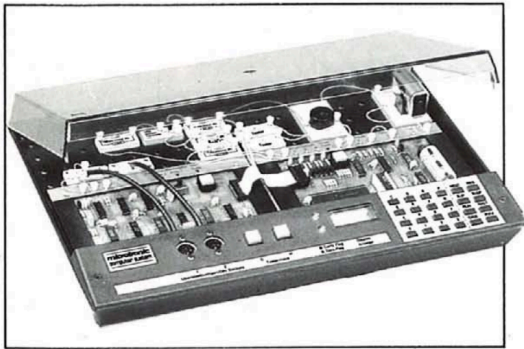


# **microtronic** computer system



## computer spiele

Strategy and puzzle games  
Games of chance and patience

17+4

Reaction games

Number Avalanche

One-Armed Bandit

Chess Clock

Code Lock

Clock with chimes

World Time Calculation

Prime number and  
exponentiation calculations

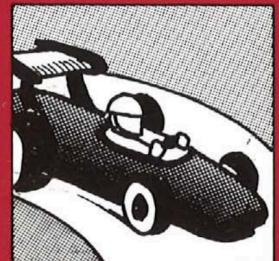
... and much more.



Black Jack



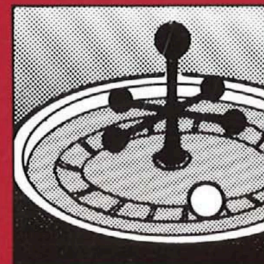
Sound Memory



Car Race



Fallgrube



Roulette



Blockade

25 games and experiments  
to program yourself for Microtronic 2090



Order no. 2094

# Table of Contents

## 1. Computer games:

## Page

|                                                               |    |
|---------------------------------------------------------------|----|
| Number Avalanche – electronic memory training                 | 4  |
| Blockade – a strategy game                                    | 6  |
| One-Armed Bandit – a game of chance                           | 8  |
| Arithmetic Trainer – the computer checks arithmetic problems  | 11 |
| Number Guessing (High-Low Game) – classic computer game       | 12 |
| A Hundred Wins – a devilish puzzle game                       | 14 |
| MICROTRONIC Roulette – a game of chance on a well-known model | 16 |
| Sound Memory – a concentration game                           | 19 |
| Pull – a strategy game                                        | 23 |
| Car Race – an interesting reaction game                       | 26 |
| Sorting Digits – a tricky brain-teaser                        | 28 |
| Forming Words and Sentences – the funny computer              | 30 |
| 17 + 4 (Black Jack) – a game of chance against the computer   | 32 |
| Fallgrube (Pitfall) – a tactical game of chance               | 34 |
| Chess Clock – the computer checks thinking times              | 37 |

## 2. Interesting computer experiments:

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| Code Lock – a microelectronic alarm centre                                     | 40 |
| Light-barrier controlled time measurement – when was the light switched on?    | 43 |
| Clock with chimes, date display and alarm function                             | 46 |
| World Time Calculation – what time is it in Tokyo?                             | 50 |
| Acoustic Computer Remote Switching – switching electrical devices by telephone | 52 |

## 3. For mathematicians and statisticians:

|                                                   |    |
|---------------------------------------------------|----|
| Program for drawing pie charts                    | 54 |
| Diagonal calculation program for polygons         | 55 |
| Division program for any number of decimal places | 56 |
| Prime number calculation                          | 58 |
| Exponentiation                                    | 60 |



*microtronic* computer system

---

# computer spiele

25 games and experiments  
to program yourself  
for Microtronic 2090

**BUSCH Modellspielwaren**  
**Postfach 1360**  
**D-6806 Viernheim**

Copyright 1983 by  
BUSCH GmbH + Co. KG, Viernheim

Alle Rechte vorbehalten

Grafik  
Atelier Wuthe, Weinheim

Printed in W.-Germany  
12/84

# Before you try the first programs!

All programs are intended specifically for the BUSCH MICROTRONIC computer system 2090. A basic knowledge of the functions of the MICROTRONIC computer is assumed. You should therefore have worked through the instruction book Part 1 supplied with the computer at least as far as page 22.

Every program has been tested several times for correct operation. If a program does not behave as described, you have certainly made a mistake while programming (transposed digits or similar).

In such a case you should go back to the start of the program once more. Required key presses: HALT – NEXT – 00. The first instruction code (address 00) is now displayed. Each time you press the NEXT key the next instruction entry is displayed, and you can use the program table to check whether all instruction codes have been entered correctly. Please bear in mind that a single incorrect instruction entry may be enough to stop the program working properly. After checking, please try a new program start: HALT – NEXT – 00 – RUN.

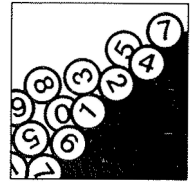
By purchasing the MICROTRONIC cassette interface 2095 you can transfer the programs entered into the computer onto a cassette recorder or tape machine. Even the longest programs are then ready to play within a few minutes, without having to be entered again. You can put together your own MICROTRONIC program library.

The programs published in this book were developed by entrants to the BUSCH MICROTRONIC programming competition of 1983. If you too have a good idea, you should send us your program for informal appraisal. Perhaps your suggestion will also be published in a following MICROTRONIC book. Please note the guidance given in the MICROTRONIC instruction book Part 2 (page 77).

We wish you continued enjoyment in programming, experimenting and learning through play how a computer works.



## Number Avalanche



The game Number Avalanche is an electronic memory exercise. It demands and develops concentrated thinking.

Individual numbers or letters light up briefly on the computer display in random order. The player's task is to memorise the numbers or letters and repeat them, in the order the computer gave them, by entering them on the keypad. After each correct entry the computer briefly shows a further number. The sequence to be repeated grows ever longer; the process continues like an avalanche, up to a maximum of 14 numbers or letters entered in the correct order.

The computer records how many repetitions were made before the first mistake. Six to eight correct entries is already a good result. Anyone who can remember all 14 has an outstanding memory.

With several players, the winner is the one who achieves the highest entry score over the agreed number of rounds.

### Function description:

After pressing HALT – NEXT – 00 the program is entered according to the program table. The piezo buzzer is then connected to the output 1 and GND sockets.

Program start: HALT – NEXT – 00 – RUN

A number (0 – 9) or a letter (A – F) is shown briefly on the computer display. The number or letter must be repeated by pressing the corresponding key. A second number is shown immediately. Both numbers must now be repeated on the keypad in the correct order. The computer shows a third number. All three numbers must again be entered one after another in the correct order.

The game continues until a mistake is made during entry.

As soon as a number or letter is entered incorrectly, or in the wrong order, the computer responds with a continuous tone and the display shows how many numbers were entered in the correct order. Pressing any number key restarts the game.

If all 14 numbers are entered in the correct order, the computer produces a rapidly warbling “tone of joy”. The game is restarted by pressing any number key.

Programmer: Frank Gothe, 2210 Itzehoe

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | F02    | DISOUT   |
| 03   | F05    | RND      |
| 04   | OD2    | MOV D,2  |
| 05   | OE3    | MOV E,3  |
| 06   | OF4    | MOV F,4  |
| 07   | F12    | DISP 1,2 |
| 08   | B39    | CALL 39  |
| 09   | F13    | DISP 1,3 |
| 0A   | B39    | CALL 39  |
| 0B   | F14    | DISP 1,4 |
| 0C   | B39    | CALL 39  |
| 0D   | F05    | RND      |
| 0E   | OD5    | MOV D,5  |
| 0F   | OE6    | MOV E,6  |

|    |     |          |
|----|-----|----------|
| 10 | OF7 | MOV F,7  |
| 11 | F15 | DISP 1,5 |
| 12 | B39 | CALL 39  |
| 13 | F16 | DISP 1,6 |
| 14 | B39 | CALL 39  |
| 15 | F17 | DISP 1,7 |
| 16 | B39 | CALL 39  |
| 17 | F05 | RND      |
| 18 | OD8 | MOV D,8  |
| 19 | OE9 | MOV E,9  |
| 1A | OFA | MOV F,A  |
| 1B | F18 | DISP 1,8 |
| 1C | B39 | CALL 39  |
| 1D | F19 | DISP 1,9 |
| 1E | B39 | CALL 39  |
| 1F | F1A | DISP 1,A |

|    |     |          |
|----|-----|----------|
| 20 | B39 | CALL 39  |
| 21 | F05 | RND      |
| 22 | ODB | MOV D,B  |
| 23 | OEC | MOV E,C  |
| 24 | F1B | DISP 1,B |
| 25 | B39 | CALL 39  |
| 26 | F1C | DISP 1,C |
| 27 | B39 | CALL 39  |
| 28 | F05 | RND      |
| 29 | F1D | DISP 1,D |
| 2A | B39 | CALL 39  |
| 2B | F1E | DISP 1,E |
| 2C | B39 | CALL 39  |
| 2D | F1F | DISP 1,F |
| 2E | B39 | CALL 39  |
| 2F | 1FF | MOVI F,F |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | 10E    | MOVI 0,E |
| 31   | 10D    | MOVI 0,D |
| 32   | FEF    | DOT F    |
| 33   | FEE    | DOT E    |
| 34   | 51D    | ADDI 1,D |
| 35   | D37    | BRC 37   |
| 36   | C32    | GOTO 32  |
| 37   | FF0    | KIN 0    |
| 38   | C00    | GOTO 00  |
| 39   | 511    | ADDI 1,1 |
| 3A   | C98    | GOTO 98  |
| 3B   | FF0    | KIN 0    |
| 3C   | 802    | CMP 0,2  |
| 3D   | E3F    | BRZ 3F   |
| 3E   | C8E    | GOTO 8E  |
| 3F   | 911    | CMPI 1,1 |

|    |     |          |
|----|-----|----------|
| 40 | E8D | BRZ 8D   |
| 41 | FF0 | KIN 0    |
| 42 | 803 | CMP 0,3  |
| 43 | E45 | BRZ 45   |
| 44 | C8E | GOTO 8E  |
| 45 | 921 | CMPI 2,1 |
| 46 | E8D | BRZ 8D   |
| 47 | FF0 | KIN 0    |
| 48 | 804 | CMP 0,4  |
| 49 | E4B | BRZ 4B   |
| 4A | C8E | GOTO 8E  |
| 4B | 931 | CMPI 3,1 |
| 4C | E8D | BRZ 8D   |
| 4D | FF0 | KIN 0    |
| 4E | 805 | CMP 0,5  |
| 4F | E51 | BRZ 51   |

|    |     |          |
|----|-----|----------|
| 50 | C8E | GOTO 8E  |
| 51 | 941 | CMPI 4,1 |
| 52 | E8D | BRZ 8D   |
| 53 | FF0 | KIN 0    |
| 54 | 806 | CMP 0,6  |
| 55 | E57 | BRZ 57   |
| 56 | C8E | GOTO 8E  |
| 57 | 951 | CMPI 5,1 |
| 58 | E8D | BRZ 8D   |
| 59 | FF0 | KIN 0    |
| 5A | 807 | CMP 0,7  |
| 5B | E5D | BRZ 5D   |
| 5C | C8E | GOTO 8E  |
| 5D | 961 | CMPI 6,1 |
| 5E | E8D | BRZ 8D   |
| 5F | FF0 | KIN 0    |

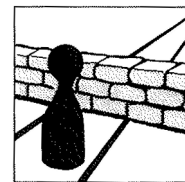
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 808    | CMP 0,8  |
| 61   | E63    | BRZ 63   |
| 62   | C8E    | GOTO 8E  |
| 63   | 971    | CMPI 7,1 |
| 64   | E8D    | BRZ 8D   |
| 65   | FF0    | KIN 0    |
| 66   | 809    | CMP 0,9  |
| 67   | E69    | BRZ 69   |
| 68   | C8E    | GOTO 8E  |
| 69   | 981    | CMPI 8,1 |
| 6A   | E8D    | BRZ 8D   |
| 6B   | FF0    | KIN 0    |
| 6C   | 80A    | CMP 0,A  |
| 6D   | E6F    | BRZ 6F   |
| 6E   | C8E    | GOTO 8E  |
| 6F   | 991    | CMPI 9,1 |

|    |     |          |
|----|-----|----------|
| 70 | E8D | BRZ 8D   |
| 71 | FF0 | KIN 0    |
| 72 | 80B | CMP 0,B  |
| 73 | E75 | BRZ 75   |
| 74 | C8E | GOTO 8E  |
| 75 | 9A1 | CMPI A,1 |
| 76 | E8D | BRZ 8D   |
| 77 | FF0 | KIN 0    |
| 78 | 80C | CMP 0,C  |
| 79 | E7B | BRZ 7B   |
| 7A | C8E | GOTO 8E  |
| 7B | 9B1 | CMPI B,1 |
| 7C | E8D | BRZ 8D   |
| 7D | FF0 | KIN 0    |
| 7E | 80D | CMP 0,D  |
| 7F | E81 | BRZ 81   |

|    |     |          |
|----|-----|----------|
| 80 | C8E | GOTO 8E  |
| 81 | 9C1 | CMPI C,1 |
| 82 | E8D | BRZ 8D   |
| 83 | FF0 | KIN 0    |
| 84 | 80E | CMP 0,E  |
| 85 | E87 | BRZ 87   |
| 86 | C8E | GOTO 8E  |
| 87 | 9D1 | CMPI D,1 |
| 88 | E8D | BRZ 8D   |
| 89 | FF0 | KIN 0    |
| 8A | 80F | CMP 0,F  |
| 8B | E8D | BRZ 8D   |
| 8C | C8E | GOTO 8E  |
| 8D | F07 | RET      |
| 8E | 1F0 | MOVI F,0 |
| 8F | FE0 | DOT 0    |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | 711    | SUBI 1,1 |
| 91   | 01D    | MOV 1,D  |
| 92   | 10E    | MOVI 0,E |
| 93   | 10F    | MOVI 0,F |
| 94   | F03    | HXDZ     |
| 95   | F3D    | DISP 3,D |
| 96   | FF0    | KIN 0    |
| 97   | C00    | GOTO 00  |
| 98   | F01    | NOP      |
| 99   | F01    | NOP      |
| 9A   | F01    | NOP      |
| 9B   | F01    | NOP      |
| 9C   | F01    | NOP      |
| 9D   | F01    | NOP      |
| 9E   | F02    | DISOUT   |
| 9F   | C3B    | GOTO 3B  |

# Blockade



Blockade is a strategy game played on two tracks of 5 and 10 squares. The computer and the player each have one piece on each track; the computer's two pieces are placed at the far left of the tracks and the player's at the far right. The pieces may then be pushed freely forwards and backwards along either track, the computer and the player taking turns. Jumping over the opponent's pieces is not permitted. The aim is to force both of the computer's pieces back into their starting position, on the leftmost squares, so that it has no moves left. This takes a good deal of tactics and strategy, since the computer also tries to block the player's pieces and force them back into the right-hand half of the board.

A Blockade playing field with counters, to serve as pieces, is supplied with this book.

## Function description:

First the program is entered after HALT – NEXT – 00 according to the table. The piezo buzzer is then connected to the GND and output 1 sockets on the computer board.

Program start: HALT – NEXT – 00 – RUN. The display shows 00.

The computer's pieces are placed on square 1 of each track. The player's pieces go on square A of track 1 and square 5 of track 2. The player's first move can then be entered into the computer. To do this, first enter 1 or 2 for the track, then the number of the square the player wishes to move to.

To place a piece on track 1, square 7, for example, the entry 17 is required. The computer shows its move (13), that is, the computer piece

is to be placed on track 1, square 3. The player now enters 22, say, and places a piece on square 2 of the second track. The computer announces that it wants to place its piece on track 1, square 6 (display: 16).

The computer already has the upper hand. The player can no longer advance further to the left, because the computer's pieces block the way. The player must retreat and enters, for example, 18. The computer immediately moves its piece to square 7 of the first track (display: 17). The player is pushed back further.

If the computer wins in this example, the display shows FFF and a short continuous signal sounds. A new game can be started by pressing key 0.

## If you cheat during entry:

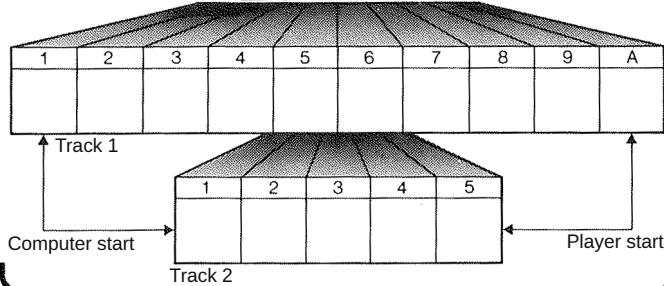
If, for example, 26 is entered (square 6 does not exist on track 2), or a move is entered onto a square already occupied by a piece, the game is broken off at once and the display shows 00 – the game must be started again.

If you manage to beat the computer, which is possible once you have worked out the right strategy, the display shows CCC and the computer produces an interrupted signal tone.

If you cannot win the game after several attempts, you will find the strategy explained on page 13.

Programmer: Christoph Fuchs, 6920 Sinsheim

# BLOCKADE



| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 40   | 943    | CMPI 4,3 |
| 41   | E43    | BRZ 43   |
| 42   | C06    | GOTO 06  |
| 43   | 1FA    | MOVI F,A |
| 44   | FEA    | DOT A    |
| 45   | 51B    | ADDI 1,B |
| 46   | D48    | BRC 48   |
| 47   | C43    | GOTO 43  |
| 48   | FE9    | DOT 9    |
| 49   | 1FB    | MOVI F,B |
| 4A   | 1FC    | MOVI F,C |
| 4B   | F3A    | DISP 3,A |
| 4C   | FF0    | KIN 0    |
| 4D   | C00    | GOTO 00  |
| 4E   | 1FA    | MOVI F,A |
| 4F   | FEA    | DOT A    |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 70   | 11F    | MOVI 1,F |
| 71   | F07    | RET      |
| 72   | 664    | SUB 6,4  |
| 73   | 645    | SUB 4,5  |
| 74   | 12F    | MOVI 2,F |
| 75   | F07    | RET      |
| 76   | 92F    | CMPI 2,F |
| 77   | E80    | BRZ 80   |
| 78   | 9AE    | CMPI A,E |
| 79   | D00    | BRC 00   |
| 7A   | 8E2    | CMP E,2  |
| 7B   | D00    | BRC 00   |
| 7C   | E00    | BRZ 00   |
| 7D   | 80E    | CMP 0,E  |
| 7E   | E00    | BRZ 00   |
| 7F   | F07    | RET      |

## Adr.Befehl Mnemonic

|    |     |          |
|----|-----|----------|
| 00 | F08 | CLEAR    |
| 01 | FE0 | DOT 0    |
| 02 | 1A0 | MOVI A,0 |
| 03 | 151 | MOVI 5,1 |
| 04 | 112 | MOVI 1,2 |
| 05 | 113 | MOVI 1,3 |
| 06 | F2E | DISP 2,E |
| 07 | FFF | KIN F    |
| 08 | FFE | KIN E    |
| 09 | B76 | CALL 76  |
| 0A | 91F | CMPI 1,F |
| 0B | E0F | BRZ 0F   |
| 0C | 92F | CMPI 2,F |
| 0D | E11 | BRZ 11   |
| 0E | C00 | GOTO 00  |
| 0F | 0E0 | MOV E,0  |

## Adr.Befehl Mnemonic

|    |     |          |
|----|-----|----------|
| 20 | B72 | CALL 72  |
| 21 | 905 | CMPI 0,5 |
| 22 | E28 | BRZ 28   |
| 23 | 9F5 | CMPI F,5 |
| 24 | E28 | BRZ 28   |
| 25 | 9E5 | CMPI E,5 |
| 26 | E28 | BRZ 28   |
| 27 | C2D | GOTO 2D  |
| 28 | B5B | CALL 5B  |
| 29 | B6E | CALL 6E  |
| 2A | C2D | GOTO 2D  |
| 2B | B5B | CALL 5B  |
| 2C | B6E | CALL 6E  |
| 2D | 904 | CMPI 0,4 |
| 2E | E34 | BRZ 34   |
| 2F | 9F4 | CMPI F,4 |

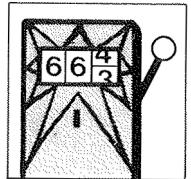
|    |     |          |
|----|-----|----------|
| 10 | C12 | GOTO 12  |
| 11 | 0E1 | MOV E,1  |
| 12 | B5B | CALL 5B  |
| 13 | 664 | SUB 6,4  |
| 14 | 675 | SUB 7,5  |
| 15 | 845 | CMP 4,5  |
| 16 | E18 | BRZ 18   |
| 17 | C1B | GOTO 1B  |
| 18 | B5B | CALL 5B  |
| 19 | B60 | CALL 60  |
| 1A | C06 | GOTO 06  |
| 1B | B5B | CALL 5B  |
| 1C | 516 | ADDI 1,6 |
| 1D | 864 | CMP 6,4  |
| 1E | D2B | BRC 2B   |
| 1F | B5B | CALL 5B  |

|    |     |          |
|----|-----|----------|
| 30 | E34 | BRZ 34   |
| 31 | 9E4 | CMPI E,4 |
| 32 | E34 | BRZ 34   |
| 33 | C36 | GOTO 36  |
| 34 | B5B | CALL 5B  |
| 35 | B72 | CALL 72  |
| 36 | 91F | CMPI 1,F |
| 37 | D3B | BRC 3B   |
| 38 | 042 | MOV 4,2  |
| 39 | 02E | MOV 2,E  |
| 3A | C3D | GOTO 3D  |
| 3B | 053 | MOV 5,3  |
| 3C | 03E | MOV 3,E  |
| 3D | 992 | CMPI 9,2 |
| 3E | E40 | BRZ 40   |
| 3F | C42 | GOTO 42  |

|    |     |          |
|----|-----|----------|
| 50 | 10A | MOVI 0,A |
| 51 | FEA | DOT A    |
| 52 | 51B | ADDI 1,B |
| 53 | D55 | BRC 55   |
| 54 | C4E | GOTO 4E  |
| 55 | 1CA | MOVI C,A |
| 56 | 1CB | MOVI C,B |
| 57 | 1CC | MOVI C,C |
| 58 | F3A | DISP 3,A |
| 59 | FF0 | KIN 0    |
| 5A | C00 | GOTO 00  |
| 5B | 004 | MOV 0,4  |
| 5C | 015 | MOV 1,5  |
| 5D | 026 | MOV 2,6  |
| 5E | 037 | MOV 3,7  |
| 5F | F07 | RET      |

|    |     |          |
|----|-----|----------|
| 60 | 11F | MOVI 1,F |
| 61 | 516 | ADDI 1,6 |
| 62 | 864 | CMP 6,4  |
| 63 | E67 | BRZ 67   |
| 64 | 062 | MOV 6,2  |
| 65 | 02E | MOV 2,E  |
| 66 | F07 | RET      |
| 67 | 726 | SUBI 2,6 |
| 68 | 062 | MOV 6,2  |
| 69 | 02E | MOV 2,E  |
| 6A | 924 | CMPI 2,4 |
| 6B | E6D | BRZ 6D   |
| 6C | F07 | RET      |
| 6D | C4E | GOTO 4E  |
| 6E | 675 | SUB 7,5  |
| 6F | 654 | SUB 5,4  |

# One-Armed Bandit

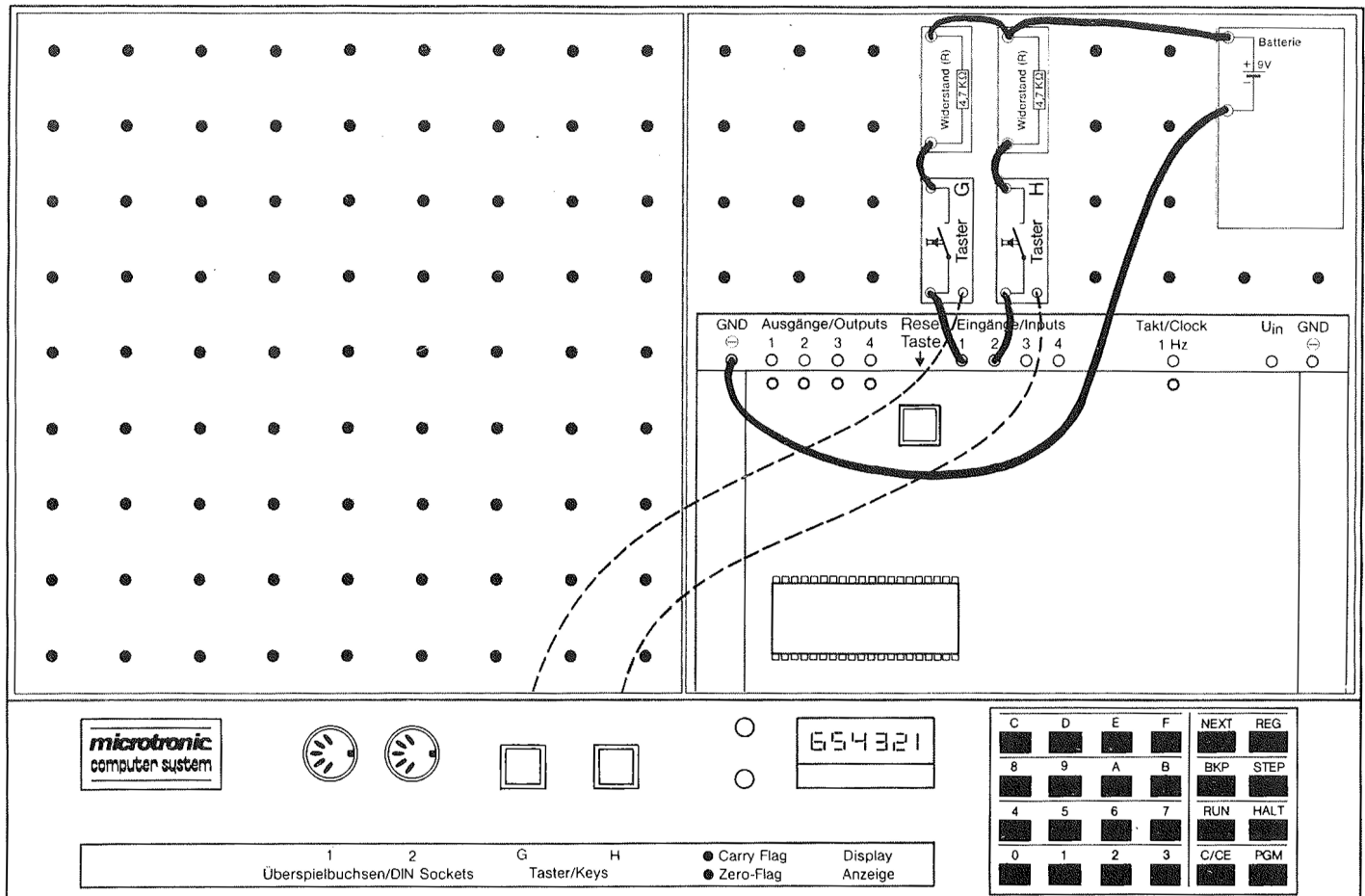


The program "One-Armed Bandit" is a game of chance for 2 players, similar to the well-known machine found in pubs and amusement arcades. The computer display shows three rows of digits which change rapidly. The individual rows can be stopped by pressing the appropriate keys. The aim is to stop the display on several identical digits, or on a rising or falling sequence.

## Function description:

The program is entered after HALT – NEXT – 00 according to the program table. Push-button G must then be connected to input 1 and push-button H to input 2 (see illustration).

Program start: HALT – NEXT – 00 – RUN. The display shows: 000000.



Now 6 arbitrary digits are entered, which the computer uses as a random starting value. After the 6th digit has been entered, the number 1 (or 2) appears briefly on the display. According to this, the first or the second player may begin.

Numbers and letters are now shown in very rapid succession at three positions of the display. Each of the three positions works independently and "counts" from 0 – 9, then from A – F, and begins again at 0. The player whose turn it is can stop this counting with the red push-buttons as follows:

Button G: the left display position is stopped.

Button H: the middle display position is stopped.

Buttons G + H together: the right position is stopped.

As soon as all 3 positions have been stopped, the computer works out the score achieved and displays it. The score is determined by the computer on the following basis:

For each F on the display there are 5 points.

For 2 identical digits there are 10 points.

For 3 identical digits there are 20 points.

For a rising or falling sequence (for example 4 5 6 or 4 3 2) there are 40 points.

After the score has been shown briefly, the display shows the number of the player whose turn is next (1 or 2). The new game begins automatically and the next player can try their luck as described.

## Ending the game:

If buttons G and H are pressed together while a player's score is being displayed, the game is ended. The computer shows the total score of player 1 and, after key 0 is pressed, the total score of player 2. Pressing key 0 once more restarts the game.

Programmer: Valentin Illich, 7750 Konstanz

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | F61    | DISP 6,1 |
| 03   | FF6    | KIN 6    |
| 04   | FF5    | KIN 5    |
| 05   | FF4    | KIN 4    |
| 06   | FF3    | KIN 3    |
| 07   | FF2    | KIN 2    |
| 08   | FF1    | KIN 1    |
| 09   | F02    | DISOUT   |
| 0A   | F05    | RND      |
| 0B   | 66E    | SUB 6,E  |
| 0C   | B1A    | CALL 1A  |
| 0D   | 65E    | SUB 5,E  |
| 0E   | B1A    | CALL 1A  |
| 0F   | 64E    | SUB 4,E  |

|    |     |          |
|----|-----|----------|
| 10 | B1A | CALL 1A  |
| 11 | 63E | SUB 3,E  |
| 12 | B1A | CALL 1A  |
| 13 | 62E | SUB 2,E  |
| 14 | B1A | CALL 1A  |
| 15 | 61E | SUB 1,E  |
| 16 | B1A | CALL 1A  |
| 17 | 78E | SUBI 8,E |
| 18 | B1A | CALL 1A  |
| 19 | COA | GOTO 0A  |
| 1A | 91E | CMPI 1,E |
| 1B | E1F | BRZ 1F   |
| 1C | 92E | CMPI 2,E |
| 1D | E1F | BRZ 1F   |
| 1E | F07 | RET      |
| 1F | 101 | MOVI 0,1 |

|    |     |          |
|----|-----|----------|
| 20 | 102 | MOVI 0,2 |
| 21 | 103 | MOVI 0,3 |
| 22 | 104 | MOVI 0,4 |
| 23 | 105 | MOVI 0,5 |
| 24 | 106 | MOVI 0,6 |
| 25 | 0E7 | MOV E,7  |
| 26 | F17 | DISP 1,7 |
| 27 | 100 | MOVI 0,0 |
| 28 | 510 | ADDI 1,0 |
| 29 | E2B | BRZ 2B   |
| 2A | C28 | GOTO 28  |
| 2B | F05 | RND      |
| 2C | F3D | DISP 3,D |
| 2D | 51D | ADDI 1,D |
| 2E | 51E | ADDI 1,E |
| 2F | 51F | ADDI 1,F |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | F00    | DIN 0    |
| 31   | 930    | CMPI 3,0 |
| 32   | E38    | BRZ 38   |
| 33   | 920    | CMPI 2,0 |
| 34   | E40    | BRZ 40   |
| 35   | 910    | CMPI 1,0 |
| 36   | E48    | BRZ 48   |
| 37   | C2D    | GOTO 2D  |
| 38   | 51E    | ADDI 1,E |
| 39   | 51F    | ADDI 1,F |
| 3A   | F00    | DIN 0    |
| 3B   | 920    | CMPI 2,0 |
| 3C   | E5A    | BRZ 5A   |
| 3D   | 910    | CMPI 1,0 |
| 3E   | E55    | BRZ 55   |
| 3F   | C38    | GOTO 38  |

|    |     |          |
|----|-----|----------|
| 40 | 51D | ADDI 1,D |
| 41 | 51F | ADDI 1,F |
| 42 | F00 | DIN 0    |
| 43 | 930 | CMPI 3,0 |
| 44 | E5A | BRZ 5A   |
| 45 | 910 | CMPI 1,0 |
| 46 | E50 | BRZ 50   |
| 47 | C40 | GOTO 40  |
| 48 | 51D | ADDI 1,D |
| 49 | 51E | ADDI 1,E |
| 4A | F00 | DIN 0    |
| 4B | 920 | CMPI 2,0 |
| 4C | E50 | BRZ 50   |
| 4D | 930 | CMPI 3,0 |
| 4E | E55 | BRZ 55   |
| 4F | C48 | GOTO 48  |

|    |     |          |
|----|-----|----------|
| 50 | 51D | ADDI 1,D |
| 51 | F00 | DIN 0    |
| 52 | 930 | CMPI 3,0 |
| 53 | E60 | BRZ 60   |
| 54 | C50 | GOTO 50  |
| 55 | 51E | ADDI 1,E |
| 56 | F00 | DIN 0    |
| 57 | 920 | CMPI 2,0 |
| 58 | E60 | BRZ 60   |
| 59 | C55 | GOTO 55  |
| 5A | 51F | ADDI 1,F |
| 5B | F00 | DIN 0    |
| 5C | 910 | CMPI 1,0 |
| 5D | E60 | BRZ 60   |
| 5E | C5A | GOTO 5A  |
| 5F | F01 | NOP      |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 1DB    | MOVI D,B |
| 61   | 51A    | ADDI 1,A |
| 62   | FBB    | ADC B    |
| 63   | D65    | BRC 65   |
| 64   | C61    | GOTO 61  |
| 65   | F02    | DISOUT   |
| 66   | 108    | MOVI 0,8 |
| 67   | 109    | MOVI 0,9 |
| 68   | 9FD    | CMPI F,D |
| 69   | E6B    | BRZ 6B   |
| 6A   | C6C    | GOTO 6C  |
| 6B   | BC0    | CALL C0  |
| 6C   | 9FE    | CMPI F,E |
| 6D   | E6F    | BRZ 6F   |
| 6E   | C70    | GOTO 70  |
| 6F   | BC0    | CALL C0  |

|    |     |          |
|----|-----|----------|
| 70 | 9FF | CMPI F,F |
| 71 | E73 | BRZ 73   |
| 72 | C74 | GOTO 74  |
| 73 | BC0 | CALL C0  |
| 74 | 8DE | CMP D,E  |
| 75 | E77 | BRZ 77   |
| 76 | C7A | GOTO 7A  |
| 77 | BC7 | CALL C7  |
| 78 | 8EF | CMP E,F  |
| 79 | EC9 | BRZ C9   |
| 7A | 8EF | CMP E,F  |
| 7B | E7D | BRZ 7D   |
| 7C | C80 | GOTO 80  |
| 7D | BC7 | CALL C7  |
| 7E | 8DF | CMP D,F  |
| 7F | EC9 | BRZ C9   |

|    |     |          |
|----|-----|----------|
| 80 | 8DF | CMP D,F  |
| 81 | E83 | BRZ 83   |
| 82 | C86 | GOTO 86  |
| 83 | BC7 | CALL C7  |
| 84 | 8DE | CMP D,E  |
| 85 | EC9 | BRZ C9   |
| 86 | 71D | SUBI 1,D |
| 87 | 51F | ADDI 1,F |
| 88 | 8ED | CMP E,D  |
| 89 | E8B | BRZ 8B   |
| 8A | C8F | GOTO 8F  |
| 8B | 8EF | CMP E,F  |
| 8C | E8E | BRZ 8E   |
| 8D | C8F | GOTO 8F  |
| 8E | BCB | CALL CB  |
| 8F | 52D | ADDI 2,D |

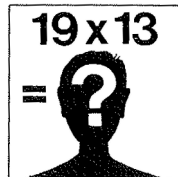
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | 72F    | SUBI 2,F |
| 91   | 8ED    | CMP E,D  |
| 92   | E94    | BRZ 94   |
| 93   | C98    | GOTO 98  |
| 94   | 8EF    | CMP E,F  |
| 95   | E97    | BRZ 97   |
| 96   | C98    | GOTO 98  |
| 97   | BCB    | CALL CB  |
| 98   | F28    | DISP 2,8 |
| 99   | 10F    | MOVI 0,F |
| 9A   | 08D    | MOV 8,D  |
| 9B   | 09E    | MOV 9,E  |
| 9C   | F04    | DZHX     |
| 9D   | 917    | CMPI 1,7 |
| 9E   | EA5    | BRZ A5   |
| 9F   | 4D4    | ADD D,4  |

|    |     |          |
|----|-----|----------|
| A0 | FB5 | ADC 5    |
| A1 | 4E5 | ADD E,5  |
| A2 | FB6 | ADC 6    |
| A3 | 117 | MOVI 1,7 |
| A4 | CAA | GOTO AA  |
| A5 | 4D1 | ADD D,1  |
| A6 | FB2 | ADC 2    |
| A7 | 4E2 | ADD E,2  |
| A8 | FB3 | ADC 3    |
| A9 | 127 | MOVI 2,7 |
| AA | 1EB | MOVI E,B |
| AB | 51A | ADDI 1,A |
| AC | FBB | ADC B    |
| AD | E26 | BRZ 26   |
| AE | F00 | DIN 0    |
| AF | 930 | CMPI 3,0 |

|    |     |          |
|----|-----|----------|
| B0 | EB2 | BRZ B2   |
| B1 | CAB | GOTO AB  |
| B2 | 01D | MOV 1,D  |
| B3 | 02E | MOV 2,E  |
| B4 | 03F | MOV 3,F  |
| B5 | F03 | HXDZ     |
| B6 | F3D | DISP 3,D |
| B7 | FF0 | KIN 0    |
| B8 | F02 | DISOUT   |
| B9 | 04D | MOV 4,D  |
| BA | 05E | MOV 5,E  |
| BB | 06F | MOV 6,F  |
| BC | F03 | HXDZ     |
| BD | F3D | DISP 3,D |
| BE | FF0 | KIN 0    |
| BF | C00 | GOTO 00  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| C0   | 558    | ADDI 5,8 |
| C1   | 998    | CMPI 9,8 |
| C2   | DC4    | BRC C4   |
| C3   | F07    | RET      |
| C4   | 568    | ADDI 6,8 |
| C5   | 519    | ADDI 1,9 |
| C6   | F07    | RET      |
| C7   | 519    | ADDI 1,9 |
| C8   | F07    | RET      |
| C9   | 519    | ADDI 1,9 |
| CA   | C98    | GOTO 98  |
| CB   | 549    | ADDI 4,9 |
| CC   | F07    | RET      |

# Arithmetic Trainer



This arithmetic trainer program lets you practise multiplication tables. The computer sets a sum. The result, worked out in your head, is entered. The computer checks whether the result is correct and then sets the next sum. If the entry is wrong, the computer shows the correct result.

## Function description:

After HALT – NEXT – 00 the program is entered according to the table. The green reset key must then be pressed briefly and the piezo buzzer connected to the GND and output 1 sockets.

Program start: HALT – NEXT – 00 – RUN. The display shows 00000. Two arbitrary numbers, which must be smaller than 20, are entered, for example 15 and 12. The display shows 15012. This is the first sum. The two left-hand positions of the display (15) must be multiplied by the two right-hand positions (12).  $15 \times 12 = 180$ .

The result (180) is entered, then key A is pressed. A short beep confirms that the entry was correct. The display automatically shows the next sum: 15013.  $15 \times 13$  is worked out and the result (195) entered, then key A pressed again.

If a wrong result is entered, the computer produces 2 short beeps. The message FE (error) appears on the display, then the correct result is shown. After a short time the next sum is set automatically.

To practise the whole multiplication table up to  $20 \times 20$ , enter 0101 after the program start (HALT – NEXT – 00 – RUN). Do not forget to press key A after every entry.

Programmer: Alexander Stadler, A-1160 Vienna

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | 10B    | MOVI 0,B |
| 02   | 10C    | MOVI 0,C |
| 03   | 10D    | MOVI 0,D |
| 04   | F56    | DISP 5,6 |
| 05   | FFE    | KIN E    |
| 06   | F02    | DISOUT   |
| 07   | 51F    | ADDI 1,F |
| 08   | 95F    | CMPI 5,F |
| 09   | E0F    | BRZ 0F   |
| 0A   | 09A    | MOV 9,A  |
| 0B   | 079    | MOV 7,9  |
| 0C   | 067    | MOV 6,7  |
| 0D   | 0E6    | MOV E,6  |
| 0E   | C04    | GOTO 04  |
| 0F   | 71F    | SUBI 1,F |

|    |     |          |
|----|-----|----------|
| 10 | 9AE | CMPI A,E |
| 11 | E17 | BRZ 17   |
| 12 | 0CD | MOV C,D  |
| 13 | 0BC | MOV B,C  |
| 14 | 0EB | MOV E,B  |
| 15 | F3B | DISP 3,B |
| 16 | C05 | GOTO 05  |
| 17 | 060 | MOV 6,0  |
| 18 | 071 | MOV 7,1  |
| 19 | F0D | EXRL     |
| 1A | 090 | MOV 9,0  |
| 1B | 0A1 | MOV A,1  |
| 1C | 102 | MOVI 0,2 |
| 1D | 103 | MOVI 0,3 |
| 1E | 104 | MOVI 0,4 |
| 1F | 105 | MOVI 0,5 |

|    |     |          |
|----|-----|----------|
| 20 | F0B | MULT     |
| 21 | 80B | CMP 0,B  |
| 22 | E24 | BRZ 24   |
| 23 | C44 | GOTO 44  |
| 24 | 81C | CMP 1,C  |
| 25 | E27 | BRZ 27   |
| 26 | C44 | GOTO 44  |
| 27 | 82D | CMP 2,D  |
| 28 | E2A | BRZ 2A   |
| 29 | C44 | GOTO 44  |
| 2A | 1FB | MOVI F,B |
| 2B | FEB | DOT B    |
| 2C | 10B | MOVI 0,B |
| 2D | FEB | DOT B    |
| 2E | F0D | EXRL     |
| 2F | 927 | CMPI 2,7 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | E32    | BRZ 32   |
| 31   | C3D    | GOTO 3D  |
| 32   | 92A    | CMPI 2,A |
| 33   | E00    | BRZ 00   |
| 34   | 107    | MOVI 0,7 |
| 35   | 116    | MOVI 1,6 |
| 36   | 519    | ADDI 1,9 |
| 37   | 9A9    | CMPI A,9 |
| 38   | E3A    | BRZ 3A   |
| 39   | C01    | GOTO 01  |
| 3A   | 569    | ADDI 6,9 |
| 3B   | FBA    | ADC A    |
| 3C   | C01    | GOTO 01  |
| 3D   | 516    | ADDI 1,6 |
| 3E   | 9A6    | CMPI A,6 |
| 3F   | E41    | BRZ 41   |

|    |     |          |
|----|-----|----------|
| 40 | C01 | GOTO 01  |
| 41 | 566 | ADDI 6,6 |
| 42 | FB7 | ADC 7    |
| 43 | C01 | GOTO 01  |
| 44 | 10C | MOVI 0,C |
| 45 | 1E3 | MOVI E,3 |
| 46 | 1F4 | MOVI F,4 |
| 47 | F50 | DISP 5,0 |
| 48 | 1FB | MOVI F,B |
| 49 | FEB | DOT B    |
| 4A | 10B | MOVI 0,B |
| 4B | FEB | DOT B    |
| 4C | 1FB | MOVI F,B |
| 4D | FEB | DOT B    |
| 4E | 10B | MOVI 0,B |
| 4F | FEB | DOT B    |

|    |     |          |
|----|-----|----------|
| 50 | 51C | ADDI 1,C |
| 51 | 9FC | CMPI F,C |
| 52 | E2E | BRZ 2E   |
| 53 | C50 | GOTO 50  |



## Number Guessing (High-Low Game)

350  
924  
76?

Number guessing is one of the classic computer games. MICROTRONIC uses its random number generator to pick a number between 0 and 999. The aim is to find this hidden random number in as few guesses as possible. After each attempt the computer indicates whether the guessed number is larger or smaller than the computer's number. With logical thinking it is possible to find it in 5 to 6 guesses.

### Function description:

After HALT – NEXT – 00 the program is entered according to the table.

Program start: HALT – NEXT – 00 – RUN. The display stays dark for a short time while the computer picks a random number between 0 and 999. It then shows: 000.

### Example of a game in progress:

Suppose 368 is the random number picked by the computer. For a first guess the number 560 is entered, then key A is pressed. The display shows: 560 000. The three left-hand positions show the number that is larger than the number sought; the three right-hand positions show the number that is smaller.

For a second guess the number 275 is entered, say, pressing key A after every entry. The display now shows: 560 275. The number sought must therefore lie between 560 and 275. If 355 is entered on the next attempt, the display shows 560 355.

With further guesses you feel your way closer and closer to the number the computer is keeping hidden. When the number is found, the display in our example shows E368E. Pressing key A again shows the number of guesses. Pressing key A once more starts the next game.

If key C is pressed while a 6-digit display is showing, the display shows the number of guesses made so far. Pressing key C again brings back the 6-digit display and guessing can continue.

If the number cannot be found, the game can be broken off at any time by pressing key B. The display then shows BXXXB (XXX = the random number picked by the computer). Pressing key A shows the number of guesses made so far. Pressing key A again starts a new game.

Programmer: Markus Jouaux, 6970 Lauda

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F02    | DISOUT   |
| 02   | F05    | RND      |
| 03   | 56D    | ADDI 6,D |
| 04   | 99D    | CMPI 9,D |
| 05   | D03    | BRC 03   |
| 06   | 56E    | ADDI 6,E |
| 07   | 99E    | CMPI 9,E |
| 08   | D06    | BRC 06   |
| 09   | 56F    | ADDI 6,F |
| 0A   | 99F    | CMPI 9,F |
| 0B   | D09    | BRC 09   |
| 0C   | 107    | MOVI 0,7 |
| 0D   | 108    | MOVI 0,8 |
| 0E   | 109    | MOVI 0,9 |
| 0F   | 19A    | MOVI 9,A |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 10   | 19B    | MOVI 9,B |
| 11   | 19C    | MOVI 9,C |
| 12   | F30    | DISP 3,0 |
| 13   | FF6    | KIN 6    |
| 14   | 9A6    | CMPI A,6 |
| 15   | E1E    | BRZ 1E   |
| 16   | 9B6    | CMPI B,6 |
| 17   | E5E    | BRZ 5E   |
| 18   | 9C6    | CMPI C,6 |
| 19   | E61    | BRZ 61   |
| 1A   | 012    | MOV 1,2  |
| 1B   | 001    | MOV 0,1  |
| 1C   | 060    | MOV 6,0  |
| 1D   | C12    | GOTO 12  |
| 1E   | 513    | ADDI 1,3 |
| 1F   | 9A3    | CMPI A,3 |

## Winning strategy for BLOCKADE

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | E22    | BRZ 22   |
| 21   | C2B    | GOTO 2B  |
| 22   | 563    | ADDI 6,3 |
| 23   | FB4    | ADC 4    |
| 24   | 9A4    | CMPI A,4 |
| 25   | E27    | BRZ 27   |
| 26   | C2B    | GOTO 2B  |
| 27   | 564    | ADDI 6,4 |
| 28   | FB5    | ADC 5    |
| 29   | 9A5    | CMPI A,5 |
| 2A   | E5E    | BRZ 5E   |
| 2B   | 829    | CMP 2,9  |
| 2C   | D5B    | BRC 5B   |
| 2D   | E2F    | BRZ 2F   |
| 2E   | C36    | GOTO 36  |
| 2F   | 818    | CMP 1,8  |
| 30   | D5B    | BRC 5B   |
| 31   | E33    | BRZ 33   |
| 32   | C36    | GOTO 36  |
| 33   | 807    | CMP 0,7  |
| 34   | D5B    | BRC 5B   |
| 35   | E5B    | BRZ 5B   |
| 36   | 82C    | CMP 2,C  |
| 37   | D41    | BRC 41   |
| 38   | E3A    | BRZ 3A   |
| 39   | C5B    | GOTO 5B  |
| 3A   | 81B    | CMP 1,B  |
| 3B   | D41    | BRC 41   |
| 3C   | E3E    | BRZ 3E   |
| 3D   | C5B    | GOTO 5B  |
| 3E   | 80A    | CMP 0,A  |
| 3F   | D41    | BRC 41   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 40   | C5B    | GOTO 5B  |
| 41   | 82F    | CMP 2,F  |
| 42   | D58    | BRC 58   |
| 43   | E45    | BRZ 45   |
| 44   | C54    | GOTO 54  |
| 45   | 81E    | CMP 1,E  |
| 46   | D58    | BRC 58   |
| 47   | E49    | BRZ 49   |
| 48   | C54    | GOTO 54  |
| 49   | 80D    | CMP 0,D  |
| 4A   | D58    | BRC 58   |
| 4B   | E4D    | BRZ 4D   |
| 4C   | C54    | GOTO 54  |
| 4D   | 1EC    | MOVI E,C |
| 4E   | 1E0    | MOVI E,0 |
| 4F   | F5C    | DISP 5,C |
| 50   | FF6    | KIN 6    |
| 51   | F33    | DISP 3,3 |
| 52   | FF6    | KIN 6    |
| 53   | C00    | GOTO 00  |
| 54   | 00A    | MOV 0,A  |
| 55   | 01B    | MOV 1,B  |
| 56   | 02C    | MOV 2,C  |
| 57   | C5B    | GOTO 5B  |
| 58   | 007    | MOV 0,7  |
| 59   | 018    | MOV 1,8  |
| 5A   | 029    | MOV 2,9  |
| 5B   | F67    | DISP 6,7 |
| 5C   | FF6    | KIN 6    |
| 5D   | C14    | GOTO 14  |
| 5E   | 1BC    | MOVI B,C |
| 5F   | 1B0    | MOVI B,0 |
| 60   | C4F    | GOTO 4F  |
| 61   | F33    | DISP 3,3 |
| 62   | FF6    | KIN 6    |
| 63   | C5B    | GOTO 5B  |

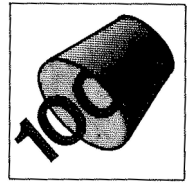
This section should only be read once the Blockade game has proved impossible to win despite repeated attempts.

Winning this game is actually quite simple: you must try to achieve the same number of empty squares between the computer's pieces and your own on both tracks.

After the program start, at the beginning of play, there are three empty squares between the pieces on track 2. With our first entry, 15, we achieve the necessary three empty squares between the pieces on track 1 as well. The computer now moves to square 2 on track 1, which means only two empty squares remain on track 1 while there are three on track 2. We can now either move our piece one square to the right on track 1, or one square to the left on track 2. In both cases the same number of empty squares between our pieces and the computer's is restored on both tracks.

The game can be continued in this way until victory. The important thing is that we consistently aim for the same number of empty squares between the pieces on both tracks.

# A Hundred Wins



A Hundred Wins is a positively fiendish puzzle game. A number between 1 and 10 is added in turn by the player and by the computer to a random number picked by the computer at the start of play. Whoever reaches 100 first has won.

## Function description:

After HALT – NEXT – 00 the program is entered carefully according to the table. The piezo buzzer is then connected to the output 1 and GND sockets. Then program start: HALT – NEXT – 00 – RUN.

The display shows: 0XXA00 (XX = the random starting number picked by the computer).

If the computer is to begin the game, the number 0 is entered. Entering zero is not allowed later in the game. If the player wants to add a number to the starting number, a number between 1 and 10 is entered. The entry is shown on the two right-hand positions of the display. Then press key A. The computer adds the entered number to the starting number and shows the result on the left-hand 3 display positions. The computer then carries on automatically and adds its own number to that result. After a short calculation a brief tone sounds and the display shows:

0XXCXX

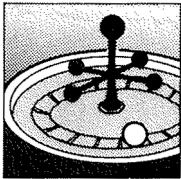
└─┐  
└─┐ → Number the computer has added to  
its last result.  
└─┐ → New total result.

Now the player enters a number again, confirming with key A after each entry. It is essential to note that the entered number must lie between 1 and 10; entering 0, or a number greater than 10, is not permitted.

The game ends as soon as the number 100 is reached. If the computer wins, the display shows: C100C. If the player wins, the display shows: E100E. A new game is started by pressing key 0.

| Adr.Befehl | Mnemonic |          | Adr.Befehl | Mnemonic |          | Adr.Befehl | Mnemonic |          | Adr.Befehl | Mnemonic |          |
|------------|----------|----------|------------|----------|----------|------------|----------|----------|------------|----------|----------|
| 00         | F02      | DISOUT   | 30         | B9E      | CALL 9E  | 60         | BA5      | CALL A5  | 90         | 915      | CMPI 1,5 |
| 01         | F08      | CLEAR    | 31         | B9E      | CALL 9E  | 61         | 1C2      | MOVI C,2 | 91         | E95      | BRZ 95   |
| 02         | FEO      | DOT 0    | 32         | C18      | GOTO 18  | 62         | B90      | CALL 90  | 92         | D97      | BRC 97   |
| 03         | F05      | RND      | 33         | BAE      | CALL AE  | 63         | 91A      | CMPI 1,A | 93         | 10A      | MOVI 0,A |
| 04         | 92E      | CMPI 2,E | 34         | 993      | CMPI 9,3 | 64         | E80      | BRZ 80   | 94         | F07      | RET      |
| 05         | D07      | BRC 07   | 35         | D37      | BRC 37   | 65         | 92A      | CMPI 2,A | 95         | 904      | CMPI 0,4 |
| 06         | C0B      | GOTO 0B  | 36         | C39      | GOTO 39  | 66         | E7D      | BRZ 7D   | 96         | E99      | BRZ 99   |
| 07         | 52F      | ADDI 2,F | 37         | 514      | ADDI 1,4 | 67         | C17      | GOTO 17  | 97         | 12A      | MOVI 2,A |
| 08         | 6FD      | SUB F,D  | 38         | 563      | ADDI 6,3 | 68         | 111      | MOVI 1,1 | 98         | F07      | RET      |
| 09         | FCE      | SUBC E   | 39         | 414      | ADD 1,4  | 69         | 100      | MOVI 0,0 | 99         | 903      | CMPI 0,3 |
| 0A         | C04      | GOTO 04  | 3A         | 994      | CMPI 9,4 | 6A         | C61      | GOTO 61  | 9A         | E9C      | BRZ 9C   |
| 0B         | 11B      | MOVI 1,B | 3B         | D3D      | BRC 3D   | 6B         | F05      | RND      | 9B         | C97      | GOTO 97  |
| 0C         | 1A2      | MOVI A,2 | 3C         | C3F      | GOTO 3F  | 6C         | 97D      | CMPI 7,D | 9C         | 11A      | MOVI 1,A |
| 0D         | 10F      | MOVI 0,F | 3D         | 515      | ADDI 1,5 | 6D         | BA8      | CALL A8  | 9D         | F07      | RET      |
| 0E         | F03      | HXDZ     | 3E         | 564      | ADDI 6,4 | 6E         | C71      | GOTO 71  | 9E         | 1F7      | MOVI F,7 |
| 0F         | B8B      | CALL 8B  | 3F         | B90      | CALL 90  | 6F         | 52D      | ADDI 2,D | 9F         | FE7      | DOT 7    |
|            |          |          |            |          |          |            |          |          |            |          |          |
| 10         | F60      | DISP 6,0 | 40         | 91A      | CMPI 1,A | 70         | C6C      | GOTO 6C  | A0         | 737      | SUBI 3,7 |
| 11         | B9E      | CALL 9E  | 41         | E7D      | BRZ 7D   | 71         | 4D3      | ADD D,3  | A1         | EA3      | BRZ A3   |
| 12         | FF6      | KIN 6    | 42         | 92A      | CMPI 2,A | 72         | 993      | CMPI 9,3 | A2         | CA0      | GOTO A0  |
| 13         | E44      | BRZ 44   | 43         | E80      | BRZ 80   | 73         | D75      | BRC 75   | A3         | FE7      | DOT 7    |
| 14         | 996      | CMPI 9,6 | 44         | 05F      | MOV 5,F  | 74         | C61      | GOTO 61  | A4         | F07      | RET      |
| 15         | D11      | BRC 11   | 45         | 04E      | MOV 4,E  | 75         | 563      | ADDI 6,3 | A5         | 080      | MOV 8,0  |
| 16         | C1B      | GOTO 1B  | 46         | 03D      | MOV 3,D  | 76         | 514      | ADDI 1,4 | A6         | 101      | MOVI 0,1 |
| 17         | B9E      | CALL 9E  | 47         | F04      | DZHX     | 77         | 994      | CMPI 9,4 | A7         | F07      | RET      |
| 18         | FF6      | KIN 6    | 48         | 8CE      | CMP C,E  | 78         | D7A      | BRC 7A   | A8         | D6F      | BRC 6F   |
| 19         | 996      | CMPI 9,6 | 49         | D4E      | BRC 4E   | 79         | C61      | GOTO 61  | A9         | 90D      | CMPI 0,D |
| 1A         | D17      | BRC 17   | 4A         | E4C      | BRZ 4C   | 7A         | 564      | ADDI 6,4 | AA         | E6F      | BRZ 6F   |
| 1B         | 100      | MOVI 0,0 | 4B         | C51      | GOTO 51  | 7B         | 515      | ADDI 1,5 | AB         | 0D0      | MOV D,0  |
| 1C         | 101      | MOVI 0,1 | 4C         | 8DB      | CMP D,B  | 7C         | C61      | GOTO 61  | AC         | 101      | MOVI 0,1 |
| 1D         | 1A2      | MOVI A,2 | 4D         | D51      | BRC 51   | 7D         | 1E6      | MOVI E,6 | AD         | F07      | RET      |
| 1E         | 060      | MOV 6,0  | 4E         | 5BB      | ADDI B,B | 7E         | 1E2      | MOVI E,2 | AE         | 403      | ADD 0,3  |
| 1F         | FF6      | KIN 6    | 4F         | FBC      | ADC C    | 7F         | C82      | GOTO 82  | AF         | D37      | BRC 37   |
|            |          |          |            |          |          |            |          |          |            |          |          |
| 20         | 996      | CMPI 9,6 | 50         | C48      | GOTO 48  | 80         | 1C6      | MOVI C,6 | B0 F07 RET |          |          |
| 21         | D25      | BRC 25   | 51         | 0B8      | MOV B,8  | 81         | 1C2      | MOVI C,2 |            |          |          |
| 22         | 001      | MOV 0,1  | 52         | 0C9      | MOV C,9  | 82         | F52      | DISP 5,2 |            |          |          |
| 23         | 060      | MOV 6,0  | 53         | 6D8      | SUB D,8  | 83         | B9E      | CALL 9E  |            |          |          |
| 24         | C1F      | GOTO 1F  | 54         | FC9      | SUBC 9   | 84         | B9E      | CALL 9E  |            |          |          |
| 25         | 911      | CMPI 1,1 | 55         | 6E9      | SUB E,9  | 85         | B9E      | CALL 9E  |            |          |          |
| 26         | D2D      | BRC 2D   | 56         | 909      | CMPI 0,9 | 86         | B9E      | CALL 9E  |            |          |          |
| 27         | E2B      | BRZ 2B   | 57         | D6B      | BRC 6B   | 87         | B9E      | CALL 9E  |            |          |          |
| 28         | 900      | CMPI 0,0 | 58         | 9A8      | CMPI A,8 | 88         | FF0      | KIN 0    |            |          |          |
| 29         | E2D      | BRZ 2D   | 59         | D6B      | BRC 6B   | 89         | C00      | GOTO 00  |            |          |          |
| 2A         | C33      | GOTO 33  | 5A         | 0BD      | MOV B,D  | 8A         | F03      | HXDZ     |            |          |          |
| 2B         | 900      | CMPI 0,0 | 5B         | 0CE      | MOV C,E  | 8B         | 0D3      | MOV D,3  |            |          |          |
| 2C         | E33      | BRZ 33   | 5C         | 10F      | MOVI 0,F | 8C         | 0E4      | MOV E,4  |            |          |          |
| 2D         | 100      | MOVI 0,0 | 5D         | B8A      | CALL 8A  | 8D         | 0F5      | MOV F,5  |            |          |          |
| 2E         | 101      | MOVI 0,1 | 5E         | 998      | CMPI 9,8 | 8E         | F07      | RET      |            |          |          |
| 2F         | B9E      | CALL 9E  | 5F         | D68      | BRC 68   | 8F         | F01      | NOP      |            |          |          |

# MICROTRONIC Roulette



As in the well-known game of roulette, you can bet on colours (red or black), on even or odd numbers, or on a particular number. Unlike ordinary roulette, MICROTRONIC roulette works with the numbers 0 – 256. This gives considerably more possibilities: you can also bet on the units, tens or hundreds digit of a number, for example. MICROTRONIC additionally acts as the bank, keeping the player's account automatically, debiting losses and crediting winnings to the starting capital. A starting capital of 10,000 play marks is available, to be increased by skilful betting.

## Function description:

After HALT – NEXT – 00 the program is entered according to the table. The piezo buzzer is then connected to the output 1 and GND sockets.

Program start: HALT – NEXT – 00 – RUN.

The display goes dark briefly. The computer uses its random generator to pick a winning number between 0 and 256. At the same time it decides whether the number is red or black, and even or odd. The winning number is of course not revealed until the stake has been entered. For this the display shows: 0000. You now enter how many play marks are being staked, for example 200 (entry 200; maximum stake 9999). Then press key A. The display shows: A. One of six different betting options must be chosen and the corresponding key pressed according to the following table:

| Betting option       | Key | further key press                                     | Winnings         |
|----------------------|-----|-------------------------------------------------------|------------------|
| Red/Black            | 1   | Key 0 for red<br>Key 1 for black                      | Stake is doubled |
| Even/Odd             | 2   | Key 1 for even number<br>Key 2 for odd number         | Stake is doubled |
| Directly on a number | 3   | Enter the winning number; then key A                  | Stake × 300      |
| Hundreds digit       | 4   | Enter the 1st digit of the winning number (0, 1 or 2) | Stake is tripled |
| Tens digit           | 5   | Enter the 2nd digit of the winning number (0 to 9)    | Stake × 10       |
| Units digit          | 6   | Enter the 3rd digit of the winning number (0 to 9)    | Stake × 5        |

Example:

Programmer: Frank Simon, 6342 Haiger 6

| Adr.  | Befehl | Mnemonic |
|-------|--------|----------|
| 00    | F02    | DISOUT   |
| 01    | F08    | CLEAR    |
| 02    | FE0    | DOT 0    |
| 03    | 11C    | MOVI 1,0 |
| 04    | FOE    | EXRM     |
| 05    | F0D    | EXRL     |
| 06    | F08    | CLEAR    |
| 07    | F05    | RND      |
| 08    | 93F    | CMPI 3,F |
| 09    | DOB    | BRC 0B   |
| 0A    | C0C    | GOTO 0C  |
| 0B    | 11C    | MOVI 1,0 |
| 0C    | 10F    | MOVI 0,F |
| 0D    | F03    | HXDZ     |
| 0E    | 0F2    | MOV F,2  |
| 0F    | 0E1    | MOV E,1  |
| <hr/> |        |          |
| 10    | 0D0    | MOV D,0  |
| 11    | F0D    | EXRL     |
| 12    | 120    | MOVI 2,0 |
| 13    | F0C    | DIV      |
| 14    | E16    | BRZ 16   |
| 15    | 11B    | MOVI 1,B |
| 16    | 100    | MOVI 0,0 |
| 17    | 101    | MOVI 0,1 |
| 18    | 102    | MOVI 0,2 |
| 19    | F0D    | EXRL     |
| 1A    | 100    | MOVI 0,0 |
| 1B    | 101    | MOVI 0,1 |
| 1C    | 102    | MOVI 0,2 |
| 1D    | F40    | DISP 4,0 |
| 1E    | FFA    | KIN A    |
| 1F    | 9AA    | CMPI A,A |
| <hr/> |        |          |
| 20    | E26    | BRZ 26   |
| 21    | 023    | MOV 2,3  |
| 22    | 012    | MOV 1,2  |
| 23    | 001    | MOV 0,1  |
| 24    | 0A0    | MOV A,0  |
| 25    | C1D    | GOTO 1D  |
| 26    | F0D    | EXRL     |
| 27    | 10A    | MOVI 0,A |
| 28    | 1A8    | MOVI A,8 |
| 29    | F18    | DISP 1,8 |
| 2A    | FF8    | KIN 8    |
| 2B    | 918    | CMPI 1,8 |
| 2C    | E38    | BRZ 38   |
| 2D    | 928    | CMPI 2,8 |
| 2E    | E3C    | BRZ 3C   |
| 2F    | 938    | CMPI 3,8 |

| Adr.  | Befehl | Mnemonic |     |
|-------|--------|----------|-----|
| 30    | E48    | BRZ      | 48  |
| 31    | 948    | CMPI     | 4,8 |
| 32    | E5D    | BRZ      | 5D  |
| 33    | 958    | CMPI     | 5,8 |
| 34    | E63    | BRZ      | 63  |
| 35    | 968    | CMPI     | 6,8 |
| 36    | E69    | BRZ      | 69  |
| 37    | C29    | GOTO     | 29  |
| 38    | B40    | CALL     | 40  |
| 39    | 89C    | CMP      | 9,C |
| 3A    | B44    | CALL     | 44  |
| 3B    | C6E    | GOTO     | 6E  |
| 3C    | B40    | CALL     | 40  |
| 3D    | 89B    | CMP      | 9,B |
| 3E    | B44    | CALL     | 44  |
| 3F    | C6E    | GOTO     | 6E  |
| <hr/> |        |          |     |
| 40    | F19    | DISP     | 1,9 |
| 41    | FF9    | KIN      | 9   |
| 42    | F02    | DISOUT   |     |
| 43    | F07    | RET      |     |
| 44    | E46    | BRZ      | 46  |
| 45    | C47    | GOTO     | 47  |
| 46    | 120    | MOVI     | 2,0 |
| 47    | F07    | RET      |     |
| 48    | 10B    | MOVI     | 0,B |
| 49    | F39    | DISP     | 3,9 |
| 4A    | FF8    | KIN      | 8   |
| 4B    | 9A8    | CMPI     | A,8 |
| 4C    | E51    | BRZ      | 51  |
| 4D    | 0AB    | MOV      | A,B |
| 4E    | 09A    | MOV      | 9,A |
| 4F    | 089    | MOV      | 8,9 |
| <hr/> |        |          |     |
| 50    | C49    | GOTO     | 49  |
| 51    | F02    | DISOUT   |     |
| 52    | 8FB    | CMP      | F,B |
| 53    | E55    | BRZ      | 55  |
| 54    | C6E    | GOTO     | 6E  |
| 55    | 8EA    | CMP      | E,A |
| 56    | E58    | BRZ      | 58  |
| 57    | C6E    | GOTO     | 6E  |
| 58    | 8D9    | CMP      | D,9 |
| 59    | E5B    | BRZ      | 5B  |
| 5A    | C6E    | GOTO     | 6E  |
| 5B    | 132    | MOVI     | 3,2 |
| 5C    | C6E    | GOTO     | 6E  |
| 5D    | B40    | CALL     | 40  |
| 5E    | 89F    | CMP      | 9,F |
| 5F    | E61    | BRZ      | 61  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | C6E    | GOTO 6E  |
| 61   | 130    | MOVI 3,0 |
| 62   | C6E    | GOTO 6E  |
| 63   | B40    | CALL 40  |
| 64   | 89E    | CMP 9,E  |
| 65   | E67    | BRZ 67   |
| 66   | C6E    | GOTO 6E  |
| 67   | 111    | MOVI 1,1 |
| 68   | C6E    | GOTO 6E  |
| 69   | B40    | CALL 40  |
| 6A   | 89D    | CMP 9,D  |
| 6B   | E6D    | BRZ 6D   |
| 6C   | C6E    | GOTO 6E  |
| 6D   | 150    | MOVI 5,0 |
| 6E   | F5B    | DISP 5,B |
| 6F   | FFF    | KIN F    |

|    |     |          |
|----|-----|----------|
| 70 | F02 | DISOUT   |
| 71 | F0D | EXRL     |
| 72 | F0E | EXRM     |
| 73 | BB9 | CALL B9  |
| 74 | F0B | MULT     |
| 75 | F50 | DISP 5,0 |
| 76 | FFF | KIN F    |
| 77 | B8D | CALL 8D  |
| 78 | F58 | DISP 5,8 |
| 79 | FFF | KIN F    |
| 7A | F02 | DISOUT   |
| 7B | 908 | CMPI 0,8 |
| 7C | E7E | BRZ 7E   |
| 7D | C04 | GOTO 04  |
| 7E | 909 | CMPI 0,9 |
| 7F | E81 | BRZ 81   |

|    |     |          |
|----|-----|----------|
| 80 | C04 | GOTO 04  |
| 81 | 90A | CMPI 0,A |
| 82 | E84 | BRZ 84   |
| 83 | C04 | GOTO 04  |
| 84 | 90B | CMPI 0,B |
| 85 | E87 | BRZ 87   |
| 86 | C04 | GOTO 04  |
| 87 | 90C | CMPI 0,C |
| 88 | E8A | BRZ 8A   |
| 89 | C04 | GOTO 04  |
| 8A | 1F0 | MOVI F,0 |
| 8B | FE0 | DOT 0    |
| 8C | CDA | GOTO DA  |
| 8D | 408 | ADD 0,8  |
| 8E | D92 | BRC 92   |
| 8F | 998 | CMPI 9,8 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | D92    | BRC 92   |
| 91   | C94    | GOTO 94  |
| 92   | 568    | ADDI 6,8 |
| 93   | 519    | ADDI 1,9 |
| 94   | 419    | ADD 1,9  |
| 95   | D99    | BRC 99   |
| 96   | 999    | CMPI 9,9 |
| 97   | D99    | BRC 99   |
| 98   | C9B    | GOTO 9B  |
| 99   | 569    | ADDI 6,9 |
| 9A   | 51A    | ADDI 1,A |
| 9B   | 42A    | ADD 2,A  |
| 9C   | DA0    | BRC A0   |
| 9D   | 99A    | CMPI 9,A |
| 9E   | DA0    | BRC A0   |
| 9F   | CA2    | GOTO A2  |

|    |     |          |
|----|-----|----------|
| A0 | 56A | ADDI 6,A |
| A1 | 51B | ADDI 1,B |
| A2 | 43B | ADD 3,B  |
| A3 | DA7 | BRC A7   |
| A4 | 99B | CMPI 9,B |
| A5 | DA7 | BRC A7   |
| A6 | CA9 | GOTO A9  |
| A7 | 56B | ADDI 6,B |
| A8 | 51C | ADDI 1,C |
| A9 | 44C | ADD 4,C  |
| AA | DAE | BRC AE   |
| AB | 99C | CMPI 9,C |
| AC | DAE | BRC AE   |
| AD | F07 | RET      |
| AE | 56C | ADDI 6,C |
| AF | 11D | MOVI 1,D |

|    |     |          |
|----|-----|----------|
| B0 | 1E0 | MOVI E,0 |
| B1 | 1E1 | MOVI E,1 |
| B2 | 1E2 | MOVI E,2 |
| B3 | 1E3 | MOVI E,3 |
| B4 | 1E4 | MOVI E,4 |
| B5 | 1E5 | MOVI E,5 |
| B6 | F60 | DISP 6,0 |
| B7 | FFF | KIN F    |
| B8 | C00 | GOTO 00  |
| B9 | 608 | SUB 0,8  |
| BA | DBC | BRC BC   |
| BB | CBF | GOTO BF  |
| BC | 768 | SUBI 6,8 |
| BD | 719 | SUBI 1,9 |
| BE | DC2 | BRC C2   |
| BF | 619 | SUB 1,9  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| C0   | DC3    | BRC C3   |
| C1   | CC6    | GOTO C6  |
| C2   | 619    | SUB 1,9  |
| C3   | 769    | SUBI 6,9 |
| C4   | 71A    | SUBI 1,A |
| C5   | DC9    | BRC C9   |
| C6   | 62A    | SUB 2,A  |
| C7   | DCA    | BRC CA   |
| C8   | CCD    | GOTO CD  |
| C9   | 62A    | SUB 2,A  |
| CA   | 76A    | SUBI 6,A |
| CB   | 71B    | SUBI 1,B |
| CC   | DD0    | BRC D0   |
| CD   | 63B    | SUB 3,B  |
| CE   | DD1    | BRC D1   |
| CF   | CD4    | GOTO D4  |

|    |     |          |
|----|-----|----------|
| D0 | 63B | SUB 3,B  |
| D1 | 76B | SUBI 6,B |
| D2 | 71C | SUBI 1,C |
| D3 | DD7 | BRC D7   |
| D4 | 64C | SUB 4,C  |
| D5 | DD8 | BRC D8   |
| D6 | F07 | RET      |
| D7 | 64C | SUB 4,C  |
| D8 | 76C | SUBI 6,C |
| D9 | F07 | RET      |
| DA | 1ED | MOVI E,D |
| DB | F68 | DISP 6,8 |
| DC | FF0 | KIN 0    |
| DD | C00 | GOTO 00  |

# Sound Memory



Sound Memory (known in English-speaking countries as “Simon”, and in Germany as “SENSO”) is an interesting game of skill that demands a great deal of attention and concentration. It works in much the same way as the game “Number Avalanche”. Instead of numbers and letters the computer plays tones, which increases the difficulty considerably. A BUSCH Electronic Studio no. 2060 (or 2065 or 2070) is required for the melody-playing circuit.

## Function description:

After pressing HALT – NEXT – 00 the program is entered according to the program table. Then press the green reset button (computer board), which switches off the 4 outputs. Now the electronic circuit “mini organ” still has to be built as shown in the illustration on page 20. Owners of an Electronic Studio 2060 or 2065 build the circuit directly in the computer housing. Owners of the larger Electronic Studio 2070 build the circuit in the studio housing and run the appropriate connecting cables to computer outputs no. 1 – 4 (do not swap the cables).

Program start: HALT – NEXT – 00 – RUN

The computer plays 4 tones one after another and shows on the display the number belonging to each tone, that is, the number 4 corresponds to the lowest tone and the number 7 to the highest tone. The display goes out – after a short pause the computer plays a tone. The key belonging to that tone (4 = lowest tone, 7 = highest tone) must now be pressed. If the correct key is pressed, the computer plays a further tone. Now the corresponding keys for both tones must be pressed in the correct order. At each key press the computer repeats the corresponding tone for about one second. It is essential to note that the following key presses must not be made until the computer has finished playing the tone entered before.

The game continues until a mistake is made during entry. The computer then plays a descending sequence of notes and shows on the display how many attempts were correct. After that the computer automatically plays the four basic tones again – a new game begins.

If you succeed in entering 12 tones in the correct order, the game is ended, which the computer signals with an ascending sequence of notes.

Although the computer plays only four different tones (4 to 7), it is not easy to repeat them in the correct order. With several players, the winner is the one who achieves the highest entry score over the agreed number of rounds.

## Increasing the difficulty

The difficulty of the game can be increased by reprogramming, extending the sequence from 4 to 8 tones. The following entries are required for the program change (at addresses 07 and 93):

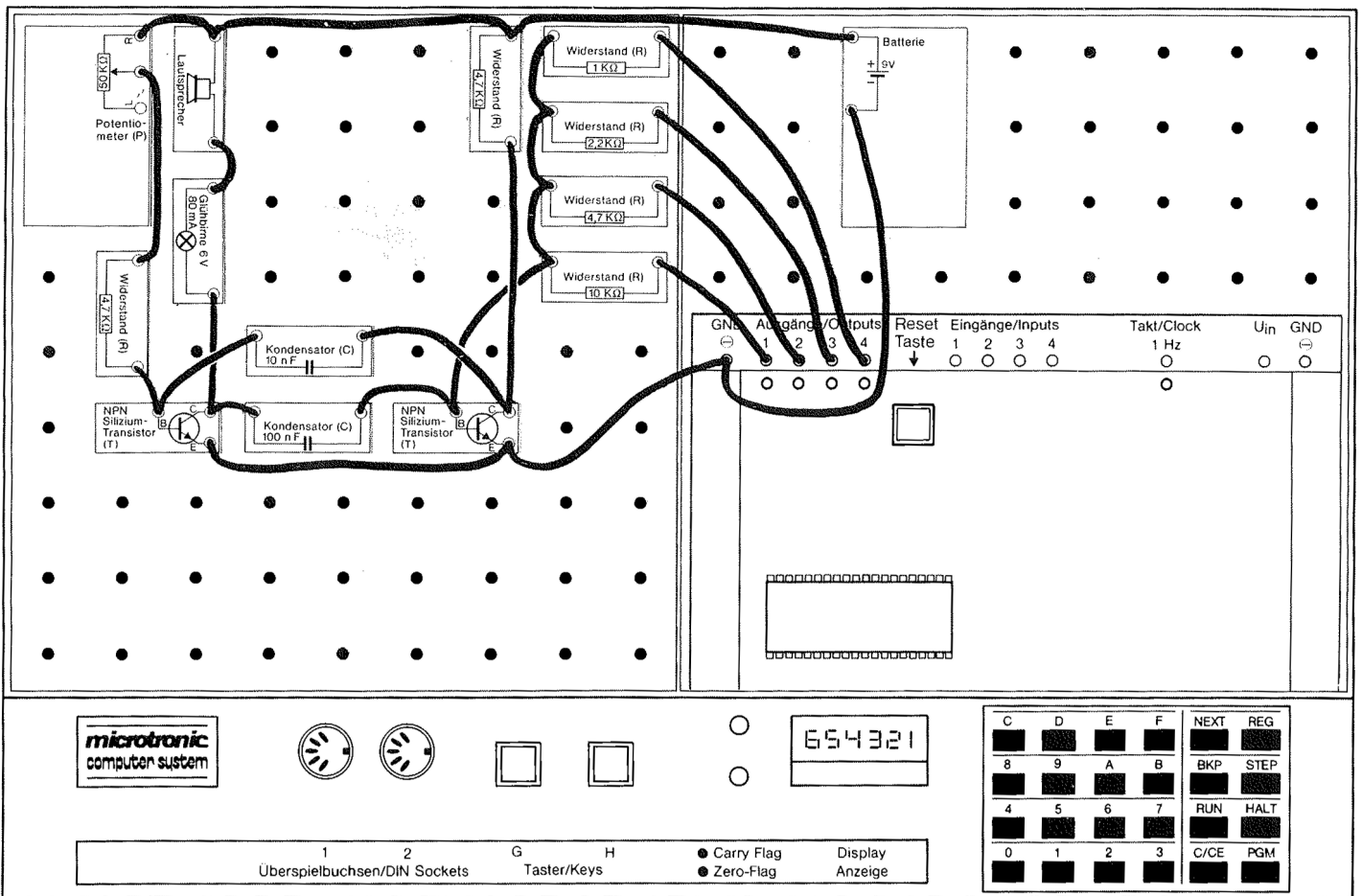
HALT – NEXT – 07 9B0 – NEXT  
HALT – NEXT – 93 9BD – NEXT

New program start: HALT – NEXT – 00 – RUN

The Sound Memory now works with eight different tones (the lowest tone is still key 4, the highest tone is now key B). Play follows the description above, except that the computer now plays eight tones (4 – 9 and A), which have to be repeated with the corresponding key presses.

Our Sound Memory has now become very difficult. To get used to it, it may be an advantage if the computer shows the tones not only acoustically but also visually on the display.





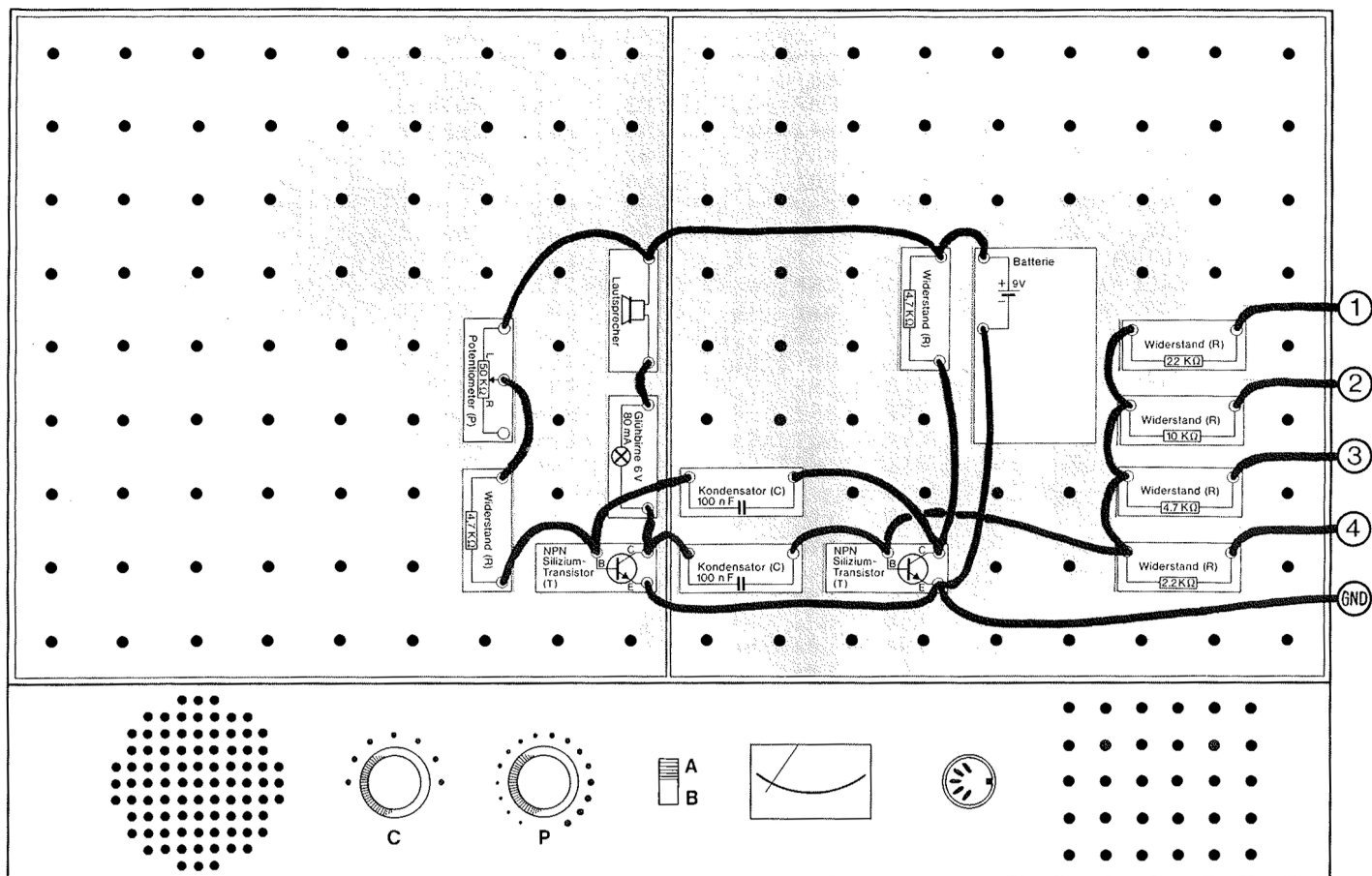
(Fig. El. Studio 2060/65 built into the MICROTRONIC housing)

To do this the instruction at address 10 has to be changed (F02 to F1D).

Entry: HALT – NEXT – 10 F1D – NEXT.

New program start: HALT – NEXT – 00 – RUN.

In addition to playing the tone, the display now also shows the corresponding key to be pressed.



(Fig. El. Studio 2070)

The connecting leads (1–4 and GND) run to the corresponding computer output sockets (1–4 and GND). The connections must not be swapped.

Since both the Studio Center 2070 and the computer have transfer sockets, the leads intended for the computer outputs can first be taken to the transfer-cable module. In that case the transfer-cable module is used on the computer as well, and the connections to the computer outputs are made from there. Care must then be taken to use the same numbering for the connections on both transfer-cable modules. The two units can now be linked to each other with a standard transfer cable.

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | 140    | MOVI 4,0 |
| 03   | F10    | DISP 1,0 |
| 04   | FE0    | DOT 0    |
| 05   | B0A    | CALL 0A  |
| 06   | 510    | ADDI 1,0 |
| 07   | 970    | CMPI 7,0 |
| 08   | D10    | BRC 10   |
| 09   | C04    | GOTO 04  |
| 0A   | 1FF    | MOVI F,F |
| 0B   | 71F    | SUBI 1,F |
| 0C   | E0E    | BRZ 0E   |
| 0D   | C0B    | GOTO 0B  |
| 0E   | FEF    | DOT F    |
| 0F   | F07    | RET      |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 10   | F02    | DISOUT   |
| 11   | B0A    | CALL 0A  |
| 12   | F08    | CLEAR    |
| 13   | B92    | CALL 92  |
| 14   | 0D2    | MOV D,2  |
| 15   | FE2    | DOT 2    |
| 16   | B39    | CALL 39  |
| 17   | 0D3    | MOV D,3  |
| 18   | FE3    | DOT 3    |
| 19   | B39    | CALL 39  |
| 1A   | 0D4    | MOV D,4  |
| 1B   | FE4    | DOT 4    |
| 1C   | B39    | CALL 39  |
| 1D   | 0D5    | MOV D,5  |
| 1E   | FE5    | DOT 5    |
| 1F   | B39    | CALL 39  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | 0D6    | MOV D,6  |
| 21   | FE6    | DOT 6    |
| 22   | B39    | CALL 39  |
| 23   | 0D7    | MOV D,7  |
| 24   | FE7    | DOT 7    |
| 25   | B39    | CALL 39  |
| 26   | 0D8    | MOV D,8  |
| 27   | FE8    | DOT 8    |
| 28   | B39    | CALL 39  |
| 29   | 0D9    | MOV D,9  |
| 2A   | FE9    | DOT 9    |
| 2B   | B39    | CALL 39  |
| 2C   | 0DA    | MOV D,A  |
| 2D   | FEA    | DOT A    |
| 2E   | B39    | CALL 39  |
| 2F   | 0DB    | MOV D,B  |

|    |     |          |
|----|-----|----------|
| 30 | FEB | DOT B    |
| 31 | B39 | CALL 39  |
| 32 | 0DC | MOV D,C  |
| 33 | FEC | DOT C    |
| 34 | B39 | CALL 39  |
| 35 | FED | DOT D    |
| 36 | B39 | CALL 39  |
| 37 | BA8 | CALL A8  |
| 38 | C00 | GOTO 00  |
| 39 | 511 | ADDI 1,1 |
| 3A | 1FF | MOVI F,F |
| 3B | 71F | SUBI 1,F |
| 3C | E3E | BRZ 3E   |
| 3D | C3B | GOTO 3B  |
| 3E | FEF | DOT F    |
| 3F | 11E | MOVI 1,E |

|    |     |          |
|----|-----|----------|
| 40 | CB1 | GOTO B1  |
| 41 | 802 | CMP 0,2  |
| 42 | E44 | BRZ 44   |
| 43 | C9A | GOTO 9A  |
| 44 | 911 | CMPI 1,1 |
| 45 | E90 | BRZ 90   |
| 46 | 12E | MOVI 2,E |
| 47 | CB1 | GOTO B1  |
| 48 | 803 | CMP 0,3  |
| 49 | E4B | BRZ 4B   |
| 4A | C9A | GOTO 9A  |
| 4B | 921 | CMPI 2,1 |
| 4C | E90 | BRZ 90   |
| 4D | 13E | MOVI 3,E |
| 4E | CB1 | GOTO B1  |
| 4F | E51 | BRZ 51   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 50   | C9A    | GOTO 9A  |
| 51   | 931    | CMPI 3,1 |
| 52   | E90    | BRZ 90   |
| 53   | 14E    | MOVI 4,E |
| 54   | CB1    | GOTO B1  |
| 55   | 805    | CMP 0,5  |
| 56   | E58    | BRZ 58   |
| 57   | C9A    | GOTO 9A  |
| 58   | 941    | CMPI 4,1 |
| 59   | E90    | BRZ 90   |
| 5A   | 15E    | MOVI 5,E |
| 5B   | CB1    | GOTO B1  |
| 5C   | 806    | CMP 0,6  |
| 5D   | E5F    | BRZ 5F   |
| 5E   | C9A    | GOTO 9A  |
| 5F   | 951    | CMPI 5,1 |

|    |     |          |
|----|-----|----------|
| 60 | E90 | BRZ 90   |
| 61 | 16E | MOVI 6,E |
| 62 | CB1 | GOTO B1  |
| 63 | 807 | CMP 0,7  |
| 64 | E66 | BRZ 66   |
| 65 | C9A | GOTO 9A  |
| 66 | 961 | CMPI 6,1 |
| 67 | E90 | BRZ 90   |
| 68 | 17E | MOVI 7,E |
| 69 | CB1 | GOTO B1  |
| 6A | 808 | CMP 0,8  |
| 6B | E6D | BRZ 6D   |
| 6C | C9A | GOTO 9A  |
| 6D | 971 | CMPI 7,1 |
| 6E | E90 | BRZ 90   |
| 6F | 18E | MOVI 8,E |

|    |     |          |
|----|-----|----------|
| 70 | CB1 | GOTO B1  |
| 71 | 809 | CMP 0,9  |
| 72 | E74 | BRZ 74   |
| 73 | C9A | GOTO 9A  |
| 74 | 981 | CMPI 8,1 |
| 75 | E90 | BRZ 90   |
| 76 | 19E | MOVI 9,E |
| 77 | CB1 | GOTO B1  |
| 78 | 80A | CMP 0,A  |
| 79 | E7B | BRZ 7B   |
| 7A | C9A | GOTO 9A  |
| 7B | 991 | CMPI 9,1 |
| 7C | E90 | BRZ 90   |
| 7D | 1AE | MOVI A,E |
| 7E | CB1 | GOTO B1  |
| 7F | 80B | CMP 0,B  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 80   | E82    | BRZ 82   |
| 81   | C9A    | GOTO 9A  |
| 82   | 9A1    | CMPI A,1 |
| 83   | E90    | BRZ 90   |
| 84   | 1BE    | MOVI B,E |
| 85   | CB1    | GOTO B1  |
| 86   | 80C    | CMP 0,C  |
| 87   | E89    | BRZ 89   |
| 88   | C9A    | GOTO 9A  |
| 89   | 9B1    | CMPI B,1 |
| 8A   | E90    | BRZ 90   |
| 8B   | 1CE    | MOVI C,E |
| 8C   | CB1    | GOTO B1  |
| 8D   | 80D    | CMP 0,D  |
| 8E   | E99    | BRZ 99   |
| 8F   | C9A    | GOTO 9A  |

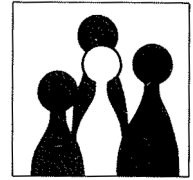
|    |     |          |
|----|-----|----------|
| 90 | 10F | MOVI 0,F |
| 91 | FEF | DOT F    |
| 92 | F05 | RND      |
| 93 | 97D | CMPI 7,D |
| 94 | D97 | BRC 97   |
| 95 | 93D | CMPI 3,D |
| 96 | D99 | BRC 99   |
| 97 | 57D | ADDI 7,D |
| 98 | C93 | GOTO 93  |
| 99 | F07 | RET      |
| 9A | 01D | MOV 1,D  |
| 9B | 10E | MOVI 0,E |
| 9C | 10F | MOVI 0,F |
| 9D | F03 | HXDZ     |
| 9E | F3D | DISP 3,D |
| 9F | 100 | MOVI 0,0 |

|    |     |          |
|----|-----|----------|
| A0 | 101 | MOVI 0,1 |
| A1 | FE1 | DOT 1    |
| A2 | 711 | SUBI 1,1 |
| A3 | DA5 | BRC A5   |
| A4 | CA1 | GOTO A1  |
| A5 | 510 | ADDI 1,0 |
| A6 | D00 | BRC 00   |
| A7 | CA1 | GOTO A1  |
| A8 | 100 | MOVI 0,0 |
| A9 | 101 | MOVI 0,1 |
| AA | FE1 | DOT 1    |
| AB | 511 | ADDI 1,1 |
| AC | DAE | BRC AE   |
| AD | CAA | GOTO AA  |
| AE | 510 | ADDI 1,0 |
| AF | D00 | BRC 00   |

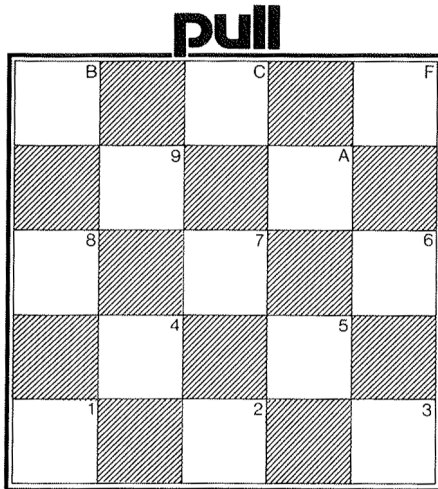
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| B0   | CAA    | GOTO AA  |
| B1   | FF0    | KIN 0    |
| B2   | FE0    | DOT 0    |
| B3   | 1FF    | MOVI F,F |
| B4   | 72F    | SUBI 2,F |
| B5   | DB7    | BRC B7   |
| B6   | CB4    | GOTO B4  |
| B7   | 10F    | MOVI 0,F |
| B8   | FEF    | DOT F    |
| B9   | 91E    | CMPI 1,E |
| BA   | E41    | BRZ 41   |
| BB   | 92E    | CMPI 2,E |
| BC   | E48    | BRZ 48   |
| BD   | 93E    | CMPI 3,E |
| BE   | E4F    | BRZ 4F   |
| BF   | 94E    | CMPI 4,E |

|    |     |          |
|----|-----|----------|
| C0 | E55 | BRZ 55   |
| C1 | 95E | CMPI 5,E |
| C2 | E5C | BRZ 5C   |
| C3 | 96E | CMPI 6,E |
| C4 | E63 | BRZ 63   |
| C5 | 97E | CMPI 7,E |
| C6 | E6A | BRZ 6A   |
| C7 | 98E | CMPI 8,E |
| C8 | E71 | BRZ 71   |
| C9 | 99E | CMPI 9,E |
| CA | E78 | BRZ 78   |
| CB | 9AE | CMPI A,E |
| CC | E7F | BRZ 7F   |
| CD | 9BE | CMPI B,E |
| CE | E86 | BRZ 86   |
| CF | C8D | GOTO 8D  |

## Pull



Pull is a strategy game. It is played on a square board of 25 individual squares. The computer gets one playing piece, which is placed on square C. We have three pieces, which are placed on squares 1, 2 and 3 at the start of the game. We may move our pieces forwards only on the dark squares, while the computer may move both forwards and backwards. We win if we manage to block the computer's piece so that it can make no further move. The computer wins as soon as it reaches one of the squares 1, 2 or 3. A Pull game board is enclosed with this book.



### Function description:

After HALT – NEXT – 00 the program is entered carefully according to the table. Then press the reset button briefly. The piezo buzzer is connected to output 3 and socket GND. Owners of the Electronic Studio 2065 or 2070 can also build the “mini organ” circuit (see page 20) and connect it to the computer outputs. Program start: HALT – NEXT – 00 – RUN.

The display shows the starting position of our pieces: 123 (our pieces are on squares 1, 2 and 3, the computer's piece on square C). We begin and make a first move. Say we want to move a piece from square 3 to square 5 (only one piece may ever be moved at a time). Our pieces are then on squares 1, 2 and 5. We enter: 125 (the new piece positions). The display now shows: A (we must place the computer's piece on square A). Then press key 0. The display shows our position 125 again. We can make the next move, for example from square 1 to square 4. We enter: 425 (the new positions of the pieces). The display now shows: 7 (the computer's piece goes to square 7). We press key 0 again and then enter our new position.

If the computer reaches square 1, 2 or 3, an interrupted tone sounds (or, with the “mini organ” connected, a falling sequence of notes). If we manage to beat the computer, a short and a long tone are produced alternately (or, with the “mini organ” connected, a short melody is played). The display then shows the positions of the four pieces in four digits. Pressing key 0 starts a new game.

## Important:

### The computer has a built-in cheat check!

If, for instance, we move one of our pieces backwards, the game is broken off at once and the display shows: F01. A new program start must be made here with HALT – NEXT – 00 – RUN.

When entering new playing positions we should also make quite sure that only one of the three piece positions is changed per move. If the display shows, say, 496 and we want to move the piece from square 9 to square B, we have to enter 4B6. If we were to enter B46, two piece positions would have changed and the computer would likewise break off the game. (Display F01 = new program start: HALT – NEXT – 00 – RUN.)

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | 130    | MOVI 3,0 |
| 02   | 121    | MOVI 2,1 |
| 03   | 112    | MOVI 1,2 |
| 04   | 1C3    | MOVI C,3 |
| 05   | BE3    | CALL E3  |
| 06   | 9C3    | CMPI C,3 |
| 07   | E20    | BRZ 20   |
| 08   | 993    | CMPI 9,3 |
| 09   | E6B    | BRZ 6B   |
| 0A   | 9A3    | CMPI A,3 |
| 0B   | E80    | BRZ 80   |
| 0C   | 983    | CMPI 8,3 |
| 0D   | E4A    | BRZ 4A   |
| 0E   | 9B3    | CMPI B,3 |
| 0F   | E95    | BRZ 95   |
| 10   | 973    | CMPI 7,3 |
| 11   | E58    | BRZ 58   |
| 12   | 943    | CMPI 4,3 |
| 13   | EA3    | BRZ A3   |
| 14   | 953    | CMPI 5,3 |
| 15   | EAC    | BRZ AC   |
| 16   | 963    | CMPI 6,3 |
| 17   | E3C    | BRZ 3C   |
| 18   | 9F3    | CMPI F,3 |
| 19   | E9C    | BRZ 9C   |
| 1A   | 913    | CMPI 1,3 |
| 1B   | EB5    | BRZ B5   |
| 1C   | 923    | CMPI 2,3 |
| 1D   | EB5    | BRZ B5   |
| 1E   | 933    | CMPI 3,3 |
| 1F   | EB5    | BRZ B5   |
| 20   | 990    | CMPI 9,0 |
| 21   | E27    | BRZ 27   |
| 22   | 991    | CMPI 9,1 |
| 23   | E27    | BRZ 27   |
| 24   | 992    | CMPI 9,2 |
| 25   | E27    | BRZ 27   |
| 26   | C2E    | GOTO 2E  |
| 27   | 9A0    | CMPI A,0 |
| 28   | EBE    | BRZ BE   |
| 29   | 9A1    | CMPI A,1 |
| 2A   | EBE    | BRZ BE   |
| 2B   | 9A2    | CMPI A,2 |
| 2C   | EBE    | BRZ BE   |
| 2D   | CD0    | GOTO D0  |
| 2E   | 9A0    | CMPI A,0 |
| 2F   | ECE    | BRZ CE   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | 9A1    | CMPI A,1 |
| 31   | ECE    | BRZ CE   |
| 32   | 9A2    | CMPI A,2 |
| 33   | ECE    | BRZ CE   |
| 34   | F05    | RND      |
| 35   | 95E    | CMPI 5,E |
| 36   | ECE    | BRZ CE   |
| 37   | 9FE    | CMPI F,E |
| 38   | ED0    | BRZ D0   |
| 39   | C34    | GOTO 34  |
| 3A   | F01    | NOP      |
| 3B   | F01    | NOP      |
| 3C   | 950    | CMPI 5,0 |
| 3D   | E43    | BRZ 43   |
| 3E   | 951    | CMPI 5,1 |
| 3F   | E43    | BRZ 43   |
| 40   | 952    | CMPI 5,2 |
| 41   | E43    | BRZ 43   |
| 42   | CDA    | GOTO DA  |
| 43   | 9A0    | CMPI A,0 |
| 44   | EBE    | BRZ BE   |
| 45   | 9A1    | CMPI A,1 |
| 46   | EBE    | BRZ BE   |
| 47   | 9A2    | CMPI A,2 |
| 48   | EBE    | BRZ BE   |
| 49   | CD0    | GOTO D0  |
| 4A   | 940    | CMPI 4,0 |
| 4B   | E51    | BRZ 51   |
| 4C   | 941    | CMPI 4,1 |
| 4D   | E51    | BRZ 51   |
| 4E   | 942    | CMPI 4,2 |
| 4F   | E51    | BRZ 51   |
| 50   | CD8    | GOTO D8  |
| 51   | 990    | CMPI 9,0 |
| 52   | EBE    | BRZ BE   |
| 53   | 991    | CMPI 9,1 |
| 54   | EBE    | BRZ BE   |
| 55   | 992    | CMPI 9,2 |
| 56   | EBE    | BRZ BE   |
| 57   | CCE    | GOTO CE  |
| 58   | 940    | CMPI 4,0 |
| 59   | E5F    | BRZ 5F   |
| 5A   | 941    | CMPI 4,1 |
| 5B   | E5F    | BRZ 5F   |
| 5C   | 942    | CMPI 4,2 |
| 5D   | E5F    | BRZ 5F   |
| 5E   | CD8    | GOTO D8  |
| 5F   | 950    | CMPI 5,0 |

Programmer: Markus Baumann, 6660 Zweibrücken 16

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | E66    | BRZ 66   |
| 61   | 951    | CMPI 5,1 |
| 62   | E66    | BRZ 66   |
| 63   | 952    | CMPI 5,2 |
| 64   | E66    | BRZ 66   |
| 65   | CDA    | GOTO DA  |
| 66   | 9A1    | CMPI A,1 |
| 67   | ECE    | BRZ CE   |
| 68   | 9A2    | CMPI A,2 |
| 69   | ECE    | BRZ CE   |
| 6A   | CD0    | GOTO D0  |
| 6B   | 970    | CMPI 7,0 |
| 6C   | E72    | BRZ 72   |
| 6D   | 971    | CMPI 7,1 |
| 6E   | E72    | BRZ 72   |
| 6F   | 972    | CMPI 7,2 |

|    |     |          |
|----|-----|----------|
| 70 | E72 | BRZ 72   |
| 71 | CD4 | GOTO D4  |
| 72 | 980 | CMPI 8,0 |
| 73 | E79 | BRZ 79   |
| 74 | 981 | CMPI 8,1 |
| 75 | E79 | BRZ 79   |
| 76 | 982 | CMPI 8,2 |
| 77 | E79 | BRZ 79   |
| 78 | CD6 | GOTO D6  |
| 79 | 9C0 | CMPI C,0 |
| 7A | ECC | BRZ CC   |
| 7B | 9C1 | CMPI C,1 |
| 7C | ECC | BRZ CC   |
| 7D | 9C2 | CMPI C,2 |
| 7E | ECC | BRZ CC   |
| 7F | CC8 | GOTO C8  |

|    |     |          |
|----|-----|----------|
| 80 | 970 | CMPI 7,0 |
| 81 | E87 | BRZ 87   |
| 82 | 971 | CMPI 7,1 |
| 83 | E87 | BRZ 87   |
| 84 | 972 | CMPI 7,2 |
| 85 | E87 | BRZ 87   |
| 86 | CD4 | GOTO D4  |
| 87 | 960 | CMPI 6,0 |
| 88 | E8E | BRZ 8E   |
| 89 | 961 | CMPI 6,1 |
| 8A | E8E | BRZ 8E   |
| 8B | 962 | CMPI 6,2 |
| 8C | E8E | BRZ 8E   |
| 8D | CD2 | GOTO D2  |
| 8E | 9C0 | CMPI C,0 |
| 8F | EDC | BRZ DC   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | 9C1    | CMPI C,1 |
| 91   | EDC    | BRZ DC   |
| 92   | 9C2    | CMPI C,2 |
| 93   | EDC    | BRZ DC   |
| 94   | CC8    | GOTO C8  |
| 95   | 990    | CMPI 9,0 |
| 96   | EBE    | BRZ BE   |
| 97   | 991    | CMPI 9,1 |
| 98   | EBE    | BRZ BE   |
| 99   | 992    | CMPI 9,2 |
| 9A   | EBE    | BRZ BE   |
| 9B   | CCE    | GOTO CE  |
| 9C   | 9A0    | CMPI A,0 |
| 9D   | EBE    | BRZ BE   |
| 9E   | 9A1    | CMPI A,1 |
| 9F   | EBE    | BRZ BE   |

|    |     |          |
|----|-----|----------|
| A0 | 9A2 | CMPI A,2 |
| A1 | EBE | BRZ BE   |
| A2 | CD0 | GOTO D0  |
| A3 | 910 | CMPI 1,0 |
| A4 | EA6 | BRZ A6   |
| A5 | CCA | GOTO CA  |
| A6 | 921 | CMPI 2,1 |
| A7 | EA9 | BRZ A9   |
| A8 | CDE | GOTO DE  |
| A9 | 973 | CMPI 7,3 |
| AA | ECC | BRZ CC   |
| AB | CD4 | GOTO D4  |
| AC | 921 | CMPI 2,1 |
| AD | EAF | BRZ AF   |
| AE | CDE | GOTO DE  |
| AF | 930 | CMPI 3,0 |

|    |     |          |
|----|-----|----------|
| B0 | EB2 | BRZ B2   |
| B1 | CE0 | GOTO E0  |
| B2 | 972 | CMPI 7,2 |
| B3 | ED2 | BRZ D2   |
| B4 | CD4 | GOTO D4  |
| B5 | 11D | MOVI 1,D |
| B6 | 71E | SUBI 1,E |
| B7 | FBD | ADC D    |
| B8 | DBB | BRC BB   |
| B9 | FEE | DOT E    |
| BA | CB6 | GOTO B6  |
| BB | F40 | DISP 4,0 |
| BC | FFF | KIN F    |
| BD | C00 | GOTO 00  |
| BE | 11D | MOVI 1,D |
| BF | 55E | ADDI 5,E |

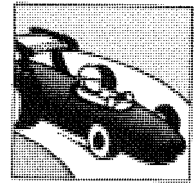
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| C0   | FBD    | ADC D    |
| C1   | DC4    | BRC C4   |
| C2   | FEE    | DOT E    |
| C3   | CBF    | GOTO BF  |
| C4   | F40    | DISP 4,0 |
| C5   | FEE    | DOT E    |
| C6   | FFF    | KIN F    |
| C7   | C00    | GOTO 00  |
| C8   | 1C3    | MOVI C,3 |
| C9   | CE1    | GOTO E1  |
| CA   | 113    | MOVI 1,3 |
| CB   | CE1    | GOTO E1  |
| CC   | 1B3    | MOVI B,3 |
| CD   | CE1    | GOTO E1  |
| CE   | 193    | MOVI 9,3 |
| CF   | CE1    | GOTO E1  |

|    |     |          |
|----|-----|----------|
| D0 | 1A3 | MOVI A,3 |
| D1 | CE1 | GOTO E1  |
| D2 | 163 | MOVI 6,3 |
| D3 | CE1 | GOTO E1  |
| D4 | 173 | MOVI 7,3 |
| D5 | CE1 | GOTO E1  |
| D6 | 183 | MOVI 8,3 |
| D7 | CE1 | GOTO E1  |
| D8 | 143 | MOVI 4,3 |
| D9 | CE1 | GOTO E1  |
| DA | 153 | MOVI 5,3 |
| DB | CE1 | GOTO E1  |
| DC | 1F3 | MOVI F,3 |
| DD | CE1 | GOTO E1  |
| DE | 123 | MOVI 2,3 |
| DF | CE1 | GOTO E1  |

|    |     |          |
|----|-----|----------|
| E0 | 133 | MOVI 3,3 |
| E1 | F13 | DISP 1,3 |
| E2 | FFF | KIN F    |
| E3 | 004 | MOV 0,4  |
| E4 | 015 | MOV 1,5  |
| E5 | 026 | MOV 2,6  |
| E6 | F30 | DISP 3,0 |
| E7 | FF2 | KIN 2    |
| E8 | FF1 | KIN 1    |
| E9 | FF0 | KIN 0    |
| EA | 804 | CMP 0,4  |
| EB | DFF | BRC FF   |
| EC | 815 | CMP 1,5  |
| ED | DFF | BRC FF   |
| EE | 826 | CMP 2,6  |
| EF | DFF | BRC FF   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| F0   | 804    | CMP 0,4  |
| F1   | EF3    | BRZ F3   |
| F2   | CF9    | GOTO F9  |
| F3   | 815    | CMP 1,5  |
| F4   | EF6    | BRZ F6   |
| F5   | CFC    | GOTO FC  |
| F6   | 826    | CMP 2,6  |
| F7   | EFF    | BRZ FF   |
| F8   | C06    | GOTO 06  |
| F9   | 815    | CMP 1,5  |
| FA   | EFC    | BRZ FC   |
| FB   | CFF    | GOTO FF  |
| FC   | 826    | CMP 2,6  |
| FD   | E06    | BRZ 06   |
| FE   | CFF    | GOTO FF  |
| FF   | F00    | HALT     |

# Car Race



The car race is an interesting reaction game. You have to try to dodge oncoming cars within a reaction time of two seconds (or, at the increased difficulty level, within one second). The road is represented by the four LEDs at the computer outputs and by four places of the display. Each LED, or each display place, corresponds to one lane. The lane occupied by your own car is marked on the display by a 1, the free lanes are marked by 000. Lit LEDs at the outputs mean that those lanes are occupied by oncoming vehicles, which change lane after every move (controlled by the computer's random generator). The player has to try to steer their car into a free lane.

## Function description:

After HALT – NEXT – 00 the program is entered according to the table. Then “input 4” is connected to the socket “Takt/Clock”.

Program start: HALT – NEXT – 00 – RUN.

The display shows, for example: 0100 = our vehicle (1) is in the second lane from the left. Individual LEDs light up at the outputs (oncoming vehicles). There are several ways of changing lane by pressing the appropriate key:

Key 1: the vehicle moves one lane to the left (the 1 on the display is shifted one place to the left).

Key 2: the vehicle stays in its lane (the display remains unchanged).

Key 3: the vehicle moves one lane to the right.

Key 5: the vehicle moves two lanes to the left.

Key 7: the vehicle moves two lanes to the right.

Key 9: the vehicle moves three lanes to the left.

Key B: the vehicle moves three lanes to the right.

The way the game works is made clear by the following example:

2 LEDs are lit at the 4 outputs:

The display shows:

1 2 3 4  
● ○ ○ ●  
0 1 0 0

Lanes 2 and 3 are occupied by oncoming cars. Our car is in the second lane, that is, in a moment it will collide with an oncoming car. An immediate evasive manoeuvre is needed: either one lane to the left (press key 1) or two lanes to the right (press key 7). The pattern of lit LEDs at the outputs changes, and we have to dodge the oncoming vehicles with quick-reaction entries.

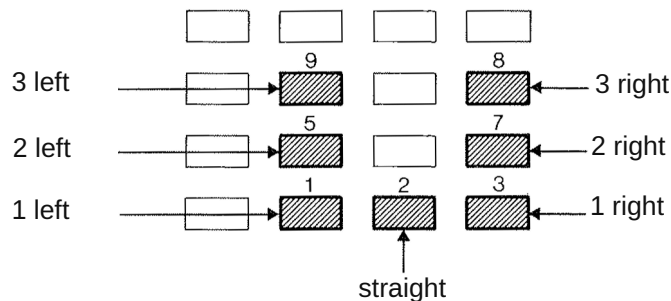
The game ends automatically if more than two seconds are needed to enter the evasive manoeuvre. The game also ends if we dodge into a lane in which a car is coming towards us, or if we move into a lane that does not exist (too far to the left or to the right). The end of the game is signalled by the flashing display, on which the kilometres driven (the number of successful evasive manoeuvres) then light up. A new game can be started by pressing key 0.

BUSCH MICROTRONIC wishes you a good trip.

For experts who would like to shorten the reaction time from two seconds to one, the instruction code 527 at address 15 can be changed to 517 in the program. Entry: HALT – NEXT – 15 517 – NEXT. New program start: HALT – NEXT – 00 – RUN.

Besides fast reactions, a little luck will now also be needed for an accident-free car race.

For fast reactions it is a good idea to cover the computer's keypad with the white keypad mask (see the accessories enclosed) and to label it as shown below:



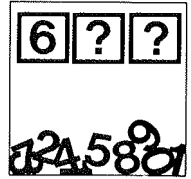
Programmer: K.-T. Eggert, 2117 Tostedt

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | 112    | MOVI 1,2 |
| 02   | 124    | MOVI 2,4 |
| 03   | F40    | DISP 4,0 |
| 04   | F05    | RND      |
| 05   | 8D9    | CMP D,9  |
| 06   | E0D    | BRZ 0D   |
| 07   | 9FD    | CMPI F,D |
| 08   | E0D    | BRZ 0D   |
| 09   | 0D9    | MOV D,9  |
| 0A   | 515    | ADDI 1,5 |
| 0B   | FB6    | ADC 6    |
| 0C   | C10    | GOTO 10  |
| 0D   | 71E    | SUBI 1,E |
| 0E   | 6ED    | SUB E,D  |
| 0F   | C05    | GOTO 05  |
|      |        |          |
| 10   | F06    | TIME     |
| 11   | 0A7    | MOV A,7  |
| 12   | 0B8    | MOV B,8  |
| 13   | 967    | CMPI 6,7 |
| 14   | D10    | BRC 10   |
| 15   | 527    | ADDI 2,7 |
| 16   | FE9    | DOT 9    |
| 17   | FFC    | KIN C    |
| 18   | 92C    | CMPI 2,C |
| 19   | E2C    | BRZ 2C   |
| 1A   | 99C    | CMPI 9,C |
| 1B   | E25    | BRZ 25   |
| 1C   | 95C    | CMPI 5,C |
| 1D   | E26    | BRZ 26   |
| 1E   | 91C    | CMPI 1,C |
| 1F   | E27    | BRZ 27   |
|      |        |          |
| 20   | 9BC    | CMPI B,C |
| 21   | E29    | BRZ 29   |
| 22   | 97C    | CMPI 7,C |
| 23   | E2A    | BRZ 2A   |
| 24   | C2B    | GOTO 2B  |
| 25   | F94    | SHR 4    |
| 26   | F94    | SHR 4    |
| 27   | F94    | SHR 4    |
| 28   | C2C    | GOTO 2C  |
| 29   | FA4    | SHL 4    |
| 2A   | FA4    | SHL 4    |
| 2B   | FA4    | SHL 4    |
| 2C   | 100    | MOVI 0,0 |
| 2D   | 101    | MOVI 0,1 |
| 2E   | 102    | MOVI 0,2 |
| 2F   | 103    | MOVI 0,3 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | 914    | CMPI 1,4 |
| 31   | E39    | BRZ 39   |
| 32   | 924    | CMPI 2,4 |
| 33   | E3B    | BRZ 3B   |
| 34   | 944    | CMPI 4,4 |
| 35   | E3D    | BRZ 3D   |
| 36   | 984    | CMPI 8,4 |
| 37   | E3F    | BRZ 3F   |
| 38   | C49    | GOTO 49  |
| 39   | 113    | MOVI 1,3 |
| 3A   | C40    | GOTO 40  |
| 3B   | 112    | MOVI 1,2 |
| 3C   | C40    | GOTO 40  |
| 3D   | 111    | MOVI 1,1 |
| 3E   | C40    | GOTO 40  |
| 3F   | 110    | MOVI 1,0 |
|      |        |          |
| 40   | F06    | TIME     |
| 41   | 87A    | CMP 7,A  |
| 42   | D49    | BRC 49   |
| 43   | 88B    | CMP 8,B  |
| 44   | E46    | BRZ 46   |
| 45   | C49    | GOTO 49  |
| 46   | 098    | MOV 9,8  |
| 47   | 248    | AND 4,8  |
| 48   | E04    | BRZ 04   |
| 49   | 10D    | MOVI 0,D |
| 4A   | 10E    | MOVI 0,E |
| 4B   | 10F    | MOVI 0,F |
| 4C   | F02    | DISOUT   |
| 4D   | 51D    | ADDI 1,D |
| 4E   | D50    | BRC 50   |
| 4F   | C4C    | GOTO 4C  |
|      |        |          |
| 50   | F40    | DISP 4,0 |
| 51   | 51E    | ADDI 1,E |
| 52   | D54    | BRC 54   |
| 53   | C50    | GOTO 50  |
| 54   | 55F    | ADDI 5,F |
| 55   | D57    | BRC 57   |
| 56   | C4C    | GOTO 4C  |
| 57   | 10F    | MOVI 0,F |
| 58   | 05D    | MOV 5,D  |
| 59   | 06E    | MOV 6,E  |
| 5A   | F03    | HXDZ     |
| 5B   | F2D    | DISP 2,D |
| 5C   | FFF    | KIN F    |
| 5D   | C00    | GOTO 00  |



# Sorting Digits



A tricky brain-teaser. The numbers 1 – 6 appear on the computer display in random order (e.g. 264315). These digits have to be sorted into ascending order (123456) by means of prescribed swap operations. The swap operations are laid down in such a way that it is not possible simply to exchange two digits with each other; instead four numbers are swapped in every swap operation. The aim of the game is to reach ascending order with as few swap operations as possible.

## Function description:

First the program is entered after HALT – NEXT – 00 according to the program table. Then the piezo buzzer is connected to output 1 and socket GND. Program start: HALT – NEXT – 00 – RUN. The display shows: 000.

The computer has now stored the number sequence 123456 and wants to know how often these numbers are to be swapped among themselves. Entering a small number (e.g. 2) gives an easier puzzle. Entering a larger number (e.g. 8) increases the difficulty. Maximum entry = 999. To begin with it is advisable to start with a small number, e.g. 2. This entry has to be completed by pressing key A.

After a short time the six numbers appear on the display in a jumbled order (e.g. 251346). Three different swap operations are available (which may also be used in alternation) to turn the jumbled sequence into an ascending one (123456). With badly jumbled sequences, “sorting digits” becomes a real brain-teaser.

The following swap operations are available:

1st swap operation = press key 0

The 1st (left-hand) number of the display is exchanged with the 4th number (half right), and at the same time the 2nd number is exchanged with the 3rd. (Example: 251346 becomes 315246.)

2nd swap operation = press key 1

The 2nd number is exchanged with the 5th and at the same time the 3rd number with the 4th. (Example: 251346 becomes 243156.)

3rd swap operation = press key 2

The 3rd number is exchanged with the 6th and the 4th number with the 5th. (Example: 251346 becomes 256431.)

A game in progress and the use of the swap operations are demonstrated by the following example:

After the program start the display shows: 000. Enter e.g. 4, then press key A. The computer now swaps the number sequence 123456 four times, in a random order of the swap operations described. The display now shows, for example: 264315

1st swap attempt  
entry e.g. key 0

2nd swap attempt  
entry e.g. key 2

3rd swap attempt  
entry e.g. key 0

4th swap attempt  
entry e.g. key 1

264315 gives 346215



346215 gives 345126



345126 gives 154326



154326 gives 123456



The correct order has been found. The piezo buzzer sounds for a short time. For the example above the display then shows: 0004, that is, 4 swap attempts were needed to find the correct number sequence. Pressing any number key restarts the program.

If key F is pressed during play (instead of 0, 1 or 2), the number of swap attempts needed so far is displayed. Pressing key F again brings back the number sequence and the game can be continued.

Programmer: Markus Jouaux, 6970 Lauda

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F02    | DISOUT   |
| 02   | FE0    | DOT 0    |
| 03   | 116    | MOVI 1,6 |
| 04   | 125    | MOVI 2,5 |
| 05   | 134    | MOVI 3,4 |
| 06   | 143    | MOVI 4,3 |
| 07   | 152    | MOVI 5,2 |
| 08   | 161    | MOVI 6,1 |
| 09   | F37    | DISP 3,7 |
| 0A   | FF0    | KIN 0    |
| 0B   | 990    | CMPI 9,0 |
| 0C   | D11    | BRC 11   |
| 0D   | 089    | MOV 8,9  |
| 0E   | 078    | MOV 7,8  |
| 0F   | 007    | MOV 0,7  |

|    |     |          |
|----|-----|----------|
| 10 | C09 | GOTO 09  |
| 11 | F02 | DISOUT   |
| 12 | 07D | MOV 7,D  |
| 13 | 08E | MOV 8,E  |
| 14 | 09F | MOV 9,F  |
| 15 | F04 | DZHX     |
| 16 | 0D7 | MOV D,7  |
| 17 | 0E8 | MOV E,8  |
| 18 | 0F9 | MOV F,9  |
| 19 | F05 | RND      |
| 1A | 93D | CMPI 3,D |
| 1B | D1D | BRC 1D   |
| 1C | C1F | GOTO 1F  |
| 1D | 53D | ADDI 3,D |
| 1E | C1A | GOTO 1A  |
| 1F | 93E | CMPI 3,E |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | D22    | BRC 22   |
| 21   | C24    | GOTO 24  |
| 22   | 53E    | ADDI 3,E |
| 23   | C1F    | GOTO 1F  |
| 24   | 93F    | CMPI 3,F |
| 25   | D27    | BRC 27   |
| 26   | C29    | GOTO 29  |
| 27   | 53F    | ADDI 3,F |
| 28   | C24    | GOTO 24  |
| 29   | 8CD    | CMP C,D  |
| 2A   | E2C    | BRZ 2C   |
| 2B   | C2E    | GOTO 2E  |
| 2C   | 51D    | ADDI 1,D |
| 2D   | C1A    | GOTO 1A  |
| 2E   | 8DE    | CMP D,E  |
| 2F   | E31    | BRZ 31   |

|    |     |          |
|----|-----|----------|
| 30 | C33 | GOTO 33  |
| 31 | 51E | ADDI 1,E |
| 32 | C1A | GOTO 1A  |
| 33 | 8EF | CMP E,F  |
| 34 | E36 | BRZ 36   |
| 35 | C38 | GOTO 38  |
| 36 | 51F | ADDI 1,F |
| 37 | C1A | GOTO 1A  |
| 38 | 0FC | MOV F,C  |
| 39 | 0DB | MOV D,B  |
| 3A | B83 | CALL 83  |
| 3B | B88 | CALL 88  |
| 3C | 0EB | MOV E,B  |
| 3D | B83 | CALL 83  |
| 3E | B88 | CALL 88  |
| 3F | 0FB | MOV F,B  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 40   | B83    | CALL 83  |
| 41   | B88    | CALL 88  |
| 42   | C19    | GOTO 19  |
| 43   | 107    | MOVI 0,7 |
| 44   | 108    | MOVI 0,8 |
| 45   | 109    | MOVI 0,9 |
| 46   | F61    | DISP 6,1 |
| 47   | FFB    | KIN B    |
| 48   | 9FB    | CMPI F,B |
| 49   | E80    | BRZ 80   |
| 4A   | 92B    | CMPI 2,B |
| 4B   | D46    | BRC 46   |
| 4C   | B88    | CALL 88  |
| 4D   | 517    | ADDI 1,7 |
| 4E   | 9A7    | CMPI A,7 |
| 4F   | E51    | BRZ 51   |

|    |     |          |
|----|-----|----------|
| 50 | C5D | GOTO 5D  |
| 51 | 567 | ADDI 6,7 |
| 52 | FB8 | ADC 8    |
| 53 | 9A8 | CMPI A,8 |
| 54 | E56 | BRZ 56   |
| 55 | C5D | GOTO 5D  |
| 56 | 568 | ADDI 6,8 |
| 57 | FB9 | ADC 9    |
| 58 | 9A9 | CMPI A,9 |
| 59 | E5B | BRZ 5B   |
| 5A | C5D | GOTO 5D  |
| 5B | 569 | ADDI 6,9 |
| 5C | FBA | ADC A    |
| 5D | 961 | CMPI 6,1 |
| 5E | E60 | BRZ 60   |
| 5F | C46 | GOTO 46  |

|    |     |          |
|----|-----|----------|
| 60 | 952 | CMPI 5,2 |
| 61 | E63 | BRZ 63   |
| 62 | C46 | GOTO 46  |
| 63 | 943 | CMPI 4,3 |
| 64 | E66 | BRZ 66   |
| 65 | C46 | GOTO 46  |
| 66 | 934 | CMPI 3,4 |
| 67 | E69 | BRZ 69   |
| 68 | C46 | GOTO 46  |
| 69 | 925 | CMPI 2,5 |
| 6A | E6C | BRZ 6C   |
| 6B | C46 | GOTO 46  |
| 6C | 916 | CMPI 1,6 |
| 6D | E6F | BRZ 6F   |
| 6E | C46 | GOTO 46  |
| 6F | 10B | MOVI 0,B |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 70   | F61    | DISP 6,1 |
| 71   | 1F0    | MOVI F,0 |
| 72   | FE0    | DOT 0    |
| 73   | 100    | MOVI 0,0 |
| 74   | FE0    | DOT 0    |
| 75   | 1F0    | MOVI F,0 |
| 76   | FE0    | DOT 0    |
| 77   | 100    | MOVI 0,0 |
| 78   | FE0    | DOT 0    |
| 79   | 51B    | ADDI 1,B |
| 7A   | 9AB    | CMPI A,B |
| 7B   | E7D    | BRZ 7D   |
| 7C   | C70    | GOTO 70  |
| 7D   | F47    | DISP 4,7 |
| 7E   | FF0    | KIN 0    |
| 7F   | C00    | GOTO 00  |

|    |     |          |
|----|-----|----------|
| 80 | F47 | DISP 4,7 |
| 81 | FFB | KIN B    |
| 82 | C46 | GOTO 46  |
| 83 | 717 | SUBI 1,7 |
| 84 | FC8 | SUBC 8   |
| 85 | FC9 | SUBC 9   |
| 86 | D43 | BRC 43   |
| 87 | F07 | RET      |
| 88 | 90B | CMPI 0,B |
| 89 | E9A | BRZ 9A   |
| 8A | 91B | CMPI 1,B |
| 8B | E93 | BRZ 93   |
| 8C | 04C | MOV 4,C  |
| 8D | 014 | MOV 1,4  |
| 8E | 0C1 | MOV C,1  |
| 8F | 03C | MOV 3,C  |

|    |     |         |
|----|-----|---------|
| 90 | 023 | MOV 2,3 |
| 91 | 0C2 | MOV C,2 |
| 92 | F07 | RET     |
| 93 | 05C | MOV 5,C |
| 94 | 025 | MOV 2,5 |
| 95 | 0C2 | MOV C,2 |
| 96 | 04C | MOV 4,C |
| 97 | 034 | MOV 3,4 |
| 98 | 0C3 | MOV C,3 |
| 99 | F07 | RET     |
| 9A | 06C | MOV 6,C |
| 9B | 036 | MOV 3,6 |
| 9C | 0C3 | MOV C,3 |
| 9D | 05C | MOV 5,C |
| 9E | 045 | MOV 4,5 |
| 9F | 0C4 | MOV C,4 |

A0 F07 RET

# Forming Words and Sentences



With this program MICROTRONIC can display simple words and form sentences.

The letters A – F are available for showing such words on the display. In addition the digit zero can be used as the letter “o”, the digit 1 as “i” and the digit 5 as “s”. This gives 9 different letters, which can be put together by suitable programming into short German words such as AFFE, EI, BOESE, BADE, SIE, SOFT, BABIE, IDA and so on.

The following program forms more or less sensible sentences from such words, whereby well-meant computer advice such as “BADE BEI EBBE” (bathe at low tide) need not necessarily be followed, and small cheeky remarks (“SIE AFFE” – you ape) should on no account be taken personally.

## Function description:

After HALT – NEXT – 00 the program is entered.  
Program start: HALT – NEXT – 00 – RUN.

The display shows the words for two sentences one after the other and then switches off. After pressing key 0 the computer brings the next sentences, until the display goes dark again. Key 0 calls up the sentences that follow. This goes on until the computer takes its leave with “ADE” (farewell). Pressing key 0 once more restarts the program.

This program is only meant to give a few ideas for programming similar comments for games and the like yourself. Once the MICROTRONIC instruction books have been worked through, the structure of the program is simple: MOVI instructions move the appropriate letters and numbers into the registers, and DISP instructions then show them. The subroutines from address AD and B3 generate pauses so that the words are shown on the display for a short time.

The use of this is demonstrated in the following game “17 + 4”, where the computer pays the winner a compliment (in English): “GOOD”, or tells the loser with “BAD” that their result was poor.

With a little thought many more words can be formed from the 9 available letters, whereby the spelling “ß” (at the end of a word) has to be turned into “ss”, e.g. “FASS” (instead of Faß). Further examples of words: ABI (Abitur), ABBA (pop group), AS (playing card), BASS (musical instrument), FASSE, ESSE, OB, SAFE, SIEB, SIE, ES, SASS and so on.

Programmer: Frank Simon, 6342 Haiger 6

| Adr.Befehl Mnemonic |     |          |
|---------------------|-----|----------|
| 00                  | F02 | DISOUT   |
| 01                  | F08 | CLEAR    |
| 02                  | BB3 | CALL B3  |
| 03                  | 1E0 | MOVI E,0 |
| 04                  | 111 | MOVI 1,1 |
| 05                  | 152 | MOVI 5,2 |
| 06                  | F30 | DISP 3,0 |
| 07                  | BAD | CALL AD  |
| 08                  | 1F1 | MOVI F,1 |
| 09                  | 1F2 | MOVI F,2 |
| 0A                  | 1A3 | MOVI A,3 |
| 0B                  | F40 | DISP 4,0 |
| 0C                  | BAD | CALL AD  |
| 0D                  | F02 | DISOUT   |
| 0E                  | BB3 | CALL B3  |
| 0F                  | 111 | MOVI 1,1 |

| Adr.Befehl Mnemonic |     |          |
|---------------------|-----|----------|
| 10                  | 152 | MOVI 5,2 |
| 11                  | F30 | DISP 3,0 |
| 12                  | BAD | CALL AD  |
| 13                  | 150 | MOVI 5,0 |
| 14                  | 1E1 | MOVI E,1 |
| 15                  | 1E3 | MOVI E,3 |
| 16                  | 114 | MOVI 1,4 |
| 17                  | 1F5 | MOVI F,5 |
| 18                  | F60 | DISP 6,0 |
| 19                  | BAD | CALL AD  |
| 1A                  | 110 | MOVI 1,0 |
| 1B                  | F20 | DISP 2,0 |
| 1C                  | BAD | CALL AD  |
| 1D                  | F02 | DISOUT   |
| 1E                  | FFF | KIN F    |
| 1F                  | BB3 | CALL B3  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | 1E0    | MOVI E,0 |
| 21   | 151    | MOVI 5,1 |
| 22   | F40    | DISP 4,0 |
| 23   | BAD    | CALL AD  |
| 24   | 150    | MOVI 5,0 |
| 25   | 1A1    | MOVI A,1 |
| 26   | 1D2    | MOVI D,2 |
| 27   | F30    | DISP 3,0 |
| 28   | BAD    | CALL AD  |
| 29   | 110    | MOVI 1,0 |
| 2A   | 1E1    | MOVI E,1 |
| 2B   | F20    | DISP 2,0 |
| 2C   | BAD    | CALL AD  |
| 2D   | F02    | DISOUT   |
| 2E   | BB3    | CALL B3  |
| 2F   | 100    | MOVI 0,0 |

|    |     |          |
|----|-----|----------|
| 30 | 1D1 | MOVI D,1 |
| 31 | 102 | MOVI 0,2 |
| 32 | 1B3 | MOVI B,3 |
| 33 | F40 | DISP 4,0 |
| 34 | BAD | CALL AD  |
| 35 | 150 | MOVI 5,0 |
| 36 | 151 | MOVI 5,1 |
| 37 | 1A2 | MOVI A,2 |
| 38 | F30 | DISP 3,0 |
| 39 | BAD | CALL AD  |
| 3A | 1A1 | MOVI A,1 |
| 3B | 1D2 | MOVI D,2 |
| 3C | F30 | DISP 3,0 |
| 3D | BAD | CALL AD  |
| 3E | 110 | MOVI 1,0 |
| 3F | 1E1 | MOVI E,1 |

|    |     |          |
|----|-----|----------|
| 40 | F20 | DISP 2,0 |
| 41 | BAD | CALL AD  |
| 42 | F02 | DISOUT   |
| 43 | BB3 | CALL B3  |
| 44 | 150 | MOVI 5,0 |
| 45 | 1A1 | MOVI A,1 |
| 46 | F30 | DISP 3,0 |
| 47 | BAD | CALL AD  |
| 48 | 110 | MOVI 1,0 |
| 49 | 1E1 | MOVI E,1 |
| 4A | F20 | DISP 2,0 |
| 4B | BAD | CALL AD  |
| 4C | 152 | MOVI 5,2 |
| 4D | F30 | DISP 3,0 |
| 4E | BAD | CALL AD  |
| 4F | 1E0 | MOVI E,0 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 50   | 1D1    | MOVI D,1 |
| 51   | 1A2    | MOVI A,2 |
| 52   | 1F3    | MOVI F,3 |
| 53   | F40    | DISP 4,0 |
| 54   | BAD    | CALL AD  |
| 55   | F02    | DISOUT   |
| 56   | FFF    | KIN F    |
| 57   | BB3    | CALL B3  |
| 58   | 150    | MOVI 5,0 |
| 59   | 1A1    | MOVI A,1 |
| 5A   | 1D2    | MOVI D,2 |
| 5B   | F30    | DISP 3,0 |
| 5C   | BAD    | CALL AD  |
| 5D   | 1E0    | MOVI E,0 |
| 5E   | 111    | MOVI 1,1 |
| 5F   | 1B2    | MOVI B,2 |

|    |     |          |
|----|-----|----------|
| 60 | 1A3 | MOVI A,3 |
| 61 | 1B4 | MOVI B,4 |
| 62 | F50 | DISP 5,0 |
| 63 | BAD | CALL AD  |
| 64 | 150 | MOVI 5,0 |
| 65 | 151 | MOVI 5,1 |
| 66 | 112 | MOVI 1,2 |
| 67 | 1B3 | MOVI B,3 |
| 68 | F40 | DISP 4,0 |
| 69 | BAD | CALL AD  |
| 6A | F02 | DISOUT   |
| 6B | BB3 | CALL B3  |
| 6C | 1E1 | MOVI E,1 |
| 6D | 152 | MOVI 5,2 |
| 6E | 1E3 | MOVI E,3 |
| 6F | 104 | MOVI 0,4 |

|    |     |          |
|----|-----|----------|
| 70 | 1B5 | MOVI B,5 |
| 71 | F60 | DISP 6,0 |
| 72 | BAD | CALL AD  |
| 73 | 1E0 | MOVI E,0 |
| 74 | 111 | MOVI 1,1 |
| 75 | 1B2 | MOVI B,2 |
| 76 | 1A3 | MOVI A,3 |
| 77 | 1B4 | MOVI B,4 |
| 78 | F50 | DISP 5,0 |
| 79 | BAD | CALL AD  |
| 7A | F02 | DISOUT   |
| 7B | FFF | KIN F    |
| 7C | BB3 | CALL B3  |
| 7D | 110 | MOVI 1,0 |
| 7E | 1D1 | MOVI D,1 |
| 7F | 1E2 | MOVI E,2 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 80   | F30    | DISP 3,0 |
| 81   | BAD    | CALL AD  |
| 82   | 150    | MOVI 5,0 |
| 83   | 151    | MOVI 5,1 |
| 84   | 1A2    | MOVI A,2 |
| 85   | 153    | MOVI 5,3 |
| 86   | F40    | DISP 4,0 |
| 87   | BAD    | CALL AD  |
| 88   | 110    | MOVI 1,0 |
| 89   | 1E1    | MOVI E,1 |
| 8A   | 1B2    | MOVI B,2 |
| 8B   | F30    | DISP 3,0 |
| 8C   | BAD    | CALL AD  |
| 8D   | 1A0    | MOVI A,0 |
| 8E   | 1D1    | MOVI D,1 |
| 8F   | 112    | MOVI 1,2 |

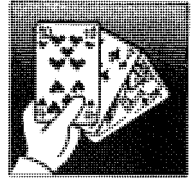
|    |     |          |
|----|-----|----------|
| 90 | BAD | CALL AD  |
| 91 | F02 | DISOUT   |
| 92 | FFF | KIN F    |
| 93 | BB3 | CALL B3  |
| 94 | 1E0 | MOVI E,0 |
| 95 | 1A2 | MOVI A,2 |
| 96 | 1B3 | MOVI B,3 |
| 97 | F40 | DISP 4,0 |
| 98 | BAD | CALL AD  |
| 99 | 110 | MOVI 1,0 |
| 9A | 1E1 | MOVI E,1 |
| 9B | 1B2 | MOVI B,2 |
| 9C | F30 | DISP 3,0 |
| 9D | BAD | CALL AD  |
| 9E | 1E0 | MOVI E,0 |
| 9F | 1B1 | MOVI B,1 |

|    |     |          |
|----|-----|----------|
| A0 | 1B2 | MOVI B,2 |
| A1 | 1E3 | MOVI E,3 |
| A2 | F40 | DISP 4,0 |
| A3 | BAD | CALL AD  |
| A4 | F02 | DISOUT   |
| A5 | BB3 | CALL B3  |
| A6 | 1D1 | MOVI D,1 |
| A7 | 1A2 | MOVI A,2 |
| A8 | F30 | DISP 3,0 |
| A9 | BB3 | CALL B3  |
| AA | F02 | DISOUT   |
| AB | FFF | KIN F    |
| AC | C00 | GOTO 00  |
| AD | 10A | MOVI 0,A |
| AE | 51A | ADDI 1,A |
| AF | 9FA | CMPI F,A |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| B0   | EB2    | BRZ B2   |
| B1   | CAE    | GOTO AE  |
| B2   | F07    | RET      |
| B3   | 10A    | MOVI 0,A |
| B4   | 51A    | ADDI 1,A |
| B5   | 9FA    | CMPI F,A |
| B6   | EB8    | BRZ B8   |
| B7   | CB4    | GOTO B4  |
| B8   | 10A    | MOVI 0,A |
| B9   | 51A    | ADDI 1,A |
| BA   | 9FA    | CMPI F,A |
| BB   | EBD    | BRZ BD   |
| BC   | CB9    | GOTO B9  |
| BD   | 10A    | MOVI 0,A |
| BE   | 51A    | ADDI 1,A |
| BF   | 9FA    | CMPI F,A |

|    |     |         |
|----|-----|---------|
| C0 | EC2 | BRZ C2  |
| C1 | CBE | GOTO BE |
| C2 | F07 | RET     |

## 17 + 4 (Black Jack)



17 + 4 is a well-known card game (a game of chance). The aim of the game is to come as close as possible to the number 21 (17 + 4) by drawing 2 or more cards. Under no circumstances may 21 be exceeded. The players receive a card in turn until one of them believes they have enough points. The winner is the one who comes closest to 21. The game (further dealing of cards) is ended when a player has either received 2 aces or has reached 21, which means that this player has won.

### Function description:

After HALT – NEXT – 00 the program is entered according to the table. Then the piezo buzzer is connected to output 1 and socket GND.  
Program start: HALT – NEXT – 00 – RUN.

The display shows: “17 ADD 4” (17 + 4). After key 1 is pressed the computer deals a “card” to us and one to itself (that is, it deals out the “card values”). The display shows the value dealt to us.

XX0X  
| |  
| X = card value we have just received.  
|  
XX = sum of the card values we have received  
altogether.

We can now decide whether we want to draw another card, or whether the game should be ended because we think we have enough card values:

Key 1 = the computer gives us another card.  
Key 0 = the game is to be ended.

At the end of the game the piezo buzzer sounds and the display shows “GOOD” (6 is used as G) if we have won, or “BAD” if the game was lost. On pressing key 1 the display shows the final score of the game:

XX00XX  
| |  
| XX = sum of our card values.  
|  
XX = sum of the computer's card values.

If the points are level the computer wins (that is how the program works). After pressing key 1 the display shows “17 ADD 4” and we can risk a new game.

Note that the numbers 1 to 9 and A as the ace (value 10) are allowed as card values. (The number 0 counts as a blank.) These values do not correspond exactly to the original cards, but the computer version of “17 + 4” is nevertheless just as interesting as the real model.

The computer ends the game automatically when the following score is reached:

1. The computer or we have reached more than 21 points.
2. The computer or we have reached exactly 21 points.
3. The first two cards drawn were 2 aces. In this case the display shows “AA” at the corresponding points position instead of the sum of the card values.

The program is written so that the computer plays as an independent player. It cannot “see” our cards. The cards drawn in turn are determined by a random generator.

With games of chance of this kind no special winning strategy can be programmed that would give the computer a better chance than us. MICROTRONIC has been programmed so that after every deal it checks whether it has already reached more than 16 points. In that case it draws no further card for itself. A "feel" for the game cannot be programmed into a computer. A person may well draw another card (although they already have 18 points) because they hope to reach exactly 21 that way, or may stop even though only 14 points have been reached.

Programmer: Johannes Freundorfer, 8354 Metten

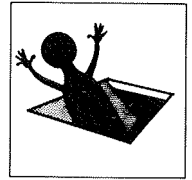
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | 14A    | MOVI 4,A |
| 03   | 1DB    | MOVI D,B |
| 04   | 1DC    | MOVI D,C |
| 05   | 1AD    | MOVI A,D |
| 06   | 17E    | MOVI 7,E |
| 07   | 11F    | MOVI 1,F |
| 08   | F6A    | DISP 6,A |
| 09   | FFF    | KIN F    |
| 0A   | F02    | DISOUT   |
| 0B   | B6C    | CALL 6C  |
| 0C   | C12    | GOTO 12  |
| 0D   | F40    | DISP 4,0 |
| 0E   | B6C    | CALL 6C  |
| 0F   | FFF    | KIN F    |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 10   | F02    | DISOUT   |
| 11   | E2C    | BRZ 2C   |
| 12   | 80D    | CMP 0,D  |
| 13   | E15    | BRZ 15   |
| 14   | C1E    | GOTO 1E  |
| 15   | 9A0    | CMPI A,0 |
| 16   | E18    | BRZ 18   |
| 17   | C1E    | GOTO 1E  |
| 18   | 902    | CMPI 0,2 |
| 19   | E1B    | BRZ 1B   |
| 1A   | C1E    | GOTO 1E  |
| 1B   | 1A2    | MOVI A,2 |
| 1C   | 1A3    | MOVI A,3 |
| 1D   | C5A    | GOTO 5A  |
| 1E   | 0D0    | MOV D,0  |
| 1F   | 402    | ADD 0,2  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | D24    | BRC 24   |
| 21   | 992    | CMPI 9,2 |
| 22   | D24    | BRC 24   |
| 23   | C26    | GOTO 26  |
| 24   | 513    | ADDI 1,3 |
| 25   | 562    | ADDI 6,2 |
| 26   | 923    | CMPI 2,3 |
| 27   | E29    | BRZ 29   |
| 28   | C2C    | GOTO 2C  |
| 29   | 912    | CMPI 1,2 |
| 2A   | E5A    | BRZ 5A   |
| 2B   | D60    | BRC 60   |
| 2C   | 917    | CMPI 1,7 |
| 2D   | D33    | BRC 33   |
| 2E   | E30    | BRZ 30   |
| 2F   | C36    | GOTO 36  |
| 30   | 966    | CMPI 6,6 |
| 31   | D33    | BRC 33   |
| 32   | C36    | GOTO 36  |
| 33   | 90F    | CMPI 0,F |
| 34   | E53    | BRZ 53   |
| 35   | C0D    | GOTO 0D  |
| 36   | 84E    | CMP 4,E  |
| 37   | E39    | BRZ 39   |
| 38   | C42    | GOTO 42  |
| 39   | 9A4    | CMPI A,4 |
| 3A   | E3C    | BRZ 3C   |
| 3B   | C42    | GOTO 42  |
| 3C   | 903    | CMPI 0,3 |
| 3D   | E3F    | BRZ 3F   |
| 3E   | C42    | GOTO 42  |
| 3F   | 1A6    | MOVI A,6 |
| 40   | 1A7    | MOVI A,7 |
| 41   | C60    | GOTO 60  |
| 42   | 0E4    | MOV E,4  |
| 43   | 446    | ADD 4,6  |
| 44   | D48    | BRC 48   |
| 45   | 996    | CMPI 9,6 |
| 46   | D48    | BRC 48   |
| 47   | C4A    | GOTO 4A  |
| 48   | 566    | ADDI 6,6 |
| 49   | 517    | ADDI 1,7 |
| 4A   | 927    | CMPI 2,7 |
| 4B   | E4F    | BRZ 4F   |
| 4C   | 90F    | CMPI 0,F |
| 4D   | E53    | BRZ 53   |
| 4E   | C0D    | GOTO 0D  |
| 4F   | 916    | CMPI 1,6 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 50   | E60    | BRZ 60   |
| 51   | D5A    | BRC 5A   |
| 52   | C4C    | GOTO 4C  |
| 53   | 837    | CMP 3,7  |
| 54   | E57    | BRZ 57   |
| 55   | D60    | BRC 60   |
| 56   | C5A    | GOTO 5A  |
| 57   | 826    | CMP 2,6  |
| 58   | E60    | BRZ 60   |
| 59   | D60    | BRC 60   |
| 5A   | 1DC    | MOVI D,C |
| 5B   | 10D    | MOVI 0,D |
| 5C   | 10E    | MOVI 0,E |
| 5D   | 16F    | MOVI 6,F |
| 5E   | F4C    | DISP 4,C |
| 5F   | C64    | GOTO 64  |
| 60   | 1DD    | MOVI D,D |
| 61   | 1AE    | MOVI A,E |
| 62   | 1BF    | MOVI B,F |
| 63   | F3D    | DISP 3,D |
| 64   | 1FB    | MOVI F,B |
| 65   | FEB    | DOT B    |
| 66   | FFB    | KIN B    |
| 67   | 104    | MOVI 0,4 |
| 68   | FE4    | DOT 4    |
| 69   | F62    | DISP 6,2 |
| 6A   | FF0    | KIN 0    |
| 6B   | C00    | GOTO 00  |
| 6C   | F05    | RND      |
| 6D   | 4FE    | ADD F,E  |
| 6E   | 9AD    | CMPI A,D |
| 6F   | D71    | BRC 71   |
| 70   | C73    | GOTO 73  |
| 71   | 57D    | ADDI 7,D |
| 72   | C6E    | GOTO 6E  |
| 73   | 9AE    | CMPI A,E |
| 74   | D76    | BRC 76   |
| 75   | F07    | RET      |
| 76   | 57E    | ADDI 7,E |
| 77   | C73    | GOTO 73  |

## Fallgrube (Pitfall)



A game board of one hundred squares holds 12 invisible pitfalls. One after another, three playing pieces have to be brought from the starting position to the GOAL without falling into a pit. Tactics and a generous helping of luck are needed for this. Every time a new game begins the computer lays out the pitfalls afresh. A special game board is enclosed with this book as an accessory.

### Function description:

After HALT – NEXT – 00 the program is entered according to the table. Then the piezo buzzer is connected to output 4 and socket GND.

Program start: HALT – NEXT – 00 – RUN.

At the start of the game the computer distributes 12 pitfalls (controlled by its random generator). The display then shows: 00. The first playing piece can start. Note that the start is always made in ROW 9. You may, however, choose which COLUMN the piece starts in.

### Example:

The first piece is to start in the 3rd column of row 9. When entering on the computer, the COLUMN is always entered first and then the ROW. In this example the entry is: 39. The display goes dark for a moment – then the value entered is shown.

### Important:

The playing piece may only ever advance one square in the direction of the GOAL. Sideways or backwards it may skip any number of squares. The piece can now go, for example, to the 4th column of row 8. Entry: 48.

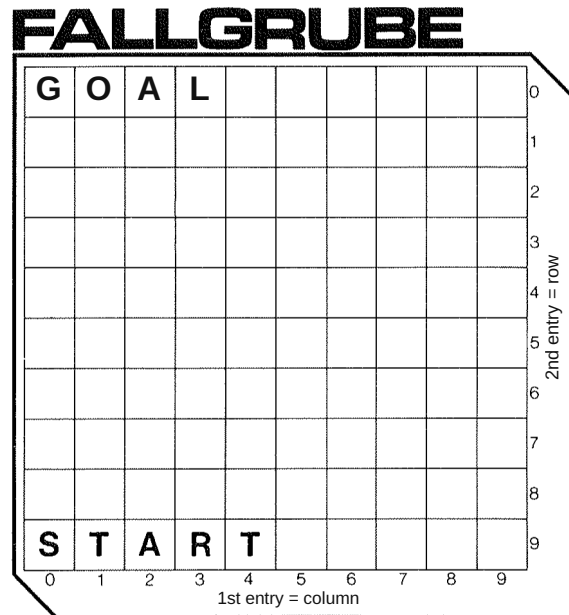
If the piezo buzzer sounds briefly after the position numbers (COLUMN and ROW) have been entered, a wrong position was entered (e.g. advanced by more than one square). In that case the position has to be repeated correctly.

If a continuous tone sounds, the playing piece has fallen into a pit. The display shows: 02. The first digit tells us how many pieces have reached the GOAL (in this case none so far). The second digit shows how many pieces are still available (in this case 2). After key 0 is pressed the display shows 00 – the next piece can be started. (Start in ROW 9.)

Once a piece has reached the goal line (row 0) there are two possibilities:

1. The piezo buzzer sounds: the piece has fallen into a pit on the goal line.
2. The piezo buzzer does not sound: the piece has arrived safely at the GOAL.

In both cases the score is displayed, e.g. 11 (1 piece at the GOAL, 1 piece still available). Pressing key 0 redistributes the pitfalls. The next piece can start.



## End of game:

The display shows: 66 F00 = all 3 pieces have fallen into a pitfall, none has reached the GOAL. New game start: HALT – NEXT – 00 – RUN.

The display shows: 1000 (or 2000 or 3000) = one piece (or 2 or 3 pieces) has arrived safely at the GOAL. The game is over, no further playing pieces are available. New game start: HALT – NEXT – 00 – RUN.

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F02    | DISOUT   |
| 01   | F08    | CLEAR    |
| 02   | F0D    | EXRL     |
| 03   | 13E    | MOVI 3,E |
| 04   | 1FC    | MOVI F,C |
| 05   | 19A    | MOVI 9,A |
| 06   | F0E    | EXRM     |
| 07   | F08    | CLEAR    |
| 08   | FE0    | DOT 0    |
| 09   | B7A    | CALL 7A  |
| 0A   | OD0    | MOV D,0  |
| 0B   | OE1    | MOV E,1  |
| 0C   | B7A    | CALL 7A  |
| 0D   | BA4    | CALL A4  |
| 0E   | D0C    | BRC 0C   |
| 0F   | OD2    | MOV D,2  |
|      |        |          |
| 10   | OE3    | MOV E,3  |
| 11   | B7A    | CALL 7A  |
| 12   | B9F    | CALL 9F  |
| 13   | D11    | BRC 11   |
| 14   | OD4    | MOV D,4  |
| 15   | OE5    | MOV E,5  |
| 16   | B7A    | CALL 7A  |
| 17   | B9A    | CALL 9A  |
| 18   | D16    | BRC 16   |
| 19   | OD6    | MOV D,6  |
| 1A   | OE7    | MOV E,7  |
| 1B   | B7A    | CALL 7A  |
| 1C   | B95    | CALL 95  |
| 1D   | D1B    | BRC 1B   |
| 1E   | OD8    | MOV D,8  |
| 1F   | OE9    | MOV E,9  |
|      |        |          |
| 20   | B7A    | CALL 7A  |
| 21   | B90    | CALL 90  |
| 22   | D20    | BRC 20   |
| 23   | ODA    | MOV D,A  |
| 24   | OEB    | MOV E,B  |
| 25   | B7A    | CALL 7A  |
| 26   | B8B    | CALL 8B  |
| 27   | D25    | BRC 25   |
| 28   | F0D    | EXRL     |
| 29   | OD0    | MOV D,0  |
| 2A   | OE1    | MOV E,1  |
| 2B   | F0D    | EXRL     |
| 2C   | B7A    | CALL 7A  |
| 2D   | B8B    | CALL 8B  |
| 2E   | D2C    | BRC 2C   |
| 2F   | F0D    | EXRL     |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | OD2    | MOV D,2  |
| 31   | OE3    | MOV E,3  |
| 32   | F0D    | EXRL     |
| 33   | B7A    | CALL 7A  |
| 34   | B8B    | CALL 8B  |
| 35   | D33    | BRC 33   |
| 36   | F0D    | EXRL     |
| 37   | OD4    | MOV D,4  |
| 38   | OE5    | MOV E,5  |
| 39   | F0D    | EXRL     |
| 3A   | B7A    | CALL 7A  |
| 3B   | B8B    | CALL 8B  |
| 3C   | D3A    | BRC 3A   |
| 3D   | F0D    | EXRL     |
| 3E   | OD6    | MOV D,6  |
| 3F   | OE7    | MOV E,7  |
|      |        |          |
| 40   | F0D    | EXRL     |
| 41   | B7A    | CALL 7A  |
| 42   | ODC    | MOV D,C  |
| 43   | B7A    | CALL 7A  |
| 44   | B86    | CALL 86  |
| 45   | D43    | BRC 43   |
| 46   | F0E    | EXRM     |
| 47   | F0F    | EXRA     |
| 48   | FE7    | DOT 7    |
| 49   | F20    | DISP 2,0 |
| 4A   | FF1    | KIN 1    |
| 4B   | FF0    | KIN 0    |
| 4C   | F02    | DISOUT   |
| 4D   | BAD    | CALL AD  |
| 4E   | F0E    | EXRM     |
| 4F   | BAD    | CALL AD  |
|      |        |          |
| 50   | F0F    | EXRA     |
| 51   | F0D    | EXRL     |
| 52   | BCA    | CALL CA  |
| 53   | F0D    | EXRL     |
| 54   | F0F    | EXRA     |
| 55   | F0E    | EXRM     |
| 56   | BE7    | CALL E7  |
| 57   | 9A5    | CMPI A,5 |
| 58   | E5A    | BRZ 5A   |
| 59   | C6A    | GOTO 6A  |
| 5A   | FE5    | DOT 5    |
| 5B   | 105    | MOVI 0,5 |
| 5C   | 716    | SUBI 1,6 |
| 5D   | 100    | MOVI 0,0 |
| 5E   | 101    | MOVI 0,1 |
| 5F   | 192    | MOVI 9,2 |



| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 906    | CMPI 0,6 |
| 61   | E63    | BRZ 63   |
| 62   | C67    | GOTO 67  |
| 63   | 907    | CMPI 0,7 |
| 64   | E66    | BRZ 66   |
| 65   | CF3    | GOTO F3  |
| 66   | F00    | HALT     |
| 67   | F26    | DISP 2,6 |
| 68   | FFF    | KIN F    |
| 69   | C48    | GOTO 48  |
| 6A   | 900    | CMPI 0,0 |
| 6B   | E6D    | BRZ 6D   |
| 6C   | C49    | GOTO 49  |
| 6D   | 100    | MOVI 0,0 |
| 6E   | 101    | MOVI 0,1 |
| 6F   | 716    | SUBI 1,6 |

|    |     |          |
|----|-----|----------|
| 70 | 517 | ADDI 1,7 |
| 71 | 906 | CMPI 0,6 |
| 72 | EF3 | BRZ F3   |
| 73 | F26 | DISP 2,6 |
| 74 | FFF | KIN F    |
| 75 | F02 | DISOUT   |
| 76 | F0F | EXRA     |
| 77 | 19A | MOVI 9,A |
| 78 | F0E | EXRM     |
| 79 | C09 | GOTO 09  |
| 7A | F05 | RND      |
| 7B | 99D | CMPI 9,D |
| 7C | D7E | BRC 7E   |
| 7D | C80 | GOTO 80  |
| 7E | 59D | ADDI 9,D |
| 7F | C7B | GOTO 7B  |

|    |     |          |
|----|-----|----------|
| 80 | 99E | CMPI 9,E |
| 81 | D83 | BRC 83   |
| 82 | C85 | GOTO 85  |
| 83 | 53E | ADDI 3,E |
| 84 | C80 | GOTO 80  |
| 85 | F07 | RET      |
| 86 | 8CA | CMP C,A  |
| 87 | E89 | BRZ 89   |
| 88 | C8B | GOTO 8B  |
| 89 | 8DB | CMP D,B  |
| 8A | EAB | BRZ AB   |
| 8B | 8EC | CMP E,C  |
| 8C | E8E | BRZ 8E   |
| 8D | C90 | GOTO 90  |
| 8E | 8FD | CMP F,D  |
| 8F | EAB | BRZ AB   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | 8D8    | CMP D,8  |
| 91   | E93    | BRZ 93   |
| 92   | C95    | GOTO 95  |
| 93   | 8E9    | CMP E,9  |
| 94   | EAB    | BRZ AB   |
| 95   | 8D6    | CMP D,6  |
| 96   | E98    | BRZ 98   |
| 97   | C9A    | GOTO 9A  |
| 98   | 8E7    | CMP E,7  |
| 99   | EAB    | BRZ AB   |
| 9A   | 8D4    | CMP D,4  |
| 9B   | E9D    | BRZ 9D   |
| 9C   | C9F    | GOTO 9F  |
| 9D   | 8E5    | CMP E,5  |
| 9E   | EAB    | BRZ AB   |
| 9F   | 8D2    | CMP D,2  |

|    |     |         |
|----|-----|---------|
| A0 | EA2 | BRZ A2  |
| A1 | CA4 | GOTO A4 |
| A2 | 8E3 | CMP E,3 |
| A3 | EAB | BRZ AB  |
| A4 | 8D0 | CMP D,0 |
| A5 | EA7 | BRZ A7  |
| A6 | CA9 | GOTO A9 |
| A7 | 8E1 | CMP E,1 |
| A8 | EAB | BRZ AB  |
| A9 | FOA | RSC     |
| AA | F07 | RET     |
| AB | F09 | STC     |
| AC | F07 | RET     |
| AD | 808 | CMP 0,8 |
| AE | EB0 | BRZ B0  |
| AF | CB4 | GOTO B4 |

|    |     |          |
|----|-----|----------|
| B0 | 819 | CMP 1,9  |
| B1 | EB3 | BRZ B3   |
| B2 | CB4 | GOTO B4  |
| B3 | 1A5 | MOVI A,5 |
| B4 | 80A | CMP 0,A  |
| B5 | EB7 | BRZ B7   |
| B6 | CB8 | GOTO BB  |
| B7 | 81B | CMP 1,B  |
| B8 | EBA | BRZ BA   |
| B9 | CB8 | GOTO BB  |
| BA | 1A5 | MOVI A,5 |
| BB | 80C | CMP 0,C  |
| BC | EBE | BRZ BE   |
| BD | CC2 | GOTO C2  |
| BE | 81D | CMP 1,D  |
| BF | EC1 | BRZ C1   |

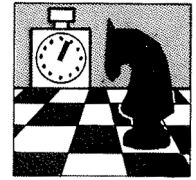
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| C0   | CC2    | GOTO C2  |
| C1   | 1A5    | MOVI A,5 |
| C2   | 80E    | CMP 0,E  |
| C3   | EC5    | BRZ C5   |
| C4   | CC9    | GOTO C9  |
| C5   | 81F    | CMP 1,F  |
| C6   | EC8    | BRZ C8   |
| C7   | CC9    | GOTO C9  |
| C8   | 1A5    | MOVI A,5 |
| C9   | F07    | RET      |
| CA   | 880    | CMP 8,0  |
| CB   | ECD    | BRZ CD   |
| CC   | CD1    | GOTO D1  |
| CD   | 891    | CMP 9,1  |
| CE   | ED0    | BRZ D0   |
| CF   | CD1    | GOTO D1  |

|    |     |          |
|----|-----|----------|
| D0 | 1AD | MOVI A,D |
| D1 | 882 | CMP 8,2  |
| D2 | ED4 | BRZ D4   |
| D3 | CD8 | GOTO D8  |
| D4 | 893 | CMP 9,3  |
| D5 | ED7 | BRZ D7   |
| D6 | CD8 | GOTO D8  |
| D7 | 1AD | MOVI A,D |
| D8 | 884 | CMP 8,4  |
| D9 | EDB | BRZ DB   |
| DA | CDF | GOTO DF  |
| DB | 895 | CMP 9,5  |
| DC | EDE | BRZ DE   |
| DD | CDF | GOTO DF  |
| DE | 1AD | MOVI A,D |
| DF | 886 | CMP 8,6  |

|    |     |          |
|----|-----|----------|
| E0 | EE2 | BRZ E2   |
| E1 | CE6 | GOTO E6  |
| E2 | 897 | CMP 9,7  |
| E3 | EE5 | BRZ E5   |
| E4 | CE6 | GOTO E6  |
| E5 | 1AD | MOVI A,D |
| E6 | F07 | RET      |
| E7 | 912 | CMPI 1,2 |
| E8 | EFO | BRZ F0   |
| E9 | 712 | SUBI 1,2 |
| EA | 802 | CMP 0,2  |
| EB | DED | BRC ED   |
| EC | CF0 | GOTO F0  |
| ED | FE4 | DOT 4    |
| EE | 512 | ADDI 1,2 |
| EF | C48 | GOTO 48  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| F0   | 002    | MOV 0,2  |
| F1   | 013    | MOV 1,3  |
| F2   | F07    | RET      |
| F3   | 106    | MOVI 0,6 |
| F4   | 105    | MOVI 0,5 |
| F5   | 104    | MOVI 0,4 |
| F6   | F44    | DISP 4,4 |
| F7   | FFF    | KIN F    |
| F8   | C00    | GOTO 00  |

# Chess Clock



So-called chess clocks are used in chess tournaments to make sure that the players have the same amount of "thinking time" available in the course of the tournament.

A chess clock consists of 2 separate stopwatches with which the time the players need to make their moves is measured alternately. While the first player is making their move, the first clock runs. When they have finished their move a button is pressed, which switches their clock off and the opponent's clock on. When the opponent's move is finished a button is pressed again, which now stops the second clock and starts the first one again. In tournaments a maximum time can be laid down that is available to each player as "thinking time". The player who reaches their maximum time first (without checkmating the opponent) has then lost the game.

Such a chess clock can also be programmed with MICROTRONIC and used for home tournaments.

## Function description:

After HALT – NEXT – 00 the program is entered as per the table. Then the two push-buttons (G and H) and the piezo buzzer are connected as shown in the illustration. A connection from input 4 to Takt/Clock must also be made.

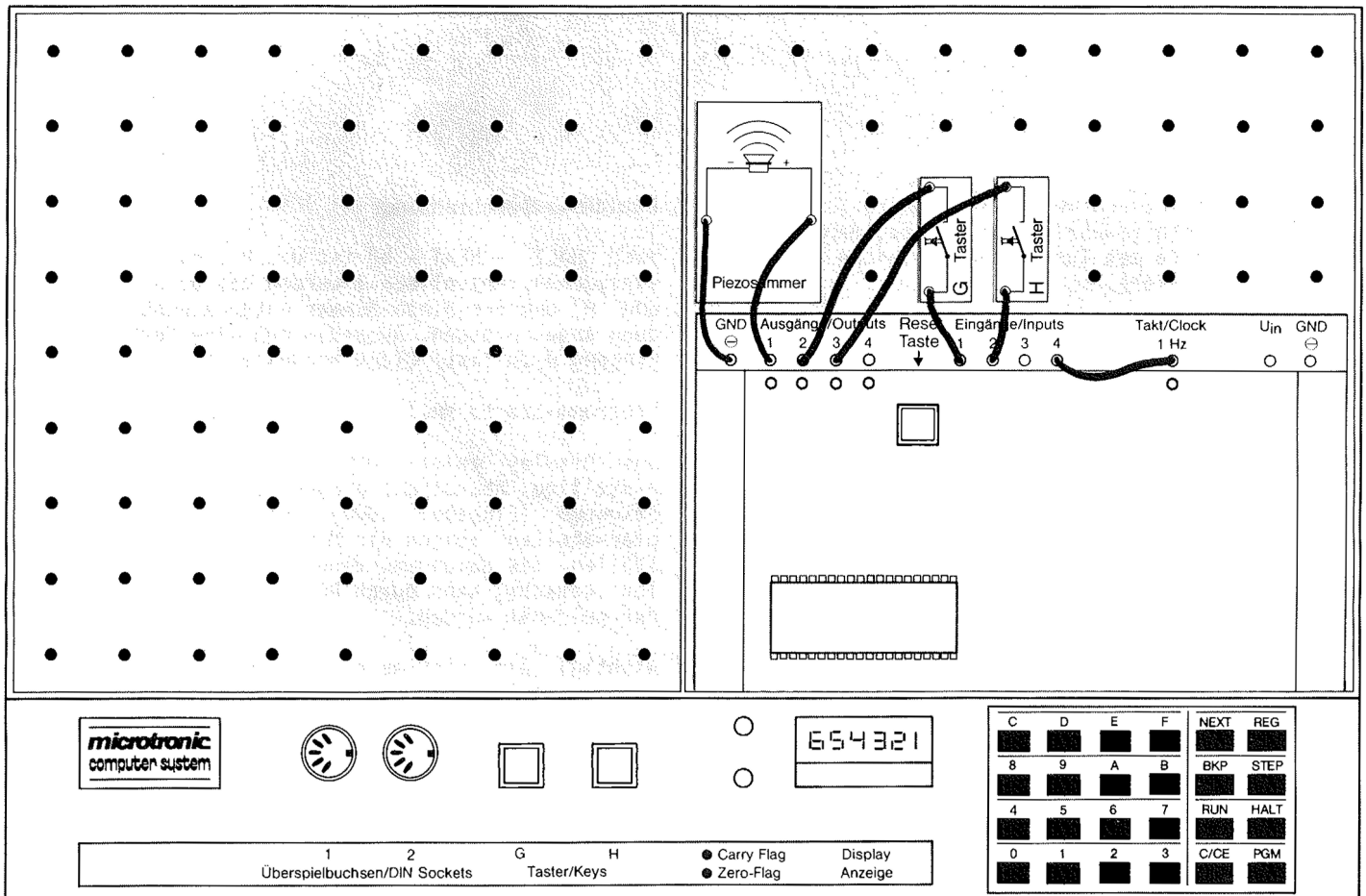
Program start: HALT – NEXT – 00 – RUN.

The display shows: 0000. We can now enter a 4-digit maximum time, e.g. 40 minutes, 00 seconds (entry: 4000). The two left-hand display places show the minutes, the two right-hand places the seconds. If a mistake is made during entry it can be cleared again by pressing key B.

Important: the seconds entered must on no account be greater than 59, as otherwise correct operation of the program is not guaranteed. The maximum time is 99 minutes and 59 seconds.

As soon as key A is pressed, the time starts to run for the first player. They can now make their move and then press the red button H (for about one second). Their time is stopped and the opponent's time starts to run, until the opponent presses button G (for about one second), which switches the first player's clock on again. The display shows how much time each player still has available.

As soon as one of the two players has used up the maximum time entered, the piezo buzzer sounds. The display shows either 1E XXX (X = time) if the first player, or 2E XXX if the second player has reached the maximum time. Pressing any number key restarts the program.



| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | 167    | MOVI 6,7 |
| 02   | FE7    | DOT 7    |
| 03   | F40    | DISP 4,0 |
| 04   | FF4    | KIN 4    |
| 05   | 9A4    | CMPI A,4 |
| 06   | EOF    | BRZ 0F   |
| 07   | 9B4    | CMPI B,4 |
| 08   | E00    | BRZ 00   |
| 09   | D01    | BRC 01   |
| 0A   | 023    | MOV 2,3  |
| 0B   | 012    | MOV 1,2  |
| 0C   | 001    | MOV 0,1  |
| 0D   | 040    | MOV 4,0  |
| 0E   | C03    | GOTO 03  |
| 0F   | F48    | DISP 4,8 |

|    |     |          |
|----|-----|----------|
| 10 | 808 | CMP 0,8  |
| 11 | E13 | BRZ 13   |
| 12 | C1B | GOTO 1B  |
| 13 | 819 | CMP 1,9  |
| 14 | E16 | BRZ 16   |
| 15 | C1B | GOTO 1B  |
| 16 | 82A | CMP 2,A  |
| 17 | E19 | BRZ 19   |
| 18 | C1B | GOTO 1B  |
| 19 | 83B | CMP 3,B  |
| 1A | E6B | BRZ 6B   |
| 1B | FD6 | DIN 6    |
| 1C | 986 | CMPI 8,6 |
| 1D | E22 | BRZ 22   |
| 1E | B74 | CALL 74  |
| 1F | 915 | CMPI 1,5 |

|    |     |          |
|----|-----|----------|
| 20 | E3D | BRZ 3D   |
| 21 | C0F | GOTO 0F  |
| 22 | F48 | DISP 4,8 |
| 23 | 518 | ADDI 1,8 |
| 24 | 9A8 | CMPI A,8 |
| 25 | E2D | BRZ 2D   |
| 26 | FD6 | DIN 6    |
| 27 | 906 | CMPI 0,6 |
| 28 | EOF | BRZ 0F   |
| 29 | B74 | CALL 74  |
| 2A | 915 | CMPI 1,5 |
| 2B | E3D | BRZ 3D   |
| 2C | C26 | GOTO 26  |
| 2D | 108 | MOVI 0,8 |
| 2E | 519 | ADDI 1,9 |
| 2F | 969 | CMPI 6,9 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | E32    | BRZ 32   |
| 31   | C26    | GOTO 26  |
| 32   | 109    | MOVI 0,9 |
| 33   | 51A    | ADDI 1,A |
| 34   | 9AA    | CMPI A,A |
| 35   | E37    | BRZ 37   |
| 36   | C26    | GOTO 26  |
| 37   | 10A    | MOVI 0,A |
| 38   | 51B    | ADDI 1,B |
| 39   | 96B    | CMPI 6,B |
| 3A   | E3C    | BRZ 3C   |
| 3B   | C26    | GOTO 26  |
| 3C   | F00    | HALT     |
| 3D   | F4C    | DISP 4,C |
| 3E   | 80C    | CMP 0,C  |
| 3F   | E41    | BRZ 41   |

|    |     |          |
|----|-----|----------|
| 40 | C49 | GOTO 49  |
| 41 | 81D | CMP 1,D  |
| 42 | E44 | BRZ 44   |
| 43 | C49 | GOTO 49  |
| 44 | 82E | CMP 2,E  |
| 45 | E47 | BRZ 47   |
| 46 | C49 | GOTO 49  |
| 47 | 83F | CMP 3,F  |
| 48 | E6D | BRZ 6D   |
| 49 | FD6 | DIN 6    |
| 4A | 986 | CMPI 8,6 |
| 4B | E50 | BRZ 50   |
| 4C | B74 | CALL 74  |
| 4D | 925 | CMPI 2,5 |
| 4E | EOF | BRZ 0F   |
| 4F | C3D | GOTO 3D  |

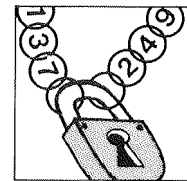
|    |     |          |
|----|-----|----------|
| 50 | F4C | DISP 4,C |
| 51 | 51C | ADDI 1,C |
| 52 | 9AC | CMPI A,C |
| 53 | E5B | BRZ 5B   |
| 54 | FD6 | DIN 6    |
| 55 | 906 | CMPI 0,6 |
| 56 | E3D | BRZ 3D   |
| 57 | B74 | CALL 74  |
| 58 | 925 | CMPI 2,5 |
| 59 | EOF | BRZ 0F   |
| 5A | C54 | GOTO 54  |
| 5B | 10C | MOVI 0,C |
| 5C | 51D | ADDI 1,D |
| 5D | 96D | CMPI 6,D |
| 5E | E60 | BRZ 60   |
| 5F | C54 | GOTO 54  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 10D    | MOVI 0,D |
| 61   | 51E    | ADDI 1,E |
| 62   | 9AE    | CMPI A,E |
| 63   | E65    | BRZ 65   |
| 64   | C54    | GOTO 54  |
| 65   | 10E    | MOVI 0,E |
| 66   | 51F    | ADDI 1,F |
| 67   | 96F    | CMPI 6,F |
| 68   | E6A    | BRZ 6A   |
| 69   | C54    | GOTO 54  |
| 6A   | F00    | HALT     |
| 6B   | 115    | MOVI 1,5 |
| 6C   | C6E    | GOTO 6E  |
| 6D   | 125    | MOVI 2,5 |
| 6E   | 1E4    | MOVI E,4 |
| 6F   | F60    | DISP 6,0 |

|    |     |          |
|----|-----|----------|
| 70 | 11F | MOVI 1,F |
| 71 | FEF | DOT F    |
| 72 | FFF | KIN F    |
| 73 | C00 | GOTO 00  |
| 74 | FD6 | DIN 6    |
| 75 | 926 | CMPI 2,6 |
| 76 | E7A | BRZ 7A   |
| 77 | 9A6 | CMPI A,6 |
| 78 | E7A | BRZ 7A   |
| 79 | C7C | GOTO 7C  |
| 7A | 115 | MOVI 1,5 |
| 7B | F07 | RET      |
| 7C | 916 | CMPI 1,6 |
| 7D | E81 | BRZ 81   |
| 7E | 996 | CMPI 9,6 |
| 7F | E81 | BRZ 81   |

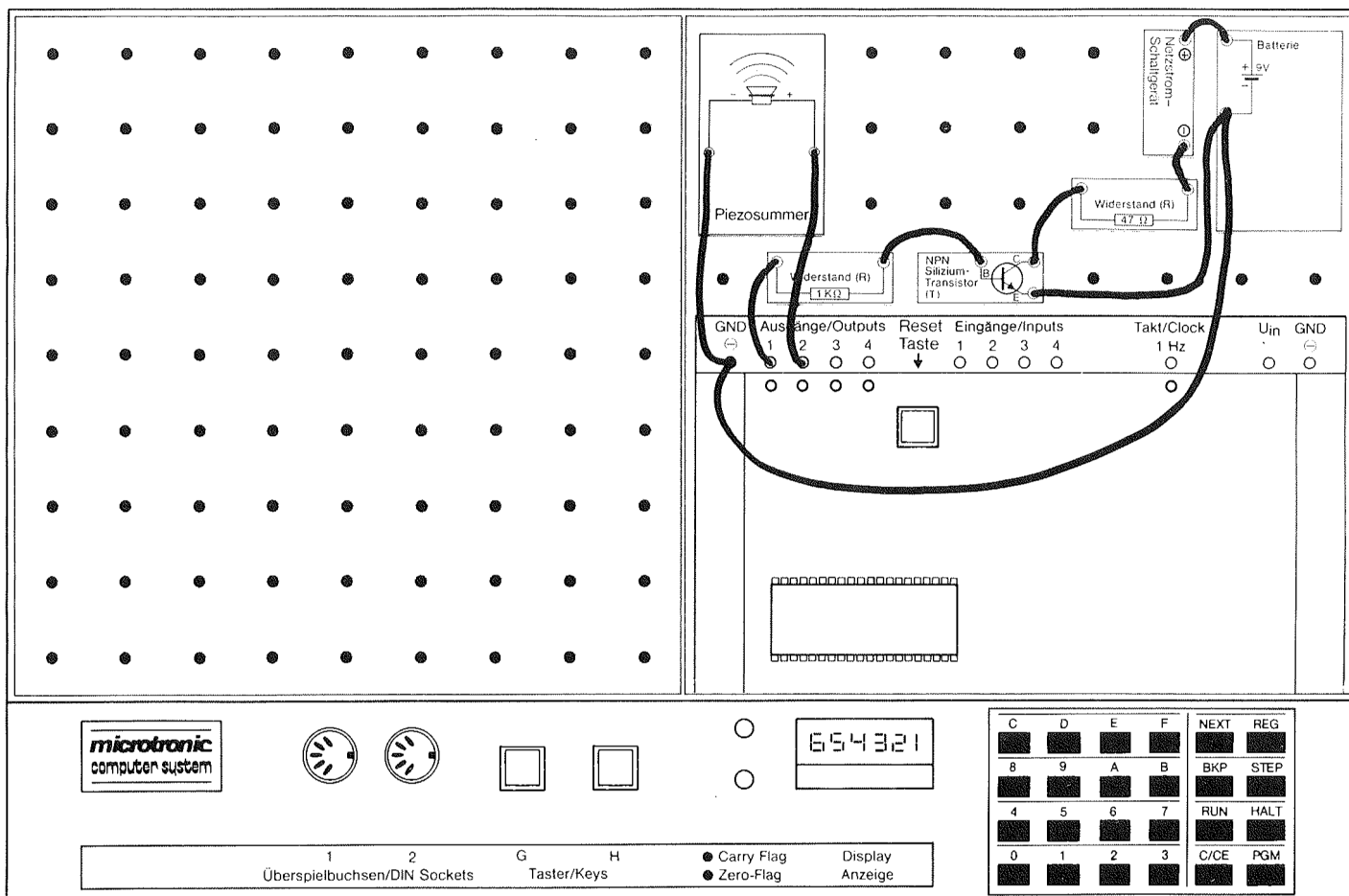
|    |     |          |
|----|-----|----------|
| 80 | C82 | GOTO 82  |
| 81 | 125 | MOVI 2,5 |
| 82 | F07 | RET      |

# Code Lock

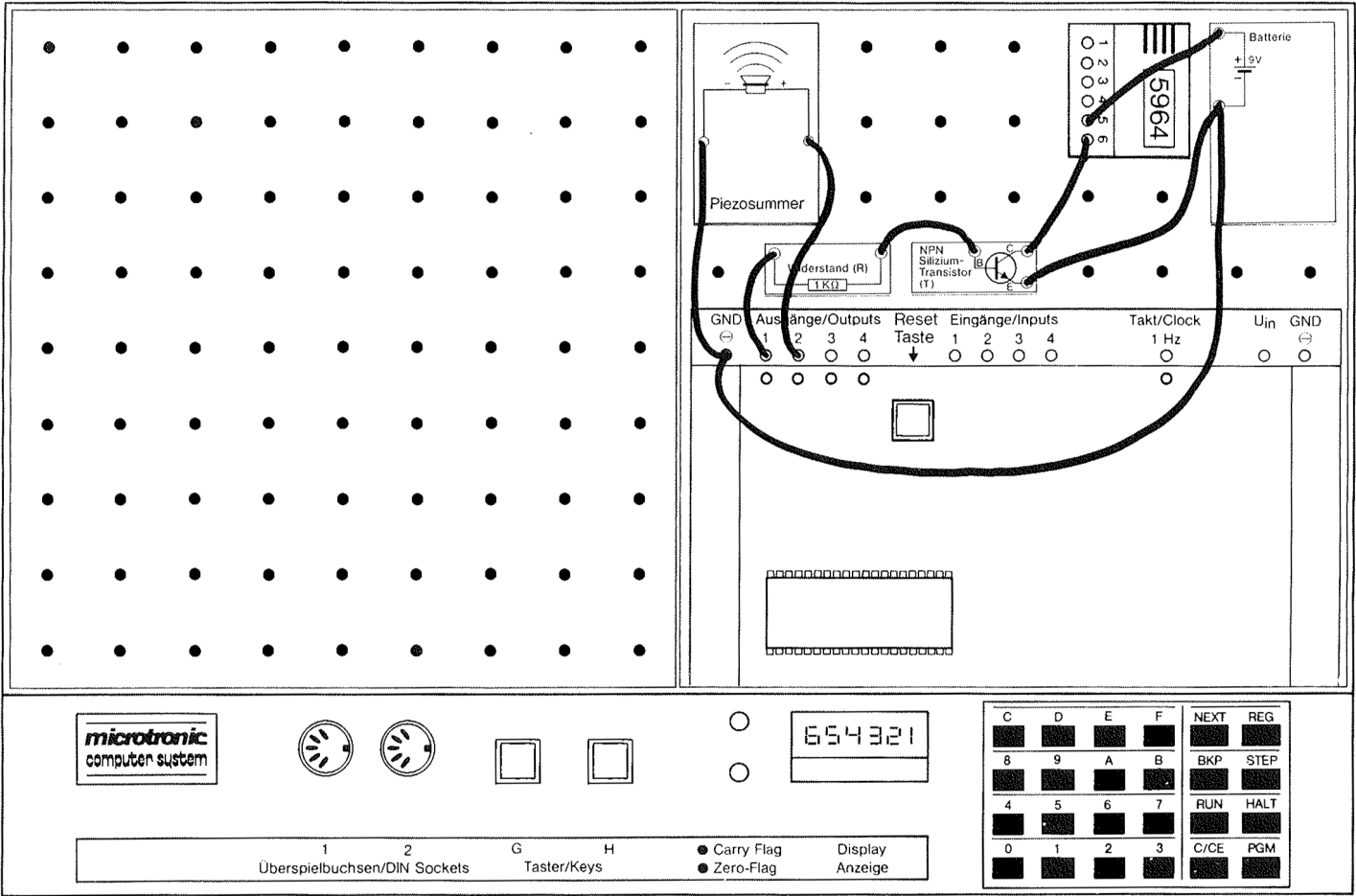


Code locks have been used for decades to secure safes, bank deposit boxes or suitcases, for example. The action of these code locks used to be achieved by complicated mechanics. For some years now there have also been electronic code locks, which have several important advantages: they work completely silently, for instance, and electronic code locks can be connected directly to an alarm system, so as to set off an alarm if a wrong code is entered, for example.

With the following program MICROTRONIC can be turned into a code lock with alarm triggering. By entering a 6-digit numerical code, electrical or electronic devices connected to the computer can be put into operation only by someone who knows the code. If a wrong code is entered, an alarm is set off.



The 6-digit code gives more than 16 million (16,777,216) different possibilities. At least 5 seconds are needed to try one possibility, that is, if you wanted to find the code number by trial and error, more than 2 1/2 years would be needed for it.



## Function description:

Before programming, the program memory must without fail be cleared by entering HALT – PGM – 5 (erase program). Then, after HALT – NEXT – 00, enter the program as per the table. Note that in the program table the middle digit of the instruction code is shown as X from address 03 to 08. In place of X one digit of the 6-digit code number is entered each time. If the code is to be 195513, for example, the following instructions have to be entered from address 03 to 08:

Address Instruction code

|    |     |
|----|-----|
| 03 | 115 |
| 04 | 194 |
| 05 | 153 |
| 06 | 152 |
| 07 | 111 |
| 08 | 130 |

When the program has been entered completely, the piezo buzzer is connected to the GND and output 2 sockets. The BUSCH special relay 5964 or the BUSCH mains switching unit 2087 can be connected to output 1 as shown in the illustrations. Electronic or electrical devices can be connected to these relays, and they can only be put into operation by entering the correct code.

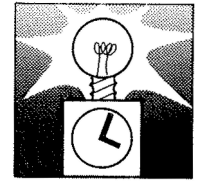
Program start: HALT – NEXT – 00 – RUN.

The display shows 000000. If the previously programmed code (195513 in our example) is now entered, output 1 is switched on together with the relay connected to it. If this output is to be switched off again, any number key has to be pressed. If a wrong code is entered, the piezo buzzer connected to output 2 sounds; it switches itself off again automatically after a short time. A new attempt at entry can then be made.

Programmer: Frank Simon, 6342 Haiger-Allendorf

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | F02    | DISOUT   |
| 03   | 1X5    | MOVI X,5 |
| 04   | 1X4    | MOVI X,4 |
| 05   | 1X3    | MOVI X,3 |
| 06   | 1X2    | MOVI X,2 |
| 07   | 1X1    | MOVI X,1 |
| 08   | 1X0    | MOVI X,0 |
| 09   | F66    | DISP 6,6 |
| 0A   | FFB    | KIN B    |
| 0B   | FFA    | KIN A    |
| 0C   | FF9    | KIN 9    |
| 0D   | FF8    | KIN 8    |
| 0E   | FF7    | KIN 7    |
| 0F   | FF6    | KIN 6    |
| 10   | 806    | CMP 0,6  |
| 11   | E13    | BRZ 13   |
| 12   | C27    | GOTO 27  |
| 13   | 817    | CMP 1,7  |
| 14   | E16    | BRZ 16   |
| 15   | C27    | GOTO 27  |
| 16   | 828    | CMP 2,8  |
| 17   | E19    | BRZ 19   |
| 18   | C27    | GOTO 27  |
| 19   | 839    | CMP 3,9  |
| 1A   | E1C    | BRZ 1C   |
| 1B   | C27    | GOTO 27  |
| 1C   | 84A    | CMP 4,A  |
| 1D   | E1F    | BRZ 1F   |
| 1E   | C27    | GOTO 27  |
| 1F   | 85B    | CMP 5,B  |
| 20   | E22    | BRZ 22   |
| 21   | C27    | GOTO 27  |
| 22   | F02    | DISOUT   |
| 23   | 11C    | MOVI 1,C |
| 24   | FEC    | DOT C    |
| 25   | FFD    | KIN D    |
| 26   | C00    | GOTO 00  |
| 27   | 12E    | MOVI 2,E |
| 28   | FEE    | DOT E    |

## Light-barrier controlled time measurement



Also required are a few components from the BUSCH Electronic Studio 2065 (or 2070).

The MICROTRONIC computer can solve problems with very small programs too. A typical case is room monitoring, for example. In combination with a simple light barrier and the following program, MICROTRONIC can determine whether and when the lighting in a room was switched on or off.

### Function description:

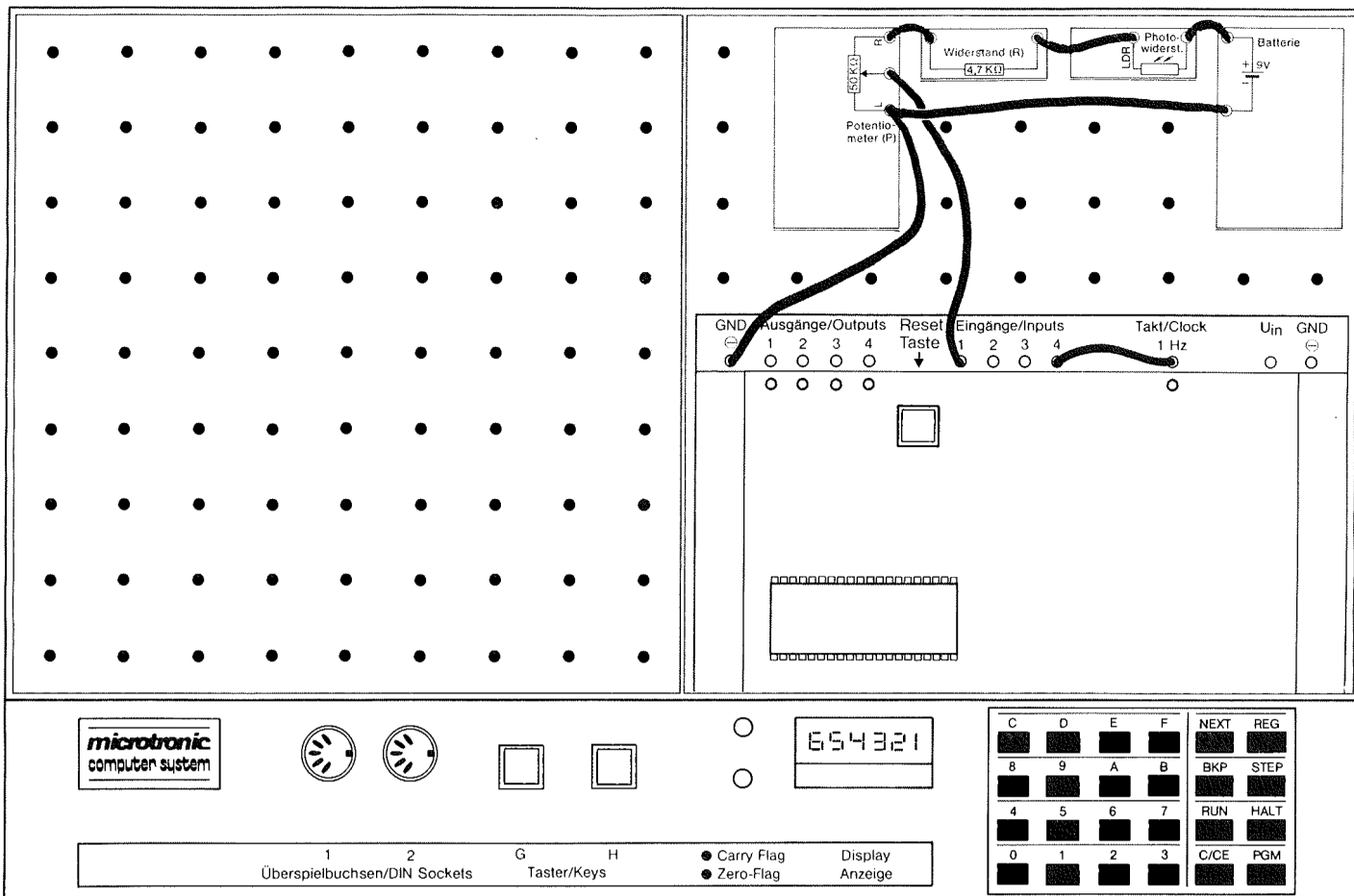
After HALT – NEXT – 00 the short program is entered as per the table. Then the small electronic circuit is built (as shown in the illustration). Then, after HALT – PGM – 3, enter the time of day as 4 digits (9.30 = 0930).

Before the program is started the photoresistor has to be adjusted correctly with the potentiometer. A helper program is available for this, which is started with HALT – NEXT – 0 – A – RUN. The potentiometer is set so that the LED at output 1 lights up under normal room lighting. With the room darkened (the photoresistor darkened) the LED at output 1 must go out.

Now the room to be monitored is darkened and the program is started with HALT – NEXT – 00 – RUN. The display shows the running time of day. As soon as a light is switched on (the photoresistor is illuminated), the time stops on the display and you can read off when the light was switched on. Pressing key 0 continues the program (with the room darkened again).

With this program you can monitor whether and when a light was switched on in a room. The time stays shown on the display even if the light is switched off again.





With a slight change to the program the opposite case can also be monitored. By entering HALT – NEXT – 01 109 – NEXT, the program is changed so that the time stops as soon as the photoresistor is darkened. In this way you can monitor whether and when the light was switched off in a lit room.

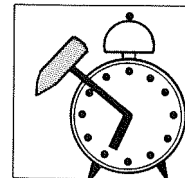
With these programs the exact time of sunrise or sunset could also be determined automatically, for example.

If a program does not work properly, check whether the potentiometer has been adjusted correctly. To do this enter HALT – NEXT – 0 – A – RUN, set the potentiometer (as described) and start the program.

| Adr.Befehl |     | Mnemonic |
|------------|-----|----------|
| 00         | F08 | CLEAR    |
| 01         | 119 | MOVI 1,9 |
| 02         | F06 | TIME     |
| 03         | F6A | DISP 6,A |
| 04         | FD8 | DIN 8    |
| 05         | 898 | CMP 9,8  |
| 06         | E08 | BRZ 08   |
| 07         | C02 | GOTO 02  |
| 08         | FF7 | KIN 7    |
| 09         | C00 | GOTO 00  |
| 0A         | F02 | DISOUT   |
| 0B         | FD0 | DIN 0    |
| 0C         | FE0 | DOT 0    |
| 0D         | C0A | GOTO 0A  |

Programmer: Holger Witt, 4006 Erkrath

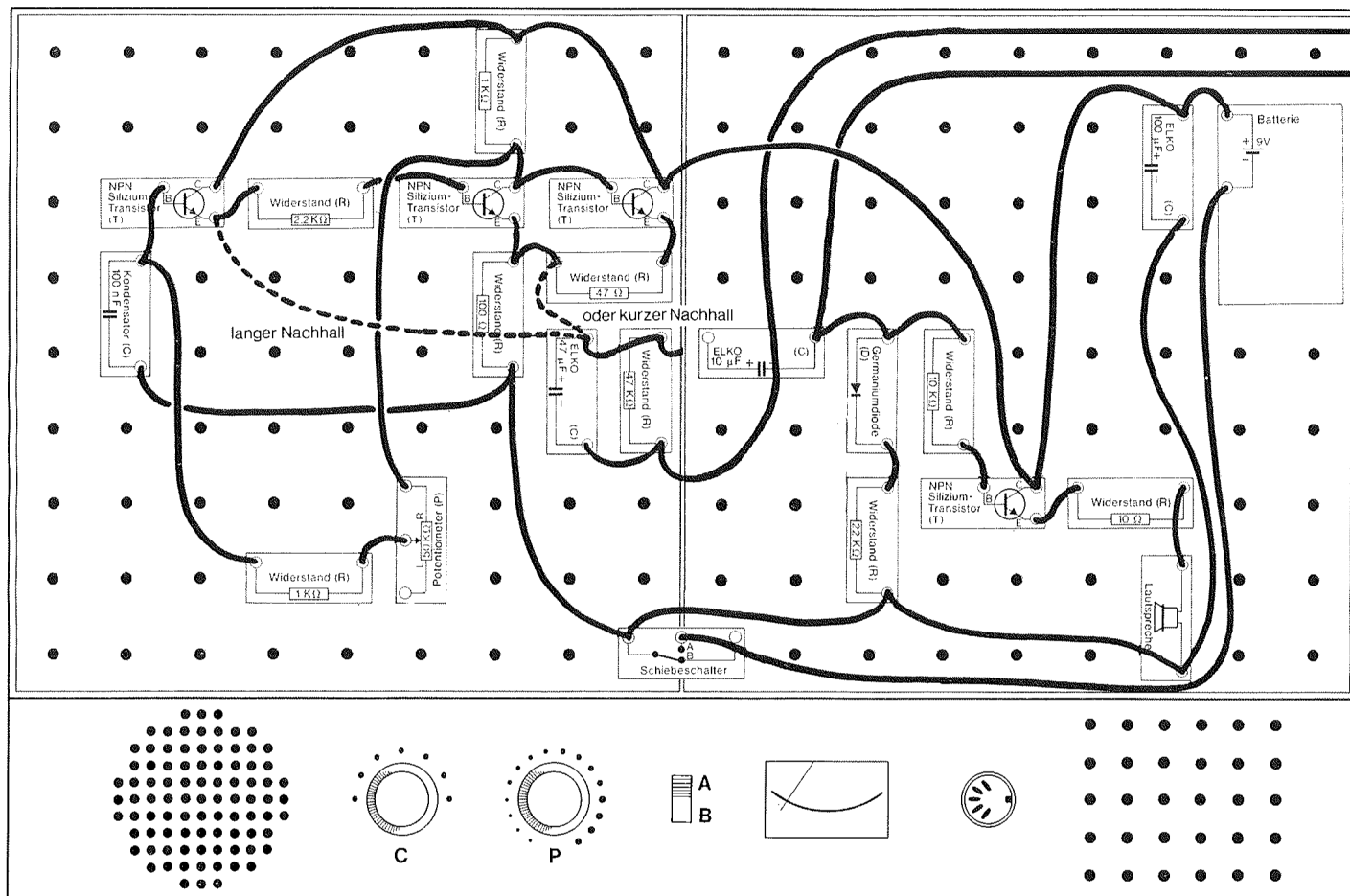
# Clock with chimes, date display and alarm function

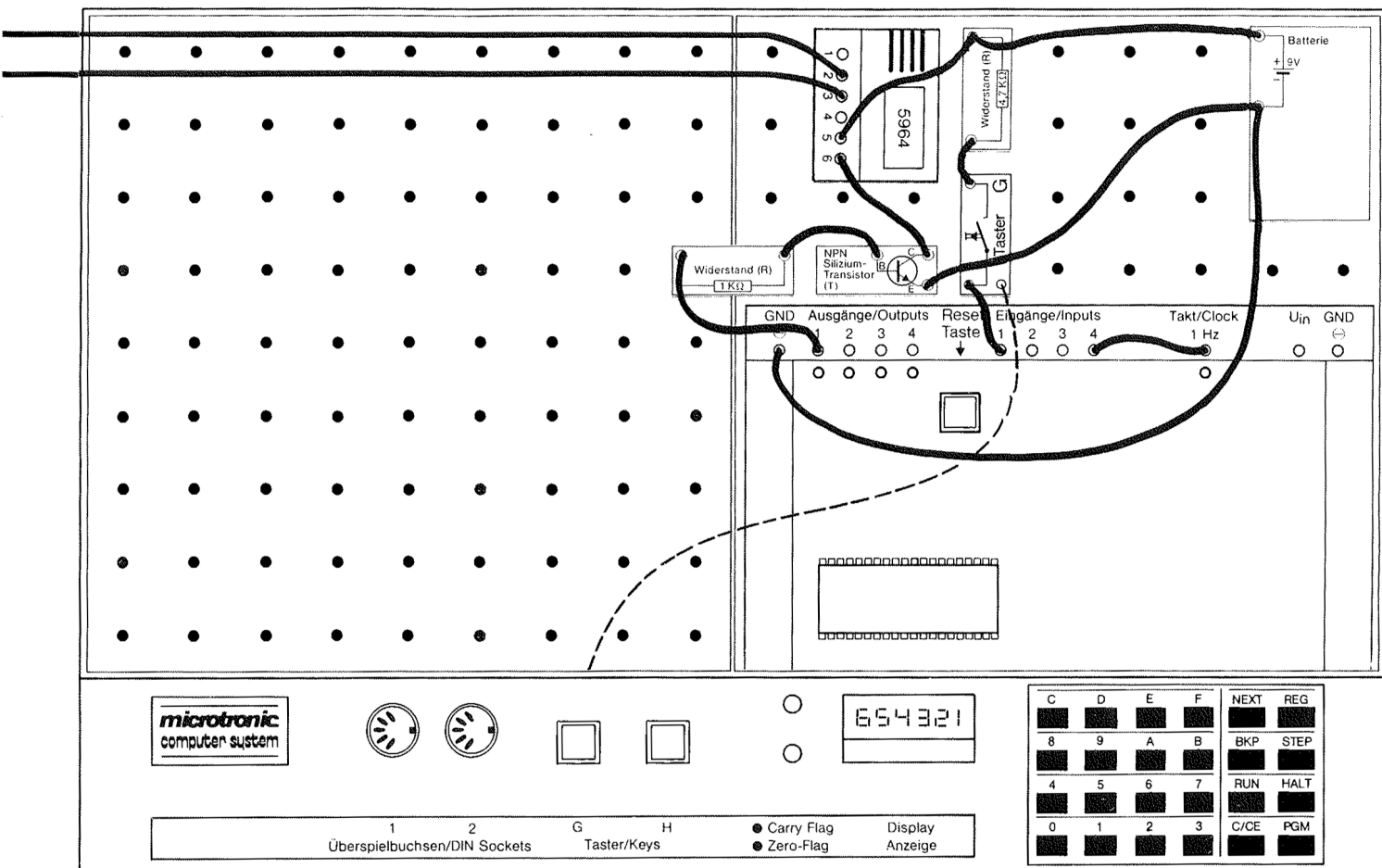


With this program MICROTRONIC becomes a “multi-function clock”. On every full hour the computer strikes the hour (like a church tower clock). The display alternately shows the time for 7 seconds and the date for 3 seconds. An alarm time can also be entered (for triggering an alarm, etc.). To produce the chime true to life, the BUSCH Electronic Studio 2070 and

special relay 5964 are needed. To try the program out (without the chime), the piezo buzzer can be used.

Additionally required: Electronic Studio 2070 and BUSCH special relay 5964.





## Function description:

After HALT – NEXT – 00 the program is entered according to the table. Then the additional electronic circuit (see illustration) is built with the Electronic Studio 2070 and connected to the computer. (If the Electronic Studio IC amplifier set 2072 is available as well, the circuit can be built as shown in illustration 20 of the 2072 instruction book. In that case the push-button is replaced by the special relay 5964.) Without an Electronic Studio 2070 the piezo buzzer can be connected to the GND and output 1 sockets as a makeshift. A true-to-life chime is then not possible, however. In every case make sure that, as shown in the illustration, push-button G is connected to input 1 and that a connection is made from the Takt/Clock socket to input 4.

The “multi-function clock” is now ready for use. First the time has to be entered. To do this press the keys HALT – PGM – 3 and enter the time as 4 digits (e.g. 7.30 = entry: 0730). Then start the program with HALT – NEXT – 00 – RUN and switch on the additional circuit on the Electronic Studio 2070 with the slide switch.

The display shows: D0000. Now follows the 4-digit entry of the date (e.g. 20 March = 2003). The two left-hand display places show the day of the month, the right-hand display places the month. Then key A still has to be pressed. The display now shows the time for 7 seconds and the date for 3 seconds alternately. On every full hour the corresponding chime is produced (e.g. at 10 o'clock in the morning 10 chimes are produced, 3 p.m. gives 3 chimes).

The multi-function clock can be used in continuous operation. The day of the month is changed automatically at midnight. The computer also takes into account that January has 31 days or February only 28 days, for example. Only in leap years does the date have to be changed manually.

An alarm time can also be entered. To do this, push-button G is held down until AAAA appears on the display. Enter the alarm time (4 digits) and press key B. If the alarm time is to be cleared, enter AAAA (in place of the alarm time already there).

At the alarm time a continuous tone is produced. It can be switched off by pressing any number key.

Programmers: Frank Simon, 6342 Haiger  
Michael Bahn, 5000 Cologne

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FE0    | DOT 0    |
| 02   | 1A1    | MOVI A,1 |
| 03   | 1A2    | MOVI A,2 |
| 04   | 1A3    | MOVI A,3 |
| 05   | 1A4    | MOVI A,4 |
| 06   | 1D9    | MOVI D,9 |
| 07   | F55    | DISP 5,5 |
| 08   | FF0    | KIN 0    |
| 09   | 990    | CMPI 9,0 |
| 0A   | D10    | BRC 10   |
| 0B   | 078    | MOV 7,8  |
| 0C   | 067    | MOV 6,7  |
| 0D   | 056    | MOV 5,6  |
| 0E   | 005    | MOV 0,5  |
| 0F   | C08    | GOTO 08  |

|    |     |          |
|----|-----|----------|
| 10 | F6A | DISP 6,A |
| 11 | F06 | TIME     |
| 12 | 90A | CMPI 0,A |
| 13 | E25 | BRZ 25   |
| 14 | 97A | CMPI 7,A |
| 15 | E23 | BRZ 23   |
| 16 | F00 | DIN 0    |
| 17 | 310 | ANDI 1,0 |
| 18 | 900 | CMPI 0,0 |
| 19 | E11 | BRZ 11   |
| 1A | F41 | DISP 4,1 |
| 1B | FF0 | KIN 0    |
| 1C | 9A0 | CMPI A,0 |
| 1D | D10 | BRC 10   |
| 1E | 034 | MOV 3,4  |
| 1F | 023 | MOV 2,3  |

|    |     |          |
|----|-----|----------|
| 20 | 012 | MOV 1,2  |
| 21 | 001 | MOV 0,1  |
| 22 | C1B | GOTO 1B  |
| 23 | F45 | DISP 4,5 |
| 24 | C16 | GOTO 16  |
| 25 | F6A | DISP 6,A |
| 26 | 90B | CMPI 0,B |
| 27 | D16 | BRC 16   |
| 28 | 90C | CMPI 0,C |
| 29 | D77 | BRC 77   |
| 2A | 90D | CMPI 0,D |
| 2B | D77 | BRC 77   |
| 2C | 90F | CMPI 0,F |
| 2D | D34 | BRC 34   |
| 2E | 90E | CMPI 0,E |
| 2F | E32 | BRZ 32   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | 0E9    | MOV E,9  |
| 31   | C40    | GOTO 40  |
| 32   | 1C9    | MOVI C,9 |
| 33   | C40    | GOTO 40  |
| 34   | 91F    | CMPI 1,F |
| 35   | D3E    | BRC 3E   |
| 36   | 92E    | CMPI 2,E |
| 37   | D3B    | BRC 3B   |
| 38   | 1A9    | MOVI A,9 |
| 39   | 4E9    | ADD E,9  |
| 3A   | C40    | GOTO 40  |
| 3B   | 0E9    | MOV E,9  |
| 3C   | 729    | SUBI 2,9 |
| 3D   | C40    | GOTO 40  |
| 3E   | 0E9    | MOV E,9  |
| 3F   | 589    | ADDI 8,9 |

|    |     |          |
|----|-----|----------|
| 40 | C89 | GOTO 89  |
| 41 | 90E | CMPI 0,E |
| 42 | D77 | BRC 77   |
| 43 | 90F | CMPI 0,F |
| 44 | D77 | BRC 77   |
| 45 | 916 | CMPI 1,6 |
| 46 | E49 | BRZ 49   |
| 47 | 109 | MOVI 0,9 |
| 48 | C4A | GOTO 4A  |
| 49 | 1A9 | MOVI A,9 |
| 4A | 459 | ADD 5,9  |
| 4B | 979 | CMPI 7,9 |
| 4C | D4E | BRC 4E   |
| 4D | C4F | GOTO 4F  |
| 4E | 759 | SUBI 5,9 |
| 4F | 929 | CMPI 2,9 |

|    |     |          |
|----|-----|----------|
| 50 | E5A | BRZ 5A   |
| 51 | 938 | CMPI 3,8 |
| 52 | E60 | BRZ 60   |
| 53 | 517 | ADDI 1,7 |
| 54 | 997 | CMPI 9,7 |
| 55 | D57 | BRC 57   |
| 56 | C77 | GOTO 77  |
| 57 | 567 | ADDI 6,7 |
| 58 | 518 | ADDI 1,8 |
| 59 | C77 | GOTO 77  |
| 5A | 987 | CMPI 8,7 |
| 5B | E5D | BRZ 5D   |
| 5C | C53 | GOTO 53  |
| 5D | 928 | CMPI 2,8 |
| 5E | E67 | BRZ 67   |
| 5F | C53 | GOTO 53  |

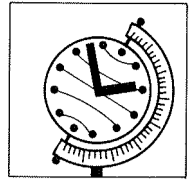
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 949    | CMPI 4,9 |
| 61   | E67    | BRZ 67   |
| 62   | 969    | CMPI 6,9 |
| 63   | E67    | BRZ 67   |
| 64   | 917    | CMPI 1,7 |
| 65   | E67    | BRZ 67   |
| 66   | C77    | GOTO 77  |
| 67   | 108    | MOVI 0,8 |
| 68   | 117    | MOVI 1,7 |
| 69   | 515    | ADDI 1,5 |
| 6A   | 995    | CMPI 9,5 |
| 6B   | D6D    | BRC 6D   |
| 6C   | C70    | GOTO 70  |
| 6D   | 516    | ADDI 1,6 |
| 6E   | 565    | ADDI 6,5 |
| 6F   | C77    | GOTO 77  |

|    |     |          |
|----|-----|----------|
| 70 | 906 | CMPI 0,6 |
| 71 | E77 | BRZ 77   |
| 72 | 935 | CMPI 3,5 |
| 73 | E75 | BRZ 75   |
| 74 | C77 | GOTO 77  |
| 75 | 106 | MOVI 0,6 |
| 76 | 115 | MOVI 1,5 |
| 77 | 8F4 | CMP F,4  |
| 78 | E7A | BRZ 7A   |
| 79 | C16 | GOTO 16  |
| 7A | 8E3 | CMP E,3  |
| 7B | E7D | BRZ 7D   |
| 7C | C16 | GOTO 16  |
| 7D | 8D2 | CMP D,2  |
| 7E | E80 | BRZ 80   |
| 7F | C16 | GOTO 16  |

|    |     |          |
|----|-----|----------|
| 80 | 8C1 | CMP C,1  |
| 81 | E83 | BRZ 83   |
| 82 | C16 | GOTO 16  |
| 83 | 1F0 | MOVI F,0 |
| 84 | FE0 | DOT 0    |
| 85 | FF0 | KIN 0    |
| 86 | 100 | MOVI 0,0 |
| 87 | FE0 | DOT 0    |
| 88 | C10 | GOTO 10  |
| 89 | 1F0 | MOVI F,0 |
| 8A | FE0 | DOT 0    |
| 8B | 710 | SUBI 1,0 |
| 8C | E8E | BRZ 8E   |
| 8D | C8B | GOTO 8B  |
| 8E | FE0 | DOT 0    |
| 8F | 510 | ADDI 1,0 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 90   | E92    | BRZ 92   |
| 91   | C8F    | GOTO 8F  |
| 92   | 510    | ADDI 1,0 |
| 93   | E95    | BRZ 95   |
| 94   | C92    | GOTO 92  |
| 95   | 719    | SUBI 1,9 |
| 96   | E41    | BRZ 41   |
| 97   | C89    | GOTO 89  |

# World Time Calculation



With the World Time Calculation program the time can be worked out for any place on earth. All you need to know for this is the meridian of longitude the place lies on.

The earth is divided into so-called time zones. These time zones mean that within an individual country the same time applies everywhere. Otherwise the time in the west of Germany would differ from the time in the east of Germany by several minutes, for example. The World Time Calculation program does not take these time zones into account, but calculates the actual time (position of the sun) for the corresponding meridian of longitude.

## Function description:

After HALT – NEXT – 00 the program is entered according to the table. Then input 4 is connected to the Takt/Clock socket. After HALT – PGM – 3 the Central European Time (CET) is entered. During summer time 1 hour has to be added to CET.

Program start: HALT – NEXT – 00 – RUN.

The display shows: 0000. Say the computer is to work out the time for Tokyo. Tokyo lies on the 138th meridian (entry: 138). Then either key 0 has to be pressed (for the time of a city that lies to the east) or key 1 (if the city lies to the west). As Tokyo lies to the east of us, key 0 is pressed.

After a short calculating time the time for this meridian appears. If it is 17.36 CET here, for example, the result for Tokyo is 2.48. For a new time calculation press key 0. The display shows 0000 again.

If CET is to be displayed, either 0 degrees west (that is, 0001) or 360 degrees east (that is, 3600) has to be entered.

Programmer: Michael Stapfer, 8850 Donauwörth

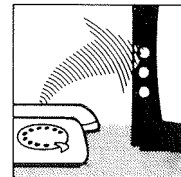
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F0D    | EXRL     |
| 02   | F0E    | EXRM     |
| 03   | F08    | CLEAR    |
| 04   | F46    | DISP 4,6 |
| 05   | FF8    | KIN 8    |
| 06   | FF7    | KIN 7    |
| 07   | FF6    | KIN 6    |
| 08   | FF9    | KIN 9    |
| 09   | 909    | CMPI 0,9 |
| 0A   | E21    | BRZ 21   |
| 0B   | F02    | DISOUT   |
| 0C   | 06D    | MOV 6,D  |
| 0D   | 07E    | MOV 7,E  |
| 0E   | 08F    | MOV 8,F  |
| 0F   | F01    | NOP      |
|      |        |          |
| 10   | F04    | DZHX     |
| 11   | 186    | MOVI 8,6 |
| 12   | 167    | MOVI 6,7 |
| 13   | 118    | MOVI 1,8 |
| 14   | 6D6    | SUB D,6  |
| 15   | FC7    | SUBC 7   |
| 16   | 6E7    | SUB E,7  |
| 17   | FC8    | SUBC 8   |
| 18   | 6F8    | SUB F,8  |
| 19   | 08F    | MOV 8,F  |
| 1A   | 07E    | MOV 7,E  |
| 1B   | 06D    | MOV 6,D  |
| 1C   | F03    | HXDZ     |
| 1D   | 0D0    | MOV D,0  |
| 1E   | 0E1    | MOV E,1  |
| 1F   | 0F2    | MOV F,2  |
|      |        |          |
| 20   | C24    | GOTO 24  |
| 21   | 060    | MOV 6,0  |
| 22   | 071    | MOV 7,1  |
| 23   | 082    | MOV 8,2  |
| 24   | F02    | DISOUT   |
| 25   | F0D    | EXRL     |
| 26   | 140    | MOVI 4,0 |
| 27   | F0B    | MULT     |
| 28   | F0D    | EXRL     |
| 29   | 161    | MOVI 6,1 |
| 2A   | 100    | MOVI 0,0 |
| 2B   | F0C    | DIV      |
| 2C   | F0D    | EXRL     |
| 2D   | F06    | TIME     |
| 2E   | 0DE    | MOV D,E  |
| 2F   | 0CD    | MOV C,D  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 30   | 10F    | MOVI 0,F |
| 31   | F04    | DZHX     |
| 32   | 0D8    | MOV D,8  |
| 33   | 0E9    | MOV E,9  |
| 34   | 10A    | MOVI 0,A |
| 35   | 00D    | MOV 0,D  |
| 36   | 01E    | MOV 1,E  |
| 37   | 10F    | MOVI 0,F |
| 38   | F04    | DZHX     |
| 39   | 4D8    | ADD D,8  |
| 3A   | FB9    | ADC 9    |
| 3B   | 4E9    | ADD E,9  |
| 3C   | FBA    | ADC A    |
| 3D   | 0AF    | MOV A,F  |
| 3E   | 09E    | MOV 9,E  |
| 3F   | 08D    | MOV 8,D  |
|      |        |          |
| 40   | F03    | HXDZ     |
| 41   | 0FA    | MOV F,A  |
| 42   | 0E9    | MOV E,9  |
| 43   | 0D8    | MOV D,8  |
| 44   | F0D    | EXRL     |
| 45   | 00D    | MOV 0,D  |
| 46   | 01E    | MOV 1,E  |
| 47   | 10F    | MOVI 0,F |
| 48   | F04    | DZHX     |
| 49   | 0A2    | MOV A,2  |
| 4A   | 091    | MOV 9,1  |
| 4B   | 080    | MOV 8,0  |
| 4C   | 0D8    | MOV D,8  |
| 4D   | 0E9    | MOV E,9  |
| 4E   | 10A    | MOVI 0,A |
| 4F   | F0D    | EXRL     |
|      |        |          |
| 50   | 161    | MOVI 6,1 |
| 51   | 100    | MOVI 0,0 |
| 52   | F0C    | DIV      |
| 53   | 00D    | MOV 0,D  |
| 54   | 01E    | MOV 1,E  |
| 55   | 02F    | MOV 2,F  |
| 56   | F04    | DZHX     |
| 57   | 4D8    | ADD D,8  |
| 58   | FB9    | ADC 9    |
| 59   | 4E9    | ADD E,9  |
| 5A   | FBA    | ADC A    |
| 5B   | F0D    | EXRL     |
| 5C   | 006    | MOV 0,6  |
| 5D   | 017    | MOV 1,7  |
| 5E   | F06    | TIME     |
| 5F   | 0ED    | MOV E,D  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 60   | 0FE    | MOV F,E  |
| 61   | 10F    | MOVI 0,F |
| 62   | F04    | DZHX     |
| 63   | 4D8    | ADD D,8  |
| 64   | FB9    | ADC 9    |
| 65   | 4E9    | ADD E,9  |
| 66   | 08D    | MOV 8,D  |
| 67   | 09E    | MOV 9,E  |
| 68   | 10F    | MOVI 0,F |
| 69   | F03    | HXDZ     |
| 6A   | 0D0    | MOV D,0  |
| 6B   | 0E1    | MOV E,1  |
| 6C   | F0D    | EXRL     |
| 6D   | 140    | MOVI 4,0 |
| 6E   | 121    | MOVI 2,1 |
| 6F   | F0C    | DIV      |
|      |        |          |
| 70   | F0D    | EXRL     |
| 71   | 008    | MOV 0,8  |
| 72   | 019    | MOV 1,9  |
| 73   | F46    | DISP 4,6 |
| 74   | FFF    | KIN F    |
| 75   | C00    | GOTO 00  |



# Acoustic Computer Remote Switching



With this program electrical devices can be switched on and off “acoustically by remote control”. An electronic circuit (a colour-organ type circuit) converts sounds into low (0) and high (1) signals. The computer evaluates these 0 and 1 signals accordingly, so that its outputs are switched on and off by sounds that can be determined in advance.

## Function description:

After HALT – NEXT – 00 the program is entered as per the table. Then the electronic circuit (as per the assembly plan) is connected to the computer's input 1 and GND sockets.

The circuit has to be adjusted with the potentiometer so that the light bulb is switched on by hand-clapping or a telephone ringing, for example, and goes out again after a short time.

Program start: HALT – NEXT – 00 – RUN.

The display shows 0. The computer is waiting for the entry of how many sounds (hand claps, telephone rings or similar) are to switch its outputs on or off. Any entry number between 1 and 9 is possible, or the letters (A = 10, B = 11, C = 12, D = 13, E = 14, F = 15).

Example: the number 4 (for four sound intervals) is entered. The display goes dark. The computer outputs can now be switched on by clapping four times. Take care, however, to leave a short pause between each clap until the light bulb (of the circuit) has switched itself off again. In the same way (four claps) the computer outputs can also be switched off.

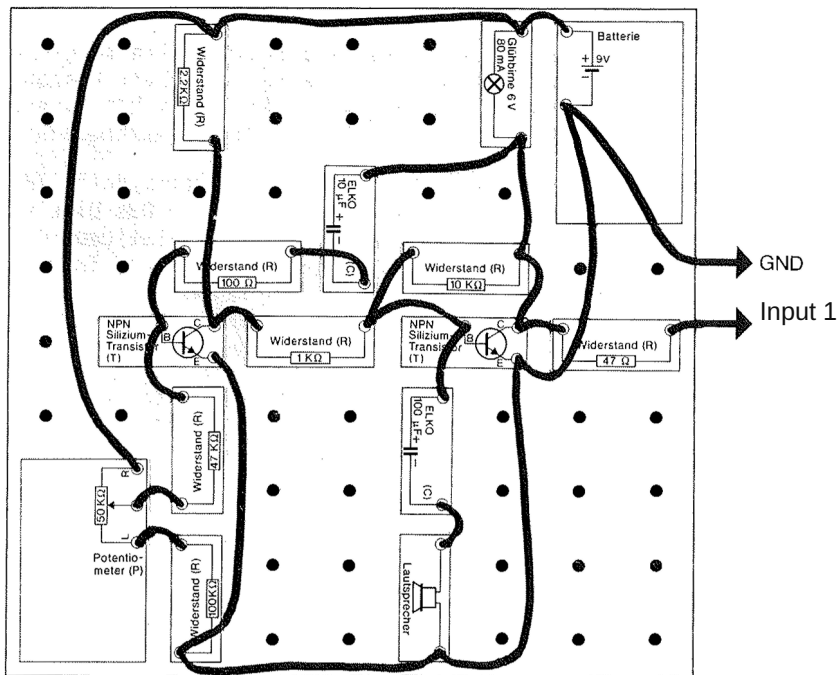
## Important!

The program has some special features so that the switching cannot be carried out by unauthorised people. In our example we entered that the switching is triggered by four sounds. The computer counts the sounds internally (which are registered through the colour-organ type circuit). It is a condition, however, that the sound intervals are made without long pauses. If there is a pause of more than 3 seconds between the second and third clap, for example, the computer assumes that the sound entry has been broken off. After a break-off the following sounds are registered as a new start, that is, a third and fourth clap do not trigger any switching.

Once the computer has registered the number of sounds entered (4 in our example), its outputs are not switched on or off immediately. It now waits about five seconds and checks whether further sounds are received during that time. If a fifth clap comes within this five-second period, for example, the program is broken off by the computer. To trigger the switching, 4 sound intervals are required in this example. Fewer (3) or more (5) are not accepted.

Thanks to this checking, the program works absolutely reliably as a telephone switch, for example. The electronic sound-receiving circuit is set up so that a telephone ringing is registered. When dialling this telephone you have to let it ring as many times as was entered into the computer. If a stranger calls, they can only trigger the switching if they happen by chance to reach the number of sounds entered into the computer.

Programmer: Peter Polzer, 7000 Stuttgart



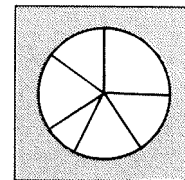
**Additionally required: BUSCH Electronic Studio 2060 (or 2065 or 2070).**

### Important note!

For telephone experiments, note that the Deutsche Bundespost forbids data transmission over its own telephone lines as a matter of principle, unless a corresponding application has been approved. Trials should therefore only be carried out with a house telephone system.

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F02    | DISOUT   |
| 02   | FE2    | DOT 2    |
| 03   | F10    | DISP 1,0 |
| 04   | FF0    | KIN 0    |
| 05   | F02    | DISOUT   |
| 06   | FD1    | DIN 1    |
| 07   | 311    | ANDI 1,1 |
| 08   | 911    | CMPI 1,1 |
| 09   | E16    | BRZ 16   |
| 0A   | 812    | CMP 1,2  |
| 0B   | E06    | BRZ 06   |
| 0C   | 012    | MOV 1,2  |
| 0D   | 911    | CMPI 1,1 |
| 0E   | E06    | BRZ 06   |
| 0F   | 103    | MOVI 0,3 |
|      |        |          |
| 10   | 804    | CMP 0,4  |
| 11   | E14    | BRZ 14   |
| 12   | 514    | ADDI 1,4 |
| 13   | C06    | GOTO 06  |
| 14   | 104    | MOVI 0,4 |
| 15   | C06    | GOTO 06  |
| 16   | B25    | CALL 25  |
| 17   | D19    | BRC 19   |
| 18   | C0A    | GOTO 0A  |
| 19   | 804    | CMP 0,4  |
| 1A   | E1E    | BRZ 1E   |
| 1B   | 104    | MOVI 0,4 |
| 1C   | 103    | MOVI 0,3 |
| 1D   | C0A    | GOTO 0A  |
| 1E   | 9FF    | CMPI F,F |
| 1F   | E22    | BRZ 22   |
|      |        |          |
| 20   | 1FF    | MOVI F,F |
| 21   | C23    | GOTO 23  |
| 22   | 10F    | MOVI 0,F |
| 23   | FEF    | DOT F    |
| 24   | C1B    | GOTO 1B  |
| 25   | 527    | ADDI 2,7 |
| 26   | D28    | BRC 28   |
| 27   | C25    | GOTO 25  |
| 28   | 513    | ADDI 1,3 |
| 29   | F07    | RET      |

# For statisticians: drawing pie charts



So-called “pie charts” are often used for the graphical presentation of tables or statistics. For this it is necessary to divide the percentage shares into “slices” of the right size (see illustration). The following program is a valuable aid in preparing pie charts for drawing. From the percentage values entered, the computer works out the angles in degrees for the corresponding sectors of the circle.

## Function description:

After HALT – NEXT – 00 the program is entered according to the table.

Program start: HALT – NEXT – 00 – RUN.

The display shows: 00.

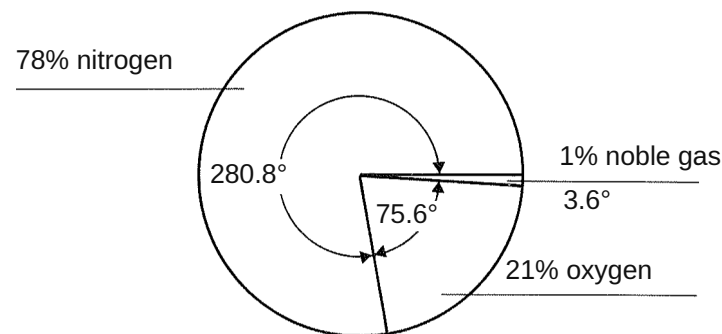
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F02    | DISOUT   |
| 01   | F08    | CLEAR    |
| 02   | 160    | MOVI 6,0 |
| 03   | 131    | MOVI 3,1 |
| 04   | F0D    | EXRL     |
| 05   | F08    | CLEAR    |
| 06   | F20    | DISP 2,0 |
| 07   | FF6    | KIN 6    |
| 08   | 9A6    | CMPI A,6 |
| 09   | E0D    | BRZ 0D   |
| 0A   | 001    | MOV 0,1  |
| 0B   | 060    | MOV 6,0  |
| 0C   | C07    | GOTO 07  |
| 0D   | F0B    | MULT     |
| 0E   | F40    | DISP 4,0 |
| 0F   | FF6    | KIN 6    |
| 10   | C00    | GOTO 00  |

## Example of use:

The proportions making up air are to be shown in a pie chart. Air consists of 78% nitrogen, 21% oxygen and 1% noble gases. These percentages have to be converted into angles in degrees.

We enter the first percentage, for nitrogen (78). Then press key A. The display shows: 2808 = 280.8 degrees. Press key A. Enter the second value (21) for oxygen, press key A, the display shows 756 = 75.6 degrees. Key A, then the last entry for noble gases (1). Key A once more – display result: 36 (3.6 degrees).

With the values in degrees calculated by the computer, a pie chart with the corresponding sectors can easily be laid out and drawn:

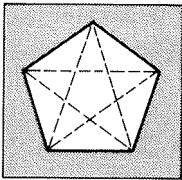


In this way any number of divisions of the circle can be made. All you have to make sure of is that all the values of the circle's contents add up to 100% (as in our example 78% + 21% + 1%).

The way the program works is simple: the percentage entered is multiplied by 3.6 and you get the result in degrees. The program demonstrates how an individually programmed computer can be used as an important aid for a wide variety of tasks.

Programmer: Thorsten Lindner, 5600 Wuppertal 2

# Diagonal calculation for polygons



With this program the computer works out how many ways there are of drawing diagonals into a polygon. It is well known, for example, that in a quadrilateral only two diagonals are possible, while in a pentagon there are already five different possibilities (see drawing). The program can calculate the number of diagonals of all polygons with up to a maximum of 99 corners.

## Function description:

After HALT – NEXT – 00 the program is entered as per the table.

Program start: HALT – NEXT – 00 – RUN.

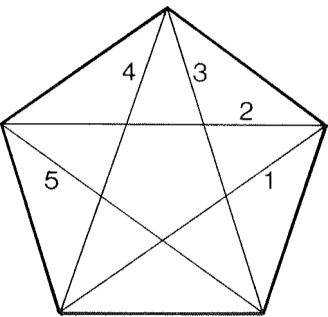
The display shows: 00. Entry e.g.: 10 (for a decagon). Then press key A. The result is shown immediately as four digits (0035 in this example), that is, 35 diagonals are possible in a decagon.

If key A is then pressed, the display shows 00 again. A new number for a polygon can be entered.

The number of diagonals is calculated by the following formula:

$$d = (n \times (n - 3)) : 2$$

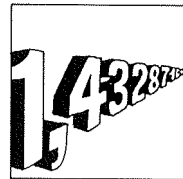
d = number of diagonals  
n = number of corners



| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F20    | DISP 2,0 |
| 02   | FFF    | KIN F    |
| 03   | 99F    | CMPI 9,F |
| 04   | D08    | BRC 08   |
| 05   | 001    | MOV 0,1  |
| 06   | 0F0    | MOV F,0  |
| 07   | C02    | GOTO 02  |
| 08   | F02    | DISOUT   |
| 09   | F70    | MAS 0    |
| 0A   | F71    | MAS 1    |
| 0B   | 730    | SUBI 3,0 |
| 0C   | D14    | BRC 14   |
| 0D   | FOB    | MULT     |
| 0E   | F0D    | EXRL     |
| 0F   | 120    | MOVI 2,0 |
| 10   | F0C    | DIV      |
| 11   | F40    | DISP 4,0 |
| 12   | FFF    | KIN F    |
| 13   | C00    | GOTO 00  |
| 14   | 760    | SUBI 6,0 |
| 15   | 711    | SUBI 1,1 |
| 16   | C0D    | GOTO 0D  |

Programmer: Peter Feltens, 5450 Neuwied 12

# For mathematicians: a super division program



This big division program shows how MICROTRONIC can carry out complicated calculations too with a simple algorithm, for example division problems with any number of decimal places.

Example:  $787 : 63 = 12.4920634920634920\dots$  With this problem MICROTRONIC will go on calculating and displaying decimal places until the program is broken off.

## Function description:

After HALT – NEXT – 00 enter the program as per the table.

Program start: HALT – NEXT – 00 – RUN.

The display shows: A0. The computer now offers three different possibilities.

### First possibility:

Calculation of three decimal places, the last place being rounded up or down automatically. This possibility is selected with key 1. The display shows: 000.

Now the divisor (three places at most) is entered (e.g. 787). The entry is completed by pressing key A. Then the dividend (e.g. 63) is entered. This entry too must be completed with key A. The computer now calculates the division  $787 : 63$ . The result before the decimal point appears on the display: 12. Pressing key 0 shows the three decimal places: 492. The complete result is therefore 12.49. A new program start is made with key 0.

### Second possibility:

Calculation of any number of decimal places. Four places are shown at a time. Pressing key 0 brings the next four places, and so on. The display shows: A0. Select this program run with key 2. Then enter the divisor and dividend. (Example  $14 : 973 =$  entry: 14 then key A, then entry: 973 and key A again.)

The display first shows the result before the decimal point (0). If key 0 is pressed, the first four decimal places are shown (0143). After each press of key 0 the next four places appear. Complete result for this example: 0.01438848... The calculation of further decimal places can be broken off with key F.

### Third possibility:

Calculation of any number of decimal places, four places always being shown one after the other automatically. The display shows: A0. Select this program run with key 3.

After entering the numbers (do not forget key A) the result before the decimal point is shown first. With key 0 four decimal places are shown for a short time. The next four places follow automatically, and so on. The program is broken off and at the same time restarted with HALT – NEXT – 00 – RUN.

## Of interest to the programmer:

The algorithm for the big division program is relatively simple: with a problem such as 787 : 63 the two numbers are divided in the working and storage registers by the division instruction. The working registers then hold the result before the decimal point, 12. The remainder, 31, stays in the storage registers.

The decimal places are calculated further by multiplying the remainder (31) by 10 and dividing the result, 310, by 63 again. The first decimal place, 4, moves into the working register; the remaining remainder (58) is again multiplied by 10 in the storage registers and divided by 63. That gives the second decimal place. This process is repeated automatically until the program is broken off, so that any number of decimal places can be calculated.

Programmer: Michael Stapfer, 8850 Donauwörth

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | 0CD    | MOV C,D  |
| 21   | 0BC    | MOV B,C  |
| 22   | 0AB    | MOV A,B  |
| 23   | 10A    | MOVI 0,A |
| 24   | 51F    | ADDI 1,F |
| 25   | C1E    | GOTO 1E  |
| 26   | F3B    | DISP 3,B |
| 27   | FFF    | KIN F    |
| 28   | C00    | GOTO 00  |
| 29   | 023    | MOV 2,3  |
| 2A   | 012    | MOV 1,2  |
| 2B   | 001    | MOV 0,1  |
| 2C   | 100    | MOVI 0,0 |
| 2D   | F0D    | EXRL     |
| 2E   | F0E    | EXRM     |
| 2F   | 0A0    | MOV A,0  |

|    |     |          |
|----|-----|----------|
| 30 | 0B1 | MOV B,1  |
| 31 | 0C2 | MOV C,2  |
| 32 | F0E | EXRM     |
| 33 | F0C | DIV      |
| 34 | 0CD | MOV C,D  |
| 35 | 0BC | MOV B,C  |
| 36 | 0AB | MOV A,B  |
| 37 | 00A | MOV 0,A  |
| 38 | 51F | ADDI 1,F |
| 39 | 94F | CMPI 4,F |
| 3A | E3C | BRZ 3C   |
| 3B | C17 | GOTO 17  |
| 3C | 91E | CMPI 1,E |
| 3D | D43 | BRC 43   |
| 3E | 95A | CMPI 5,A |
| 3F | D41 | BRC 41   |

|    |     |          |
|----|-----|----------|
| 40 | C26 | GOTO 26  |
| 41 | 51B | ADDI 1,B |
| 42 | C26 | GOTO 26  |
| 43 | 92E | CMPI 2,E |
| 44 | D4B | BRC 4B   |
| 45 | F4A | DISP 4,A |
| 46 | FFF | KIN F    |
| 47 | 9FF | CMPI F,F |
| 48 | E00 | BRZ 00   |
| 49 | 10F | MOVI 0,F |
| 4A | C16 | GOTO 16  |
| 4B | F4A | DISP 4,A |
| 4C | 518 | ADDI 1,8 |
| 4D | 9F8 | CMPI F,8 |
| 4E | E49 | BRZ 49   |
| 4F | C4C | GOTO 4C  |

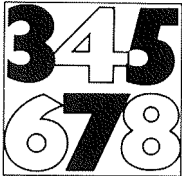
| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 50   | F10    | DISP 1,0 |
| 51   | FF6    | KIN 6    |
| 52   | 996    | CMPI 9,6 |
| 53   | D63    | BRC 63   |
| 54   | 060    | MOV 6,0  |
| 55   | FF6    | KIN 6    |
| 56   | 996    | CMPI 9,6 |
| 57   | D63    | BRC 63   |
| 58   | 001    | MOV 0,1  |
| 59   | 060    | MOV 6,0  |
| 5A   | F20    | DISP 2,0 |
| 5B   | FF6    | KIN 6    |
| 5C   | 996    | CMPI 9,6 |
| 5D   | D63    | BRC 63   |
| 5E   | 012    | MOV 1,2  |
| 5F   | 001    | MOV 0,1  |

|    |     |          |
|----|-----|----------|
| 60 | 060 | MOV 6,0  |
| 61 | F30 | DISP 3,0 |
| 62 | C5B | GOTO 5B  |
| 63 | F07 | RET      |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F0D    | EXRL     |
| 02   | F0E    | EXRM     |
| 03   | F08    | CLEAR    |
| 04   | 1AF    | MOVI A,F |
| 05   | F2E    | DISP 2,E |
| 06   | FFE    | KIN E    |
| 07   | 10F    | MOVI 0,F |
| 08   | F0E    | EXRM     |
| 09   | B50    | CALL 50  |
| 0A   | 00D    | MOV 0,D  |
| 0B   | 01E    | MOV 1,E  |
| 0C   | 02F    | MOV 2,F  |
| 0D   | F0D    | EXRL     |
| 0E   | B50    | CALL 50  |
| 0F   | 00A    | MOV 0,A  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 10   | 01B    | MOV 1,B  |
| 11   | 02C    | MOV 2,C  |
| 12   | F0E    | EXRM     |
| 13   | F0C    | DIV      |
| 14   | F30    | DISP 3,0 |
| 15   | FF0    | KIN 0    |
| 16   | F02    | DISOUT   |
| 17   | F0D    | EXRL     |
| 18   | 902    | CMPI 0,2 |
| 19   | D29    | BRC 29   |
| 1A   | 901    | CMPI 0,1 |
| 1B   | D29    | BRC 29   |
| 1C   | 900    | CMPI 0,0 |
| 1D   | D29    | BRC 29   |
| 1E   | 94F    | CMPI 4,F |
| 1F   | E26    | BRZ 26   |

# For mathematicians: prime number calculation



The definition of prime numbers is given in the “Guinness Book of Records” as follows:

Prime numbers are all whole numbers (with the exception of 1) that are divisible only by themselves and by 1, e.g. 2, 3, 5, 7 or 11.

The smallest prime number is therefore 2; the largest known prime number (a number with 13395 digits) is  $2^{44497} - 1$ . It was reported as the highest prime number on 8 April 1979, after Harry Nelson (47) and David Slowinski (25) had worked on this calculation for two months with the help of a Cray One computer at the University of California (Lawrence Livermore Laboratory).

Prime numbers play a large part in mathematics. Prime number calculation is also often used in computer technology for so-called “benchmark” programs. With benchmark programs the calculating speed of different types of computer, or of different programming languages, can be compared. A computer is made to calculate the first thousand prime numbers one after the other, for example, and the time needed for this is measured. If these calculations are carried out with different computers, you get a very good comparison of their working speed.

With the following program all prime numbers between 0 and 99,999 can be calculated, or particular numbers can be checked to see whether they are a prime number.

## Function description:

After HALT – NEXT – 00 the program is entered carefully as per the table. Then connect the piezo buzzer to output 1 and socket GND. Program start with HALT – NEXT – 00 – RUN.

The display shows: 00000. If key C is pressed, the computer calculates the first prime number. It starts automatically with 1 (although this is not really a prime number at all). The result shown is announced at the same time by a buzzer tone. After each press of key

C the next prime number is calculated (2, 3, 5, etc.). The piezo buzzer can be switched off with key D. With key A the display is set back to 0. If key B is then pressed, any numerical value can be entered for a prime number check.

Example: press keys A and B one after the other. Then enter the number 500. Press key C. The computer checks whether the number entered (500) is a prime number. If it were, the number entered would be displayed again. As in our example the number 500 is not a prime number, the next higher prime number (503) is displayed. Key C can then be pressed again for the calculation of the next prime number.

When calculating larger prime numbers the calculating time may in some circumstances amount to several minutes. The piezo buzzer draws attention to the result found (switch the buzzer off with key D).

Programmer: Klaus Hallatschek,  
9850 Neugablonz-Kaufbeuren

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | FEF    | DOT F    |
| 02   | F50    | DISP 5,0 |
| 03   | FFE    | KIN E    |
| 04   | FE5    | DOT 5    |
| 05   | 9BE    | CMPI B,E |
| 06   | D09    | BRC 09   |
| 07   | E0C    | BRZ 0C   |
| 08   | C00    | GOTO 00  |
| 09   | 9CE    | CMPI C,E |
| 0A   | D02    | BRC 02   |
| 0B   | C16    | GOTO 16  |
| 0C   | FFF    | KIN F    |
| 0D   | 9AF    | CMPI A,F |
| 0E   | EDA    | BRZ DA   |
| 0F   | D18    | BRC 18   |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 10   | 034    | MOV 3,4  |
| 11   | 023    | MOV 2,3  |
| 12   | 012    | MOV 1,2  |
| 13   | 001    | MOV 0,1  |
| 14   | 0F0    | MOV F,0  |
| 15   | C0C    | GOTO 0C  |
| 16   | F02    | DISOUT   |
| 17   | B77    | CALL 77  |
| 18   | F02    | DISOUT   |
| 19   | B90    | CALL 90  |
| 1A   | F0D    | EXRL     |
| 1B   | F08    | CLEAR    |
| 1C   | 168    | MOVI 6,8 |
| 1D   | F0D    | EXRL     |
| 1E   | F0F    | EXRA     |
| 1F   | B77    | CALL 77  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 20   | F0F    | EXRA     |
| 21   | 904    | CMPI 0,4 |
| 22   | D2A    | BRC 2A   |
| 23   | 903    | CMPI 0,3 |
| 24   | D33    | BRC 33   |
| 25   | 902    | CMPI 0,2 |
| 26   | D36    | BRC 36   |
| 27   | 901    | CMPI 0,1 |
| 28   | D3C    | BRC 3C   |
| 29   | C3F    | GOTO 3F  |
| 2A   | 92A    | CMPI 2,A |
| 2B   | D2D    | BRC 2D   |
| 2C   | C41    | GOTO 41  |
| 2D   | 909    | CMPI 0,9 |
| 2E   | D30    | BRC 30   |
| 2F   | C41    | GOTO 41  |

|    |     |          |
|----|-----|----------|
| 30 | 958 | CMPI 5,8 |
| 31 | D3F | BRC 3F   |
| 32 | C41 | GOTO 41  |
| 33 | 90A | CMPI 0,A |
| 34 | D3F | BRC 3F   |
| 35 | C41 | GOTO 41  |
| 36 | 929 | CMPI 2,9 |
| 37 | D39 | BRC 39   |
| 38 | C41 | GOTO 41  |
| 39 | 918 | CMPI 1,8 |
| 3A | D3F | BRC 3F   |
| 3B | C41 | GOTO 41  |
| 3C | 909 | CMPI 0,9 |
| 3D | D3F | BRC 3F   |
| 3E | C41 | GOTO 41  |
| 3F | 1FF | MOVI F,F |

|    |     |          |
|----|-----|----------|
| 40 | C01 | GOTO 01  |
| 41 | F70 | MAS 0    |
| 42 | F71 | MAS 1    |
| 43 | F72 | MAS 2    |
| 44 | F73 | MAS 3    |
| 45 | F74 | MAS 4    |
| 46 | 680 | SUB 8,0  |
| 47 | D49 | BRC 49   |
| 48 | C4C | GOTO 4C  |
| 49 | 760 | SUBI 6,0 |
| 4A | 711 | SUBI 1,1 |
| 4B | D4F | BRC 4F   |
| 4C | 691 | SUB 9,1  |
| 4D | D50 | BRC 50   |
| 4E | C53 | GOTO 53  |
| 4F | 691 | SUB 9,1  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 50   | 761    | SUBI 6,1 |
| 51   | 712    | SUBI 1,2 |
| 52   | D56    | BRC 56   |
| 53   | 6A2    | SUB A,2  |
| 54   | D57    | BRC 57   |
| 55   | C5D    | GOTO 5D  |
| 56   | 6A2    | SUB A,2  |
| 57   | 762    | SUBI 6,2 |
| 58   | 713    | SUBI 1,3 |
| 59   | D5B    | BRC 5B   |
| 5A   | C5D    | GOTO 5D  |
| 5B   | 763    | SUBI 6,3 |
| 5C   | 714    | SUBI 1,4 |
| 5D   | 900    | CMPI 0,0 |
| 5E   | D65    | BRC 65   |
| 5F   | 010    | MOV 1,0  |

|    |     |          |
|----|-----|----------|
| 60 | 021 | MOV 2,1  |
| 61 | 032 | MOV 3,2  |
| 62 | 043 | MOV 4,3  |
| 63 | 104 | MOVI 0,4 |
| 64 | C5D | GOTO 5D  |
| 65 | 904 | CMPI 0,4 |
| 66 | D46 | BRC 46   |
| 67 | 903 | CMPI 0,3 |
| 68 | D46 | BRC 46   |
| 69 | 82A | CMP 2,A  |
| 6A | D1D | BRC 1D   |
| 6B | E6D | BRZ 6D   |
| 6C | C46 | GOTO 46  |
| 6D | 819 | CMP 1,9  |
| 6E | D1D | BRC 1D   |
| 6F | E71 | BRZ 71   |

|    |     |          |
|----|-----|----------|
| 70 | C46 | GOTO 46  |
| 71 | 808 | CMP 0,8  |
| 72 | D1D | BRC 1D   |
| 73 | E75 | BRZ 75   |
| 74 | C46 | GOTO 46  |
| 75 | F0D | EXRL     |
| 76 | C17 | GOTO 17  |
| 77 | 510 | ADDI 1,0 |
| 78 | 990 | CMPI 9,0 |
| 79 | D7B | BRC 7B   |
| 7A | C90 | GOTO 90  |
| 7B | 560 | ADDI 6,0 |
| 7C | 511 | ADDI 1,1 |
| 7D | 991 | CMPI 9,1 |
| 7E | D80 | BRC 80   |
| 7F | C90 | GOTO 90  |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 80   | 561    | ADDI 6,1 |
| 81   | 512    | ADDI 1,2 |
| 82   | 992    | CMPI 9,2 |
| 83   | D85    | BRC 85   |
| 84   | C90    | GOTO 90  |
| 85   | 562    | ADDI 6,2 |
| 86   | 513    | ADDI 1,3 |
| 87   | 993    | CMPI 9,3 |
| 88   | D8A    | BRC 8A   |
| 89   | C90    | GOTO 90  |
| 8A   | 563    | ADDI 6,3 |
| 8B   | 514    | ADDI 1,4 |
| 8C   | 994    | CMPI 9,4 |
| 8D   | D8F    | BRC 8F   |
| 8E   | C90    | GOTO 90  |
| 8F   | 564    | ADDI 6,4 |

|    |     |          |
|----|-----|----------|
| 90 | 910 | CMPI 1,0 |
| 91 | EA9 | BRZ A9   |
| 92 | 920 | CMPI 2,0 |
| 93 | E9D | BRZ 9D   |
| 94 | 930 | CMPI 3,0 |
| 95 | E9D | BRZ 9D   |
| 96 | 970 | CMPI 7,0 |
| 97 | EA9 | BRZ A9   |
| 98 | 990 | CMPI 9,0 |
| 99 | EA9 | BRZ A9   |
| 9A | 950 | CMPI 5,0 |
| 9B | E9D | BRZ 9D   |
| 9C | C77 | GOTO 77  |
| 9D | 904 | CMPI 0,4 |
| 9E | DA6 | BRC A6   |
| 9F | 903 | CMPI 0,3 |

|    |     |          |
|----|-----|----------|
| A0 | DA6 | BRC A6   |
| A1 | 902 | CMPI 0,2 |
| A2 | DA6 | BRC A6   |
| A3 | 901 | CMPI 0,1 |
| A4 | DA6 | BRC A6   |
| A5 | CCF | GOTO CF  |
| A6 | 930 | CMPI 3,0 |
| A7 | EA9 | BRZ A9   |
| A8 | C77 | GOTO 77  |
| A9 | F70 | MAS 0    |
| AA | F71 | MAS 1    |
| AB | F72 | MAS 2    |
| AC | F73 | MAS 3    |
| AD | F74 | MAS 4    |
| AE | 410 | ADD 1,0  |
| AF | 101 | MOVI 0,1 |

| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| B0   | 10F    | MOVI 0,F |
| B1   | DD2    | BRC D2   |
| B2   | 990    | CMPI 9,0 |
| B3   | DD2    | BRC D2   |
| B4   | 420    | ADD 2,0  |
| B5   | 102    | MOVI 0,2 |
| B6   | 11F    | MOVI 1,F |
| B7   | DD2    | BRC D2   |
| B8   | 990    | CMPI 9,0 |
| B9   | DD2    | BRC D2   |
| BA   | 430    | ADD 3,0  |
| BB   | 103    | MOVI 0,3 |
| BC   | 12F    | MOVI 2,F |
| BD   | DD2    | BRC D2   |
| BE   | 990    | CMPI 9,0 |
| BF   | DD2    | BRC D2   |

|    |     |          |
|----|-----|----------|
| C0 | 440 | ADD 4,0  |
| C1 | 104 | MOVI 0,4 |
| C2 | 13F | MOVI 3,F |
| C3 | DD2 | BRC D2   |
| C4 | 990 | CMPI 9,0 |
| C5 | DD2 | BRC D2   |
| C6 | 901 | CMPI 0,1 |
| C7 | DAE | BRC AE   |
| C8 | 930 | CMPI 3,0 |
| C9 | ED0 | BRZ D0   |
| CA | 960 | CMPI 6,0 |
| CB | ED0 | BRZ D0   |
| CC | 990 | CMPI 9,0 |
| CD | ED0 | BRZ D0   |
| CE | F0D | EXRL     |
| CF | F07 | RET      |

|    |     |          |
|----|-----|----------|
| D0 | F0D | EXRL     |
| D1 | C77 | GOTO 77  |
| D2 | 560 | ADDI 6,0 |
| D3 | 511 | ADDI 1,1 |
| D4 | 92F | CMPI 2,F |
| D5 | DC4 | BRC C4   |
| D6 | EBE | BRZ BE   |
| D7 | 90F | CMPI 0,F |
| D8 | DB8 | BRC B8   |
| D9 | CB2 | GOTO B2  |
| DA | F08 | CLEAR    |
| DB | C0C | GOTO 0C  |



## For mathematicians: exponentiation

$$\begin{array}{c} 15 \times 15 \times 15 \\ 15^3 \end{array}$$

The following program shows that besides the four basic arithmetic operations (see the instruction books) MICROTRONIC can also carry out exponentiation. An example of exponentiation is  $57^3$  ( $57 \times 57 \times 57 = 185,193$ ).

### Function description:

After HALT – NEXT – 00 the program is entered according to the table.

Program start with HALT – NEXT – 00 – RUN. The display shows 000.

Example: the problem  $57^3$  is to be calculated. First the number 57 is entered – then key A. The display shows: 00. Now the exponent 3 is entered – key A. After a short calculating time the display shows 185193 as the result. After pressing any number key a new calculation can be carried out.

If the result is too large (more than 6 digits), the display shows: EEEEEEE.

Note that the power 0 to the power of 0 is not calculated correctly, because it is not mathematically defined.

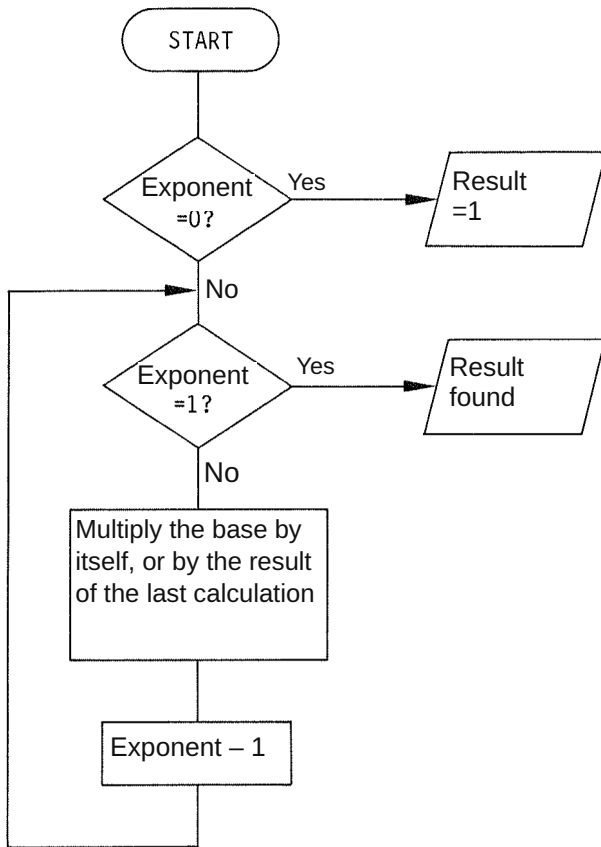
### Of interest to the programmer:

On many pocket calculators exponentiation can be called up by pressing a key. In these calculators the microprocessor is programmed to carry out the exponentiation by repeated multiplication. The algorithm needed for this is quite simple. First it is checked whether the exponent is a 1. If yes, the base (57 in our example) is the result. If the exponent is greater than 1, the base is multiplied by itself ( $57 \times 57$  in our example) and 1 is subtracted from the exponent entered. If the exponent is now 1, the result has been found. If no, the result of the last calculation ( $57 \times 57 = 3,249$ ) is multiplied by the base entered (57 in our example) again. 1 is again subtracted from the exponent and it is then checked whether the exponent is now equal to 1 and the result has thus been found.

The flow chart makes this principle clear. Note that the flow chart shows only the actual calculating part. Special input and output routines are of course also needed for the program. The flow chart shows one more special feature: in exponentiation it has been defined that the result is 1 if the exponent is 0. For this reason it is checked before the actual calculation whether the exponent is equal to 0 – if yes, the result is automatically 1.

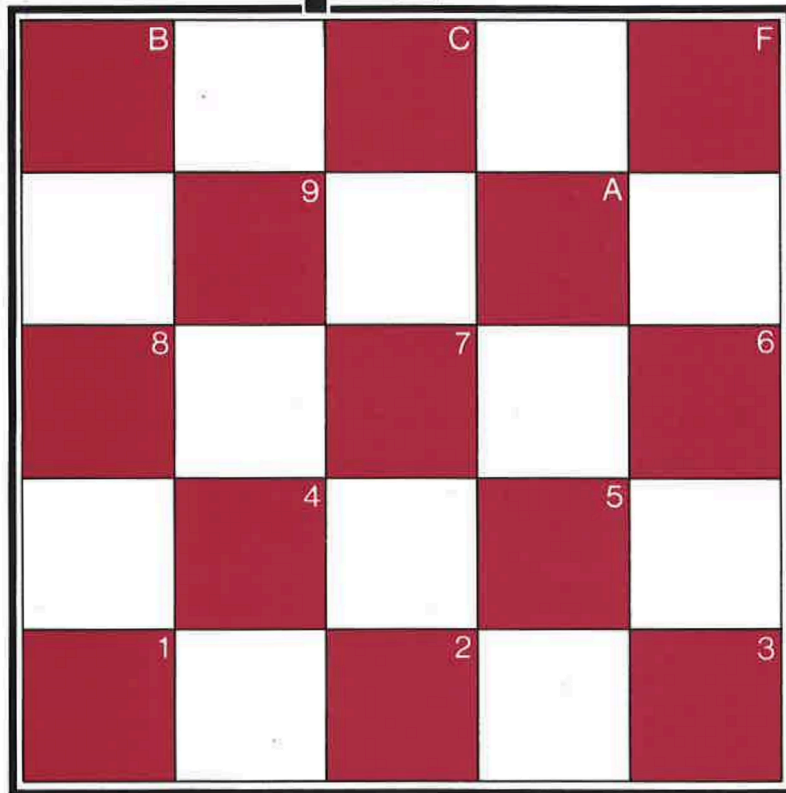
Programmer: Matthias Gorek, 3210 Elze

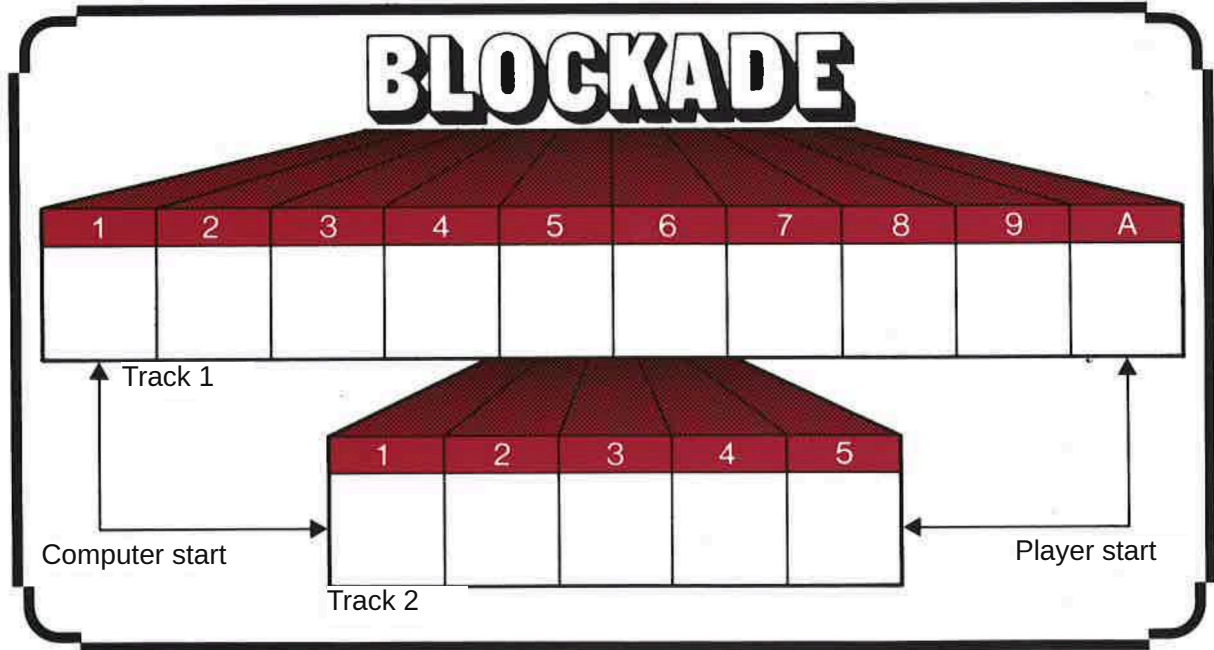
Flow chart



| Adr. | Befehl | Mnemonic |
|------|--------|----------|
| 00   | F08    | CLEAR    |
| 01   | F30    | DISP 3,0 |
| 02   | FF6    | KIN 6    |
| 03   | 996    | CMPI 9,6 |
| 04   | D09    | BRC 09   |
| 05   | 012    | MOV 1,2  |
| 06   | 001    | MOV 0,1  |
| 07   | 060    | MOV 6,0  |
| 08   | C02    | GOTO 02  |
| 09   | 02A    | MOV 2,A  |
| 0A   | 019    | MOV 1,9  |
| 0B   | 008    | MOV 0,8  |
| 0C   | F2D    | DISP 2,D |
| 0D   | FFC    | KIN C    |
| 0E   | 99C    | CMPI 9,C |
| 0F   | D13    | BRC 13   |
|      |        |          |
| 10   | 0DE    | MOV D,E  |
| 11   | 0CD    | MOV C,D  |
| 12   | C0D    | GOTO 0D  |
| 13   | 90E    | CMPI 0,E |
| 14   | E16    | BRZ 16   |
| 15   | C18    | GOTO 18  |
| 16   | 90D    | CMPI 0,D |
| 17   | E27    | BRZ 27   |
| 18   | F02    | DISOUT   |
| 19   | F04    | DZHX     |
| 1A   | 71D    | SUBI 1,D |
| 1B   | FCE    | SUBC E   |
| 1C   | E1E    | BRZ 1E   |
| 1D   | C20    | GOTO 20  |
| 1E   | 90D    | CMPI 0,D |
| 1F   | E29    | BRZ 29   |
|      |        |          |
| 20   | F0D    | EXRL     |
| 21   | 0A2    | MOV A,2  |
| 22   | 091    | MOV 9,1  |
| 23   | 080    | MOV 8,0  |
| 24   | F0D    | EXRL     |
| 25   | F0B    | MULT     |
| 26   | C1A    | GOTO 1A  |
| 27   | F08    | CLEAR    |
| 28   | 110    | MOVI 1,0 |
| 29   | F60    | DISP 6,0 |
| 2A   | FFB    | KIN B    |
| 2B   | C00    | GOTO 00  |

# pull





# FALLGRUBE

| G | O | A | L |   |   |   |   |   |   | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   | 1 |
|   |   |   |   |   |   |   |   |   |   | 2 |
|   |   |   |   |   |   |   |   |   |   | 3 |
|   |   |   |   |   |   |   |   |   |   | 4 |
|   |   |   |   |   |   |   |   |   |   | 5 |
|   |   |   |   |   |   |   |   |   |   | 6 |
|   |   |   |   |   |   |   |   |   |   | 7 |
|   |   |   |   |   |   |   |   |   |   | 8 |
| S | T | A | R | T |   |   |   |   |   | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |

1st entry = column

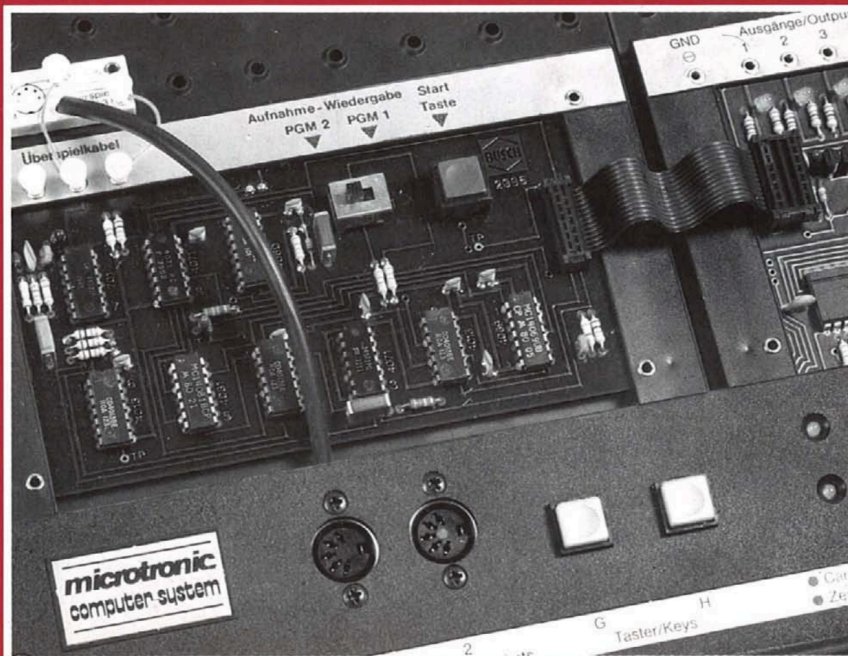
2nd entry = row

# microtronic computer system



Add-on for 2090

## cassette interface 2095



Ready-to-use plug-in module for easy installation.

For transferring (saving) computer programs onto cassettes or tape.

The digital computer signals are recorded by ordinary cassette recorders or tape machines. Transfer time for a full Microtronic program memory (256 addresses) approx. 4 minutes. 12-14 large computer programs can be stored on one C60 cassette.

On playback the interface turns the audio signals back into digital computer signals.