

LC80ex-BASIC *USB

Version 2.0

Stand: 11. Mai 2016

Diese spezielle BASIC-Version setzt folgende Hardware voraus:

- Kommunikation: [TV-Terminal](#) an SIO-B
- Drucken: [Druckadapter](#) an SIO-A
- USB-Funktionen: [USB-Adapter](#)
- Systemuhr: [Timekeeper](#) auf 6000...7FFF
- Treiber-ROM: [LCTOOLS](#) auf A000...BFFF
- RAM: 2000...7FFFh durchgehend
- LC-80ex-Jumper-Einstellungen:
 - Takt 1,8 MHz
 - CTC/SIO: Kanal0

```
LC80ex-BASIC 2.0*USB
23470 Bytes free
ok
>
```

Inhaltsverzeichnis

Eckdaten	2
Speicherbelegung.....	2
USB-Schnittstelle.....	2
Serielle Schnittstelle.....	2
Systemuhr.....	2
Steuerzeichen/-tasten.....	3
Hinweise zu speziellen Anweisungen.....	5
Kompatibilität.....	7
Anlage	8
Übersicht der Anweisungen.....	8
Fehlermeldungen.....	10
Tokenliste.....	11
Hardwarezubehör.....	12
LCTOOLS2.....	12
TV-Terminal.....	12
USB-Adapter.....	12
Druckadapter.....	12
Timekeeper.....	12

Eckdaten

- Kommunikation per RS232 (SIO Kanal B) mit speziellem Hardware-Terminal
- Druckausgabe per RS232 (SIO Kanal A)
- Soundausgabe (LC-80-Monitorfunktion)
- SAVE/LOAD per USB
- Variablen: Namen haben 2 signifikante Zeichen
Typen:
 - numerisch Gleitkomma (einfache Genauigkeit, 7-8 geltende Ziffern)
 - Zahlenbereich: ca. $+ - 2^{-127} \approx 5,8E-39$... $+ - 2^{+125} \approx 4,2E+37$
 - String (max. 255 Zeichen)
 - Feld numerisch und String (Größe/Anzahl Dimensionen je nach verfügbarem RAM)
- Druckfunktionen LLIST, LPRINT und LFILES
- Funktionen für Datum und Zeit (basierend auf TIMEKEEPER-RAM)
- Als Ausgangsmaterial diente das vom AC1/LLC2 bekannte "GS-BASIC" (Ludwig/SCCH).
 - Syntax und Steuercodes sind weitgehend kompatibel mit GS-BASIC.
 - Da dieses jedoch in einigen Anweisungen sehr hardwarenah (Bildschirm und Tastatur) arbeitet, sind einige Besonderheiten und Einschränkungen vorhanden (siehe [Hinweise](#) und auch [Tokenliste](#)).
 - Ansonsten gilt die Beschreibung von GS-BASIC zu den Anweisungen entsprechend.

Speicherbelegung

- #2000...#20FF Arbeitszellen BASIC
- #2100...#22E2 (...#23FF) freier Speicher z.B. für MC-Routinen
- #2400...#7FFF BASIC-Programm, Obergrenze ist fest eingestellt, kann aber geändert werden (in #C00C/#C00D) oder auch per CLEAR-Anweisung temporär herabgesetzt werden (z.B. um Platz für MC-Programm zu schaffen)
- #A000...#BFFF USB- und serielle Treiber (& rdk-Tiny-BASIC)
- #C000...#DFFF BASIC-ROM
- Start #C000, Warmstart: #C003

USB-Schnittstelle

- LOAD und SAVE benutzen Dateien mit der Endung *.Z80, die jedoch in den Anweisungen nicht mit anzugeben ist.
- Fehler beim USB-Zugriff (z.B. kein USB-Stick dran) werden als "USB error" angezeigt

Serielle Schnittstelle

- Protokoll 2400 Baud, 8 Datenbits, 1 Stoppbit, keine Parität, kein Handshake
- gilt für LC-80ex-Takt von 1,8 MHz, bei 900 kHz nur 1200 Baud
- Druckausgabe: gleiches Protokoll (Drucker auf 2400 bzw. 1200 Baud einstellen!)
 - Die Druckanweisungen LPRINT, LLIST und LFILES sollten nur dann verwendet werden, wenn auch wirklich ein Drucker angeschlossen und bereit ist. Ansonsten hängt ggf. BASIC fest.
 - Es erfolgt weder ein Test auf vorhandenen Drucker noch ist ein "timeout" eingebaut

Systemuhr

- Voraussetzung für die Anweisungen DATE\$(0) und TIME\$(0) ist ein auf Adresse 6000...7FFF installierter Timekeeper-RAM M48T08. Ist dieser nicht vorhanden oder in einem undefinierten Zustand, so werden Fragezeichen angezeigt.
- Das Stellen der Uhr per BASIC ist nicht implementiert, dazu existiert ein separates Tool.

Steuerzeichen/-tasten

Ob und wie Steuerzeichen benutzt werden können und wie bestimmte Tasten reagieren, hängt weitgehend vom benutzten Terminal ab. Diese BASIC-Version ist auf einen angepassten [Hardware-Adapter](#) (TV-Terminal, AVR-kontrolliert) ausgelegt.

Code	Taste	Bildschirmreaktion	Bemerkungen
#01	Strg+A	Kursor home (links oben)	
#03	Strg+C	Abbruch bei LIST, RUN, EDIT und Kommando	<input checked="" type="checkbox"/>
#04	Strg+D	DELETE, Zeichen auf Kursorposition löschen	<input checked="" type="checkbox"/>
#05	Strg+E	Zeichen an Kursorposition einfügen	<input checked="" type="checkbox"/>
#06	Strg+F	Kursor an den Zeilenanfang	<input checked="" type="checkbox"/>
#08	Kursoraste links	Kursor 1 Schritt nach links	<input checked="" type="checkbox"/>
#09	Kursoraste rechts	Kursor 1 Schritt nach rechts	<input checked="" type="checkbox"/>
#0A	Strg+J	Zeilenschaltung	
#0C	Strg+L	Bildschirm löschen+home	
#0D	ENTER	neue Zeile	<input checked="" type="checkbox"/>
#0F	TAB	Tabulator 8 Zeichen	
#13	Strg+S	Kursorpositionierung x	nächstes Zeichen = Spaltennummer
#14	Strg+T	Kursorpositionierung y	nächstes Zeichen = Zeilennummer
#7F	BACKSPACE	letztes eingegebenes Zeichen löschen	<input checked="" type="checkbox"/>

Besonderheiten:

- In der Kommandozeile bzw. im Zeileneditor von BASIC wirken nur die mit markierten Codes bzw. Tasten. Mittels PRINT CHR\$(code) lassen sich jedoch alle übrigen erzeugen und ausgeben.
- Die Tastatur arbeitet im "Dauer-CAPS LOCK" (Computer-)Modus = Großbuchstaben und Ziffern. Mit SHIFT werden Kleinbuchstaben und Sonderzeichen erzeugt. SHIFT LOCK bewirkt Verriegelung auf Kleinbuchstaben+Sonderzeichen.
- Mit Strg+(A...Z) lassen sich alle Zeichen im Bereich 01...1Ah erzeugen, aber nur die linke Strg-Taste wird ausgewertet!
- Die Funktionstasten F1...F12 lassen sich nicht mit kompletten Befehlen belegen. Sie liefern lediglich feste Codes F1h...FCh, die z.B. mit der INKEY-Funktion abzufragen sind.
- Der Ziffernblock liefert die "Dauer-Numlock"-Werte (0...9,...).
- Die Sondertasten Einfg/Pos1, Bild hoch, Bild runter, Entf, Ende, Druck, Rollen, Pause sowie R_Strg, L_WIN, R_WIN und MENU werden wie evtl. vorhandene weitere Multimedia-Tasten aktuell nicht unterstützt.
- Die Tastatur-LEDs sind nicht aktiv.
- Weitere Details siehe Beschreibung TV-Terminal.

Achtung:

Mit der [INP-Funktion](#) lässt sich im Vergleich zu AC1/LLC2 u.a. die Tastatur nicht abfragen, da diese keinem Port direkt zugeordnet ist!

Ausnahmen vom deutschen Tastaturlayout:

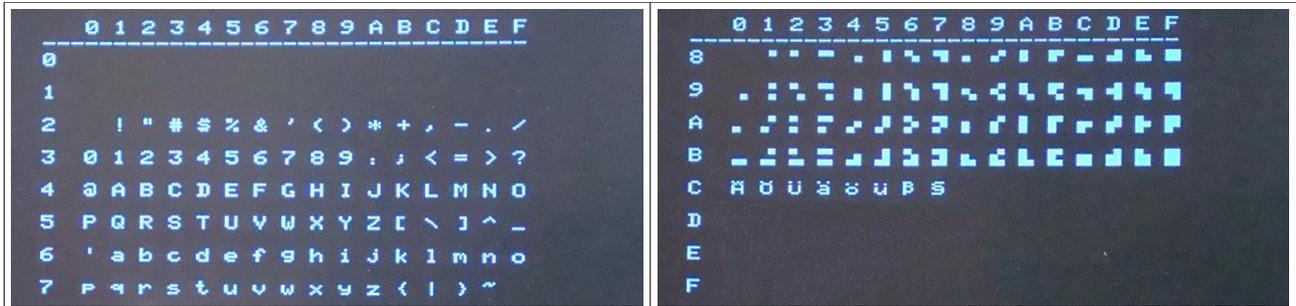
Code	Tastatur	Wirkung
40h	§	@
5Bh	Ä	[
5Ch	Ö	\
5Dh	Ü]
5Eh	^	^
7Bh	ä	{
7Ch	ö	
7Dh	ü	}
7Eh	°	~

Code	Tastatur	Wirkung
C0h	ALT+C0	Ä
C1h	ALT+C1	Ö
C2h	ALT+C2	Ü
C3h	ALT+C3	ä
C4h	ALT+C4	ö
C5h	ALT+C5	ü
C6h	ß, ALT+C6	ß
C7h	ALT+C7	§

Bildschirm:

- Es gibt 25 Zeilen zu je 40 Zeichen.
- Der Bildwiederholpeicher liegt im AVR des TV-Terminals und ist nicht Teil des z80-Adressraumes.
- Es kann nur schreibend auf den Bildwiederholpeicher zugegriffen werden (PRINT-Anweisung bzw. Steuerzeichen und darauf beruhende Anweisungen).
- PEEK und POKE für einen direkten Bildschirmzugriff lassen sich nicht anwenden.
- kein "Onscreen-Editor" verfügbar

Zeichensatz:



Codierung Semigrafik:

<p>99hex</p> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td> </tr> </table>	1	0	0	1	1	0	0	1	<p>6 Blöcke pro Zeichen= 80 x 75 "Bildpunkte"</p> <p>Codes #80...#BF: #80 = Leerfeld #81 = Pixel rechts oben ... #BF = Vollfeld</p>
1	0	0	1	1	0	0	1		

Hinweise zu speziellen Anweisungen

CLEAR	Speicherreservierung einstellen	
	CLEAR	Ohne Parameter werden nur alle Variablen auf Null gesetzt. An Stringraum und RAM-Grenzen wird nichts geändert.
	CLEAR 500	So wird nur der Stringraum gesetzt. Hier werden 500 Bytes für Strings reserviert (Standardwert = 50 Bytes). Das bisherige RAM-Ende bleibt unverändert. Zulässiger Wertebereich: (0)...32767
	CLEAR 500,32767	Reserviert 500 Bytes für Strings <u>und</u> setzt das obere RAM-Ende auf 16383 = #3FFF.
DATE\$(0)	Datum als String anzeigen	
	Ausgabeform: "01.01.2016"	
DEEK	Doppelbyte lesen	nicht möglich für Bildschirmspeicher!
DOKE	Doppelbyte schreiben	
EDIT	Zeileneditor	
	EDIT zzz	Die Zeile zzz wird zur Bearbeitung geholt und komplett angezeigt. Der Cursor steht hinter dem letzten Zeichen. Die Zeile kann nun bei Bedarf ergänzt/geändert werden.
	Kursor <- ->	Navigieren in der Zeile, ohne Zeichen zu ändern
	BACKSPACE	das Zeichen links vom Cursor wird gelöscht
	Strg+D	(Delete) Zeichen an Cursorposition löschen, Rest rückt auf
	Strg+E	Einfügen Zeichen an Cursorposition, Rest rückt nach
	Strg+F	Cursor an Zeilenanfang
	Strg+C	Abbruch, evtl. Änderungen werden verworfen
	<ENTER>	Zeile wird übernommen und Editor beendet
	Buchstaben, Ziffern und Sonderzeichen überschreiben das Zeichen an der aktuellen Cursorposition. Es kann auch die Zeilennummer überschrieben werden, um Zeilen zu duplizieren. Löschen einer Zeile durch Eingabe Zeilennummer und <ENTER>.	
FILES LFILES	Anzeige USB-Inhaltsverzeichnis Ausdrucken -"-	
	Es werden <u>alle</u> Dateien auf dem USB-Stick angezeigt. Keine Parameter, keine Sortierung (Reihenfolge entspricht der Schreibfolge der Dateien).	
INKEY	Tastaturabfrage ohne Halt	
	Die INKEY-Funktion reagiert leider etwas träge. Für zügige Abarbeitung (z.B. Reaktionsspiele) muss daher die Unterbrechungsmöglichkeit (siehe MODE) zumindest zeitweise <u>ausgeschaltet</u> werden. Beispiel: 10 MODE 1:REM SCHALTET UNTERBRECHUNGSMOEGlichkeit AUS! 20 INKEY A\$:IF A\$=""THEN 20 30 PRINT A\$ 40 MODE 0:REM SCHALTET UNTERBRECHUNGSMOEGlichkeit EIN! ...	
INP	Portabfrage	
	Da die serielle QWERTZ-Tastatur am LC-80ex keiner Portnummer 00...FFh direkt zugeordnet ist, lässt sich diese <u>nicht</u> mit der INP-Funktion abfragen.	

LIST	BASIC-Listing auf Schirm anzeigen	
	Es werden soviel Zeilen angezeigt, wie mit LINES n eingestellt (Standard: 20). Anschließend erfolgt ein HALT (Anzeige "!"). "Strg+C" bricht ab, alle anderen Tasten setzen fort.	
LLIST	Drucken BASIC-Listing	
	Drucker vorher bereitmachen! Kein HALT zwischendurch, Endlospapier nötig!	
LOAD	Laden BASIC-Programm per USB LOAD "TEST"	
	geladen werden nur z80-Files mit Dateityp "B" (wie sie SAVE schreibt)	
LOCATE	Kursor-Direktpositionierung	
	Gilt nur für das vorgesehene ("PECHAL"-)TV-Terminal! Man muss selber auf die Einhaltung der jeweils gültigen Grenzen für x (0...39) und y (0...24) achten!	
LPRINT	Drucken Zeichen(-kette)	
	Drucker vorher bereitmachen!	
MODE 0 MODE 1	Ein-/Ausschalten der Unterbrechungsmöglichkeit	
	Standardeinstellung (nach jedem ok >) ist "MODE 0" (Unterbrechungsmöglichkeit per "Strg C" ein). "MODE 1" schaltet diese ab. Ist die Unterbrechungsmöglichkeit ausgeschaltet und das Programm "hängt fest", kann durch ein "RES" am LC-80ex und anschließenden Warmstart (#C003) neu gestartet werden.	
PEEK	Byte lesen	nicht möglich für Bildschirmspeicher!
POKE	Byte schreiben	
SAVE	Sichern BASIC-Programm per USB SAVE "TEST"	
	Existiert bereits ein gleichnamiges File, so erfolgt Rückfrage, ob überschrieben werden soll. Datentyp "B" wird geschrieben.	
SOUND	Tonausgabe	
	Die Parameterwerte für Dauer und Höhe entsprechen der LC-80-Monitorroutine: SOUND D,H Ausgabe eines Tones ("Note") mit der Dauer D (1...255) und der Höhe H (0...31, 32= Pause entsprechender Länge) SOUND ohne Parameterangabe ergibt einen "Aufmerksamkeitston" als 3-Ton-Folge. Die absolute Tonhöhe sowie Tonlängen sind von der Rechnertaktfrequenz abhängig.	
TIME\$(0)	Systemzeit anzeigen	
	Ausgabeform: "12:00:00"	
USR	Aufrufen einer Maschinencode-Routine mit Parameterübergabe	
	1. MC-Routine gestalten, Speicherbereich und Startadresse festlegen (nnnn):	
	Innerhalb der MC-Routine gelten folgende Randbedingungen:	
	<ul style="list-style-type: none"> • Der übergebene Parameter muss im Wertebereich von -32768... 32767 liegen. • Ein Sichern der Register ist nicht nötig, alle Register können verwendet werden. • Parameterübernahme in MC-Routine durch BASIC-Systemcall "DEINT" (->DE). • Rücksprung nach BASIC mit RET. Soll ein Wert zurückgegeben werden, ist dieser in den Registern A (höherwertiges) und B (niederwertiges Byte) abzulegen und anschließend der BASIC-Systemcall mit "JP ABPASS" auszuführen. • Rückgabewerte >=#8000 erscheinen in der BASIC-Variablen als negative Werte, d.h. #7FFF =>32767, #8000 =>-32768, ..., #FFFF => -1 • Die Systemcalls lauten: DEINT: #C006 ABPASS: #C009 	
	2. im Basicprogramm Startadresse des MC setzen: DOKE 8195, nnnn	
	3. im BASIC-Programm aufrufen: A=USR(B)	

Pseudografik

Im Bereich der Codes 80h...C7h stehen eine Pixelgrafik sowie die deutschen Umlaute zur Verfügung. Diese lassen sich mit der Funktion PRINT CHR\$(xx) ausgeben. CHR\$(198) liefert beispielsweise das "ß" als Sonderzeichen.

Vergleichbar mit der vom AC1/LLC2 bekannten "Grafiktaste" ist auch eine Eingabe mittels Tastenkombination "ALT+xx" möglich. "ALT BF" ergibt z.B. ein Vollfeld an der Cursorposition. Das funktioniert auch in PRINT-Anweisungen, z.B. PRINT "█".

Bei LIST einer solchen Zeile mit **Pseudografikzeichen** erscheinen diese als TOKEN.

Eingabe: 10 PRINT "█" ALT BF fügt das Zeichen █ ein

LIST: 10 PRINT "LN" "LN" = Klartext des Tokens BF

Solange die Zeile nicht bearbeitet wird, ist das unschädlich und nur etwas verwirrend im Listing. Eine Bearbeitung der Zeile mit "EDIT 10" lässt das Grafikzeichen jedoch verschwinden und übernimmt den Token-Klartext!

Bedingt durch die völlig andere Bildschirmverwaltung (kein direkter Zugriff auf den Bildschirmspeicher) sind die Befehle SET/RESET/POINT für die Pixelgrafik nicht verfügbar.

Kompatibilität

Das "LC80ex-BASIC *USB" benutzt fast den gleichen Befehlsvorrat wie das GS-BASIC. Dennoch bestehen Einschränkungen/Besonderheiten, die bei der Umsetzung vorhandener GS-BASIC-Programme zu beachten sind.

Ein direkter Import per USB-Stick ist nicht möglich, da beide BASICs in unterschiedlichen Speicherbereichen arbeiten und die Adressen als Zeiger im BASIC-Programm verankert sind.

Nicht lauffähig sind die GS-BASIC-Programme, die:

- hardwarenah direkt per PEEK oder POINT vom Bildschirm lesen bzw. per POKE oder SET/RESET darauf schreiben,
- per INP-Anweisungen auf die Tastatur zugreifen,
- einen Joystickanschluss benötigen
- oder entsprechende MC-Routinen für den Zugriff auf Bildschirm/Tastatur benutzen.

Eingeschränkt lauffähig sind Programme, die

- mehr als 40 Zeichen Bildschirmbreite erfordern oder
- bestimmte Pseudografikzeichen (z.B. für "Spielfiguren") benutzen.
- Sie sind -sofern möglich- erst entsprechend zu modifizieren.

Alle übrigen GS-BASIC-Programme sollten grundsätzlich lauffähig sein.

Zu rdk-Tiny-BASIC besteht keine Kompatibilität. Solche Programme werden auch nicht tokenisiert abgelegt sondern (mit Ausnahme der Zeilennummern) als direkter ASCII-Text.

Achtung:

Die Programme beider BASICs werden als z80-Dateien abgelegt. Es ist auf den ersten Blick nicht ersichtlich, welches BASIC für welche Programmdatei nötig ist. Beide BASICs prüfen jedoch den Dateityp (Tiny-BASIC: "b", 8k-BASIC: "B"). Stimmt der beim Laden nicht, so wird das Laden verweigert ("*Typ error*" bei 8k-BASIC bzw. "*No Tiny Basic*" dort).

Es empfiehlt sich daher, z.B. mit verschiedenen Speichermedien zu arbeiten oder die Namensgebung entsprechend zu gestalten (z.B. Tiny-BASIC-Programme mit "T" beginnend).

Anlage

Übersicht der Anweisungen

Befehl	Bedeutung	Parameter/Hinweise	Syntaxbeispiel
ABS	Betrag	numerisch	A=ABS(B)
AND	logisches UND	bitweise	A=A AND 1
ASC	ASCII-Code des 1. Zeichens	String	A=ASC("H")
ATN	Arcustangens	numerisch	PI=4*ATN(1)
AUTO	automatische Zeilennummerierung	Anfangswert, Schrittweite	AUTO 100,2
BYE	Verlassen BASIC		BYE
CALL	Aufruf Maschinenprogramm	Adresse dezimal oder hex	CALL8448 o. CALL*2100
CHR\$	Zeichenumwandlung	numerisch 0...255	? CHR\$(12)
CLEAR	Speicherreservierung	numerisch, siehe Anmerk.	CLEAR 500,32000
CLS	Bildschirm löschen = ? CHR\$(12)		CLS
CONT	Fortsetzung nach STOP		CONT
COS	Cosinus	numerisch (Bogenmaß)	A=SIN(0.11)
DATA	Werteliste für READ	numerisch oder String	DATA 100,"X"
DATE\$	Timekeeper Datum auslesen	String	A\$=DATE\$(0)
DEEK	Doppelspeicherplatz (16bit) lesen	Adresse dezimal	A=DEEK(32000)
DEF	Definieren Nutzerfunktion		DEF FNA(x)=x*x*x
DIM	Dimensionieren Array		DIM A(3,3,3)
DOKE	Doppelspeicherplatz schreiben	Adresse dezimal	DOKE(32000),A
EDIT	Zeile bearbeiten	Zeilennummer	EDIT 100
END	Programm-Ende		END
EXP	Exponentialfunktion	numerisch	A= EXP(2)
FILES LFILES	Anzeige Inhalt USB-Stick Drucken -"-		FILES
FN	Definition Anwenderfunktion		A=FNY(B)
FOR...TO... STEP... NEXT	Schleife		FOR I=0 TO 9 STEP 2 ... NEXT
FRE	Anzeige freier Speicher	0=alles, "0" = Stringraum	? FRE(0)
GOSUB	Unterprogrammaufruf	Zeilennummer	GOSUB 2000
GOTO	Unbedingter Sprung	Zeilennummer	GOTO 2000
IF...THEN..	Bedingung	kein ELSE verfügbar!	IF A=B THEN C=0
INKEY	Tastaturabfrage ohne Warten	Stringvariable	INKEY A\$
INP	Port abfragen (keine Tastatur!)	numerisch, 0...255	A=INP(4)
INPUT	Eingabe Zahl oder Zeichenkette	ggf. mit Vortext	INPUT"Wert";X,X\$
INT	Ganzzahlanteil	numerisch	A=INT(12/5)
LEFT\$	linkes Teilstück eines Strings	String, Anzahl Zeichen	A\$=LEFT\$("NAME",3)
LEN	Länge eines Strings	String	A=LEN(B\$)
LET	Wertzuweisung, kann entfallen		LET A=1 A=1
LINES	Anzahl bei LIST ausgegebenen Zeilen, ehe ein Halt erfolgt	Zeilenanzahl numerisch	LINES 10
LIST LLIST	Listet Basicprogramm druckt das Listing	(ab Zeilennummer)	LIST (100)
LN	natürlicher Logarithmus	numerisch	A=LN(B)

LOAD	lädt Basicprogramm per USB, altes Programm wird vorher gelöscht	Dateiname nur als Literal, keine Variable!	LOAD "TEST"
LOCATE	Kursordirektpositionierung	siehe Hinweise!	LOCATE (10,5)
MID\$	mittleres Teilstück eines Strings	String, Position, Anzahl	A\$=LEFT\$("NAME",3,2)
MODE	Ein-/Ausschalten der Unterbrechungsmöglichkeit	Standard: "ein", bei "aus" Programm schneller	MODE 0 (ein) MODE 1 (aus)
NEW	löscht Basicprogramm im Speicher		NEW
NOT	logisches NICHT		A= NOT B
NULL	Anzahl Dummyzeichen	numerisch	NULL 60
ON	Sprungverteiler	numerische Variable <>0	ON A GOTO 100,200,300
OR	logisches ODER	bitweise	A= A OR 1
OUT	Portausgabe	Portnummer, Wert	OUT 4,A
PEEK	Speicherzelle lesen	numerisch	A=PEEK(32000)
POKE	Speicherzelle schreiben	numerisch	POKE 32000,0
POS	liefert Cursorposition in Zeile	0=dummy	A=POS(0)
PRINT LPRINT	Ausgabe auf Bildschirm/Drucker	Trenner , tabuliert Trenner ; unterdrückt NL	PRINT "HALLO",A ? "HALLO"
READ	Lesen Element aus Werteliste	Variable je nach Daten	READ A,B\$
REM	Kommentar	Freitext	REM Kommentar
RENUMBER	Neunummerieren Listing von, bis, Start, Schrittweite	ohne Parameter Standard (1/32767/10/10)	RENUMBER 10,100,1000,1
RESTORE	DATA-Zeiger setzen	(Zeilennummer)	RESTORE (100)
RETURN	Rückkehr aus einem Unterprogramm		RETURN
RIGHT\$	rechtes Teilstück eines Strings	String, Anzahl	A\$=LEFT\$("NAME",2)
RND	Zufallszahl 0...<1 erzeugen Argument bestimmt Startwert	0 =letzte generierte Zahl 1 =nächste Zufallszahl -1=immer dieselbe Zahl	A=RND(1)
RUN	Programm starten (ab Zeile n)	(Zeilennummer)	RUN
SAVE	sichert Basicprogramm per USB	Dateiname als String	SAVE "TEST"
SGN	Vorzeichen (0 oder -1)	numerisch	A=SGN(B)
SIN	Sinusfunktion	numerisch, Bogenmaß	A=SIN(3.14)
SOUND	Tonausgabe	Dauer D=0..255 Höhe H=0...31 (32=Pause)	SOUND D,H
SPC(n Leerzeichen ausgeben	numerisch	? SPC(10)
SQR	Quadratwurzel	numerisch	A=SQR(B)
STOP	Hält Programm an		STOP
STR\$	wandelt Zahl in String	numerisch	A\$=STR\$(10)
TAB(Tabulator ausgeben	numerisch	? TAB(4)
TAN	Tangensfunktion	numerisch, Bogenmaß	A=TAN(B)
TIME\$	Timekeeper Uhrzeit auslesen	String	A\$=TIME\$(0)
TROFF	Kontrollmodus aus		TROFF
TRON	Kontrollmodus ein	mit Zeilennummernanzeige	TRON
USR	MC-Routine aufrufen, Wertrückgabe	numerisch	A=USR(B)
VAL	numerischer Wert eines Strings	String	A=VAL(B\$)
WAIT	Warten auf Portereignis	Port, Wert1, Wert2	WAIT 4,10,255
WIDTH	Einstellen Ausgabebreite	Zeilenlänge, nach der automatisch ein Umbruch erfolgt	WIDTH 20

Fehlermeldungen

/Ø	unzulässige Division durch Ø
BS	Feldelement außerhalb dimensionierten Bereich aufgerufen
CN	Programmfortsetzung mittels CONT war nicht möglich
DD	Feld mehrfach dimensioniert
FC	unzulässiger Funktionsaufruf
ID	Fehlerhafte Direkteingabe
LS	String länger als 255 Zeichen
MO	Operand fehlt
NF	Variablen in FOR NEXT Schleife fehlerhaft
OD	zu wenig Daten für READ
OM	Basicspeicher zu klein für nächste Operation
OS	Stringspeicher zu klein
OV	Überlauf (Ergebnis größer 1.70141E38)
RG	RETURN ohne GOSUB
SN	Syntax Fehler
ST	String zu lang oder zu komplex
TM	Typenunverträglichkeit (String - Zahl)
UF	Funktion nicht definiert
UL	Zeilennummer existiert nicht

Klartextmeldungen:

?Extra ignored	Zu viele Werte zu INPUT, zulässige Werte ignoriert
?Redo from start	Fehlerhafte INPUT - Eingabe (String anstatt Zahl oder umgekehrt)
Break	Angabe der Zeile, wo STOP-Anweisung erfolgte
Nothing to save.	Kein Basicprogramm zum sichern vorhanden
Canceled.	Abbruch einer Überschreiben-Anfrage
File exists, overwrite (Y)?	Datei beim Sichern per USB bereits vorhanden, Rückfrage, ob überschrieben werden soll (erfolgt nur bei "Y")
Not found	zu ladende Datei nicht gefunden
Typ error	zu ladende Datei war kein LC80ex-Basic-Programm
USB error	Fehler beim USB-Zugriff (z.B. kein Stick dran)

Tokenliste

LC80ex-BASIC *USB						
	8x	9x	Ax	Bx	Cx	Dx
Ø	END	OUT	LIST	^	EXP	SET
1	FOR	ON	CLEAR	AND	COS	RESET
2	NEXT	NULL	LOAD	OR	SIN	RENUMBER
3	DATA	WAIT	SAVE	>	TAN	LOCATE
4	INPUT	DEF	NEW	=	ATN	SOUND
5	DIM	POKE	TAB(<	PEEK	INKEY
6	READ	DOKE	TO	SGN	DEEK	MODE
7	LET	AUTO	FN	INT	POINT	TRON
8	GOTO	LINES	SPC(ABS	LEN	TROFF
9	RUN	CLS	THEN	USR	STR\$	FILES
A	IF	WIDTH	NOT	FRE	VAL	LFILES
B	RESTORE	BYE	STEP	INP	ASC	LLIST
C	GOSUB	KEY	+	POS	CHR\$	LPRINT
D	RETURN	CALL	-	SQR	LEFT\$	TIME\$
E	REM	PRINT	*	RND	RIGHT\$	DATE\$
F	STOP	CONT	/	LN	MID\$	EDIT

Token und Schreibweisen wie GS-BASIC, Ausnahmen und Besonderheiten:

- **Nicht existent**
 - Blockgrafk-Anweisungen (SET/RESET/POINT)
 - F-Tastenanzeige (KEY)
- **eingeschränkt**
 - INP nicht für Tastaturabfrage!
 - INKEY ist etwas träge und benötigt MODE 1 ([siehe hier](#))
- **Geändert:**
 - LOAD und SAVE benutzen USB-Schnittstelle, Schreibweise ohne führendes "C"!
 - "MODE" wirkt nur auf Ein-/Ausschalten der Unterbrechung, nicht auf den Zeichensatz:
 - MODE 0: Unterbrechungsmöglichkeit ein (Standard)
 - MODE 1: Unterbrechungsmöglichkeit aus
 - MODE 2: wie MODE 1
 - MODE ohne Parameter ist nicht zulässig
- **zusätzlich**
 - FILES-Kommando für Inhalt USB-Stick
 - LFILES/LLIST/LPRINT Druckausgabe der jeweiligen Grundkommandos
 - TIME\$/DATE\$ für Datums-/Zeitfunktionen (Timekeeper)
 - EDIT:Zeileneditor

Hardwarezubehör

Dieses 8k-BASIC ist nur zusammen mit folgenden Baugruppen (vollständig) lauffähig:

LCTOOLS2

Funktion: (E)EPROM mit Treiberpaket & rdk-Tiny-BASIC
Adressbereich: A000...BFFFh
Details: siehe Beschreibung "LCTOOLS2"

TV-Terminal

Funktion: Ein-/Ausgabe über SIO Kanal B (2400 Baud bei 1,8 MHz)
Spezial-IC: ATMega8
Details: siehe Beschreibung "LC80ex-TV-Terminal"

USB-Adapter

Funktion: USB-Massenspeicher
Spezial-IC: VDIP1 + z80-PIO
Peripherieadressen: #BC...#BF
Treiber: in LCTOOLS2
Details: siehe Beschreibung "LCTOOLS2"

Druckadapter

Funktion: Ausgabe über SIO Kanal A (2400 Baud bei 1,8 MHz)
Treiber: in LCTOOLS2
Details: siehe Beschreibung "Drucken am LC80ex"

Timekeeper

Funktion: (intern batteriegestützter) RAM + Echtzeituhr
Spezial-IC: MT48T08
Adressbereich: 6000...7FFFh, Anmerkung:
Es ist möglich den RAM bis auf 9FF0 zu erweitern und als BASIC-Speicher zu benutzen. Für diesen Fall muss der Timekeeper an das obere Ende (8000....9FFF). Andernfalls werden seine Uhrenzellen vom BASIC-Programm unbeabsichtigt überschrieben und die Uhr funktioniert nicht mehr!
Es ist dann eine andere BASIC-Version erforderlich!
Details: siehe Beschreibung "Timekeeper"