

Code	Befehl	Bemerkung
0	NOP	No Operation
1	+	
2	-	
3	*	
4	/	
5	.	Dezimalpunkt
6	ENTER	
7	STO <mem>	X-Register in Speicher <mem> sichern
8	RCL <mem>	Speicher <mem> nach Rx lesen
9	SQRT	Quadratwurzel
10	GOTO <prg_adr>	Unbedingter Sprung zu Program,madresse <prg_adr>
11	GOSUB <prg_adr>	Unterprogrammsprung Sprung zu Program,madresse <prg_adr>
12	/-/	Vorzeichenwechsel Rx
13	1/x	Kehrwert von Rx
14	HALT	Run/Stop
15	CX	Rx löschen
16	y^x	Ry ^ Rx
17	x<->y	Inhalt Rx und Ry tauschen
18	STO+ <mem>	Rechnender Speicher, zum Speicher <mem> wird der Wert von Rx addiert
19	STO- <mem>	Rechnender Speicher, vom Speicher <mem> wird der Wert von Rx subtrahiert
20	STO* <mem>	Rechnender Speicher, Speicher <mem> wird mit dem Wert von Rx multipliziert
21	STO/ <mem>	Rechnender Speicher, Speicher <mem> wird durch den Wert von Rx geteilt
23	END	Programmende
28	SIN	Sinus(Rx)
29	COS	Cosinus(x)
30	TAN	tangens(x)
31	EEX	Exponenteingabe
32	10^X	
34	e^X	
35	INT	Ganzzahliger Anteil
36	RDN	Roll-Down-Funktion der Rechenregister, Zyklische Vertauschung ohne Datenverlust
48	0	
49	1	
50	2	
51	3	
52	4	
53	5	
54	6	
55	7	
56	8	
57	9	
58	A	
59	B	

60	C	
61	D	
62	E	
63	F	
65	RND	Zufallszahl
67	grd <key>	Umschaltung Radiant <key = 1> / Grad <key=0> für die Winkelfunktionen
68	Fix <key>	Umschaltung des Anzeigemodus: <0> = Float, <1> = Fix, <2>= Eng <3>=HH:MM <4>=HEX
70	LSTx	Last X
71	SFILE <key>	Speicher des Programmspeichers ab PC bis END in die Datei PROG_<key>.B5M
72	LFILE <key>	Lesen der die Datei PROG_<key>.B5M in den Programmspeicher
73	SQR	Quadratwurzel
74	PAUSE	Programmabarbeitung für 1 s pausieren
75	RETURN	Return aus einem Unterprogramm
76	PI	Pi natürlich
78	RUN	Umschalten in den Programmiermodus/Ausführungsmodus
79	CReg	Löschen der Zahlenspeicher, Achtung: Geändert
80	x^y	
81	CPrg	Löschen der Programmspeicher, Achtung: Geändert
92	ASIN	Arcussinus
93	ACOS	Arcuscosinus
94	ATAN	Arcustangens
95	ABS	Absolutwert von Rx
96	LOG	Zehnerlogarithmus von Rx
98	LN	Natürlicher Logarithmus
99	FRAC	Gebrochener Anteil von Rx
100	ROLLUP	Roll-Up-Funktion der Rechenregister, Zyklische Vertauschung ohne Datenverlust
113	IF x=y <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
114	IF x>y <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
115	IF x<y <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
116	IF x=0 <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
117	IF x>0 <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
118	IF x<0 <prg_adr>	Bedingter Sprung zu Programadresse <prg_adr>
119	Loop 7 <prg_adr>	Loop, wenn der Inhalt von Speicher 7 größer als 0 ist, wird dieser um 1 vermindert und dann zu <prg_adr> gesprungen
120	Loop 8 <prg_adr>	Loop, wenn der Inhalt von Speicher 8 größer als 0 ist, wird dieser um 1 vermindert und dann zu <prg_adr> gesprungen
121	Loop 9 <prg_adr>	Loop, wenn der Inhalt von Speicher 9 größer als 0 ist, wird dieser um 1 vermindert und dann zu <prg_adr> gesprungen
135	STO Ix <index_mem>	Speichere Rx in die Speicherzelle, deren Adresse in <index_mem> steht
136	RCL Ix <index_mem>	Lese die Speicherzelle, deren Adresse in <index_mem> steht nach Rx
138	GOTO Ix <index_mem>	Unbedingter Sprung zu dem Programmspeicher, dessen Adress in <index_mem> steht

139	GOSUB lx <index_mem>	Unterprogrammsprung zu dem Programmspeicher, dessen Adress in <index_mem> steht
146	STO+ lx <index_mem>	Rechnender, indexierter Speicher, addiere Rx zu dem Speicher, dessen Adresse in <index_mem> steht
147	STO- lx <index_mem>	Rechnender, indexierter Speicher
148	STO* lx <index_mem>	Rechnender, indexierter Speicher
149	STO/ lx <index_mem>	Rechnender, indexierter Speicher
195	grd lx <index_mem>	Umschaltung Radiant <key = 1> / Grad <key=0> für die Winkelfunktionen, der Key steht im Speicher
196	Fix lx <index_mem>	Umschaltung des Anzeigemodus: <0> = Float, <1> = Fix, <2>= Eng <3>=HH:MM <4>=HEX, der Key steht im Speicher <index_mem>
199	SFILE lx <index_mem>	Speichern des Programmspeichers ab PC bis END in die Datei PROG_<key>.B5M, der Key steht im Speicher <index_mem>
200	LFILE lx <index_mem>	Lesen der Datei PROG_<key>.B5M, in den Programmspeichers , der Key steht im Speicher <index_mem>
241	IF x=y lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
242	IF x>y lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
243	IF x<y lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
244	IF x=0 lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
245	IF x>0 lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
246	IF x<0 lx <index_mem>	Bedingter Sprung zu dem Programmspeicher, dessen Adresse in <index_mem> steht
247	Loop 7 lx <index_mem>	Loop, wenn der Inhalt von Speicher 7 größer als 0 ist, wird dieser um 1 vermindert und dann zu der Adresse gesprungen, die in <index_mem> steht
248	Loop 8 lx <index_mem>	Loop, wenn der Inhalt von Speicher 8 größer als 0 ist, wird dieser um 1 vermindert und dann zu der Adresse gesprungen, die in <index_mem> steht
249	Loop 9 lx <index_mem>	Loop, wenn der Inhalt von Speicher 8 größer als 0 ist, wird dieser um 1 vermindert und dann zu der Adresse gesprungen, die in <index_mem> steht
255	NOP	

<key>	numerischer Steuerparameter, einstellig
<mem>	Speicherzellenadresse 00..63 - zweistellig
<index_mem>	Zahlenspeicheradresse (indexspeicher) 00..63, zweistellig
<prg_adr>	Programmspeicheradresse 000...254 - dreistellig