

AVR Chipbasic 17

V1.0

BASIC PROGRAMME UND SPIELE

Hinweis

Alle Programme unterliegen der GPL (GNU General Public License) Version 3 oder höher, jede Nutzung der Software/Informationen nonkonform zur GPL oder außerhalb des Geltungsbereiches der GPL ist untersagt!

Die Veröffentlichung dieses Projekts erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE IRGEND EINE GARANTIE, auch ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK.

Alle im Text genannten Marken sind Eigentum des entsprechenden Inhabers.

Das Grundkonzept von Hard- und Software wurde von Jörg Wolfram entwickelt.
<http://www.jcwolfram.de/projekte/avr/chipbasic2/main.php>

Version 1.0 vom 15.03.2017

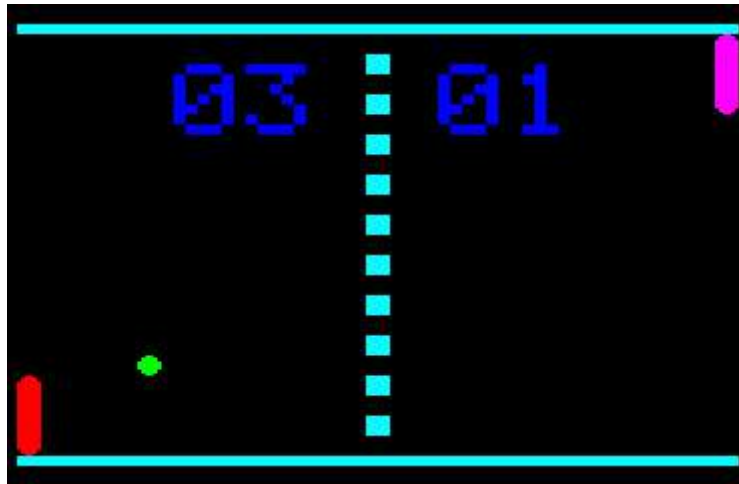
12:44:08

Inhalt

1 Spiele	3
1.1 Classic Pong	3
1.2 Racer	5
1.3 Pacman	7
1.4 Pong One	9
1.5 Batnum	12

1 Spiele

1.1 Classic Pong



Autor: Jörg Wolfram

Das klassische Pong für 2 Spieler. Gesteuert wird mit der linken und rechten Shift und CTRL Taste.

```
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
#
```

1.2 Racer



Autor: Jörg Wolfram

Gesteuert wird mit den Cursortasten links und 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
#
```

1.3 Pacman



Autor: Jörg Wolfram

Gesteuert wird mit den Cursortasten. Wenn man ein rotes Obst erwischt, werden die Geister für eine kurze Zeit weiß und können selbst gejagt werden.

```
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
#

```

1.4 Pong One



Autor: Dominik Kristen

Das klassische Pong, jedoch für einen Spieler gegen den Computer.

```

PROGRAM:Pong
01 GOSUB 65
02 GOSUB 30
03
04 SYNC 0:SPRITE 42,Y,X
05 A=A-KEY(7)
06 'A=A+1
07 LIMIT A,1,18:B=Y-U:LIMIT B,1,18
08 SPRITE 0,A,0:SPRITE 21,B,29
09 P=(P+1)%2:IF P=0:X=X+E:Y=Y+F
10 IF (Y=0)#(Y=22) NO 20:F=-F:Y=Y+F
11 IF X>0 GOTO 14
12 IF (Y<A)#(Y>(A+3)) GOTO 17
13 X=1:E=-E:NOTE 78:GOTO 4
14 IF X<29 GOTO 4
15 IF (Y<B)#(Y>(B+3)) GOTO 17
16 X=28:E=-E:NOTE 78:GOTO 4
17
18 SPRITE 42,Y,X:NOTE 127
19 IF X=0 THEN R=R+1:X=27
20 IF X=29 THEN L=L+1:X=2
21 GOSUB 50
22 IF L=15 THEN GOTO 53
23 IF R=15 THEN GOTO 59
24 WAIT 5:U=RND(V):GOTO 4
25
26
27
28

```

```
29
30 IF T=12:V=6:O=2
31 IF T=14:V=4:O=2
32 DA 0,2,-1,-1,4,1,18,15,15,19
33 DA 9,14,14,14,14
34 DA 21,2,-1,-1,4,1,18,15,15,19
35 DA 30,12,12,12,12
36 DA 42,2,-1,-1,1,1,17,$F4
37 COLOR 7,0:CLS
38 DRAW 1,0,1,59
39 DRAW 44,0,44,59:COLOR 5,1
40 IF T=16:V=3:O=1
41
42
43 X=3:Y=10:E=1:F=1
44 IF RND(2)=1 THEN E=-1:X=25
45 SPRITE 0,1,0
46 SPRITE 21,1,29
47 DATA 51,64,"To start Game",0
48 'ALERT 51
49 COLOR 5,0
50 ? !66;@2,6;L;@2,17;R:RETURN
51 RETURN
52
53 DATA 51,39
54 DATA 52,"Player 1 wins. New?",0
55 ASK P,51
56 IF P=1 THEN L=0:R=0:GOTO 1
57 CLS :COLOR 2,0
58 END
59 DATA 51,55
60 DATA 52,"AVR wins. New?",0
61 ASK P,51
62 IF P=1 THEN L=0:R=0:GOTO 1
63 CLS :COLOR 2,0
64 DA 109,39,8,44,4,49,8,44,4
65 CLS :COLOR 1:FBOX 0,0,55,60
66 COL 6,1: ? !66;@2,8;"PONG"
67 ? @7,8;%136;%143;%143;%143
68 ? @7,12;%143;%143;%143;%143
69 ? @7,16;%143;%143;%143;%137
70 COLOR 7:T=12: ? @1,12;"DK's"
71 COL 0,6: ? @7,12"ONE":COL 7,1
72 ? @12,11;"Amateur"
73 ? @14,11;"Novice"
74 ? @16,11;"Profi"
75 'PLAY 109,4,4,128
76 COLOR 4: ? @T,9;%17:Q=KEY(7)
77 IF Q=-1 THEN NO 30:GOS 83
78 IF Q=1 THEN NO 30:GOS 87
79 ? @20,5;"Press FIRE to begin"
80
81 IF KEY(0)=234 THEN NO 100:RET
82 SYNC 2:GOTO 76
83 SYNC 4
84 IF T=12:T=14: ? @12,9;%32:RET
85 IF T=14:T=16: ? @14,9;%32:RET
86 IF T=16:T=12: ? @16,9;%32:RET
87 SYNC 4
88 IF T=12:T=16: ? @12,9;%32:RET
```

```
89 IF T=14:T=12:? @14,9;%32:RET
90 IF T=16:T=14:? @16,9;%32:RET
#
```

1.5 Batnum



Autor: John Kemeny

Portierung: Dominik Kristen

Bei diesem Spiel wählen Sie zu Beginn die Anzahl Münzen, Steine, was auch immer, die auf dem Tisch liegen sollen. Wählen Sie, wie viele Steine man pro Zug wegnehmen kann (z.B. 1-3).

Ziel des Spiels ist es, entweder durch Wegnahme des letzten Steins das Spiel zu gewinnen oder es zu vermeiden den letzten Stein liegen zu lassen.

Gespielt wird gegen den AVR.

```
PROGRAM:Batnum
01 CLS :GOSUB 84
02 CLS
03 COL 1:FBOX 0,0,50,60:COL 7,1
04 ? @2,2;INP "Enter Pile size: ";N
05 IF N>=1 THEN GOTO 7
06 GOTO 4
07 CLS :COL 1:FBOX 0,0,50,60
08 COL 7,1
09 ? @2,2;"Enter Win options"
10 ? @4,2;"1 - to take last"
11 ? @5,2;"2 - to avoid last"
12 M=KEY(1)
13 IF M=49 THEN GOTO 15
14 IF M<>50 THEN GOTO 7
15 CLS :COL 1:FBOX 0,0,50,60
16 COL 7,1
17 ? @2,2;INP "Enter min: ";A
18 ? @3,2;INP "Enter max: ";B
19 IF A>B THEN GOTO 15
20 IF A<1 THEN GOTO 15
21 CLS :COL 1:FBOX 0,0,50,60
22 COL 7,1
23 ? @2,2;"Enter start option"
```

```

24 ? @4,2;"1 - AVR first"
25 ? @5,2;"2 - You first"
26 S=KEY(1)
27 IF S=49 THEN GOTO 29
28 IF S<>50 THEN GOTO 21
29 C=A+B
30 IF S=50 THEN GOTO 33
31 GOSUB 36
32 IF W=1 THEN GOTO 2
33 GOSUB 57
34 IF W=1 THEN GOTO 2
35 GOTO 31
36 Q=N
37 IF M=49 THEN GOTO 39
38 Q=Q-1
39 IF M=49 THEN GOTO 44
40 IF N>A THEN GOTO 48
41 W=1
42 ? :? "AVR takes ";N;" and lose""
43 WAIT 2:L=KEY(1):RETURN
44 IF N>B THEN GOTO 48
45 W=1
46 ? :? "AVR takes ";N;" and wins"
47 WAIT 2:L=KEY(1):RETURN
48 P=Q-C*(Q/C)
49 IF P>=A THEN GOTO 51
50 P=A
51 IF P<=B THEN GOTO 53
52 P=B
53 N=N-P
54 CLS :? "AVR takes";P;". Rest:";N
55 W=0
56 RETURN
57 PRINT :PRINT "Your move ";
58 INPUT P
59 IF P<>0 THEN GOTO 63
60 PRINT "Not zero!"
61 W=1
62 RETURN
63 IF P>=A THEN GOTO 66
64 IF P=N THEN GOTO 71
65 GOTO 67
66 IF P<=B THEN GOTO 69
67 PRINT "Illegal Move";
68 GOTO 58
69 N=N-P
70 IF N<>0 THEN GOTO 78
71 IF M=49 THEN GOTO 75
72 PRINT "You lose!"
73 W=1:WAIT 1:L=KEY(1)
74 RETURN
75 PRINT "YOU WIN!!!"
76 W=1:WAIT 5
77 RETURN
78 IF N>=0 THEN GOTO 81
79 N=N+P
80 GOTO 67
81 W=0
82 RETURN
83 END

```

```
84 CLS :COL 9:FBOX 0,0,50,60
85 COL 0,9:? @2,5;!66;"BATNUM"
86 ? @8,1;"Battle of numbers game"
87 ? @9,1;"where the AVR is your"
88 ? @10,1;"opponent."
89
90
91
92
93 COL 0,7:? @20,9;"PRESS A KEY"
94 WAIT 2:L=KEY(1)
95 RETURN
#
```