.
0530  F755
0540  F755  68                  PLA             ;X-REG.
0550  F756  AA                  TAX             ;UND
0560  F757  68                  PLA             ;ACCU HOLEN.
0570  F758  60                  RTS             ;UNTERPROGRAMM-ENDE.
0580  F759
0590  F759              STICK   .END

ERRORS = 0000
```

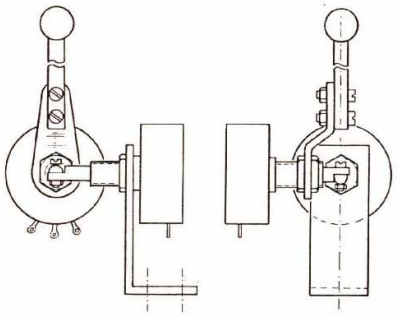


Bild 1.

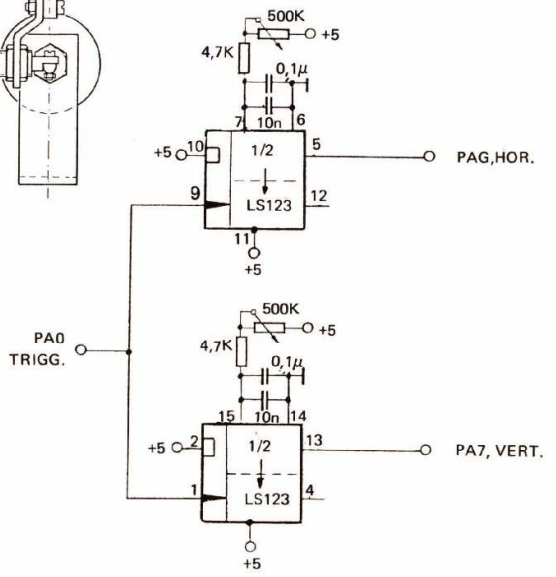


Bild 2:

R. Schöne ■

Programmbeschreibung "FLAG-OPERATION"
(TRS-80, Level II, 16 K)

Das Programm zeigt, wie man FLAGS (Marken, Fähnchen oder Bits) setzen kann, um später z. B. bei längeren Programmen bestimmte Unterprogramme auszulösen. Dieses Programm hier ist ziemlich lang, und man soll daran auch nur sehen, wie es geht.

In den "eigentlichen" Programmen läßt man alle Befehle weg, die dem Betrachter am Bildschirm die Funktionsweise zeigen.

Die Programmzeilen bedeuten:

Zeile 200: Was soll getan werden? FLAG setzen (1), FLAG löschen (2) oder FLAG testen (3). Die Eingabe veranlaßt den Sprung zu den entsprechenden Unterprogrammen.

Zeile 210: SF% steht hier für die Bitnummer im FLAG-Wort F%. Mit F% stehen 16 einzelne Bits von Nr. "0" bis Nr. "15" zur Verfügung.

Zeile 250: Eingabe des zu löschenden FLAGs (Bit).

Zeile 290: Eingabe des zu testenden FLAGs

Zeilen 5000 bis 5042: Umwandlung der Dezimalzahl Z% in die Dualzahl Z$ (eigentlich keine "Zahl", sondern eine Zeichenkette — ein String).

Zeile 5050: Umwandlung der Dualzahl Z$ in die Dezimalzahl Z%.

Zeile 5200: Kontrollzeile, die hier nach der Entscheidung in Zeile 200 entfallen kann.

Zeile 5220: Die OR- (ODER)Funktion verursacht, daß das SF%-te Bit in F% zu einer "1" wird. Hier wird also das FLAG Nr. SF% (ursprüngliche Eingabe) "gesetzt".

Zeile 5240: Die AND-Funktion im Zusammenhang mit der NOT-Funktion von K%, wobei das DF%-te Bit in K% zu "0" alle anderen zu "1" werden, erzwingt eine "0" an der DF%-ten Stelle von F%.

Zeile 5250: K% enthält nur an der TF%-ten Stelle eine "1", dadurch kann das Ergebnis K% nur den Wert des entsprechenden Bits in F% enthalten.

Zeile 5300 bildet die 2er-Potenz von IE%, da in BASIC das Setzen bestimmter Bits nicht möglich ist. (Daher dieses Programm überhaupt!)

```
100 CLS:A$="DIE FLAGS SIND NUN SO GESETZT: ":B$="FLAG ":C$="GESETZT":D$="NICHT ":PRIN
T"FLAG-OPERATION, EIN DEMO-PROGRAMM"
200 PRINT:PRINT"1=SET FLAG
2=DELETE FLAG
3=TEST FLAG";:INPUT"   ";I%:ONI%GOTO210,250,290:REM * ALLE OPERATIONEN BEZIEHEN SICH
AUF DIE 2-BYTE-ZAHL F%, DEREN BITS VON RECHTS ="0" NACH LINKS="14" ALS  **  FLAGS  **
 BENUTZT WERDEN.
201 'IN EINEM "RICHTIGEN" PRG. SIND NUR DIE  5 ZEILEN 5220-5310 NOTWENDIG!
210 INPUT"SF= ";SF%:SF%=SF%+1:DF%=0:GOSUB5200:Z%=F%:GOSUB5000:PRINTA$Z$:GOTO200:REM *
 SF = >SETZE >FLAG(=BIT-)NR. SF%
250 INPUT"DF= ";DF%:SF%=0:DF%=DF%+1:GOSUB5200:Z%=F%:GOSUB5000:PRINTA$Z$:GOTO200:REM *
 DF= LOESCHE (=>DELETE) >FLAG-nR.: DF%
290 INPUT"TF= ";TF%:GOSUB5250:IFK%=0THENPRINTB$;TF%;D$C$ELSEPRINTB$;TF%;C$:REM * TF=
>TEST DES >FLAGS NR. TF%
300 GOTO200
5000 Z$="":ZX%=Z%:REM * UMWANDLUNG DER DEZIMALZAHL Z% IN DIE
DUALZAHL Z$
5020 ZY%=ZX%:ZX%=ZX%/2:ZY%=ZY%-2*ZX%:Z$=RIGHT$(STR$(ZY%),1)+Z$
5040 IFZX%=0THEN RETURN ELSE IF ZX%=-1THEN5042ELSE5020
5042 IFLEN(Z$)>15THEN RETURN ELSE5020
5050 Z%=0:FORI%=1TOLEN(Z$):Z%=Z%*2:Z%=Z%+VAL(MID$(Z$,I%,1)):NEXT:RETURN:REM * UMWANDL
UNG DER DUALZAHL Z$ IN DIE DEZIMALZAHL Z%
= UMKEHRUNG DER ROUTINE "GOSUB 5000"! "GOSUB 5050" WIRD HIER
NICHT GEBRAUCHT
5200 IFSF%>0ANDDF%=0THEN5220ELSE IF DF%>0ANDSF%=0THEN5240ELSEPRINT"SF UND DF >0 = FAL
SCH!":STOP
5220 IE%=SF%-1:GOSUB5300:F%=K%ORF%:RETURN:REM * BITFOLGE VON K%
WIRD IN F% EINGEFUEGT
5240 IE%=DF%-1:GOSUB5300:F%=F%AND(NOTK%):RETURN:REM * ERZWINGT 0,WO K% EINE LOGISCHE
1 STEHEN HAT
5250 IE%=TF%:GOSUB5300:K%=K%ANDF%:RETURN:REM * NUR WENN DAS K%-
BIT *UND* DAS ENTSPR. F%-BIT =1 SIND, IST DAS ERGEBNIS 1 SONST 0
5300 K%=1:IFIE%=0THEN5310 ELSE FORI%=1TOIE%:K%=K%*2:NEXT:REM *
==> K%^IE%
5310 RETURN
```

Dr. E. Brewig, Overath