

LCTOOLS

für den LC-80ex

Version 2.0

WeRo 2016, Stand: 02.05.2016

Inhaltsverzeichnis

Übersicht.....	2
<i>Sprungverteiler.....</i>	<i>2</i>
<i>Anwenderprogramme.....</i>	<i>2</i>
<i>Dateien.....</i>	<i>3</i>
<i>Meldungen.....</i>	<i>3</i>
RS232-Schnittstelle.....	4
<i>Terminal-Anschluss.....</i>	<i>4</i>
<i>Drucker-Anschluss.....</i>	<i>4</i>
<i>Laden von Daten per RS232.....</i>	<i>5</i>
<i>Sichern von Daten per RS232.....</i>	<i>6</i>
USB-Schnittstelle.....	7
<i>USB-SAVE/LOAD aus dem LC-80ex-Monitor.....</i>	<i>7</i>
<i>USB-SAVE/LOAD extern.....</i>	<i>8</i>
<i>USB-SAVE/LOAD aus Tiny Basic.....</i>	<i>8</i>
Anwendungen.....	9
<i>Timekeeper-Uhr.....</i>	<i>9</i>
<i>AD-Umsetzer.....</i>	<i>9</i>
<i>rdk Basic V3.3*LC-80ex*USB.....</i>	<i>10</i>
<i>Eckdaten.....</i>	<i>10</i>
<i>Übersicht der Anweisungen.....</i>	<i>11</i>
<i>Fehler- und sonstige Meldungen.....</i>	<i>12</i>
<i>Anmerkungen.....</i>	<i>13</i>
Anlage.....	14
<i>z80 Headersave.....</i>	<i>14</i>
<i>Realisierung USB-Adapter.....</i>	<i>15</i>
<i>Inbetriebnahme.....</i>	<i>15</i>
<i>Schaltplan.....</i>	<i>16</i>
<i>Realisierung Druckadapter.....</i>	<i>17</i>
<i>Schaltplan.....</i>	<i>17</i>
<i>V.24-Transceiver.....</i>	<i>18</i>

Übersicht

Nach der erfolgreichen Erprobung der seriellen Übertragung (Laden/Sichern von Dateien) zwischen LC-80ex und einem (PC-)Terminal soll dieses Werkzeug weiter ausgebaut werden. Fast alle Funktionen sind auch per LC-80ex-Hextastatur aufrufbar.

Die LCTOOLS sind in einem 8k-(E)PROM auf A000...BFFF untergebracht. Alle Funktionen sind unabhängig von der benutzten Monitorversion; es werden lediglich Standard-Einsprünge benutzt.

Sprungverteiler

In der Version 2 stehen nunmehr zur Verfügung:

Einsprungs- adresse	Funktion	Aufruf per
A000	Laden per USB	LC-80ex-HEX-Tastatur
A003	Sichern per USB	
A006	Laden per RS232	
A009	Sichern per RS232	
A00C	externer Aufruf Laden per USB	MC-Programm
A00F	externer Aufruf Sichern per USB	
A018	Initialisieren USB	
A01B	Initialisieren SIO-B	
A01E	Initialisieren SIO-A (Drucker)	LC-80ex-HEX-Tastatur
A030	Timekeeper-Uhr	
A033	AD-Wandler	
B000	Aufruf "rdk Tiny-Basic")	
B003	Warmstart rdk Tiny-Basic	

*) mit USB- und Druck-Unterstützung

Anwenderprogramme

Aktuell sind folgende Anwenderprogramme in LCTOOLS2 enthalten:

	Start	Ende/Abbruch
Timekeeper-Uhr	ADR A030 EX	Taste "-"
AD-Wandler gem. Originalanleitung	ADR A033 EX	Taste "RES"
rdk Tiny-Basic	ADR B000 EX	BYE

Dateien

Dateiformat:	Headersave (z80) Andere Dateiformate werden nicht unterstützt.	
Dateitypen:	Kenner A...F je nach Dateinhalt, siehe Anlage	
Dateinamen:	RS232	Dateien enthalten "LC-80EX" als Standard im Header (der "richtige" Dateiname wird am PC durch Auswahl der zu sendenden Datei genommen).
	USB	Der Dateiname wird immer abgefragt. Er ist im Header eingetragen, die Datei selbst entsprechend benannt. Es ist immer darauf zu achten, dass firmwarebedingt Dateinamen maximal 8.3 Zeichen lang sein können und keine Leer-/Sonderzeichen aufweisen. Die Dateierweiterung ".z80" ist bei SAVE/LOAD unter BASIC jedoch <u>nicht</u> mit anzugeben!

Meldungen

Je nach Art des Aufrufs von Funktionalitäten können folgende Meldungen auftreten:

Bedienung per Hex-Tastatur	Ausschriften auf LED-Anzeige:		
	ADR -S	Anfangsadresse eingeben	nach Meldung Werte-Eingabe, dann jeweils weiter mit + Taste
	ADR -E	Endeadresse eingeben	
	ADR -A	(Auto-)Startadresse eingeben	
	FILE -F	Filenames eingeben	
	TYP	Dateityp eingeben	
	Fehlermeldungen:		
	Err -U	USB-Fehler	danach weiter mit beliebiger Taste =>
	Err -F	Dateifehler (kein korrektes z80)	
	noFILE	File nicht gefunden	Rückkehr zum Monitor (Warmstart)
	ExFILE	File existiert bereits	
externer Aufruf	Schirmausschrift: (Register A enthält Fehlercode, ansonsten A=0)		
	Not found.	File nicht gefunden	
	No z80.	Dateifehler	
	Typ Error.	Datei war nicht vom angeforderten Typ	
	File exists, overwrite ? (Y)	File existiert bereits, "Überschreiben? " (andere Taste als "Y" bricht ab)	
	USB Error.	allgemeiner USB-Fehler	

RS232-Schnittstelle

Terminal-Anschluss

Per RS232 ist sowohl ein "dummes" Hardware-Terminal am LC-80ex anschließbar als auch ein PC mit Terminalemulation (z.B. HTerm, PuTTY). Es wird gegenwärtig das Protokoll

2400 Baud, 8 Datenbits, 1 Stopbit, keine Parität, kein Handshake

benutzt, welches für einen Systemtakt des LC-80ex von 1,8432 MHz gilt. Die LC-80ex-Hardware ist dazu wie folgt einzustellen:

JP9: Systemtakt 1/2, JP12: SIO-Port B0

Die Verwendung des Standardtaktes 921,6 kHz ist ebenso möglich, allerdings halbiert sich die Baudrate auf 1200.

Pin	V1: Stecker ("DTE")	V2: Buchse ("DCE")
2	RxD	TxD
3	TxD	RxD
7	RTS	CTS
8	CTS	RTS
Kabel	Nullmodem	Verlängerung

Achtung!

Die Platinenrevisionen V1 und V2 verwenden eine unterschiedliche Belegung des RS232-Anschlusses!

Daraus resultiert die Art des Verbindungskabels zum PC.

Die parallele Nutzung der seriellen Schnittstelle für Bildschirm und Tastatur zur Bedienung des TinyBasic steht etwas im Widerstreit zur Benutzung für das Laden und Speichern von Daten:

- Entweder benutzt man zusätzlich den zweiten Kanal der SIO (was einer zweiten seriellen Strippe am LC-80ex und einem zweiten Terminal entspricht) oder
- man schafft sich ein "Spezialterminal", welches per manueller Umschaltung auf Kommunikation (Bildschirm und Tastatur) oder Datenablage (Ein-/Ausgabepuffer) zu schalten ist.

Da beides einen gewissen Aufwand erfordert, entstand u.a. deshalb die USB-Schnittstelle für die einfache Datenablage...

Drucker-Anschluss

Neu in Version 2 ist die Unterstützung für einen seriellen Drucker. Dazu wird das gleiche o.a. Protokoll benutzt, jedoch nur für Ausgabe an SIO Kanal A. Details siehe [Anhang](#) und gesondertes Dokument "Drucken_LC80ex". Anwendung findet das Drucken z.B. im "rdk Basic 3.3" mit den Kommandos:

- LLIST BASIC-Listing ausdrucken
- LPRINT Druckausgabe Zahlen, Strings
- LFILES USB-Inhaltsverzeichnis ausdrucken

Achtung!

Wird ein Druckbefehl verwendet, wenn

- kein Druckeradapter angesteckt oder
- kein Drucker am Adapter angeschlossen oder
- der Drucker nicht eingeschaltet oder nicht "online" ist ,

blockiert der LC-80ex!

=> Drucker bereit machen oder Abbruch mit RESET.

Laden von Daten per RS232

1. Nach RESET am LC-80 eingeben:

Taste	=> Anzeige	Beschreibung
ADR	X.X.X.X.X X	Adresseingabe
A 0 0 6	A.0.0.6.X X	A006 = Laderoutine starten
EX	(dunkel)	Empfang beginnt, LC-80 wartet auf serielle Daten.
	E r r - F	Dateifehler (keine korrekte z80-Datei) bzw.
	X X X X X	Anzeige Ladeadresse + Dateityp bzw.
		Autostart, wenn Dateityp= "A"

2. Am PC wird nun mittels Terminalprogramm das gewünschte in den LC-80 zu ladende Programm gesendet. Die empfangenen Daten kommen automatisch an der richtigen Speicherposition des LC-80ex zur Ablage.
3. Ist die im z80-File verankerte Anzahl Bytes übertragen, so wird das geladene Programm automatisch an der im z80-File angegebenen Startadresse gestartet, falls der Typeintrag "A" (= ausführbares Programm mit Autostart) lautete. Bei allen anderen Typen unterbleibt der Autostart. In der Anzeige steht dann die Ladeadresse sowie der Dateityp. Eine beliebige Taste führt zum Grundzustand. Man kann dann andere Aktivitäten vornehmen (z.B. weitere Daten nachladen).

Für das Laden., d.h. die Übertragung von Dateien PC => LC-80ex per serieller Schnittstelle benutzte ich gerne meinen "[V.24-Transceiver](#)".

Sichern von Daten per RS232

Das erfolgt analog zur Bedienung der Kassettensicherung, wobei die Eingabe eines Dateinamens entfällt und stattdessen zwei neue Parameter ("Autostartadresse" und "Dateityp") abgefragt werden.

1. Am PC ein Terminalprogramm starten (und wenn nötig auf Empfang stellen).
2. Nach RESET am LC-80 eingeben:

Taste	=> Anzeige	Beschreibung
ADR	X.X.X.X.X X	Adresseingabe
A 0 0 9	A.0.0.9.X X	A009 = Sicherungsroutine starten
EX	a d r - S	S = Startadresse eingeben!
2 0 0 0	2 0 0 0 - S	Anfangsadresse, z.B. 2000H
+	a d r - E	E = Endadresse eingeben!
2 1 F F	2 1 F F - E	Endadresse, z.B. 21FFH
+	a d r - A	A =Autostartadresse eingeben!
2 0 0 3	2 0 0 3 - A	Autostartadresse, z.B. 2003 (nur bei Dateityp A bedeutsam, ansonsten beliebig)
+	t Y P -	Dateityp eingeben
C	t Y P - C	C = <u>Dateityp</u> (A...F) siehe Anlage Übergeht man die Typ-Eingabe einfach mit [+], so wird standardmäßig der Typ "A" verwendet.
+		Anzeige dunkel Datenübertragung startet, TAPE-LED leuchtet
	L C - 80	wenn fertig: TAPE-LED aus automatischer Reset => Monitor

3. Am PC kann nun der Empfangspuffer als Datei abgespeichert werden. Nur dort ist ein Dateiname zur Ablage erforderlich (z.B. TEST.Z80).

Für das Sichern, d.h. die Übertragung von Dateien LC-80ex => PC per serieller Schnittstelle benutzte ich gerne meinen "[V.24-Transceiver](#)".

USB-Schnittstelle

Ein USB-Anschluss (VINCULUM) am Systembus ermöglicht sowohl Laden und Speichern aus dem (per Terminal bedientem) TinyBasic als auch extern (z.B. per LC-80ex-Monitor). Der Anschluss anderer USB-Geräte außer Speichermedien ist nicht vorgesehen.

USB-SAVE/LOAD aus dem LC-80ex-Monitor

Der Aufruf ist vom LC-80ex wie folgt möglich (wobei im Gegensatz zu den RS232-Routinen immer ein Dateiname anzugeben ist).

Nach **EX** dauert es immer einen Moment (Anzeige dunkel), bis USB initialisiert und getestet ist. Erst dann erscheint die Aufforderung zur Eingabe des Filenamens!

Laden z80-File:

Taste	=> Anzeige	Beschreibung
ADR	X.X.X.X.X X	Adresseingabe
A 0 0 0	A.0.0.0.X X	A000 = Laderoutine starten
EX	F I L E - F	F = Filename eingeben *)
+		Anzeige dunkel, USB-LEDs blinken, dann entweder:
	E r r - U	Fehler (z.B. kein USB-Stick dran) oder
	n o F I L E	Datei wurde nicht gefunden oder
	2 0 0 0 C	Anzeige der Ladeadresse + Typ oder
	x x x x x x	Autostart (Dateityp war A)

Sichern z80-File:

Taste	=> Anzeige	Beschreibung
ADR	X.X.X.X.X X	Adresseingabe
A 0 0 3	A.0.0.3.X X	A003 = Sicherungsroutine starten
EX	a d r - S	S = Startadresse eingeben!
2 0 0 0	2 0 0 0 - S	Anfangsadresse, z.B. 2000H
+	a d r - E	E = Endadresse eingeben!
2 1 F F	2 1 F F - E	Endadresse, z.B. 21FFH
+	a d r - A	A =Autostartadresse eingeben!
2 0 0 3	2 0 0 3 - A	Autostartadresse, z.B. 2003 (nur bei Dateityp A bedeutsam, ansonsten beliebig)
+	t Y P -	Dateityp eingeben!
C	t Y P - C	C= <u>Dateityp</u> (Eingabe mit + übergehen: "A")
+	F I L E - F	F = Filename eingeben *)
+		Anzeige dunkel, USB-LEDs blinken
	E x F I L E	Datei existiert bereits (Abbruch)
	L C - 8 0	wenn fertig: RESET ("LC-80"-Meldung)

*) Infolge der Beschränkungen durch die HEX-Tastatur sind die "Namen" nur aus den Zeichen 0...9 und A...F zu bilden und nur 4 Zeichen lang.

USB-SAVE/LOAD extern

In eigenen MC-Programmen kann ebenfalls die USB-Schnittstelle benutzt werden. Zuvor ist diese mit INIT_USB zu initialisieren und ggf. dabei auftretende Fehler abzufangen. Alle nötigen Angaben sind in Registern bereitzustellen:

Laden:

```
CALL #A018 ;Init_USB
CP 0 ;fehlerfrei?
JP NZ,reakt ;nein=> entsprechende Reaktion...
LD A,"C" ;Dateityp
LD IX,nnnn ;Anfangsadresse Dateiname
CALL #A00C ;Laden aufrufen
;geladen wird immer an die im z80 enthaltene Anfangsadresse
;wenn geladen, Rückkehr zum Caller. Bei Fehler A<>0
;kein Autostart!
```

Sichern:

```
CALL #A018 ;Init_USB
CP 0 ;fehlerfrei?
JP NZ,reakt ;nein=> entsprechende Reaktion...
LD A,"A" ;Dateityp
LD HL,aadr ;Anfangsadresse
LD DE,eadr ;Endeadresse
LD BC,sadr ;Startadresse
LD IX,nnnn ;Anfangsadresse Dateiname
CALL #A00F ;Sichern aufrufen
;wenn fertig, Rückkehr zum Caller; bei Fehler oder
;existenter Datei A<>0
```

Der Dateiname ist in einem Puffer als 0-terminierter ASCII-String bereitzustellen, wobei IX dann auf den Pufferanfang weist. Beispiel:

"T"	"E"	"S"	"T"	"."	"Z"	"8"	"0"	#0D	#00
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

USB-SAVE/LOAD aus Tiny Basic

Die USB-Schnittstelle wird auch von "rdk Basic V3.3*LC-80ex*USB" benutzt. Dafür gibt es folgende Anweisungen:

- SAVE Sichern des aktuellen BASIC-Files
- LOAD Laden eines BASIC-Files
- FILES Anzeige des Inhaltsverzeichnisses

TinyBasic benutzt im z80-Format den Dateityp "b". Andere Dateiformate werden abgewiesen. Weitere Beschreibung siehe unter TinyBasic.

Die USB-Aufrufe von LOAD/SAVE per HEX-Tastatur sowie aus TinyBasic führen automatisch zunächst eine Initialisierung der USB-Schnittstelle durch.

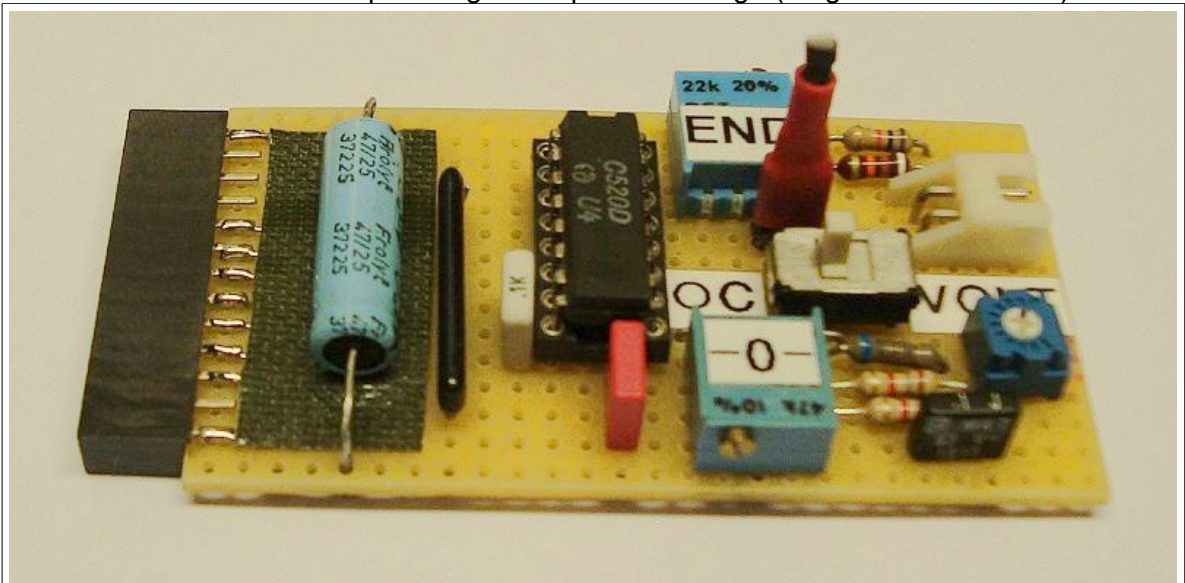
Anwendungen

Timekeeper-Uhr

- Für diese Anwendung ist ein auf 6000...7FFFh installierter Timekeeper M48T08 erforderlich.
- Der Aufruf mit **ADR A030 EX** startet die Zeitanzeige.
- Beendet wird die Zeitanzeige durch Drücken der Taste "-".
- Das Stellen der Uhr ist nicht enthalten und muss mit einem externen Programm vorgenommen werden.

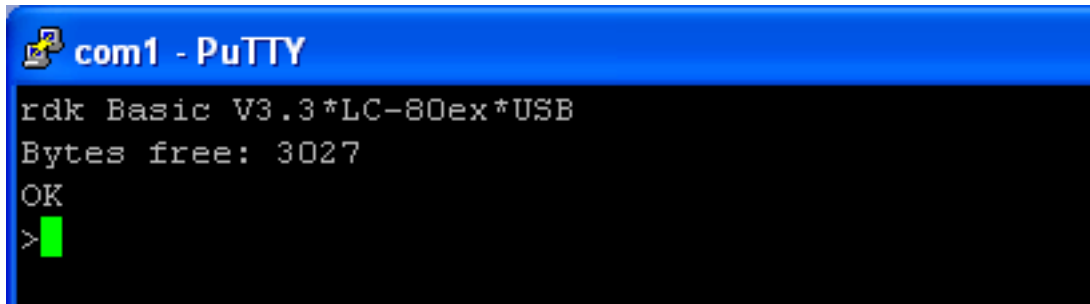
AD-Umsetzer

- Hierfür ist der Hardwarezusatz am USER-Port gem. Originalanleitung erforderlich.
- Probeaufbau: umschaltbar Spannungs-/Temperaturanzeige (aufgestecktes PT100)



- Der Aufruf mit **ADR A033 EX** startet den Umsetzer.
- Bei nicht angeschlossener Hardware bleibt die Anzeige dunkel.
- Verlassen wird die Software mit der Taste "RESET".

rdk Basic V3.3*LC-80ex*USB



Basis:

- TinyBASIC (TDL-Derivat) "rdk Prompt Basic V3.2 3k"
- Beschreibung siehe "Rolf Dieter Klein: BASIC-INTERPRETER" (FRANZIS) oder z.B. hier: <http://hc-ddr.hucki.net/wiki/doku.php/z1013:software:tinybasic:rdk>

Anpassung für LC-80ex:

- Speicherbelegung geändert
- ? als Abkürzung für PRINT sowie CLS und SOUND
- USB-Schnittstelle mit SAVE, LOAD & FILES als Direktkommandos
- IO-Routinen für Kommunikation per seriell Terminal (bei 1,8MHz: 2400 Baud,8,N,1)
- Druckende Anweisungen LLIST, LPRINT, LFILES

Eckdaten

Speicherbelegung:

2000...<23FFh	Arbeitszellen, BASIC-Stack
2400... MAXRAM	BASIC-Text (+42 Byte Eingabepuffer am oberen Ende)
A000...(AFFF)h	USB- und serielle Treiber
B000...(BFFF)h	Interpreter
B000h	Kaltstart, B003h =Warmstart

Variablen:

A...Z numerisch, -32768...32767

@(I) numerisches Feld

Stringvariablen als solche existieren nicht. Es können jedoch Strings mit der Anweisung **I\$ nnnn** eingegeben und **O\$ nnnn** ausgegeben werden. nnnn ist hierbei die manuell anzugebende Speicheradresse des Strings (z.B. auch "TOP").

Wichtige Steuerzeichen/-tasten:

^C	03	Abbruch bei LIST, RUN
Backspace	7F	Löschen des letzten eingegebenen Zeichens

Besonderheiten:

- Der bescheidene freie Platz von 3027 Bytes resultiert aus dem in der Grundvariante verfügbaren RAM (bis 2FFF). Er kann durch Hinzufügen von **zusammenhängendem RAM** ab 3000 erhöht werden. Es gibt jedoch keine automatische Erkennung des RAM-Endes. Dieses ist in Speicherplatz **B006** einzutragen, wenn mehr RAM verfügbar!
- Es werden nur **Großbuchstaben** (sowie Ziffern und Sonderzeichen) korrekt verarbeitet. Sofern möglich, ist daher am PC zweckmäßigerweise CAPS-LOCK einzustellen!
- Die Kursortasten liefern bei Software-Terminals ESC-Folgen und sind für die Bedienung des TinyBasic tabu!
- Es gibt keinen Zeileneditor. Einzig zulässige Korrekturtaste ist "Backspace".
- Mehrere Anweisungen in einer Zeile: Trennung durch **Semikolon!**
- höchste Zeilennummer: 32767
- max. Zeilenlänge: Standard 40 Zeichen (130 Zeichen nach END)

Übersicht der Anweisungen

Direktbefehle

BYE	Beenden Basic	LED-Display leuchtet dann wieder
END nnnn	Speicherende neu festlegen	maximal: END 32767
FILES	Listet Files des USB-Sticks	unsortierte Ausgabe
LFILES	USB-Inhaltsverzeichnis ausdrucken	
LIST	Listen Basicprogramm	auch LIST zeilennummer
LLIST	Listing auf Drucker ausgeben	
LOAD	Programm per USB laden	Dateiname wird hinterher abgefragt
NEW	Löschen des Basicprogramm	
RUN	Ausführen Basicprogramm	
SAVE	Programm per USB sichern	Dateiname wird hinterher abgefragt

Programmierbare Anweisungen:

BYTE	Ausgabe hexadezimal in Byteform	BYTE(15) ergibt "0F"
CALL nnnn	Maschinenprogramm aufrufen	CALL nnnn
CLS	Bildschirm löschen	
FOR... TO...STEP... NEXT...	Schleife Angabe Schleifenindex nötig!	FOR I =0 TO 10 STEP 2 ... NEXT I
GOSUB	Unterprogrammsprung	
GOTO	Unbedingter Sprung	GOTO 100 oder X=10;GOTO X
I\$ nnnn	Eingabe eines Textes	
IF	Bedingung	IF A=1 B=2 oder IF A=0 GOTO 100
INPUT	Eingabe numerisch, keine Strings!	INPUT A oder INPUT "Zahl=",A
LET	arithmetische Zuweisung	A=1 (das LET kann auch entfallen)
O\$ nnnn	Ausgabe eines Textes ab einer angegebenen Speicheradresse	O\$ nnnn bzw. I\$ nnnn nnnn =Speicheradresse nnnn=TOP verwendet 1. freie Adresse
OUT	Ausgabe an Port	OUT(4)=A
OUTCHAR	Einzelzeichenausgabe	OUTCHAR(12)
POKE	Direkter Speicherzugriff	POKE nnnn,bb
PRINT	Ausgabe Variable oder Literal	Abkürzung "?" möglich, Trenner ", " unterdrückt anschließendes NL #n für Formatierung auf n Stellen
LPRINT	wie Print, aber (auch) auf den Drucker	
REM	Kommentar	
RETURN	Unterprogrammrücksprung	
SOUND	Tonerzeugung, ohne Argumente "3-Ton-Signal"	SOUND 10,10 SOUND
STOP	Beenden des Programmablaufs	
TAB	Tabulator ausgeben	TAB(10)
WORD	Ausgabe hexadezimal in Wortform	WORD(2000) ergibt "07D0"

Funktionen:		
'	Einzelzeichenumwandlung	A='A' ergibt in A den Wert 65
ABS(n)	Betrag	A=ABS(B)
CSTS	Tastaturstatus	A=CSTS
FRE	Freier Speicher	A=FRE
HEX(n)	Umwandlung hex=>dez	? HEX(FFFF)
IN(n)	Port lesen	A=IN(4)
INCHAR	Einzelzeicheneingabe	A=INCHAR
LEN	Länge des Strings (bei I\$)	A=LEN
PEEK(n)	Speicherzelle lesen	A=PEEK(nnnn)
RND(n)	Zufallszahl zwischen 0 und n	A=RND(100)
TOP	Erste freie Speicherzelle ermitteln	A=TOP
Vergleichsoperatoren:		
>=	größer oder gleich	
<>	ungleich	
>	größer	
<=	kleiner oder gleich	
<	kleiner	
=	gleich	beachte: auch Zuweisung!

Fehler- und sonstige Meldungen

Es gibt wie bei den meisten TinyBasics nur drei Fehlermeldungen:

HOW?	Der Interpreter hat die Anweisung verstanden, sie fordert jedoch etwas Unmögliches...(z.B. Division durch Null)
WHAT?	Der Interpreter versteht die Anweisung nicht (Befehl falsch geschrieben bzw. nicht existent).
SORRY?	Eine (eigentlich korrekte) Anforderung ist vom Interpreter unter den aktuellen Bedingungen (z.B. wegen Speichermangel) nicht zu erfüllen.

Im Zusammenhang mit der USB-Schnittstelle gibt es folgende (Klartext-)Meldungen:

Name too long (8)	Dateiname ist zu lang, darf maximal 8 Zeichen lang sein. Die Dateierweiterung z80 ist <u>nicht</u> mit anzugeben!
Not found.	Beim Laden angegebene Datei wurde nicht gefunden
No TinyBASIC.	Zu ladende Datei war keine TinyBasic-Datei
No z80.	Zu ladende Datei war keine (korrekte) z80-Datei
File exists, overwrite? (Y)	Beim Sichern wurde ein bereits vorhandener Dateiname festgestellt. mit der Eingabe "Y" kann diese Datei überschrieben werden. Ansonsten erfolgt Abbruch des Speicherns (Meldung "canceled")
USB error.	Beim USB-Zugriff ist ein Fehler aufgetreten (zB. kein Adapter angesteckt, USB-Stick nicht angesteckt oder sonstige Ursachen).
No save.	kein Programm zum Sichern im Speicher

Anmerkungen

- Die für das TinyBasic nötigen Terminaleinstellungen entsprechen den [o.a. genannten](#).
- Die Anweisungen **BYTE, WORD, OUTCHAR, TAB und O\$** geben direkt auf dem Schirm aus. Sie sind keine Funktionen, denen man zur Ausgabe ein PRINT voranstellen müsste.
- Mit **END** lässt sich programmtechnisch die benutzte obere RAM-Grenze verändern.
 - ➔ Maximal darf nnnn=32767 betragen, was einem freien Speicher von 24317 Bytes entspricht (Voraussetzung: RAM lückenlos von 2000 bis 7FFF bestückt).
 - ➔ Mit END wird gleichzeitig die maximale Zeilenlänge automatisch von standardmäßig 40 Zeichen auf 130 Zeichen erhöht.
- Mit **BYE** wird die Steuerung an den LC-80ex zurückgegeben (Warmstart).
- **FILES** zeigt alle auf dem USB-Stick vorhandenen Dateien an (Reihenfolge entspricht dem Schreiben auf USB, keine Sortierung). **LFILES** druckt die Dateinamen als Liste aus.
- **SAVE** Das aktuell im Speicher befindliche Programm wird als z80-File auf USB-Stick abgelegt. Ist bereits ein gleichnamiges File vorhanden, so erfolgt Rückfrage, ob Überschreiben. Wenn <>"J", Abbruch der Speicheroperation.
- **LOAD** Laden eines BASIC-Programms. Wird im Ladeversuche keine z80-Datei oder kein Typ "b" gefunden, wird mit entsprechender Meldung abgebrochen. Ansonsten wird das Programm geladen.
- Dateinamen bei SAVE und LOAD sind nicht hinter der Anweisung anzugeben sondern werden separat nachfolgend abgefragt!
- FILES, SAVE und LOAD initialisieren sicherheitshalber immer erst die USB-Schnittstelle, deswegen dauern die Operationen etwas länger als eigentlich nötig.
- Bei Kaltstart wird automatisch ein eventuell angeschlossener Drucker initialisiert. Die Anweisungen **LLIST, LPRINT** sowie **LFILES** sollten natürlich **nur benutzt werden, wenn auch ein Drucker vorhanden und bereit ist**. Ansonsten "klemmt" die weitere Ausgabe, denn auf "timeout" u.dgl. wurde der Einfachheit halber verzichtet.
- Mit **SOUND** lassen sich Töne wie mit der Monitorroutine erzeugen:
 - ➔ SOUND D,H mit D=Dauer (0...255) und H=Höhe (0..31, 32=Pause)
 - ➔ SOUND ohne Parameter ergibt ein "3-Ton-Aufmerksamkeitssignal"

Anlage

z80 Headersave

Offsets:

0	ANFANGSADRESSE	LOW	
1	ANFANGSADRESSE	HIGH	
2	ENDEADRESSE	LOW	
3	ENDEADRESSE	HIGH	
4	AUTOSTARTADRESSE	LOW	
5	AUTOSTARTADRESSE	HIGH	
6...0B	(FREI)		"LC80ex"
0C	TYP ("A"... "F")		"b" BEI TINY-BASIC-PROGRAMM
0D...0F	3X 0XD3 "MAGIC-BYTE" (Z80-HEADERSAVE-KENNUNG)		
10...1F	PROGRAMM-NAME ASCII		SPEICHERNAME BEI TINY-BASIC-PROGRAMM

PUFFER: EQU 2348H ;32 Byte Headersave-Puffer
LADADR: EQU PUFFER+#20 ;2 Bytes für Ladeadresse
MENGE: EQU PUFFER+#22 ;4 Bytes für Lademenge
TYP: EQU PUFFER+#26 ;Dateityp

Zur Verwendung der Dateitypen gibt es bislang (je nach Rechner) unterschiedliche Definitionen, jedoch keine verbindlichen Vorgaben. Für den LC-80ex wird vorgeschlagen (wie schon bei den RS232-Routinen erwähnt):

A ausführbares MC-Programm, Autostart
C ausführbares MC-Programm
D Daten
b TinyBasic-Programm ab 2400h
B 8k-Basic-Programm ab 2400h

Sollen z80-Headersave-Programme per HEX-Tastatur eingetippt werden, so sind die ersten 32 Byte wegzulassen!

Realisierung USB-Adapter

Es wurde die herkömmliche Beschaltung und Ansteuerung eines VDIP1 (oder auch VDIP2) im Parallelmodus gewählt, wie sie bereits in zahlreichen anderen Kleincomputern (AC1, LLC2, Z9001,...) zum Einsatz kam.



Da hierfür zwei komplette Ports einer PIO benötigt werden, schied der Einsatz am USER-Port des LC-80ex aus (was jedoch den Vorteil hat, dass dieser weiterhin verfügbar bleibt).

Das VDIP wurde mit einer 3. PIO am Systembus angeschlossen. Der Aufwand ist minimal ([Schaltplan](#)).

Die Adressierung erfolgt analog der bisherigen Systematik mit /CE=A6. Damit erhält das VDIP die IO-Adressen BC...BF. Interrupt wird nicht benutzt.

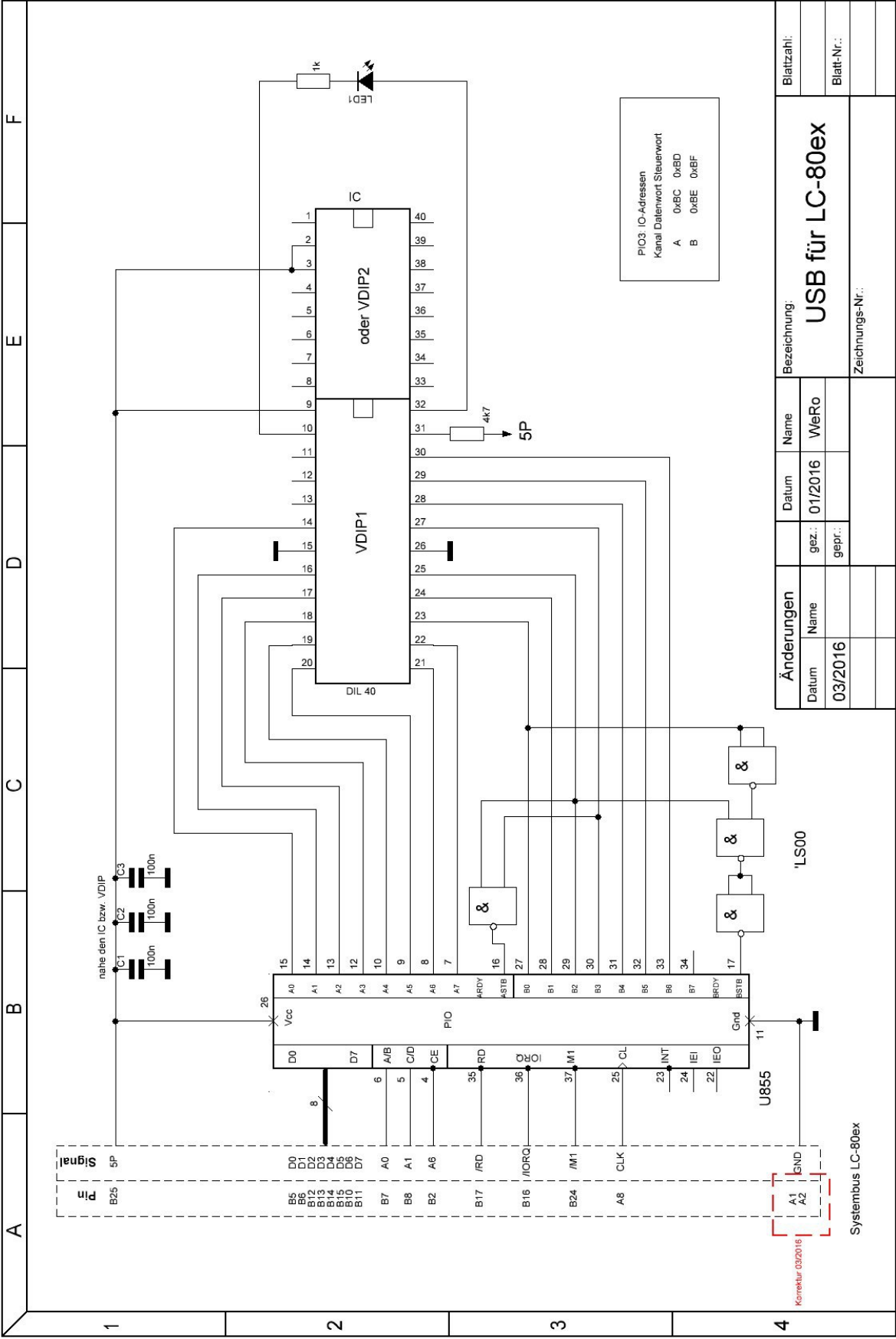
Musteraufbau ("gefädelt")

Inbetriebnahme

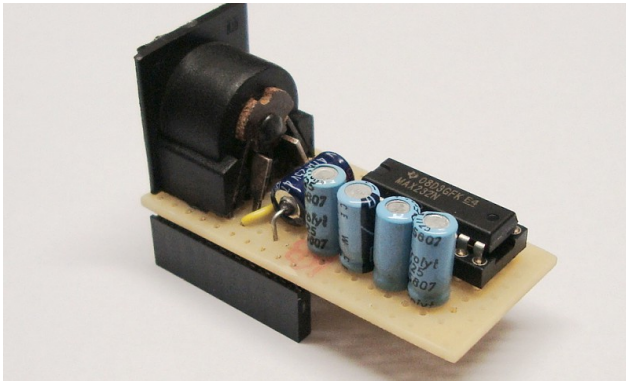
(ich beschreibe hier meine Vorgehensweise, es geht sicher auch anders...)

1. Hardware-Adapter aufbauen und an LC-80ex anstecken.
2. Einen USB-Stick frisch mit FAT formatieren.
3. Als erste Datei die aktuelle Firmware (FTRFB.FTD, Version 3.69 im Paket) auf den USB-Stick aufspielen.
4. Jumper auf dem VDIP beide auf 1-2, dann USB-Stick an Adapter anstecken
5. LC-80ex einschalten. Das VDIP lädt die Firmware vom USB-Stick und blinkt dann unaufhörlich. Eine Weile blinken lassen...
6. LC-80ex ausschalten, am VDIP Jumper ändern: **J3: 1-2, J4: 2-3** (parallele Betriebsart)
7. Nun ist das USB-Interface betriebsbereit. Bei jedem neuen Einschalten (sowie beim Zugriff durch Software) blinken die LEDs kurz.

Schaltplan



Realisierung Druckadapter

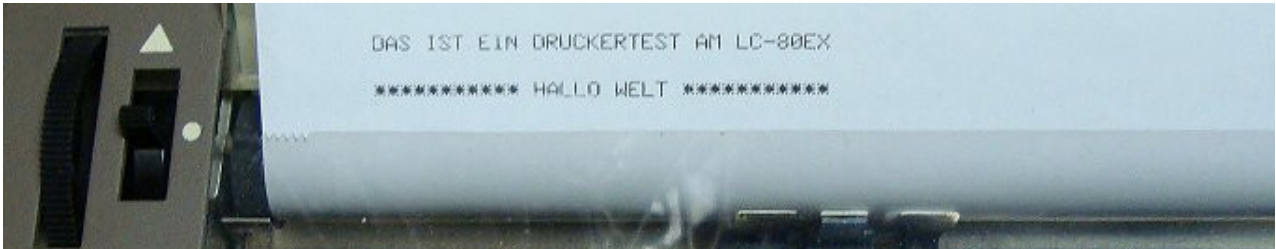


Funktionsmuster des Druckadapters

Er wird bei Bedarf einfach auf die Stiftleisten links und rechts neben der SIO aufgesteckt.

Der Anschluss des Druckers erfolgt bei mir per DIN-Buchse. Die Belegung entspricht der von Modul 003 am KC85.

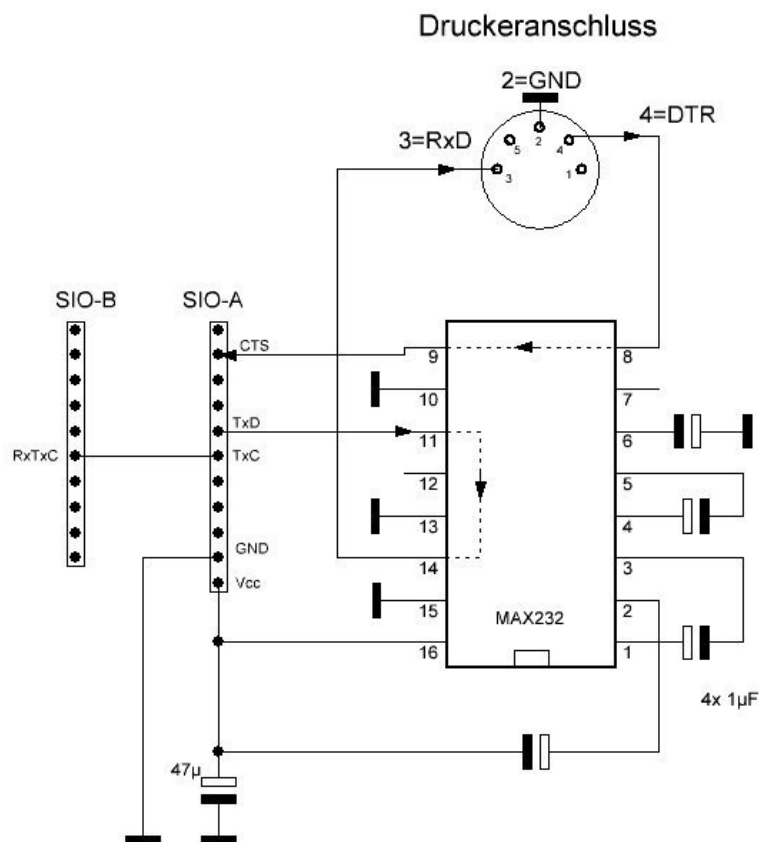
Getestet wurde mit einem Thermodrucker K6304.



Einstellungen:

- JP9: "1/2" = Takt 1,8 MHz, JP12: "SIO-0"
- Drucker mit serieller Schnittstelle auf 2400 Baud (K6304: DIL7=on, DIL6=off)

Schaltplan



Detailliertere Infos und Druckertestprogramm siehe gesondertes Dokument!

V.24-Transceiver

Mit diesem PC-Werkzeug kann über die serielle Schnittstelle bequem eine Datei gesendet oder eine empfangene Datei abgespeichert werden. Prüfsummen werden berechnet, angezeigt und können als Indiz für eine korrekte Übertragung dienen. Die bei anderen Werkzeugen zu findenden zahlreichen (für Gelegenheitsnutzer teilweise auch verwirrenden) Einstellmöglichkeiten entfallen.

V.24 Transceiver V. 0.8

Datei Test

Port Baudrate Parität Datenbits Stoppbits Flusskontrolle

COM schließen COM1 2400 keine 8 1 keine

RX

Art

AADR EADR

SADR

CRC h (o.Header)

Bytes

Name

Typ

Sichern

TX

BIN/Z80 laden Refresh

Art Geladen: z80

AADR 2000 EADR 2216

SADR 2000

CRC E9B0 h (o.Header)

Bytes 0217

Name UHR

Typ A = ausführbar, Auto

Datei senden

☒ Fortschrittsanzeige

Pause n. Header: keine

☐ als BIN senden (o. Header)

☐ Endlos senden

Puffer: E:\temp\UHR.z80

Das Werkzeug ist nur unter WINDOWS (XP bis WIN10/64bit) lauffähig, muss nicht installiert werden und schreibt außer einer INI-Datei keine Dateien unkontrolliert auf den PC.