

Atari ST - MIDI-Programmierung



Omikron Basic

Das Programm wurde kostenlos mit jedem Atari ST ausgeliefert.

Programm Laden (Doppelklick auf OMIKR.PRG), dann HELP drücken → „Bildschirmeditor“

Besonderheiten gegenüber sonstigem Texteschreiben:

Textzeile schreiben + **RETURN** speichert den Zeileninhalt, führt als nicht zu einem neuen Absatz...

Shift + F9 löscht eine Zeile!!

Control + RETURN fügt eine neue Zeile zwischen bereits bestehende Zeilen.

Menue Block:

Markieren von „Blöcken“ mit der Maus ist möglich, dann den Cursor an eine Stelle bringen, auf die der Block verschoben oder kopiert werden soll... (es gibt also keine Zwischenspeicher copy-paste).

Menue Run:

selbstredend zum Starten des im Editor eingegebenen Programms.

HELP führt zum Bildschirmeditor zurück.

Der Befehl „system“ (im Editor nach „ok“ geschrieben + RETURN) beendet Omikron.

Vom Schreiben zum Programm: Maus als Zeichenstift

REPEAT

MOUSEON

IF MOUSEBUT = 1 THEN **MOUSEOFF**: CIRCLE MOUSEX, MOUSEY, 5

UNTIL LEN(INKEY\$)

oder

IF MOUSEBUT = 1 THEN

MOUSEOFF

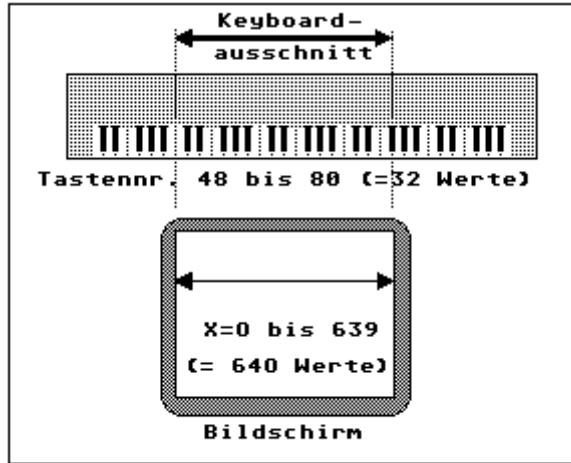
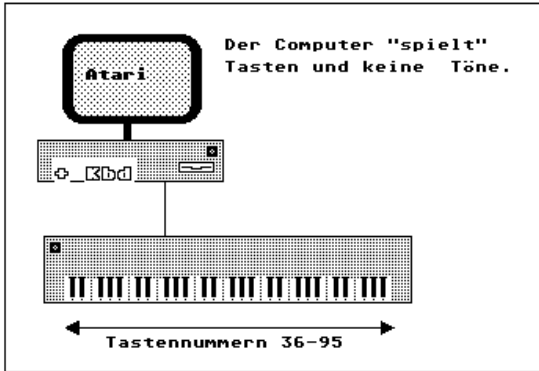
CIRCLE MOUSEX, MOUSEY, 5

ENDIF

Midi-Programmierung

```
BIOS(,3,3,144):BIOS(,3,3,60):BIOS(,3,3,90)  
WAIT 2.5  
BIOS(,3,3,144):BIOS(,3,3,60):BIOS(,3,3,0)
```

144 → 143 + Midikanal (hier = 1). Es gibt 16 Midikanäle. Wir verwenden nur Kanal 1.
60 → eingestrichenes C = 60. Taste des virtuellen Keyboards mit 127 Tasten
90 bzw. 0 = „Velocity“, Lautstärke (von 0 bis 126).



Zeichnen unterschiedlicher Tonhöhen

```
REPEAT
  MOUSEON
  IF MOUSEBUT = 1
    THEN MOUSEOFF : CIRCLE MOUSEX, MOUSEY, 5
    Taste = 48 + MOUSEX/20
    BIOS(,3,3,144) : BIOS(,3,3,Taste) : BIOS(,3,3,90)
    WAIT .1
    BIOS(,3,3,144) : BIOS(,3,3,Taste) : BIOS(,3,3,0)
  ENDIF
UNTIL LEN( INKEY$ )
```

$0 \leq \text{MOUSEX} \leq 639 \quad \rightarrow \quad 0 \leq \text{MOUSEX}/20 \leq 31 \quad \rightarrow \quad 48 \leq \text{Taste} \leq 79$

Tonhöhen und Lautstärke im X-Y-Diagramm

```
REPEAT
  MOUSEON
  IF MOUSEBUT = 1
    THEN MOUSEOFF : CIRCLE MOUSEX, MOUSEY, 5
    Taste = 48 + MOUSEX/20: Lautst = MOUSEY/4
    BIOS(,3,3,144) : BIOS(,3,3,Taste) : BIOS(,3,3,Lautst)
    WAIT .1
    BIOS(,3,3,144) : BIOS(,3,3,Taste) : BIOS(,3,3,0)
  ENDIF
UNTIL LEN( INKEY$ )
```

0 <= MOUSEY <= 399 → 0 <= Lautst <= 99

Zwei Tonhöhen im X-Y-Diagramm

```
REPEAT
  MOUSEON
  IF MOUSEBUT=1
    THEN MOUSEOFF : CIRCLE MOUSEX, MOUSEY, 5
      Taste1 = 48+ MOUSEX/20
      Taste2 = 96 -MOUSEY/10
      BIOS(,3,3,144):BIOS(,3,3,Taste1):BIOS(,3,3,90)
      BIOS(,3,3,144):BIOS(,3,3,Taste2):BIOS(,3,3,90)
      Wait .1
      BIOS(,3,3,144):BIOS(,3,3,Taste1):BIOS(,3,3,0)
      BIOS(,3,3,144):BIOS(,3,3,Taste2):BIOS(,3,3,0)
  ENDIF
UNTIL LEN (IKEY$)

0 <= MOUSEY <= 399   →   96 => Taste2 => 58
```

Akkord und Klang (Sound) im X-Y-Diagramm (PRG2.BAS)

```
REPEAT
  MOUSEON
  IF MOUSEBUT=1
  THEN MOUSEOFF : CIRCLE MOUSEX, MOUSEY, 5
    Taste1 = 36 + MOUSEX/12
    Taste2 = 40 + MOUSEX/12
    Taste3 = 43 + MOUSEX/12
    Sound =MOUSEY/4
    BIOS(,3,3,192):BIOS(,3,3,Sound)
    BIOS(,3,3,144):BIOS(,3,3,Taste1):BIOS(,3,3,90)
    BIOS(,3,3,144):BIOS(,3,3,Taste2):BIOS(,3,3,90)
    BIOS(,3,3,144):BIOS(,3,3,Taste3):BIOS(,3,3,90)
    Wait .1
    BIOS(,3,3,144):BIOS(,3,3,Taste1):BIOS(,3,3,0)
    BIOS(,3,3,144):BIOS(,3,3,Taste2):BIOS(,3,3,0)
    BIOS(,3,3,144):BIOS(,3,3,Taste3):BIOS(,3,3,0)
  ENDIF
UNTIL LEN(INKEY$)
```



Feste Relation der
Tonhöhen!

**Problem der Tonwiederholungen bei gedrückter Maus und auch der umständlichen Schreibweise.
Lösung durch Prozeduren und einen Trick (PRG3.BAS):**

```

REPEAT
MOUSEON
  IF MOUSEBUT=1
    THEN MOUSEOFF : CIRCLE MOUSEX, MOUSEY, 5
    Sound =MOUSEY/4
    Taste = 36 + MOUSEX/12:Play
  ENDIF
UNTIL LEN (INKEY$)
BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)  [→ Abschalten des Tons am Ende!]
END

DEF PROC Play
  BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)  [→ der „alte Ton“ wird abgeschaltet,
  BIOS(,3,3,192):BIOS(,3,3,Sound)                 bevor
  BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)    der „neue Ton“ angeschaltet wird. ]
  Taste_alt=Taste
RETURN

```

Aufnahme und Wiedergabe eines Zeichenvorganges (PRG3A)

```
DIM X(1000):DIM y(1000):DIM T(1000)
```

Aufnahme

```
Startzeit= TIMER
```

```
REPEAT
```

```
  MOUSEON
```

```
    IF MOUSEBUT=1
```

```
      X(N)=MOUSEX:Y(N)=MOUSEY:T(N)=TIMER-Startzeit
```

```
      THEN MOUSEOFF : CIRCLE X(N), Y(N), 5
```

```
      Sound =Y(N)/4
```

```
      Taste = 36 + X(N)/12:Play
```

```
    ENDIF
```

```
  UNTIL LEN (INKEY$)
```

```
  BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
```

```
END
```

```
T(N+1)=TIMER-Startzeit
```

```
Anzahl=N
```

```
CLS
```

```
REPEAT:UNTIL LEN(INKEY$)
```

```
  (=Warteschleife „weiter mit SPACE“...)
```

Wiedergabe

```
-Anfang
```

```
N=0:Startzeit=TIMER
```

```
REPEAT
```

```
  IF TIMER-Startzeit>=T(N+1)
```

```
    THEN MOUSEOFF : CIRCLE X(N), Y(N), 5
```

```
    Sound =Y(N)/4
```

```
    Taste = 36 + X(N)/12:Play
```

```
  ENDIF
```

```
  UNTIL N=Anzahl +1
```

```
  BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
```

Wiederholung?

```
REPEAT
```

```
  IF MOUSEBUT=1 THEN GOTO Anfang
```

```
  UNTIL LEN (INKEY$)
```

```
DEF PROC Play
```

```
  BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
```

```
  BIOS(,3,3,192):BIOS(,3,3,Sound)
```

```
  BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)
```

```
  Taste_alt=Taste
```

```
RETURN
```

Zufalls-Grafiken

```
X = 320 : Y = 200
```

```
REPEAT
```

```
  X = X + RND(5) - 2
```

```
  Y = Y + RND(5) - 2
```

```
  DRAW X, Y
```

```
UNTIL LEN ( INKEY$ )
```

[→ „Rekursive Formel“: die neuen Werte werden aufgrund der alten Werte berechnet.]

$\text{RND}(5) \rightarrow \{0, 1, 2, 3, 4\}$ $\text{RND}(5)-2 \rightarrow \{-2, -1, 0, 1, 2\}$

Variationen:

```
X = X + RND(7) - 3 : Y = Y + RND(3) - 1
```

```
X = X + RND(9) - 5 : Y = Y + RND(5) - 3
```

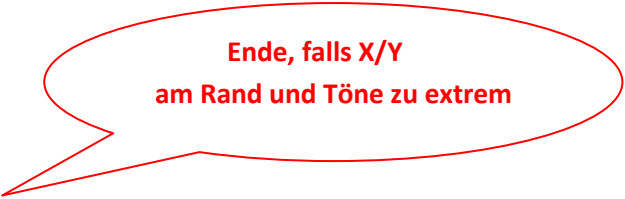
Zufallstonfolgen einstimmig (PRG4.BAS)

```

X = 320 : Y = 200
REPEAT
  X= X + RND(5) - 2 : Y= Y + RND(5) - 2
  IF X>0 AND X<640 AND Y>0 AND Y<400
  THEN  DRAW X, Y
        Taste = 48+X/20
        IF Taste_alt <>Taste THEN Play

  ELSE
    BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
  ENDIF
UNTIL LEN ( INKEY$ )
BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
END
DEF PROC Play
  BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
  BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)
  Taste_alt=Taste
RETURN

```



Ende, falls X/Y
am Rand und Töne zu extrem

Zufallsfolge in „Partiturform“ (PRG5.BAS)

REPEAT

X = X + 6 : Y = RND(36)

PBOX X, Y*10+5, 6,6

Taste = 36 + Y

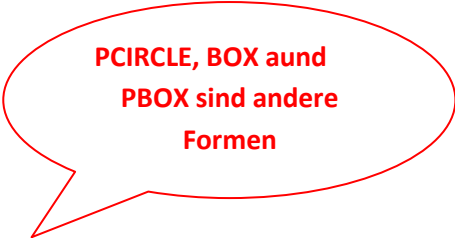
BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)

WAIT .1

BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,0)

ENDIF

UNTIL LEN (INKEY\$)



PCIRCLE, BOX und
PBOX sind andere
Formen

Der Bildschirm wird bei den meisten Atari's mehrfach überschrieben. Wird $X > 639$, dann wird „vorne angefangen“, d.h. $X = X - 640$ gerechnet und gezeichnet.

Zufallsfolge und fraktale Grafiken

REPEAT

Y_alt = Y*10+5

X = X + 6 : Y = RND(36)

DRAW X-6, Y_alt TO X, Y*10+5

Y_alt = Y

Taste = 36 + Y

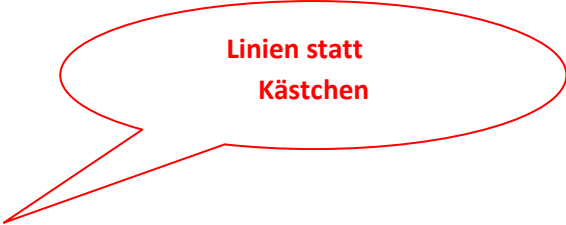
BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)

WAIT .1

BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,0)

ENDIF

UNTIL LEN (INKEY\$)



Linien statt
Kästchen

Variationen: DRAW 320, 200 to X, Y*10+5

(Echte) Fraktale Grafiken vertonen

Die „Hüpfer-Grafik“

DEFSNG "A-Z"

A = 3000 : B = 0.31 : C = -999

Xo = 300 : Yo = 175

REPEAT

$X = Y_{\text{neu}} - \text{SGN}(X_{\text{neu}}) * (B * X_{\text{neu}} - C) \text{ MOD } 64000$

$Y = A - X_{\text{neu}} \text{ MOD } 40000$

Xneu= X : Yneu = Y

DRAW Xo + X/100, Yo + Y/110

UNTIL LEN(INKEY\$)



Rekursionsformel

Anfangswerte:

A = 5000, B = 0.1, C = 5025

A = 5000, B = 0.2, C = 5025

A = 10000, B = - 0.01, C = - 10000

A = 123, B = 0.456, C = 7890

A = 6000, B = 0.02, C = 6000

A = 7010, B = - 0.3, C = 4000

Bildschirm zwischendurch löschen**ON HELP GOSUB Loeschen****DEFSNG "A-Z"****A = 3000 : B = 0.31 : C = -999****Xo = 300 : Yo = 175****REPEAT****X = Yneu - SGN(Xneu)*(B*Xneu - C) MOD 64000****Y = A - Xneu MOD 40000****Xneu= X : Yneu = Y****DRAW Xo + X/100, Yo + Y/110****UNTIL LEN(INKEY\$)****-Loeschen****CLS****RETURN**

Das Hüpfprogramm mit Musik (PRG6.BAS)

```
ON HELP GOSUB Loeschen
DEFSNG "A-Z"
A = 3000 : B = 0.31 : C = -999
Xo = 300 : Yo = 175
REPEAT
    X = Yneu - SGN(Xneu)*(B*Xneu - C) MOD 64000
    Y = A - Xx MOD 40000
    Xneu= X : Yneu = Y
    DRAW Xo + X/100, Yo + Y/110
    IF N MOD 30 = 0 AND X>0 AND X<12000:THEN Taste = 36 + X/200: Play
    N=N+1
UNTIL LEN( INKEY$ )
DEF PROC Play
    BIOS(,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
    BIOS(,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)
    Taste_alt=Taste
RETURN
-Loeschen:CLS:RETURN
```



nur jeder 30. Bildpunkt
soll vertont werden...

„Fraktales Zeichnen“ vortont (PRG7.BAS)

```
REPEAT
MOUSEON
  IF MOUSEBUT = 1
  THEN  MOUSEOFF
        REPEAT
          A=RND(20) - 10 : B = RND(20) - 10 : R = RND(20)
          CIRCLE MOUSEX + 3*A, MOUSEY + 3*B, R+5
          Taste = 36 + 3*A/10
          Play
        UNTIL MOUSEBUT = 0
      ENDIF
UNTIL LEN( INKEY$ )
DEF PROC Play
  BIOS (,3,3,144):BIOS(,3,3,Taste_alt):BIOS(,3,3,0)
  BIOS (,3,3,144):BIOS(,3,3,Taste):BIOS(,3,3,90)
  Taste_alt=Taste
RETURN
```

Atari ST - MIDI-Programmierung

für heute ist erst mal

ENDE