Keine Angst vor'm LCD

Ein Text-LCD (auch) am LC-80(ex)

WeRo, Stand: 30.08.2016

Die LED-Anzeige am LC-80(ex) ist etwas „spartanisch“. Mit der 7-Segment-Darstellung sind auch nicht alle ASCII-Textzeichen korrekt darstellbar. Nachfolgender Beitrag zeigt eine Möglichkeit, wie ein Text-LCD (im Beispiel 16x2 Zeichen) an den LC-80ex angeschlossen werden kann. An einem Maschinencode-Programm und einem 8k-BASIC-Programm wird die Ansteuerung erläutert.

Ich habe hier beschrieben, wie man es machen kann und dass es funktioniert. Wozu man das benutzen könnte und ob jemand eine solche Anzeige überhaupt braucht, das steht auf einem anderen Blatt :-)

Sinngemäß lässt sich die Ansteurung auch auf jeden anderen z80-Computer übertragen, der über einen freien Port verfügt.

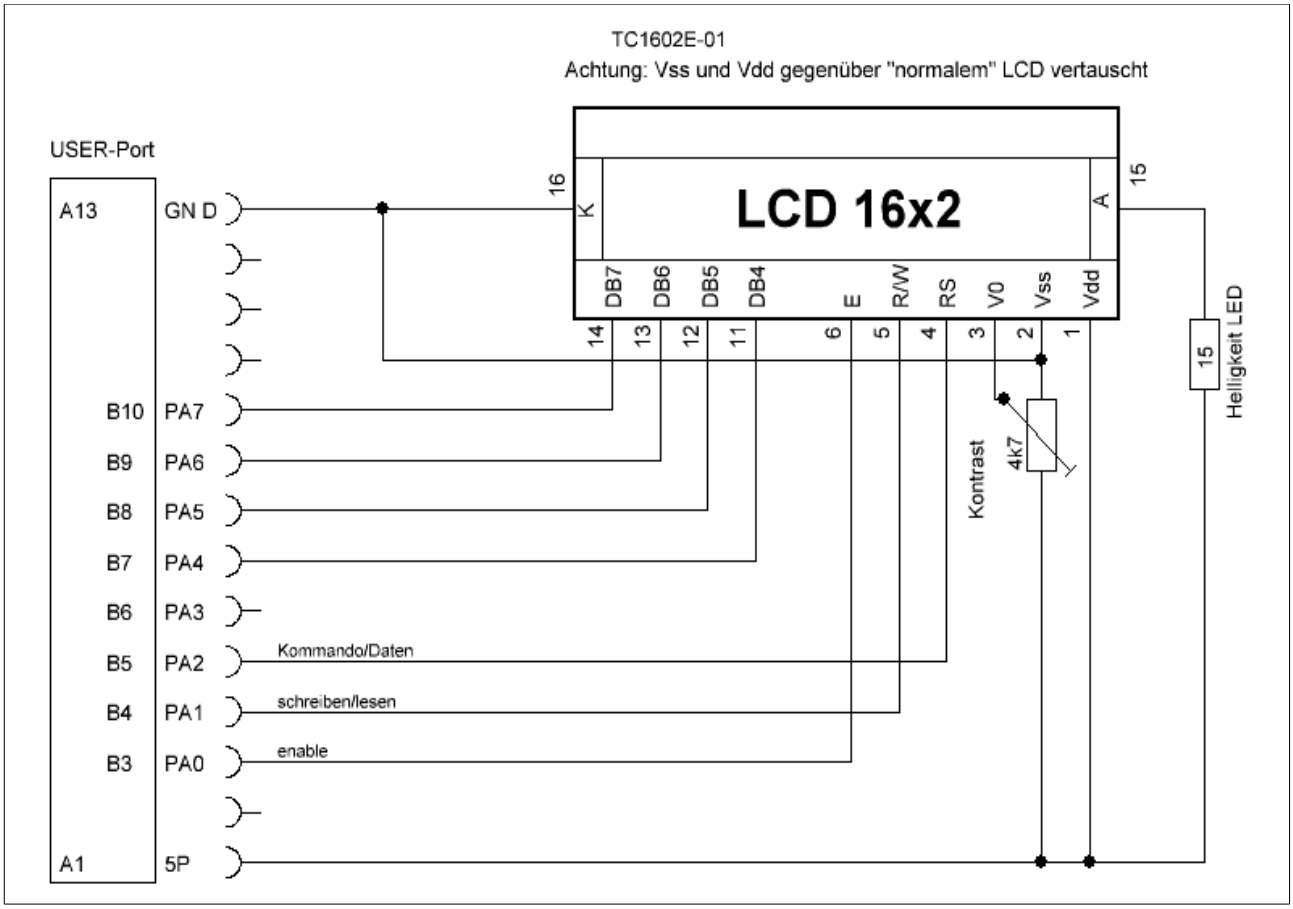
Wer sich weiter zur Problematik „Text-LCD“ informieren will, findet hier eine gute Quelle:

http://www.sprut.de/electronic/lcd/

Inhaltsverzeichnis

Hardware

Bei der Ansteuerung wurde der 4-Bit-Modus gewählt. Damit kann das LCD sehr einfach am USER-Port angeschlossen werden. Die Leitungen DB0...DB3 des LCD bleiben offen.



Display

Als Display sind alle geeignet, die den Kontrollertyp **HD44780 oder KS0070B** (kompatibel) aufweisen. Im Funktionsmuster wurde ein „TC1602E-01“ (Pollin) benutzt. Achtung, die Pinbelegung weicht bei diesem Typ von Standard ab (Pin1 und 2 vertauscht).

Prinzipiell sind auch andere LCD verwendbar. Dabei ist die im Datenblatt angegebene Pinbelegung zu beachten. Preiswerte Displays benötigen oft eine negative Kontrastspannung! Bei Anlegen der Betriebsspannung und noch ohne Ansteuerung müssen mit richtiger Kontrastspannung dunkle Kästchen zu sehen sein.

Andere LCD als 16x2 haben vom hier beschriebenen Schema ggf. abweichende Besonderheiten in der Ansteuerung (z.B. die Zeichen-Positionierung)!

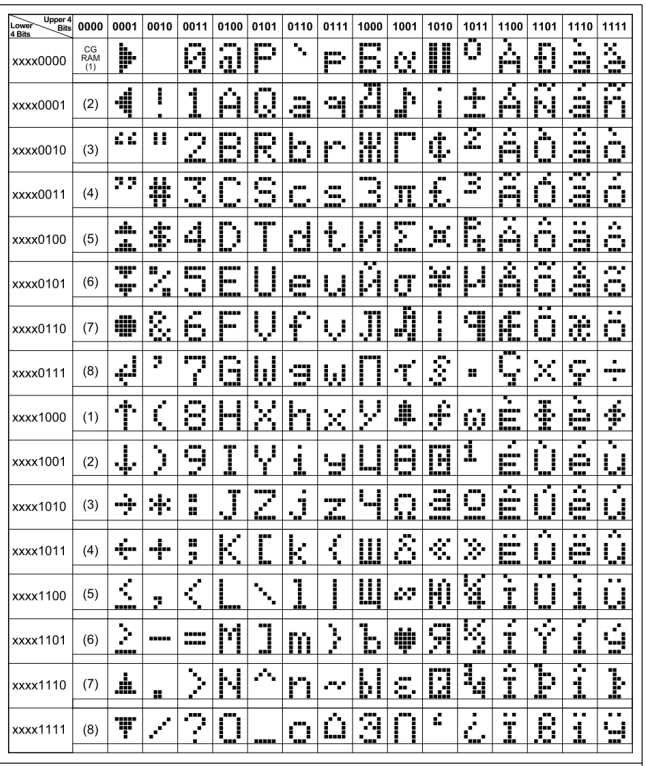
Der Vorwiderstand für die LED-Hintergrundbeleuchtung ist so zu wählen, dass eine ausreichende Helligkeit erreicht, der maximal zulässige Strom aber nicht überschritten wird.

Zeichensatz

Es gibt leider keine einheitliche Zeichendarstellung für die Text-LCD. Der Zeichensatz ist vom verwendeten Controller abhängig. Die Standard-ASCII-Zeichen #20 (Leerzeichen) bis #7A (z) sind jedoch auf den meisten LCD etwa gleich aussehend belegt.

Für den Kontroller **HD44780** (Quelle: Datenblatt) wurde von HITACHI beispielsweise festgelegt:

ROM-Code A00



ROM-Code A02

Welcher Zeichensatz konkret vorhanden ist, sollte aus dem Datenblatt des jeweiligen LCD hervorgehen oder kann durch Testausgabe eines bestimmten Zeichens durch Probieren herausgefunden werden.

Die meisten LCD zeigen im Grundzustand die einzelnen Zeichen in einer Matrix von 5x8 Punkten an. Mit speziellen Kommandos kann man sich auch selbst bis zu acht eigene Zeichen definieren. Wie das geht, findet man auch bei SPRUT.

Software

Die Ansteuerung ist eigentlich simpel. Am LCD sind die nötigen Daten (Kommando oder Ausgabedaten) anzulegen. Dann wird ein 0=>1=>0 Impuls an den E-Eingang gegeben. An der High-Flanke dieses Impulses liest das LCD den Pegel am RW-Eingang. Der entscheidet, ob die Daten vom LCD als Kommando oder als Anzeigedaten zu interpretieren sind. Gleichfalls an dieser High-Flanke wird der Pegel am RS-Eingang gelesen. Damit stellt das LCD fest, ob in das LCD geschrieben oder vor dort gelesen (wird hier nicht benutzt) werden soll. An der fallenden Flanke des Impulses erfolgt die Kommandoausführung bzw. es wird das Zeichen angezeigt.

Ablauf:

- USER-PIO A programmieren
- LCD initialisieren/vorbereiten
- Aufruf Ausgaberoutine (Kommando oder Daten)

Auf eine Abfrage des LCD, ob dieses bereit ist, wird verzichtet. Bei einer Ansteuerung mit CPU-Takt 1,8 MHz werden im Zusammenhang mit den dargestellten Routinen offensichtlich die Bedingungen an die nötigen Signalverzögerungen erfüllt (nicht weiter untersucht, es funktioniert).

Die nachfolgenden Ansteuerbeispiele gelten für eine Signalzuordnung gem. obiger Schaltung:

Port-Bit	LCD-Pin	Bem.
PA0	E	Enable-Impuls 0=>1=>0
PA1	RW	0=Schreiben, 1 = Lesen
PA2	RS	0=Kommando, 1= Daten
PA3	-	unbelegt
PA4	D4	Daten (LSB)
PA5	D5	Daten
PA6	D6	Daten
PA7	D7	Daten (MSB)

Einige Kommandos

#30	Rücksetzen (3x ausführen!)
#20	4-Bit-Modus einschalten
#28	2-Zeilen-Modus
#0C	Display ein, Kursor unsichtbar
#06	Kursorinkrement (rückt nach Zeichenausgabe automatisch weiter)
#01	Display löschen, Kursor an den Zeilenanfang
#02	Kursor home (Zeilenanfang)
#10	Kursor 1 Schritt zurück
#14	Kursor 1 Schritt vor
#80	Kursor positionieren auf Anfang 1. Zeile
#C0	Kursor positionieren auf Anfang 2. Zeile

Im 4-Bit-Modus sind zuerst das obere Nibble und dann das untere Nibble des Kommando- bzw. Datenbytes an PA4...PA7 anzulegen und auszugeben. Das wird von der Senderoutine automatisch vorgenommen. Für das Senden eines Kommandos und von Daten gibt es zwei getrennte Routinen, da der RS-Eingang des LCD anders bedient werden muss.

Ansteuerung in Maschinencode

Komplettbeispiel "Hallo Welt", geschrieben mit „LC80ex-Assembler“

```
 10         ORG  #2400
 20         ENT  $
 30 UPIOAC  EQU  #FA         ;USER PIO A, STEUERUNG
 40 UPIOAD  EQU  #F8         ;USER PIO A, DATEN
 50         JR   START
 60 TEXT1   DEFM "HALLO WELT!";AUSGABETEXTE
 70 TEXT2   DEFM "LC-80EX"
 80 ENA     OUT  (UPIOAD),A  ;DATEN ANLEGEN + ENABLE-IMPULS
 90         OR   1           ;Bit0 = ENABLE-LEITUNG
100         OUT  (UPIOAD),A  ;EIN
110         AND  #FE
120         OUT  (UPIOAD),A  ;AUS
130         RET
140 SENDK   PUSH AF          ;AUSGABE KOMMANDO, KOMMT MIT A=KOMMANDOCODE
150         AND  #F0         ;NUR OBERES NIBBLE + RS=0 (KOMMANDO)
160         CALL ENA         ;ENABLE-IMPULS
170         POP  AF
180         RLA
190         RLA
200         RLA
210         RLA
220         AND  #F0         ;UNTERES NIBBLE, VERSCHOBEN NACH OBEN
230         CALL ENA         ;ENABLE-IMPULS
240         RET
250 SENDD   PUSH AF          ;AUSGABE DATEN, KOMMT MIT A=DATENBYTE
260         AND  #F0         ;NUR OBERES NIBBLE
270         OR   4           ;RS=1 (DATEN)
280         CALL ENA         ;ENABLE-IMPULS
290         POP  AF
300         RLA
310         RLA
320         RLA
330         RLA
340         AND  #F0         ;UNTERES NIBBLE, VERSCHOBEN NACH OBEN
350         OR   4           ;RS=1 (DATEN)
360         CALL ENA         ;ENABLE-IMPULS
370         RET
380 START   LD   A,#CF       ;USER-PIO A PROGRAMMIEREN
390         OUT  (UPIOAC),A  ;EINZELBITSTEUERUNG
400         XOR  A
410         OUT  (UPIOAC),A  ;ALLES AUSGÄNGE
420         OUT  (UPIOAD),A  ;ALLE AUSGÄNGE AUF 0
430         LD   B,3
440 INIT    LD   A,#30       ;LCD STARTUP AUS JEDEM ZUSTAND
450         CALL ENA
460         DJNZ INIT
470         LD   A,#20       ;4-BIT-MODUS
480         CALL ENA
490         LD   A,#28       ;2-ZEILEN-MODUS
500         CALL SENDK
510         LD   A,#0C       ;DISPLAY EIN, KURSOR UNSICHTBAR
520         CALL SENDK
```

```
530         LD   A,#06       ;KURSOR INKREMENT
540         CALL SENDK
550 LOOP    LD   A,#80       ;KURSOR AUF ANGFANG ZEILE 1
560         CALL SENDK
570         LD   B,11        ;LÄNGE TEXT 1
580         LD   HL,TEXT1    ;ANFANGSADRESSE TEXT 1
590 AUS1    LD   A,(HL)      ;ZEICHEN HOLEN
600         CALL SENDD       ;AUSGEBEN
610         INC  HL          ;NÄCHSTES
620         DJNZ AUS1
630         LD   A,#C0       ;KURSOR AUF ANGFANG ZEILE 2
640         CALL SENDK
650         LD   B,7
660         LD   HL,TEXT2
670 AUS2    LD   A,(HL)
680         CALL SENDD
690         INC  HL
700         DJNZ AUS2
710         LD   A,#80       ;ANZEIGEPAUSE
720         CALL PAUSE
730         LD   A,1         ;KOMMANDO: LCD LÖSCHEN
740         CALL SENDK
750         LD   A,#40       ;PAUSE MIT LEEREM LCD
760         CALL PAUSE
770         CALL #0483       ;DAK2 ABFRAGE TASTATUR
780         CP   #17         ;"-"-TASTE GEDRÜCKT?
790         JR   NZ,LOOP     ;NEIN
800         JP   0           ;JA   (FÜR TESTBETRIEB IM ASM DURCH RET ERSETZEN)
810 PAUSE   PUSH BC          ;PAUSE, KOMMT MIT LÄNGE IN A
820         LD   B,A
830         LD   C,0
840 PAUSE1  DEC  BC
850         LD   A,B
860         OR   C
870         JR   NZ,PAUSE1
880         POP  BC
890         RET
```

Nach dem Programmstart blinkt auf dem LCD „HALLO WELT“, bis die **-** Taste auf der HEX-Tastatur gedrückt wird.

Ansteuerung in 8k-BASIC

```
  1  REM LCD-DEMO
 10  OUT 250,207           :REM USER PIO A PROGRAMMIEREN
 15  OUT 250,0             :REM EINZELBIT, ALLES AUSGÄNGE
 20  OUT 248,0             :REM ALLE AUSGÄNGE AUF 0

 25  FOR I=1 TO 3          :REM DISPLAY VORBEREITEN
 30  DA=48:GOSUB200        :REM INIT AUS BELIEBIGEM ZUSTAND
 35  NEXT
 40  DA=32:GOSUB200        :REM 4-BIT-MODUS
 45  DA=40:GOSUB300        :REM KOMMANDO: 2-ZEILENMODUS
 50  DA=12:GOSUB300        :REM KOMMANDO: DISPLAY EIN, KURSOR UNSICHTBAR
 55  DA=6: GOSUB300        :REM KOMMANDO: KURSOR INKREMENT
 60  T1$="HALLO WELT!"     :REM TEXT FÜR ZEILE 1
 65  T2$="LC-80EX"         :REM TEXT FÜR ZEILE 2
 70  DA=1:GOSUB300         :REM LCD LEEREN
 75  T$=T1$:GOSUB100       :REM TEXT 1 AUSGEBEN
 80  DA=192:GOSUB300       :REM POSITIONIEREN AUF ANFANG ZEILE 2
 85  T$=T2$:GOSUB100       :REM TEXT 2 AUSGEBEN
 92  FORP=1TO500:NEXT      :REM ANZEIGEPAUSE
 95  GOTO70                :REM SCHLEIFE (ABBRUCH MIT STRG+C)

100  FOR I=1TO LEN(T$)     :REM TEXTAUSGABE, KOMMT MIT T$=AUSGABETEXT
105  DA=ASC(MID$(T$,I,1))  :REM ZEICHENCODE HOLEN
110  GOSUB400:NEXT         :REM ZEICHEN ANZEIGEN
115  RETURN

200  OUT 248,DA            :REM ENABLE-IMPULS, KOMMT MIT DA=SENDEDATEN
205  DA=DA OR 1            :REM ENABLE EIN
210  OUT 248,DA            :
215  DA=DA AND 254         :REM ENABLE AUS
220  OUT 248,DA            :
225  RETURN

300  DB=DA                 :REM KOMMANDO SENDEN, KOMMT MIT DA=KOMMANDOCODE
305  DA=DA AND 240         :REM OBERES NIBBLE, RS=0 (KOMMANDO)
310  GOSUB 200             :REM AUSGEBEN
315  DA=16*(DB AND 15)     :REM UNTERES NIBBLE, VERSCHIEBEN, RS=0 (KOMMANDO)
320  GOSUB 200             :REM AUSGEBEN
325  RETURN

400  DB=DA                 :REM DATEN SENDEN, KOMMT MIT DA=DATENBYTE
405  DA=DA AND 240         :REM OBERES NIBBLE
410  DA=DA OR 4            :REM RS=1 (DATEN)
415  GOSUB 200             :REM AUSGEBEN
420  DA=16*(DB AND 15)     :REM UNTERES NIBBLE, VERSCHIEBEN
425  DA=DA OR 4            :REM RS=1 (DATEN)
430  GOSUB 200             :REM AUSGEBEN
435  RETURN
```

Gegenüber dem Beispiel in Maschinencode erfolgt die Ansteuerung in BASIC relativ langsam.

LCTOOLS4-Unterstützung

Mit dem Treiberpaket ***LCTOOLS4*** lässt sich das Ganze noch vereinfachen. Es stehen dazu im Sprungverteiler folgende Grundfunktionen zur Verfügung:

```
#A015:      LCD_I       ;Initialisieren (2x16)
#A036:      LCD_K       ;senden Kommando an LCD
#A039:      LCD_D       ;senden Anzeigedaten an LCD
```

Die o.a. Beispiele fallen damit kürzer aus:

```
      ORG   #2400       ;STARTADRESSE
      ENT   $
LCD_I EQU   #A015       ;LCTOOLS4 ERFORDERLICH!
LCD_K EQU   #A036
LCD_D EQU   #A039
      JR    START
TEXT1 DEFM  "HALLO WELT!";AUSGABETEXTE
TEXT2 DEFM  "LC-80EX"

START CALL  LCD_I       ;PIO PROGRAMMIEREN + LCD INITIALISIEREN
LOOP  LD    A,#80       ;KURSOR AUF ANGFANG ZEILE 1
      CALL  LCD_K       ;=KOMMANDO AUSGEBEN
      LD    B,11        ;LÄNGE TEXT 1
      LD    HL,TEXT1    ;ANFANGSADRESSE TEXT 1
AUS1  LD    A,(HL)      ;ZEICHEN HOLEN
      CALL  LCD_D       ;=DATEN AUSGEBEN
      INC   HL          ;NÄCHSTES
      DJNZ  AUS1
      LD    A,#C0       ;KURSOR AUF ANGFANG ZEILE 2
      CALL  LCD_K       ;=KOMMANDO AUSGEBEN
      LD    B,7
      LD    HL,TEXT2
AUS2  LD    A,(HL)
      CALL  LCD_D       ;=DATEN
      INC   HL
      DJNZ  AUS2
      LD    A,#80       ;ANZEIGEPAUSE
      CALL  PAUSE
      LD    A,1         ;LCD LÖSCHEN
      CALL  LCD_K       ;=KOMMANDO AUSGEBEN
      LD    A,#40       ;PAUSE LEERES LCD
      CALL  PAUSE
      CALL  #0483       ;DAK2 ABFRAGE TASTATUR
      CP    #17         ;“-“-TASTE GEDRÜCKT?
      JR    NZ,LOOP     ;NEIN
      RET               ;JA => ENDE (FÜR FERTIGES PROGRAMM DURCH JP 0 ERSETZEN)


PAUSE PUSH  BC          ;PAUSE, KOMMT MIT LÄNGE IN A
      LD    B,A
      LD    C,0
PAUS1 DEC   BC
      LD    A,B
      OR    C
      JR    NZ,PAUS1
      POP   BC
      RET
```

BZW. IN BASIC:

```
  1  REM LCD-DEMO MIT LCTOOLS4
 10  CALL #A015             :LCD INITIALISIEREN
 60  T1$="HALLO WELT!"      :REM TEXT FÜR ZEILE 1
 65  T2$="LC-80EX"          :REM TEXT FÜR ZEILE 2
 70  DA=1:  CALL #A036      :KOMMANDO: LCD LEEREN
 75  T$=T1$:GOSUB100        :REM TEXT 1 AUSGEBEN
 80  DA=192:CALL #A036      :KOMMANDO: POSITIONIEREN AUF ANFANG ZEILE 2
 85  T$=T2$:GOSUB100        :REM TEXT 2 AUSGEBEN
 92  FORP=1TO500:NEXT       :REM ANZEIGEPAUSE
 95  GOTO70                 :REM SCHLEIFE (ABBRUCH MIT STRG+C)
100  FOR I=1TO LEN(T$)      :REM TEXTAUSGABE, KOMMT MIT T$=AUSGABETEXT
105  DA=ASC(MID$(T$,I,1))   :REM ZEICHENCODE HOLEN
110  CALL #A039             :DATEN SENDEN
115  NEXT:RETURN
```