

LCTOOLS

Version 4.0

für den LC-80ex

WeRo, Stand: 03.01.2017

Wesentliche Änderungen gegenüber Version 3:

- Treiber für Text-LCD aufgenommen
- Sprungverteiler um erweitert um **neue Grundfunktionen** (Tastatureingabe mit Cursorblinken, Zeichenkettenausgabe)
- Im Anschluss an INIT_SIO (CALL #A01B) wird das Druckerflag automatisch rückgesetzt.

Inhaltsverzeichnis

Übersicht.....	2
<i>Sprungverteiler.....</i>	<i>2</i>
<i>Anwenderprogramme.....</i>	<i>3</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>LCD-Anschluss.....</i>	<i>4</i>
<i>RS232-SAVE/LOAD mit HEX-Tastatur.....</i>	<i>5</i>
<i>Laden über RS232.....</i>	<i>5</i>
<i>Sichern über RS232.....</i>	<i>5</i>
<i>Bedienung der seriellen Schnittstelle mit Maschinencode.....</i>	<i>6</i>
<i>Initialisierung SIO.....</i>	<i>6</i>
<i>QWERTZ-Tastatur.....</i>	<i>6</i>
<i>Bildschirmausgabe.....</i>	<i>6</i>
<i>Drucker.....</i>	<i>7</i>
USB-Schnittstelle.....	8
<i>USB-SAVE/LOAD mit HEX-Tastatur.....</i>	<i>8</i>
<i>Laden über USB.....</i>	<i>8</i>
<i>Sichern über USB.....</i>	<i>8</i>
<i>USB-Arbeit mit Maschinencode.....</i>	<i>9</i>
<i>Initialisieren USB.....</i>	<i>9</i>
<i>USB-Inhaltsverzeichnis anzeigen.....</i>	<i>9</i>
<i>Dateiname vereinbaren.....</i>	<i>9</i>
<i>Laden & Speichern von z80-Dateien.....</i>	<i>10</i>
<i>Sequenzielles Lesen/Schreiben.....</i>	<i>10</i>
Anwendungen.....	11
<i>Maschinencode-Editor.....</i>	<i>11</i>
<i>Timekeeper-Uhr.....</i>	<i>11</i>
<i>AD-Umsetzer.....</i>	<i>11</i>
Anlage.....	12
<i>z80 Headersave.....</i>	<i>12</i>
<i>Realisierung USB-Adapter.....</i>	<i>13</i>
<i>Realisierung Druckadapter.....</i>	<i>15</i>
<i>Realisierung Text-LCD-Anschluss.....</i>	<i>16</i>

Übersicht

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 4 (abwärtskompatibel zu Version 3) stehen nunmehr zur Verfügung:

Einsprungsadresse	Funktion	Aufruf per
A000	Laden über USB	LC-80ex-HEX-Tastatur
A003	Sichern über USB	
A006	Laden über RS232	
A009	Sichern über RS232	
A00C	externes Laden über USB	MC-Programm
A00F	externes Sichern über USB	
A015	Initialisieren LCD	
A018	Initialisieren USB	
A01B	Initialisieren SIO-B	
A01E	Initialisieren SIO-A (Drucker)	
A021	serielle Eingabe (A)	
A024	serielle Ausgabe (A)	
A027	Druckausgabe ausschalten	
A02A	Druckausgabe einschalten	
A02D	USB-Directory anzeigen	
A036	LCD-Kommando senden	
A039	LCD-Daten senden	
A040	Ausgabe A als HEX-Zahl	
A043	Ausgabe DE als HEX-Zahl	
A046	Ausgabe DE als Dez-Zahl	
A049	Dateinamen abfragen	
A04C	Öffnen für Schreiben sequenziell	
A04F	Öffnen für Lesen sequenziell	
A052	Schließen Datei	
A055	Löschen Datei	
A058	1 Byte schreiben sequenziell	
A05B	1 Byte lesen sequenziell	
A05E	Filepointer positionieren	
A061	Eingabe mit Blinkkursor	
A064	Ausgabe Zeichenkette ab (HL) bis 00	

Es versteht sich von selbst, das unter Verwendung dieser Routinen erstellte Programme nur auf solchen LC-80ex lauffähig sind, die LCTOOLS4 installiert haben.

Anwenderprogramme

Aktuell sind folgende Anwenderprogramme in LCTOOLS4 enthalten:

	Start	Ende/Abbruch
Timekeeper-Uhr	ADR A030 EX	Taste "-"
AD-Wandler gem. Originalanleitung	ADR A033 EX	Taste "RES"
Maschinencode-Editor	ADR B000 EX	Q

Dateien

Dateiformat:	Headersave (z80)	
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 jedoch <u>nicht</u> mit anzugeben!

Darüber hinaus steht ein rohes BIN-Format beim sequenziellen Schreiben/Lesen zur Verfügung.

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	Filenamen eingeben	
	TYP	Dateityp eingeben	
	Err -U	USB-Fehler	danach weiter mit beliebiger Taste => Rückkehr zum Monitor (Warmstart)
	Err -F	Dateifehler (kein korrektes z80)	
	noFILE	File nicht gefunden	
	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)	
	Name too long (8)	Dateiname zu lang	
	USB Error.	allgemeiner USB-Fehler	

RS232-Schnittstelle

Terminal-Anschluss

Per RS232 ist sowohl ein 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 steht etwas im Widerstreit zur deren 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

Dazu wird das gleiche o.a. Protokoll benutzt, jedoch nur für Ausgabe an SIO Kanal A. Details siehe [Anhang](#) und gesondertes Dokument "Drucken_LC80ex".

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.

LCD-Anschluss

LCTOOLS4 unterstützt nun die Ansteuerung eines Text-LCD am User-Port. Einzelheiten dazu siehe [Anhang](#) und gesondertes Dokument „LCD.PDF“.

RS232-SAVE/LOAD mit HEX-Tastatur

Laden über 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.

Sichern über 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 =Dateityp (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).

Bedienung der seriellen Schnittstelle mit Maschinencode

Die serielle Schnittstelle des LC-80ex kann mit den Calls #A021 (Eingabe) und #A024 (Ausgabe) beliebige Geräte bedienen.

Die Beschreibung erfolgt für Eingabe per Tastatur und Ausgabe an einen Bildschirm. Für diese beiden Geräte stehen weitere nützliche Funktionen zur Verfügung:

Initialisierung SIO

Vor der Verwendung der SIO muss diese (und die CTC) programmiert werden:

CALL	#A01B	;init SIO-B:	2400/8/1/N, kein Handshake
CALL	#A01E	;init SIO-A:	für Drucker 2400/8/1/N, CTS-Handshake

QWERTZ-Tastatur

CALL	#A021	;Zeichen holen ohne Warten auf Taste ;Keine Taste: Rückkehr mit gesetztem Z-Flag, A=0 ;Taste gedrückt: Rückkehr Z=0, ASCII-Zeichen in A	
CALL	#A061	;Zeichen holen, wartet mit blinkendem Cursor ;Zeichenrückgabe in A ;vor Aufruf in DE Hindergrundzeichen (#20 o.a.) definieren: ;LD DE,#2370 ;freie Zelle (vorzugsweise) ;LD A,"?" ;Blinkzeichen ;LD (DE),A ;CALL #A061	

Bildschirmausgabe

CALL	#A024	;Zeichen in A seriell als ASCII ausgeben ;bei Steuerzeichen (<#20) und Sonderzeichen >#7F ist die ;Reaktion vom benutzten Terminal abhängig	
CALL	#A040	;Ausgabe als HEX-Zahl auf Schirm, kein Reg. verändert	
CALL	#A043	;Hexadezimalausgabe Register DE auf Schirm (bei HL=0000) bzw. ;Ablage DE im RAM ab Adresse HL als HEX-ASCII-String ;LD DE,WERT ;hexadezimal auszugebende 16Bit-Zahl ;LD HL,0 ;Bildschirmausgabe gewünscht ;CALL #A046	
CALL	#A046	;Dezimalausgabe Register DE auf Schirm (bei HL=0000) bzw. ;Ablage DE im RAM ab Adresse HL als DEZ-ASCII-String ;LD DE,WERT ;dezimal auszugebende 16Bit-Zahl ;LD HL,0 ;Bildschirmausgabe gewünscht ;CALL #A046	
CALL	#A064	;Ausgabe einer Zeichenkette ab HL bis 00 erkannt ;TEXT: DEFM „HALLO“,0 ;... ;LD HL,TEXT ;Anfangsadresse gewünschter Text ;CALL #A064 ;Ausgabe!	

Drucker

Der Drucker kann ein- oder ausgeschaltet werden. Mit diesen zwei Calls wird lediglich ein Flag beeinflusst. Die Ausgaberroutine prüft dieses und gibt (wenn gesetzt) zusätzlich zum Bildschirm auch auf den Drucker aus.

CALL	#A027	;Drucken ein (parallel zum Bildschirm)
CALL	#A02A	;Drucken aus

In folgenden Fällen ist eine eigene Druckausgabe nötig:

- Senden von Steuerzeichen außer #0D/#0A (z.B. für die Schrifteinstellung eines Druckers)
- Druckausgabe „solo“ und nicht parallel zum Bildschirm

Das muss dann ohne Benutzung von Drucker ein/aus "zu Fuß" erfolgen, z.B. so:

```
SIOAC: EQU #DE
SIOAD: EQU #DC
      CALL #A01E      ;Initialisieren Drucker
...      ;auszugebendes (Steuer-)Zeichen in A
      PUSH AF          ;Ausgabezeichen sichern
CHK:    LD  C,SIOAC     ;SIO Steuerung A
      IN   A,(C)        ;lesen RR0
      BIT  2,A          ;sendebereit?
      JR   Z,CHK        ;nein => weiter abfragen
      POP  AF           ;ja  => Ausgabezeichen holen
      LD   C,SIOAD      ;SIO Daten A
      OUT  (C),A        ;Zeichen an Drucker senden
;...
```

USB-Schnittstelle

Ein USB-Anschluss (VINCULUM) am Systembus ermöglicht Laden und Speichern von Dateien. Das kann sowohl per HEX-Tastatur als auch Aufruf per Maschinenprogramm geschehen. Der Anschluss anderer USB-Geräte außer Speichermedien ist nicht vorgesehen.

USB-SAVE/LOAD mit HEX-Tastatur

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 über USB

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 über USB

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 - 80	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-Arbeit mit Maschinencode

Initialisieren USB

In eigenen MC-Programmen ist die USB-Schnittstelle zunächst mit INIT_USB zu initialisieren und ggf. dabei auftretende Fehler abzufangen.

CALL	#A018	;Init_USB
OR	A	;fehlerfrei?
JP	NZ,xxxx	;nein=> entsprechende Reaktion...

USB-Inhaltsverzeichnis anzeigen

CALL	#A018	;Init_USB	
OR	A	;fehlerfrei?	
JP	NZ,xxxx	;nein=> entsprechende Reaktion...	
CALL	#A02D	;USB-Directory anzeigen	
		;bei >20 Dateien wird gestoppt (Anzeige "!")	NEU!
		;Abbruch mit Strg+C, weiter mit anderer Taste	NEU!

Dateiname vereinbaren

Die Dateiarbeit erfordert die Angabe eines Dateinamens. Nachfolgende Routine realisiert das:

CALL	#A049	;Dateinamen für SAVE/LOAD abfragen, max. 8 Zeichen
OR	A	;Fehler?
JP	NZ,xxxx	;ja, Abbruch
LD	A,(IX+0)	;1. Zeichen prüfen
CP	"."	;Name eingegeben?
JP	Z,xxxx	;nein, war nur <ENTER> => Abbruch!
...		;in Puffer bereitgestellt, terminiert durch #0D/#00
		;IX weist auf 1. Zeichen

Der mit o.a. Routine eingegebene Dateiname trägt standardmäßig immer die Erweiterung *.z80. Ist das z.B. für die sequenzielle Dateiarbeit nicht erforderlich, so kann das wie folgt geändert werden:
Beispiel: ab IX steht "TEST.z80", ändern in "TEST.BIN":

	PUSH	IX	;Anfang Dateiname
	POP	HL	
ct1:	LD	A,(HL)	
	INC	HL	
	CP	"."	
	JR	nz,ct1	;Punkt suchen
	LD	(HL),"B"	;B ablegen
	INC	HL	
	LD	(HL),"I"	;I ablegen
	INC	HL	
	LD	(HL),"N"	;N ablegen
	RET		

Laden & Speichern von z80-Dateien

Hierbei wird eine Datei (<64kB) komplett in den Speicher eingelesen bzw. ein Speicherbereich komplett als Datei gespeichert. Alle nötigen Angaben sind in Registern bereitzustellen:

Laden:		
CALL	#A018	;Init_USB
OR	A	;fehlerfrei?
JP	NZ,xxxx	;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 mit A=0
		;wenn Datei nicht vorhanden ("NOT FOUND"), dann A=0
		;kein Autostart!
Speichern:		
CALL	#A018	;Init_USB
OR	A	;fehlerfrei?
JP	NZ,xxxx	;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	;Speichern aufrufen
		;wenn fertig und fehlerfrei, Rückkehr mit A=0
		;bei existenter Datei (Überschreiben= "nein") A=2

Sequenzielles Lesen/Schreiben

Die nachfolgenden Routinen ermöglichen einen wahlfreien lesenden oder schreibenden Zugriff auf beliebige Inhalte einer beliebigen Datei auf dem USB-Stick. Die Dateigröße ist dabei nicht auf 64kB beschränkt (max. ca. 2GB bei FAT16). Anwendungsbeispiel siehe gesondertes Dokument.

Wie bei den z80-Funktionen muss auch hier (vor dem Öffnen/Schließen/Löschen/Positionieren) ein Dateiname bereitgestellt werden und **IX dabei konstant auf das 1. Zeichen des Dateinamens weisen** (s.o.)

CALL	#A04C	;Öffnen für Schreiben	
		;Existiert Datei nicht, wird sie angelegt, ansonsten geöffnet	
		;und der Filepointer hinter das Dateiende gesetzt	
CALL	#A04F	;Öffnen für Lesen, Filepointer am Dateianfang	
OR	A	;fehlerfrei? (dann Dateilänge in #236A, 32 Bit!)	
JP	NZ,xxxx	;nein=> entsprechende Reaktion (A=2 wenn "Not found.")	
CALL	#A05E	;Filepointer positionieren	
		;absolute Position in HL (high) und DE (low)	
CALL	#A052	;Schließen Datei	
CALL	#A055	;Löschen Datei	
CALL	#A058	;1 Byte schreiben sequenziell	Byte in A
CALL	#A05B	;1 Byte lesen sequenziell	

Eine geöffnete Datei muss am Ende der Bearbeitung unbedingt geschlossen werden, ansonsten Gefahr von Datenverlust.

Anwendungen

Maschinencode-Editor

Dieser bietet u.a. folgende Möglichkeiten:

- Speicheransicht/ -Modifikation,
- Hex- & ASCII-Dump,
- Byteeingabe, Kopieren, Füllen, Suchen,
- Registeransicht/ -Modifikation,
- Disassembler,
- Debugger (Haltepunkte, Einzelschritt),
- Starten von Programmen,
- Ermittlung von Prüfsummen,
- Filearbeit per USB,
- Druckausgabe per RS232,
- Kommunikation per RS232.

Start mit **ADR B000 EX**

```
*PC 2400 CD 18 A0 B7
SP 2100 00 00 00 00
IX 0000 FF FF FF FF
IX 0000 FF FF FF FF
HL 0000 FF FF FF FF
DE 0000 FF FF FF FF
BC 0000 FF FF FF FF
AF 0044 N 31 56 18
IR 0252 V 5 10

23FC 00 .
23FD 00 .
23FE 00 .
23FF 00 .
>2400 CD< H
2401 18 .
2402 A0 -
2403 B7 ■

2400 CD18A0 CALL #A018
#
```

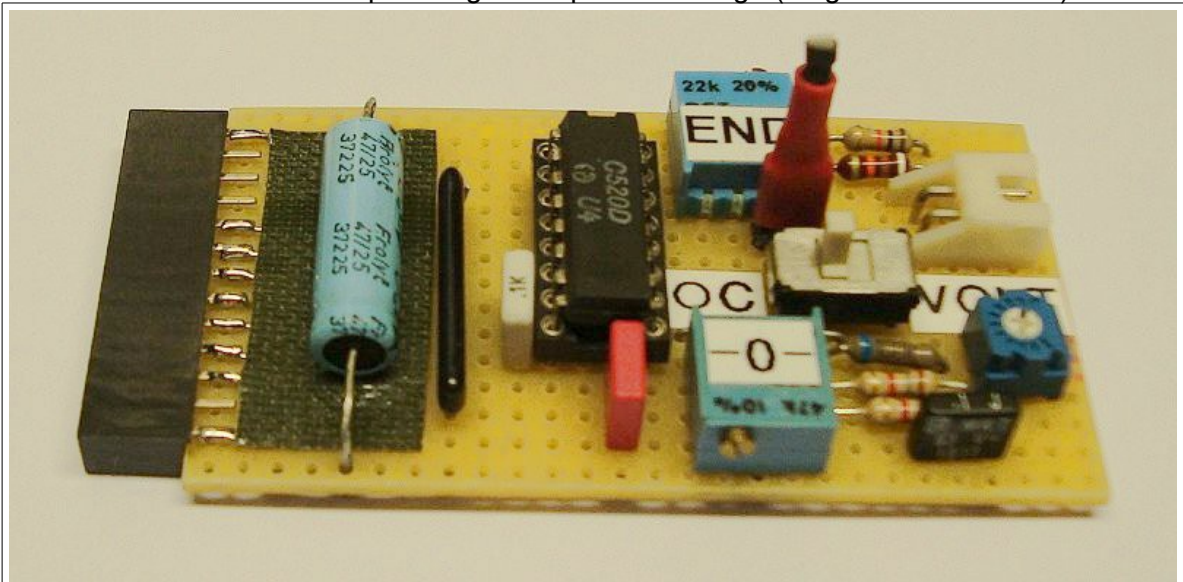
Beschreibung siehe gesondertes Dokument!

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".

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)	ggf. "LC80ex"
0C	TYP	("A"... "F")
0D...0F	3X 0XD3	"MAGIC-BYTE" (Z80-HEADERSAVE-KENNUNG)
10...1F	PROGRAMM-NAME	ASCII

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.

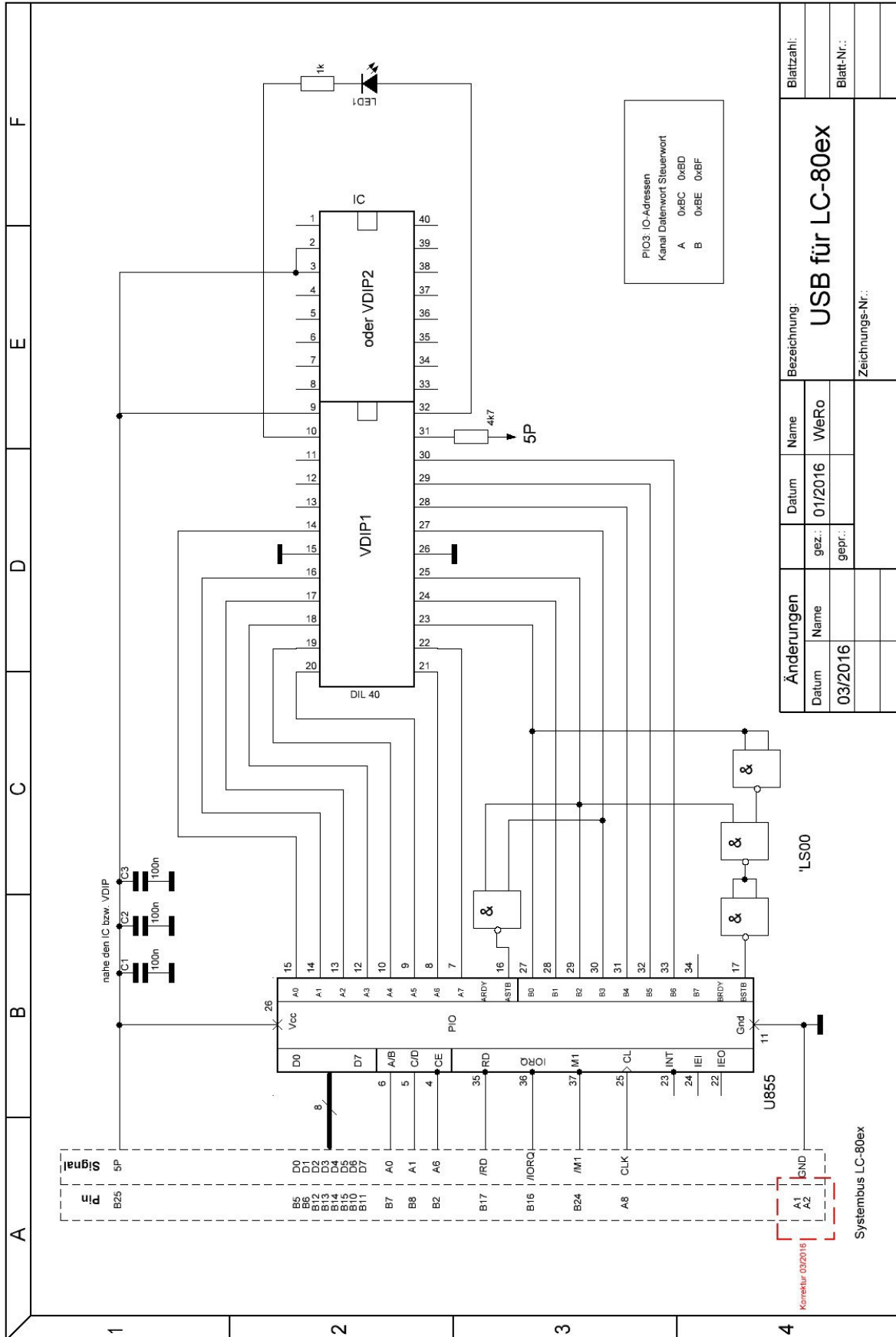
Musteraufbau ("gefädelt")

Inbetriebnahme:

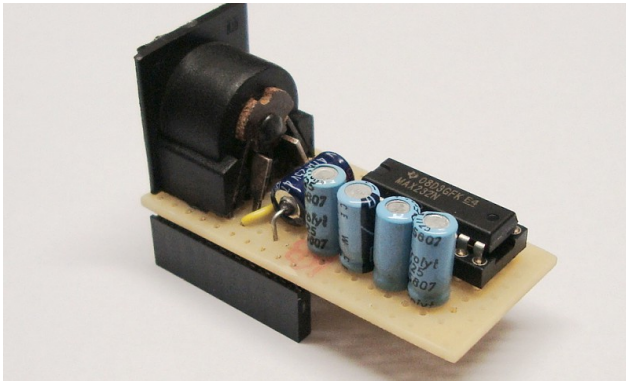
(ich beschreibe hier meine Vorgehensweise, es geht sicher auch anders...)

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 (sicherheitshalber ca. 10 sek.) 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

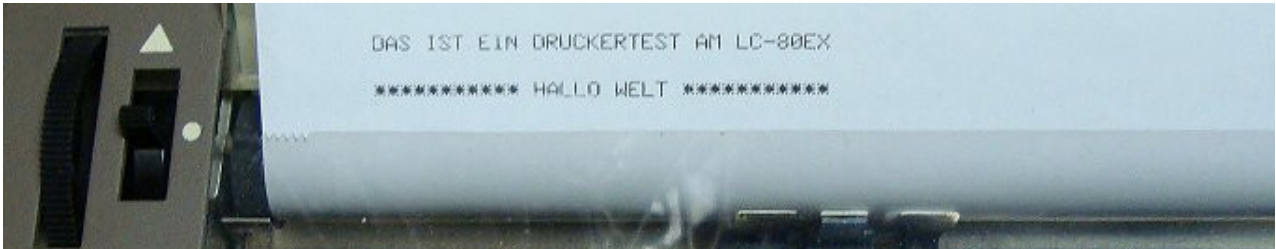


Funktionsmuster des Druckadapters

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.

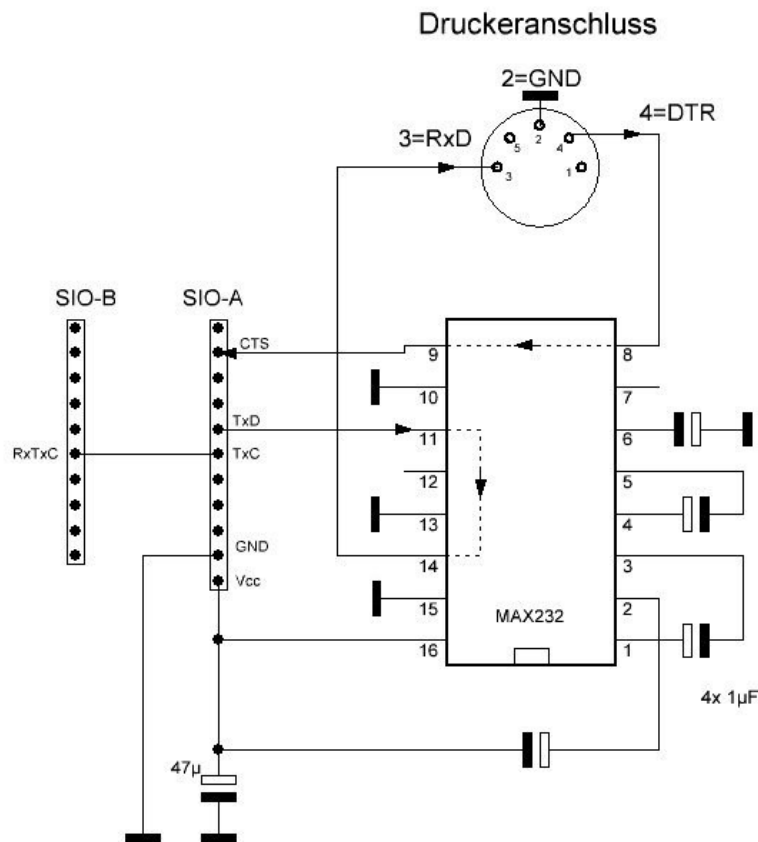
Getestet wurde mit einem Thermodrucker K6304.



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!

Realisierung Text-LCD-Anschluss

Funktionsmuster mit Demo-Programm		
Grundroutinen im Sprungverteiler	#A015	initialisiert PIO + LCD
	#A036	Ausgabe Kommandobyte in A an LCD
	#A039	Ausgabe Datenbyte in A an LCD
Anschlussplan		
<p style="text-align: center;">TC1602E-01 Achtung: Vss und Vdd gegenüber "normalem" LCD vertauscht</p> 