AVR-ChipBasic2: Beispielprogramme

V1.45 (c) 2006-2012 Jörg Wolfram

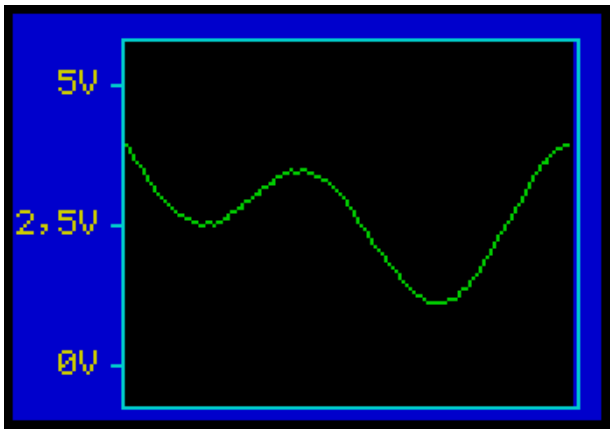
In diesem Abschnitt werden einige Beispiel-Programme vorgestellt. Diese stellen mehr oder weniger abgeschlossene Einheiten dar und demonstrieren die Leistungsfähigkeit des Systems. Sie sollen aber auch vor allem Anreiz für eigene Programme sein.

1 Darstellung des Zeichensatzes


```
PROGRAM 4:Charset
01 DATA 0,"0123456789ABCDEF"
02 CLS :COLOR 6:BOX 3,13,36,46
03 FOR Y=0 TO 15:? @Y+2,5;%AR(Y)
04 FOR X=0 TO 15:? @0,X+7;%AR(X)
05 COLOR 7,1
06 IF (X%2)=(Y%2) THEN COLOR 5,0
07 C=16*Y+X
08 ? @Y+2,X+7;%C;:COLOR 6,0
09 NEXT :NEXT
#
```

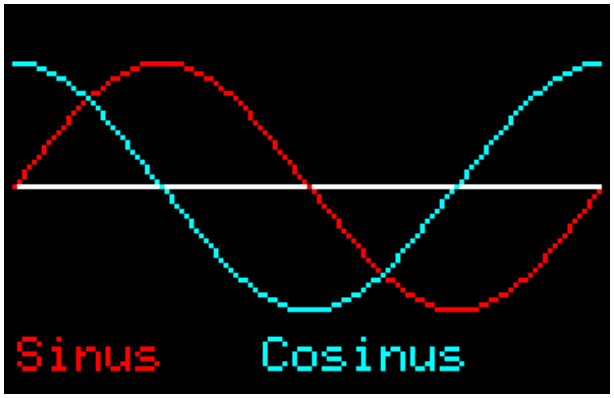
2 Ein kleines Oszilloskop

Ausgewertet wird der Analogeingan 0 (PIN2 der Printer-Buchse), die Abtastung erfolgt 50 mal pro Sekunde. Da die Referenzspanng defaultmäßig 2,56V beträgt, wird der Parameterbereich 0x100...0x1FF (direkter Zugriff auf ADMUX) benutzt um AVCC als Referenzspannung einzustellen. Zum Abschalten der Demonstration muss Zeile 16 auskommentiert werden.




```
PROGRAM:MODE1-Oszi
01 'init screen
02 COLOR 5,1 :VMODE 1
03 BOX 7,31,112,161
04 DRAW 20,28,20,31
05 DRAW 60,28,60,31
06 DRAW 100,28,100,31
07 COL 6
08 COL 6:? @ 15,0;"  5V";
09 COL 6:? @ 55,0;"2,5V";
10 COL 6:? @ 95,0;"  0V";
11 COLOR 4,0:CBOX 1,4,13,19
12 Y=60
13 'the main loop
14 A=100-ADC(64)/12
15 J=J+1: IF J=3600 THEN J=0
16 A=52-SIN(J*2)/16+COS(J*5)/16
17 BCOPY 1,8,33,13,16,8,32
18 DRAW Y,158,A,159:Y=A
19 SYNC 1:GOTO 14
#
```

3 Sinus und Cosinus im Grafikmode 2

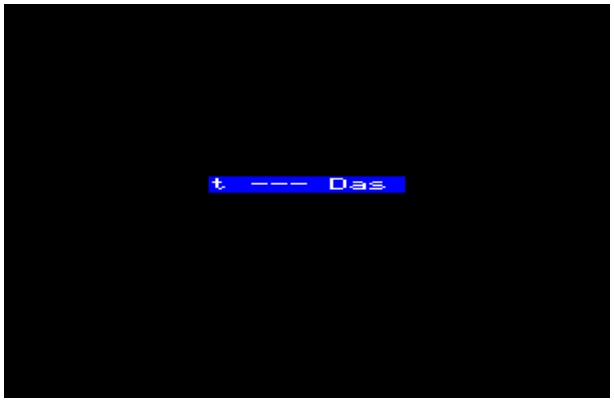



```
PROGRAM:SinusCosinus
01 CLS :VMODE 2:COLOR 1
02 DRAW 35,0,35,119,3
03 PLOT 35,0
04 FOR X=0 TO 119
05 W=3*X:V=SIN(W)/10
06 DRAWTO 35-V,X
07 NEXT
08 ? @65,0;"Sinus"
09 PLOT 10,0:COLOR 2
10 FOR X=0 TO 119
11 W=3*X:V=COS(W)/10
12 DRAWTO 35-V,X
13 NEXT
14 ? @65,50;"Cosinus"
15 Z=KEY(1)
#
```

4 Ein ”Gesicht” im Grafikmode 2

```
PROGRAM:gesicht
01 VM 2:PAL 3,6
02 FCIRCLE 38,40,36,36,3
03 FOR K=25 TO 30
04 ARC 38,40,K,30,90,270,1:NEXT
05 FCIRCLE 26,20,8,8,2
06 FCIRCLE 26,60,8,8,2
07 FCIRCLE 44,40,12,8,1
08 FCIRCLE 28,21,4,4,0
09 FCIRCLE 28,59,4,4,0
10 GO 10
#
```

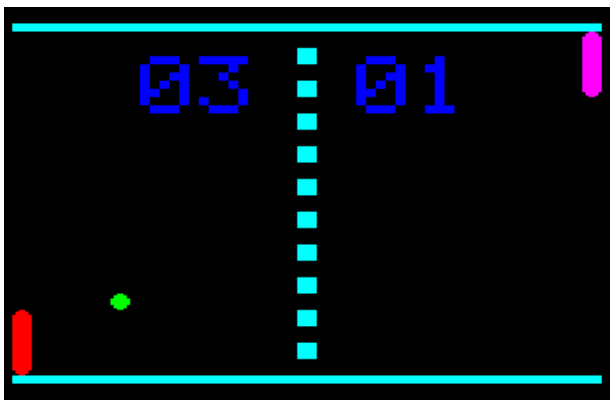
5 Laufschrift




```
PROGRAM:Laufschrift
01 DA 0,0,-1,-1,1,10
02 DA 5,"xxxxxxxxxx"
03 DA 15,$17,$17,$17,$17,$17
04 DA 20,$17,$17,$17,$17,$17
05 DA 50,"Das ist ein lustiger"
06 DA 70," Test --- Das ist ein"
07 FOR E=50 TO 79
08 ACOPY E,5,10
09 SPRITE 0,10,10
10 SYNC 4
11 NEXT
12 GOTO 7
#
```

6 Das altbekannte PONG

Die Schläger werden mit den Ctrl- und Shifttasten links und rechts gesteuert.




```
PROGRAM:Pong  V0.2
01 'Pong Version 0.2
02 GOSUB 30 :'Init
03 'the main loop
04 SYNC 2:SPRITE 42,Y,X
05 A=A-KEY(4):LIMIT A,1,18
06 B=B-KEY(5):LIMIT B,1,18
07 SPRITE 0,A,0:SPRITE 21,B,29
08 P=(P+1)%2:IF P=0:X=X+E:Y=Y+F
09 IF (Y=0)#(Y=22) NO 20:F=-F:Y=Y+F
```

```
10 IF X>0 GOTO 13
11 IF (Y<A)#(Y>(A+3)) GOTO 17
12 X=1:E=-E:NOTE 78:GOTO 4
13 IF X<29 GOTO 4
14 IF (Y<B)#(Y>(B+3)) GOTO 17
15 X=28:E=-E:NOTE 78:GOTO 4
16 'a point
17 SPRITE 42,Y,X:NOTE 127
18 IF X=0 THEN R=R+1:X=27
19 IF X=29 THEN L=L+1:X=2
20 GOSUB 49
21 IF L=15 THEN GOTO 52
22 IF R=15 THEN GOTO 58
23 WAIT 5:GOTO 4
24
25
26
27
28
29
30 'Init sprites and draw playfield
31 DA 0,2,-1,-1,4,1,18,15,15,19
32 DA 9,$F2,$F2,$F2,$F2
33 DA 21,2,-1,-1,4,1,18,15,15,19
34 DA 30,$F3,$F3,$F3,$F3
35 DA 42,2,-1,-1,1,1,17,$F4
36 COLOR 5,0:CLS
37 DRAW 1,0,1,59
38 DRAW 44,0,44,59:COLOR 5,1
39 FOR I= 1 TO 10
40  BOX 4*I,29,4*I+1,30
41 NEXT
42 X=3:Y=10:E=1:F=1
43 IF RND(2)=1 THEN E=-1:X=25
44 SPRITE 0,1,0
45 SPRITE 21,1,29
46 DATA 51,64,"To start Game",0
47 ALERT 51
48 COLOR 1,0
49 ? !$42;@2,6;L;@2,17;R:RETURN
50 RETURN
51
52 DATA 51,39
53 DATA 52,"Player 1 wins. New?",0
54 ASK P,51
55 IF P=1 THEN L=0:R=0:GOTO 1
56 CLS :COLOR 2,0
57 END
58 DATA 51,55
59 DATA 52,"Player 2 wins. New?",0
60 ASK P,51
61 IF P=1 THEN L=0:R=0:GOTO 1
62 CLS :COLOR 2,0
#
```

7 Ein kleines Autorennspiel

Gesteuert wird mit den Cursortasten nach links und nach rechts. Ein violetter Bonuspunkt bringt 20 Punkte, Highscore wird im internen EEPROM gespeichert.


```
PROGRAM:Racer
01 'init game
02 COLOR 7,1:CLS :BOX 3,3,42,56
03 COLOR 7,0:CBOX 2,2,20,27
04 DATA 0,2,-1,-1,2,2,20,21,22,23
05 DATA 9,133,133,133,133
06 X=13:S=15:Q=0:C=0:D=0:T=$88
07 B=0:E=0:P=0:V=6:GOSUB 60
08 DATA 25,39,"--GAME OVER--",0
09 G=0:'EEPROM Highscore address
10
11 'the main loop
12 Q=Q+1:IF Q=5:D=1-RND(3):Q=0
13 X=X+KEY(6):LIMIT X,4,24:O=S
14 S=S+D:LIMIT S,7,22:COLOR 4,0
15 SPRITE 0,-1,-1:SCROLL 2
16 SPRITE 0,14,X:COLOR 6,6
17
18 'draw street
19 FOR I=2 TO S-4:? @2,I;%T:NEXT
20 FOR I=S+4 TO 27:? @2,I;%T:NEXT
21 COLOR 6,0:IF O>S ? @2,S-3;%131
22 COLOR 0,6:IF O<S ? @2,S-4;%130
23 COLOR 0,6:IF O>S ? @2,S+4;%131
24 COLOR 6,0:IF O<S ? @2,S+3;%130
25
26 'bonus points
27 J=RND(60):COLOR 3,0
28 IF J=5 THEN ? @2,S-2;%27;
29 IF J=25 THEN ? @2,S;%27;
30 IF J=45 THEN ? @2,S+2;%27;
31
32 'trees
33 B=B+1:IF B=4 THEN B=0
34 IF B<>1 GOTO 39
35 H=RND(2)
36 IF H=0 THEN F=2+RND(S-7)
37 IF H=1 THEN F=27-RND(22-S)
38 COLOR 2,6:? @2,F;%161;:GOTO 43
39 IF B<>2 GOTO 43
40 COLOR 4,6:? @2,F;%162;
41
42 'check collision
43 W=30-3*V:IF (AR(0)&1)=0 GOTO 47
44 N=AR(13)+AR(14)+AR(15)+AR(16)
45 IF N>150 GOTO 55
46 P=P+19:DATA 13,32,32:W=94
47 P=P+1:COLOR 7,1
48 ? @0,2;"Score:";P
49 IF P>500 THEN V=5
50 IF P>1000 THEN V=4
```

```
51 NOTE W:SYNC V:DATA 0,2
52 GOTO 12
53
54 'game over
55 NOTE 255: IF P<M GOTO 57
56 EPOKE G,LO(P):EPOKE G+1,HI(P)
57 GOSUB 60:ALERT 25: GOTO 1
58
59 'print High-score
60 IF EPEEK(G+1)<128 GOTO 62
61 EPOKE G,0:EPOKE G+1,0
62 M=EPEEK(G)+256*EPEEK(G+1)
63 COLOR 6,1
64 ? @0,15;"High:";M:RETURN
#
```

8 Eatman - ein Pacman Clone

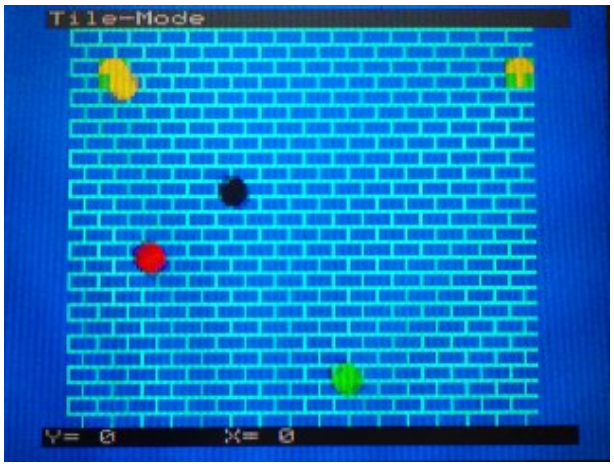
Dieses Spiel zeigt was mit 95 Zeilen BASIC machbar ist, aber auch die Grenzen. Denn es werden fast alle Zeilen gebraucht und der Programmtext ist alles andere als übersichtlich. Gesteuert wird mit den Cursortasten nach links und nach rechts, wenn man ein rotes Obst erwischt, werden die Geister für eine kurze Zeit weiß und können selbst gejagt werden. Die Geister sind allerdings ”dumm”, denn sie bewegen sich nur vom Zufall gesteuert durch das Labyrinth.


```
PROGRAM:eatman3
01 BO 0:P=0:GOS 64:DA 64,64,32,0
02 ALERT 64:X=AR(1):Y=AR(0):D=21
03 GOSUB 48:FOR F=1 TO 4:A=F*5
04 IF AR(A)<>AR(3) THEN GO 9
05 IF AR(A+1)<>AR(4) THEN GO 9
06 IF G>0 THEN NO 98:P=P+20:GO 35
07 ? @AR(3)+1,AR(4)+4;%250;
08 NO 255: BO 2:A=KEY(2):END
09 NEXT :F=0:GOSUB 40
10 T=1-T: IF T=0 THEN GOTO 17
11 K=KEY(6):M=KEY(7)
12 IF (K=-1) & (L=1) THEN X=X-1
13 IF (K=1) & (R=1) THEN X=X+1
14 IF (M=1) & (U=1) THEN Y=Y-1
15 IF (M=-1) & (D=1) THEN Y=Y+1
16 DA 0,Y,X:GO 32
17 FOR F=1 TO 4:GOSUB 40
18 S=U+D+L+R:M=N
19 IF S>2 THEN GOSUB 37
20 IF S=1 THEN N=(N+2)&3:GO 26
21 IF (N=0) & (R=1) THEN GO 26
22 IF (N=2) & (L=1) THEN GO 26
23 IF (N=1) & (D=1) THEN GO 26
24 IF (N=3) & (U=1) THEN GO 26
25 GOSUB 37: GO 18
26 IF N=0 THEN DA A+1,AR(A+1)+1
27 IF N=1 THEN DA A,AR(A)+1
28 IF N=2 THEN DA A+1,AR(A+1)-1
29 IF N=3 THEN DA A,AR(A)-1
30 DA A+2,N:NEXT
31 IF G>0 THEN G=G-1
32 COLOR 7,0:? @0,10;"Punkte:";P
33 SYNC 2: IF Q=1 THEN GO 35
34 GO 3
35 DA A,10,10:N=1:GO 31
36 GOSUB 64:DA 64,0,4,32,0:GOTO 2
37 Z=RND(3):IF Z=2 THEN Z=3
38 N=(M+Z)&3:RETURN
39 'check if figure f can move
40 L=0:R=0:U=0:D=0:A=F*5
```

```
41 N=AR(A+2):E=100+AR(A)*21+AR(A+1)
42 IF AR(E-21)<>49 THEN U=1
43 IF AR(E+21)<>49 THEN D=1
44 IF AR(E-1)<>49 THEN L=1
45 IF AR(E+1)<>49 THEN R=1
46 RETURN
47 'draw figures
48 FOR W=1 TO 4:A=W*5
49 J=AR(A+3):I=AR(A+4):GOSUB 91
50 J=AR(A+0):I=AR(A+1):COL W+1,0
51 DA A+3,J,I
52 IF (G>0)&(G<>7) THEN COL 7,0
53 ? @1+J,4+I;%25;:NEXT
54 J=AR(3):I=AR(4):GOS 91:COL 6,0
55 E=100+21*AR(0)+AR(1)
56 IF AR(E)=42 THEN GOSUB 61
57 IF AR(E)=46 THEN GOSUB 60
58 DA E,0:? @1+AR(0),4+AR(1);%26;
59 DA 3,AR(0),AR(1):RET
60 NO 104: P=P+1:Q=Q-1:RET
61 NO 120: P=P+10:G=50:RET
62
63 'Draw playfield and set data
64 DA 0,15,10,1,0,0,10,9,1,0,0
65 DA 10,10,9,1,0,0,10,11,1,0,0
66 DA 20,10,8,1,0,0:M=0
67 DA 100,"11111111111111111111"
68 DA 121,"1.....*...1........*1"
69 DA 142,"1.11.1111.1.1111.11.1"
70 DA 163,"1.11.1111.1.1111.11.1"
71 DA 184,"1...................1"
72 DA 205,"1.11.1.1111111.1.11.1"
73 DA 226,"1.*..1....1....1....1"
74 DA 247,"1.11.1111...1111.11.1"
75 DA 268,"1..1.1.........1.1..1"
76 DA 289,"11.1...111 111...1*11"
77 DA 310,"1..1.1.1     1.1.1..1"
78 DA 331,"1.11.1.1111111.1.11.1"
79 DA 352,"1....1.........1....1"
80 DA 373,"1.11.1.1111111.1.11*1"
81 DA 394,"1..1.1.1111111.1.1..1"
82 DA 415,"11.1....*........1.11"
83 DA 436,"11.1.1.1111111.1.1.11"
84 DA 457,"1....1....1....1....1"
85 DA 478,"1.1111111.1.1111111.1"
86 DA 499,"1.*............*....1"
87 DA 520,"11111111111111111111"
88 FOR J=0 TO 20:FOR I=0 TO 20
89 GOSUB 91:NEXT :NEXT :Q=M:RETURN
90 'draw one array element
91 C=AR(100+21*J+I):D=0:F=6
92 IF C=49 THEN D=15:F=1
93 IF C=46 THEN D=226:F=6:M=M+1
94 IF C=42 THEN D=235:F=2
95 COLOR F,0:? @1+J,4+I;%D;:RETURN
#
```

9 Tilemode-Testprogramm

Dieses kleine Programm zeigt die Nutzung des Tilemode. Mittels der Cursortasten kann der sichtbare Ausschnitt im 32x32 Tiles großen Feld gescrollt werden. Da in diesem Modus kein Screenshot möglich ist, wurde das Bild von einem NEC TFT abfotografiert.




```
PROGRAM:tile01 test
01 LFIND L,144:IF L> 0 GOTO 4
02 PRINT "NO TILE-DRIVER"
03 END
04 VM 7:BORDER 1:? "Tile-Mode"
05 VPOKE 0,0:VPOKE 1,0
06 CALL L,14,0,1:CALL L,15,0,1,5
07 CALL L,14,1,4:CALL L,15,1,11,6
08 CALL L,14,2,1:CALL L,15,2,2,2
09 CALL L,14,3,1:CALL L,15,3,2,6
10 CALL L,14,4,1:CALL L,15,4,2,0
11 CALL L,14,5,1:CALL L,15,5,2,4
12 CALL L,25,0,4 ':end init
13 CA L,26,0,2,1:CA L,27,0,13,3,1,1
14 CA L,26,1,3,1:CA L,27,1,25,5,1,1
15 CA L,26,2,4,1:CA L,27,2,37,7,1,1
16 CA L,26,3,5,1:CA L,27,3,49,9,1,1
17 VPOKE 73,1: VP 87,1:CA L,32
18 E=E+1:IF E<2 GOTO 21
19 C=(VPEEK(0)+KEY(7)) & 255
20 D=(VPEEK(1)-KEY(6)) & 255:E=0
21 VPOKE 0,C:VPOKE 1,D
22 CALL L,35:FOR I=0 TO 3:K=I*16
23 IF AR(K+1)=120-I DA K+11,255
24 IF AR(K+1)=I DA K+11,1
25 IF AR(K)=96-I DA K+10,255
26 IF AR(K)=I DA K+10,1
27 NEXT :? @1,0;"Y=";C;"   "
28 ? @1,10;"X=";D;"   "
29 GOTO 18
#
```