

Quick-Start Information

After plugging the power supply into a 220 V outlet, the computer is ready to operate. The display (LED display) shows: **00 000**. The power supply is VDE-approved. It is a safety transformer that converts the 220 V mains voltage into the 10 V operating voltage required by the computer.

Built-in Programs

A number of easily accessible built-in programs are stored in the computer. They are called up by pressing the function keys (on the right-hand side of the dashboard panel): **HALT – PGM – 0** (or by entering the corresponding number). To understand the built-in-program functions you should consult the matching pages of the instruction manual:

- * PGM 0 = Test program (see page 4–6)
- ** PGM 1 = Load program from cassette into computer (see note)
- ** PGM 2 = Load program from computer onto cassette (see note)
- * PGM 3 = Enter time of day (page 8–9)
- * PGM 4 = Display time of day (page 8–9)
- PGM 5 = Delete self-programmed programs (page 31)
- PGM 6 = Write NOP into every program-memory address (page 63)
- * PGM 7 = "Nim" game (page 7–8)

* These programs are suitable as demonstration programs.

** If PGM 1 or PGM 2 are started without an attached interface, the computer must be re-activated for operation with the RESET key (see page 4).

To select a different built-in program, first press **HALT**, then **PGM**, and finally the program number.

Interesting and Easily Programmable Demonstration Programs

The corresponding instructions from the manual must be programmed (entered). Programming always starts with the keys **HALT – NEXT** followed by pressing the number key **00** twice. Then the input instructions are entered with the number and letter keys. An instruction always consists of a 3-digit code, e.g. F05. After each instruction entry the **NEXT** key must be pressed. The next instruction is then accepted. The "address" (shown by the two left-most digits of the display) automatically increments after every instruction entry (from 00 to 01, etc.).

Electronic die	Page 10–11
Automatic counter	12
"The Moon Landing"	23–24
Timer — computer as time switch	38–39
Lotto-number generator	42
Computer as alarm clock	45–46
Reaction test for two persons	55
Reaction test "de luxe"	55–57
Arithmetic program "Multiplication"	61
Arithmetic program "Division"	62
Tic-Tac-Toe (board game)	Manual, Pt. 2
Sea battle	Manual, Pt. 2
Code breaker	Manual, Pt. 2
"Nim 2"	Manual, Pt. 2
Morse-code decoder	Manual, Pt. 2

Function-Key Short Description

(detailed description see page 72–73)

- HALT** Cancel program execution
- NEXT** Displays the next instruction in program memory. The instruction can be modified and, after pressing NEXT again, stored back in program memory.
- BKP** Break-point (consult page 72)
- REG** Inspection & modification key for the working registers
- C/CE** Clear key for incorrectly entered instructions
- RUN** Program-start key (in combination with HALT – NEXT and entry of the start address, e.g. B 00)
- STEP** Single-step key for stepwise checking of program execution
- PGM** Call-up key for the built-in programs

Number and digit keys: 1–9 and A–F — entry keys for programming and value input.

Red function keys G and H: Special-purpose keys for connecting input signals.

Green RESET key: Located on the computer circuit board; re-activates the operating system after faulty programming.

Comparison Decimal ↔ Hexadecimal

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
0	00	64	40	128	80	192	C0
1	01	65	41	129	81	193	C1
2	02	66	42	130	82	194	C2
3	03	67	43	131	83	195	C3
4	04	68	44	132	84	196	C4
5	05	69	45	133	85	197	C5
6	06	70	46	134	86	198	C6
7	07	71	47	135	87	199	C7
8	08	72	48	136	88	200	C8
9	09	73	49	137	89	201	C9
10	0A	74	4A	138	8A	202	CA
11	0B	75	4B	139	8B	203	CB
12	0C	76	4C	140	8C	204	CC
13	0D	77	4D	141	8D	205	CD
14	0E	78	4E	142	8E	206	CE
15	0F	79	4F	143	8F	207	CF
16	10	80	50	144	90	208	D0
17	11	81	51	145	91	209	D1
18	12	82	52	146	92	210	D2
19	13	83	53	147	93	211	D3
20	14	84	54	148	94	212	D4
21	15	85	55	149	95	213	D5
22	16	86	56	150	96	214	D6
23	17	87	57	151	97	215	D7
24	18	88	58	152	98	216	D8
25	19	89	59	153	99	217	D9
26	1A	90	5A	154	9A	218	DA
27	1B	91	5B	155	9B	219	DB
28	1C	92	5C	156	9C	220	DC
29	1D	93	5D	157	9D	221	DD
30	1E	94	5E	158	9E	222	DE
31	1F	95	5F	159	9F	223	DF
32	20	96	60	160	A0	224	E0
33	21	97	61	161	A1	225	E1
34	22	98	62	162	A2	226	E2
35	23	99	63	163	A3	227	E3
36	24	100	64	164	A4	228	E4
37	25	101	65	165	A5	229	E5
38	26	102	66	166	A6	230	E6
39	27	103	67	167	A7	231	E7
40	28	104	68	168	A8	232	E8
41	29	105	69	169	A9	233	E9
42	2A	106	6A	170	AA	234	EA
43	2B	107	6B	171	AB	235	EB
44	2C	108	6C	172	AC	236	EC
45	2D	109	6D	173	AD	237	ED
46	2E	110	6E	174	AE	238	EE
47	2F	111	6F	175	AF	239	EF
48	30	112	70	176	B0	240	F0
49	31	113	71	177	B1	241	F1
50	32	114	72	178	B2	242	F2
51	33	115	73	179	B3	243	F3
52	34	116	74	180	B4	244	F4
53	35	117	75	181	B5	245	F5
54	36	118	76	182	B6	246	F6
55	37	119	77	183	B7	247	F7
56	38	120	78	184	B8	248	F8
57	39	121	79	185	B9	249	F9
58	3A	122	7A	186	BA	250	FA
59	3B	123	7B	187	BB	251	FB
60	3C	124	7C	188	BC	252	FC
61	3D	125	7D	189	BD	253	FD
62	3E	126	7E	190	BE	254	FE
63	3F	127	7F	191	BF	255	FF

Bookmark

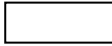
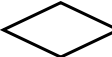
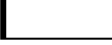
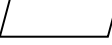
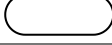
Quick-Reference Information for Programming

The Microtronic Instruction Set (see also instruction manual page 74 to 76)

Mnemonic	Code	Description	Meaning	Page
MOV	0 s d	s into d	Move contents of register s into register d	22
MOVI	1 n d	n into d	Move constant n into register d	33
AND	2 s d	s AND d = d	Logical AND operation	59–60
ANDI	3 n d	n AND d = d	Logical AND with a constant	59–60
ADD	4 s d	s ⊕ d = d	Add register contents; result in d	18–21
ADDI	5 n d	n ⊕ d = d	Add constant n to reg. d; result in d	19–21
SUB	6 s d	d ⊖ s = d	Subtract register contents; result in d	35
SUBI	7 n d	d ⊖ n = d	Subtract constant n; result in d	35
CMP	8 s d	Compare s, d	Compare contents of register s with register d	74
CMPI	9 n d	Compare n, d	Compare constant n with register d	26
OR	A s d	s OR d = d	Logical OR operation	59–60
CALL	B a a	Jump to subroutine a a	Jump to subroutine beginning at address a a	63–64
GOTO	C a a	Jump a a	Jump to any address a a	14–15
BRC	D a a	Carry-jump a a	Jump to a a if carry-flag set	26
BRZ	E a a	Zero-jump a a	Jump to a a if zero-flag set	33
MAS	F 7 d	A-Reg. → S-Reg.	Copy working register d into memory register d	61
INV	F 8 d	Invert d	Invert the contents of register d	50
SHR	F 9 d	Shift d right	Shift register d binarily right	49
SHL	F A d	Shift d left	Shift register d binarily left	48
ADC	F B d	d ⊕ C = d	Set carry-flag adds "1" to register d	19–21
SUBC	F C d	d ⊖ C = d	Set carry-flag subtracts "1" from register d	35
DIN	F D d	Data in	Store input data into register d	37
DOT	F E s	Data out	Output data from register s	36
KIN	F F d	Key value	Wait for key entry; store value in register d	14–15
DISP	F n s	Display on	Show n digits starting at register s	14–15
HALT	F 0 0	Program stop	Resume halted program with STEP key	63
NOP	F 0 1	No operation	An instruction code may be inserted later	63
DISOUT	F 0 2	Display off	Switch display off (for faster execution)	14–15
HXDZ	F 0 3	Hex → Dec	Convert hex to decimal in register D-E-F	43
DZHX	F 0 4	Dec → Hex	Convert decimal to hex in register D-E-F	43
RND	F 0 5	Random generator	Random numbers in registers D-E-F	41
TIME	F 0 6	Clock time	Copy time of day into registers A-B-C-D-E-F	44–45
RET	F 0 7	Return	Return from a subroutine	63–64
CLEAR	F 0 8	Clear registers	Set all register values to "0"	21
STC	F 0 9	Set carry	Set carry-flag (value 1)	52
RSC	F 0 A	Reset carry	Reset a set carry-flag (0)	52
MULT	F 0 B	Multiplication	Multiplication performed in registers 0–5	61–62
DIV	F 0 C	Division	Division performed in registers 0–3 (reg. 4 & 5 = "0")	62–63
EXRL	F 0 D	Exchange A/S 0–7	Swap working regs 0–7 with memory regs 0–7	61
EXRM	F 0 E	Exchange A/S 8–F	Swap working regs 8–F with memory regs 8–F	61
EXRA	F 0 F	Exchange 0–7 / 8–F	Swap working regs 0–7 with working regs 8–F	33

For d the register number is entered. Register d holds the result after instruction execution.	15
For s the register number is entered. Register s is not modified by instruction execution.	16
For n a constant (the same in all operations) value is entered.	74
For a a the jump address (address number to jump to) is entered.	15/74

The Number Systems			
Decimal	Hex	Binary	Inverted bin / hex
0	0	0000	1111 F
1	1	0001	1110 E
2	2	0010	1101 D
3	3	0011	1100 C
4	4	0100	1011 B
5	5	0101	1010 A
6	6	0110	1001 9
7	7	0111	1000 8
8	8	1000	0111 7
9	9	1001	0110 6
10	A	1010	0101 5
11	B	1011	0100 4
12	C	1100	0011 3
13	D	1101	0010 2
14	E	1110	0001 1
15	F	1111	0000 0

Symbols for Flowcharts	
	General operation
	Branching
	Subroutine
	Input / Output
	Start or end of a program
	Connector to other program parts
> means: greater < means: smaller = means: equal value	

The binary system is represented by the LEDs at the 4 outputs: **4 3 2 1**. The 4 outputs have the values **8 4 2 1**.



BUSCH GmbH
Postfach 1360
D-6806 Viernheim

Notes for Microtronic Programmers on Using the Program Tables

The enclosed program lists are for the development of self-designed programs.

Register allocation: It is important to keep track of the registers required by a program and their contents.

Jump target: Entry for short labels indicating at which addresses special program parts begin that are jumped to from another program part.

Address-Nr.: It is practical to start a program at address 00. When programming, the computer automatically assigns the further address numberings. The address number is given to 2 hex digits, and it is recommended that the address number be supplemented, e.g. 10, 1F, etc.

Input instruction code: The instruction codes to be programmed are entered in this column.

Mnemonic (Befehls-Kürzel): Noting the mnemonic, such as MOV, GOTO etc., is advisable so that, in case of troubleshooting, the program is clearer.

Jump-to: Here it can be indicated by a short label from where individual program jumps are made.

Explanations: Important lines of thought can be recorded in this column.

The bookmark (in the manual) is an important helper containing all necessary quick-reference information.

Have fun programming and experimenting.



Program List for Microtronic Computer System 2090

Program name: _____

Sheet no.: _____

Programmer: _____

Date: _____

Register allocation

Register-Nr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Register contents																

Jump target	Addr.	Input code	Mnemonic	Jump to	Explanations
	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	A				
	B				
	C				
	D				
	E				
	F				
	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	A				
	B				
	C				
	D				
	E				
	F				
	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	A				
	B				
	C				
	D				
	E				
	F				