Thomas Schlenger-Klink

Basic-EMUF mit LCD und Tasten

Bei vielen Anwendungen von Einplatinen-Computern ist die Ein-/Ausgabe von Daten während des Betriebs notwendig, so auch beim Basic-EMUF. Die Programmierung einer solchen Ein-Ausgabe-Einheit ist bei Realisierung in Maschinensprache relativ umfangreich und zeitaufwendig. Die hier vorgestellte Lösung LCDKEY ist einfach zu handhaben, da die Programmierung mit den Basic-Befehlen „PRINT" und „INPUT" erfolgt. Die Daten können dabei alphanumerisch angezeigt und eingegeben werden.

Die LCDKEY-Baugruppe wurde in erster Linie als Ein-/Ausgabe-Baugruppe für den Basic-EMUF entwickelt. Der Anschluß an andere Computer ist jedoch ebenfalls möglich. Die Ansteuerung der Tastatur und des LC-Moduls erfolgt über einen PIO-Baustein 8255. Dieser Baustein stellt 24 TTL Ein-Ausgabeleitungen zur Verfügung. Damit kann die Platine auch an andere Computer angeschlossen werden (z.B. 8085, Z80 etc.). Die Anzeige besteht aus einem fertig montierten LC-Modul, das in den verschiedensten Ausführungsformen erhältlich ist (mit und ohne Beleuchtung, 16-40 stellig). Die Taster des Tastenfeldes sind so konstruiert, daß sie mit einlegbaren Papierschildern sehr einfach und dauerhaft selbst beschriftet werden können, es sind jedoch auch fertig beschriftete Tastenkappen lieferbar.

Schaltungsbeschreibung

Bild 1 zeigt das Schaltbild des gesamten LCDKEY-Moduls. Das Anzeigemodul enthält bereits die gesamte Ansteuerungselektronik für das Display. Auf der Platine kann eine Anzeige mit zwei Zeilen zu je 16 Stellen montiert werden. Das Platinenformat wurde so gewählt, daß sich das LCDKEY-Modul sehr gut in einem 19-Zoll Einschubsystem unterbringen läßt. Andere Module (20, 32, 40 Stellen) werden über ein Kabel an die Grundplatine angeschlossen, da die unterschiedlichen Module meist die gleiche Ansteuerungselektronik enthalten. Das Anzeigemodul stellt den kompletten ASCII-Zeichensatz dar. Darüberhinaus verfügt die Anzeige über einen selbstprogrammierbaren Zeichengenerator zur Anzeige selbstentworfener Zeichen (z.B. Umlaute). Das LC-Modul ist über einen PIO-Baustein 8255 an den Datenbus des Basic-EMUF angeschlossen (über PIO-Stecker ST4). Dadurch werden auf dem Basic-EMUF keine I/O-Leitungen belegt. Der Datenbus der Anzeige ist an Port A des 8255 angeschlossen, die restlichen Steuerleitungen (RS, E, R/W) an Port C.

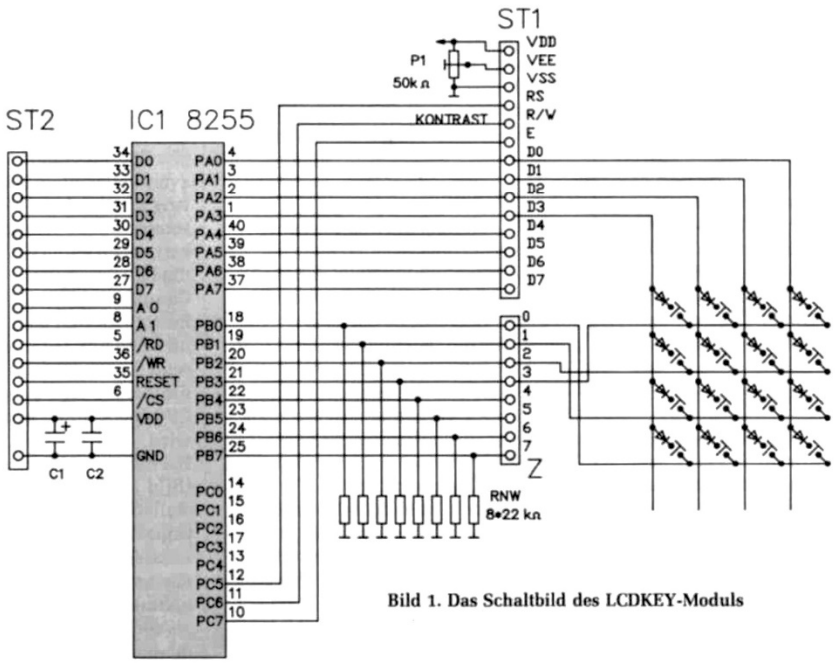

Bild 1. Das Schaltbild des LCDKEY-Moduls

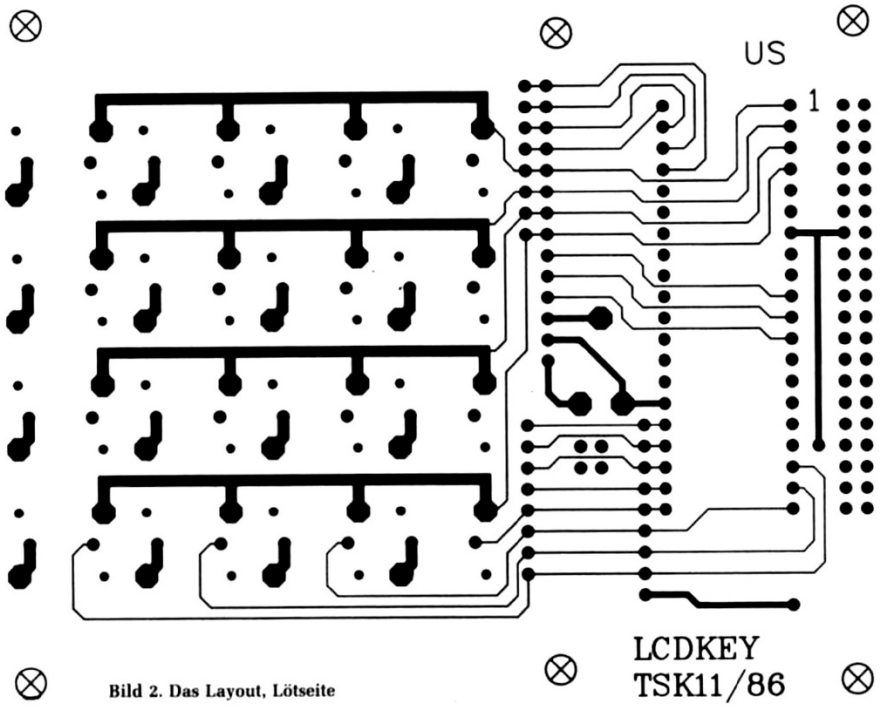

Bild 2. Das Layout, Lötseite

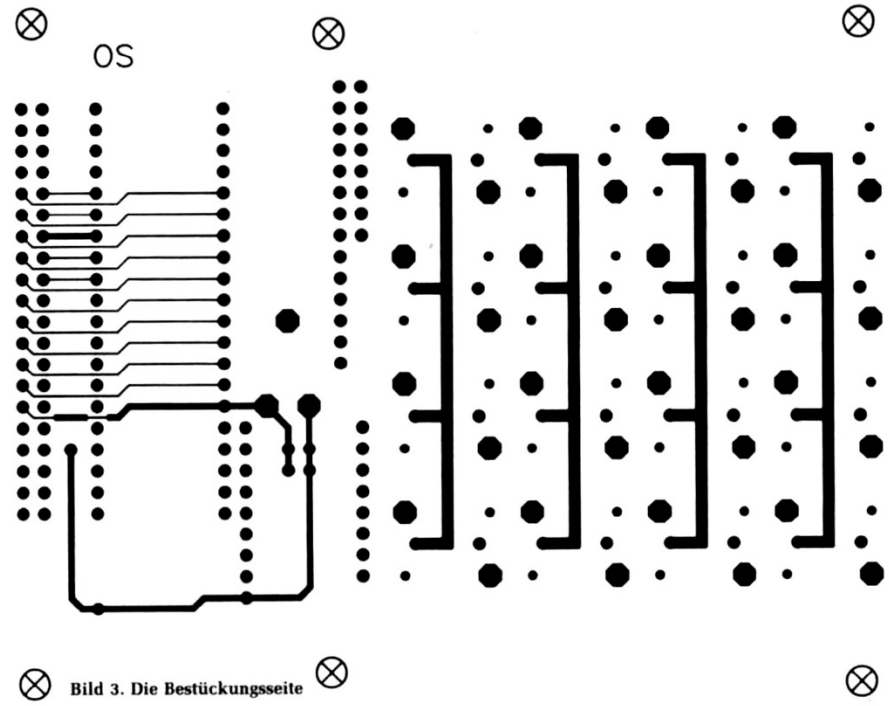

Bild 3. Die Bestückungsseite

Für den PIO-Baustein sollte nach Möglichkeit eine CMOS-Version (82C55, 71055) eingesetzt werden, um den Stromverbrauch so gering wie möglich zu halten. Er beträgt dann nur einige Milliampere (ca. 3 mA).
Die Ansteuerung der Tastenmatrix erfolgt mit Port A (Ausgänge) und Port B (Eingänge). Zur Entkoppelung der einzelnen Tasten untereinander dient jeweils eine Diode an jedem Tastenkontakt. Damit ist es möglich auch Tastenkombinationen von zwei und mehr Tasten auszuwerten (z.B. SHIFT-Taste). Die Abfrage erfolgt in der Weise, das am Port A jeweils eine Leitung auf „HIGH" gesetzt wird und dann am Port B geprüft wird, ob irgendein Eingang ebenfalls auf „HIGH" liegt.

Aufbau

In *Bild 2 und 3* ist das Layout der zweiseitigen Platine angegeben. *Bild 4* zeigt den Bestückungsplan der Baugruppe. Zur Verwendung einer eigenen Tastatur sind entsprechende Kontakte auf der Platine vorgesehen. Das Tastenfeld kann dazu abgetrennt werden. Bei Verwendung des vorgesehenen Tastenfeldes muß darauf geachtet werden, daß zuerst die Dioden unter den Tastern , dann erst die Taster selbst eingebaut werden. Die restlichen Bauteile nach Stückliste werden von der Bestückungsseite der Platine eingebaut. Vor der Montage des Displays ist die Platine sorgfältig auf Löt- und Bestückungsfehler zu überprüfen. Das LC-Modul wird mechanisch mit kleinen Abstandsbolzen befestigt, erst danach erfolgt der elektrische Anschluß.

Inbetriebnahme

Nach genauer Überprüfung auf Fehler wird die LCDKEY Baugruppe an den Basic-EMUF mit einem 40 poligen Flachbandkabel am PIO-Stecker (ST4) angeschlossen. Die Länge dieses Kabels sollte 30 cm nicht übersteigen. Ein EPROM mit dem angebenen Treiberprogramm muß im Assembler-EPROM-Sokkel (IC 8) des Basic-EMUF eingesteckt sein. Nach dem Einschalten mit angeschlossenem LCDKEY sollte sich der Basic-EMUF wie gewohnt am Terminal melden. Der Kontrasteinsteller auf der Rückseite der Platine wird jetzt so eingestellt, daß die obere Reihe der Anzeige dunkle Kästchen anzeigt (fast rechter Anschlag von P1). Nach der Initialisierung mit CALL 1 ist jetzt ein blinkender Cursor in der oberen linken Ecke sichtbar. Mit dem Befehl „UO1" kann nun die Ausgabe auf das Display umgeleitet werden. Mit dem Befehl „UO0" kann jederzeit wieder auf das Terminal zurückgeschaltet werden.

Universal Input/Output Programm

Ein Bausatz mit Anleitung und universellem Input-/Output-Programm ist beim Elektronikladen in Detmold erhältlich. Die mc-Redaktion verschickt das Listing von UIO auf Anfrage.
Das Ansteuerprogramm für die LCDKEY-Baugruppe belegt nur einige hundert Byte des Eproms. Es wurden noch einige andere Unterprogramme im EPROM untergebracht. Diese dienen hauptsächlich der Ansteuerung verschiedener Peripheriebaugruppen des Basic-EMUFs. Weiterhin wurde ein Unterprogramm hinzugefügt, das die Funktion VALUE (Umwandlung eines Strings in eine Fließkommazahl) durchführt.
Für die jeweiligen Funktionen sind in *Bild 5* verschiedene einfache Testprogramme angegeben, die die Verwendung dieser kleinen Hilfsprogramme zeigen.

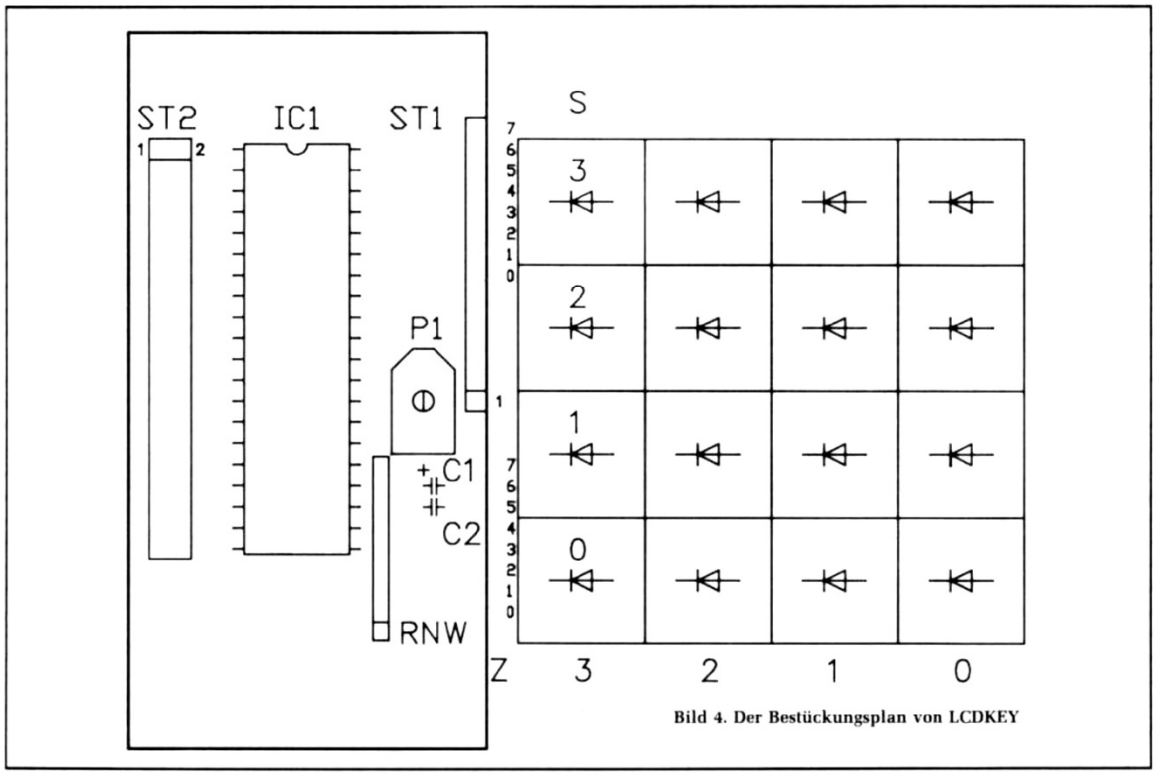

Bild 4. Der Bestückungsplan von LCDKEY

```
Aufruf    NAME       Funtion
CALL 0    DORES      Fuhrt einen absoluten Sprung nach 0h aus.
                     Entspricht in etwa einem Software Reset
CALL 1    LCDINI     Initialisierung LCDKEY
CALL 2    LCDDAT     Ausgabe von [TOS] an das Daten-Register
                     des LCD-Moduls. Dient zur Ansteuerung des
                     programmierbaren Zeichengenerators usw.
CALL 3    LCDCON     Ausgabe von [TOS] an das Control-Register
                     des LC-Moduls. Dient hauptsachlich zur
                     freien CURSOR-Positionierung
CALL 4    ADC        Messen eines AD-Kanals. Kanal in [TOS].
                     Ergebnis wird in [TOS] zuruckgeliefert
CALL 5    READTM     Einlesen der Uhrzeit nach $(0).
                     Strings mussen mit min. 17 Zeichen definiert
                     sein (z.B. STRING 100,20)
CALL 6    RDNBAS     Lesen eines 4-Bit Wertes (Nibble) der Uhr RTC58321.
                     Aufruf mit [TOS]=Adresse des Nibbles.
                     Ruckkehr mit [TOS]=Daten des Nibbles
CALL 7    WRNBAS     Schreiben eines 4-Bit Wertes (Nibble) der RTC58321.
                     Aufruf mit [TOS-1] = zu schreibender Wert
                                [TOS]   = Adresse des Nibbles
CALL 8    VALUE      Wandelt $(0) in eine Fließkommazahl
                     Ruckkehr [TOS] = Status, bei Fehler<>0
                     [TOS-1] = Fließkommazahl wenn Status=0
CALL 9    FORINI     Initialisierung der Parameter fur die Drucker-
                     schnittstelle mit [TOS]. Freigabe des PRINT@,LIST@
                     Treiberprogramms


Programmbeispiele:

10      REM TESTPROGRAMM ADC,LCD
20      PRINT "BEENDEN DURCH BELIEBIGEN TASTENDRUCK"
30      CALL 1 :  UO 1 :  REM INITIALISIERUNG UND UMSCHALTEN AUF LCDKEY
40      PRINT CHR(5) :  REM CURSOR AUS
50      PUSH 0 :  CALL 4 :  REM MESSEN KANAL 0
60      POP W :  REM W = MESSWERT
70      PUSH 80H :  CALL 3 :  REM CURSOR 1.ZEICHEN 1.ZEILE POS.
80      PRINT W :  REM AUSGABE DES MESSWERTES
90      FOR I=1 TO 100 :  NEXT I :  REM WARTESCHLEIFE
100     A=GET :  IF A=0 THEN 50
110     UO 0

10      REM TESTPROGRAMM UHR
20      STRING 106,20 :  REM STRINGRESERVIERUNG 5*20 BYTE
30      PRINT "ZUM STELLEN DER UHR MUSS JUMPER J5 GESCHLOSSEN SEIN"
40      PRINT
50      INPUT "WELCHES NIBBLE SOLL BESCHRIEBEN WERDEN ? ",AD
60      PUSH AD :  CALL 6 :  REM LESEN RTC
70      POP DA :  REM WERT HOLEN
80      PRINT "MOMENTANER WERT =",DA,
90      INPUT "NEUER WERT ? ",DA
100     PUSH DA,AD :  CALL 7 :  REM SCHREIBEN DES NEUEN WERTES
110     CALL 5 :  PRINT $(0) :  REM AUSGABE DER UHRZEIT
120     PRINT "WEITERMACHEN (J/N) ? ",
130     A=GET :  IF A=0 THEN 130 :  REM WARTEN AUF TASTENDRUCK
140     IF A=ASC(J) THEN 40

10      REM TESTPROGRAMM UHR,LCD
20      STRING 106,20 :  REM STRINGRESERVIERUNG 5*20 BYTE
30      PRINT "BEENDEN DURCH BELIEBIGEN TASTENDRUCK"
40      CALL 1 :  UO 1 :  REM INITIALISIERUNG UND UMSCHALTEN AUF LCDKEY
50      PRINT CHR(5) :  REM CURSOR AUS
60      CALL 5 :  REM LESEN DER UHRZEIT NACH $(0)
70      PUSH 80H :  CALL 3 :  REM CURSOR 1.ZEICHEN 1.ZEILE POS.
80      FOR I=1 TO 9 :  PRINT CHR(ASC($(0),I)), :  NEXT I :  REM DATUM
90      PUSH 0C0H :  CALL 3 :  REM CURSOR 1.ZEICHEN 2.ZEILE POS.
100     FOR I=10 TO 18 :  PRINT CHR(ASC($(0),I)), :  NEXT I :  REM ZEIT
110     A=GET :  IF A=0 THEN 50 :  REM WARTEN AUF TASTENDRUCK
120     UO 0

10      REM TESTPROGRAMM TASTENFELD
20      STRING 106,20 :  REM RESERVIERUNG STRINGSPEICHERPLATZ
30      CALL 1 :  REM INITIALISIERUNG LCDKEY
40      PRINT "GEBEN SIE EINEN STRING AM 16ER-TASTENFELD EIN"
50      UI 1 :  REM UMSCHALTEN TASTENFELD
60      INPUT $(0)
70      UI 0 :  REM ZURUECKSCHALTEN AUF TERMINAL
80      PRINT $(0)

10      REM TEST VALUE FUNKTION
20      STRING 100,10
30      INPUT $(0)
40      CALL 8
50      POP ST
60      IF ST<>0 THEN  PRINT "Falsche Eingabe! Bitte wiederholen" :  GOTO 30
70      POP Z
80      PRINT "Die eingegebene Zahl ist ",Z
```

Bild 5. Aufzählung der Funktionsunterprogramme

Ansteuerung des LCDKEY

Zur Initialisierung muß vor Gebrauch der Ausgabe-Routinen das Display initialisiert werden. Dies erfolgt mit dem Befehl „CALL 1". Dieser Befehl braucht nur einmal in einem Programm durchgeführt zu werden. Die normale Ausgabe des „PRINT" Statements, kann nach Ausführung von „UO1" auf das LC-Modul umgeleitet werden. Jederzeit kann jedoch mit „UO0" wieder auf die serielle Terminalschnittstelle zurückgeschaltet werden.

An Steuerzeichen werden folgende ASCII-Werte ausgewertet:

04H blinkender Cursor ein
05H Cursor aus
08H ein Zeichen nach links (max. bis Zeilenanfang)
0AH neue Zeile (Wechsel zwischen 1. und 2. Zeile, kein Scroll)
0DH Cursor an Zeilenanfang

Andere Steuercodes sind zur Zeit nicht implementiert und werden ignoriert (sonstige ASCII-Werte zwischen 0 und 1FH). Zur Ansteuerung der internen Register des LC-Moduls dienen die Unterprogramme „CALL 2" und „CALL 3".

Das Tastenfeld

Das Unterprogramm zur Ansteuerung des Tastenfeldes auf dem LCDKEY-Modul ist mit einer SHIFT- und CONTROL-Funktion ausgestattet. Dabei ist die Tastenposition dieser SHIFT- und CONTROL-Tasten frei programmierbar. Es kann für jede Tastenkombination (Normal, Shift, Control, Shift+Control) ein Eintrag in einer Tabelle abgelegt werden, und damit auch der jeweilige Tastencode. Die derzeitige Belegung kann *Bild 6* entnommen werden.

VALUE

Diese Routine dient der Umwandlung eines Strings in eine Fließkommazahl. Bei INPUT-Befehlen am LCDKEY wird bei Eingabe einer falschen Fließkommazahl die Meldung „TRY AGAIN" ausgegeben und dabei das Display gelöscht. Um dies zu vermeiden, kann jetzt ein String eingegeben werden und danach per Programm gewandelt werden.

Gleichzeitig wird eine Prüfung des Strings vorgenommen, wodurch Eingabefehler in der Software abgefangen werden können.

```
! A   " B   # C
N 7   O 8   P 9    shift

$ D   % E   & F
Q 4   R 5   S 6    control

( G   ) H   = I   - J
T 1   U 2   V 3   W +

? K   * L   / M
X del Y 0   Z .   CR

shift+control   shift
control         normal
```

Bild 6. Die gegenwärtige Belegung der Funktionstasten

PRINT@, LIST@ ... Routinen

Beim Anschluß eines seriellen Druckers und Ansteuerung mit LIST#, und PRINT# ist es bisher erforderlich, die Schnittstelle mit einer niedrigen Baudrate zu betreiben, da im internen Drukkerprogramm keine Überprüfung eines Busy-Signals vorgesehen war. Dies wurde durch Verwendung des PRINT@ Treibers ermöglicht. Außerdem ist es bei diesem Treiberprogramm vorgesehen, die Datenübertragungsparameter einzustellen. Dies geschieht mit Hilfe eines 8-Bit Datenbytes, das eine Codierung nach *Bild 7* besitzt.

Zur Initialisierung der Schnittstelle muß der Wert des Format-Bytes auf den Argument-Stack gelegt werden, danach erfolgt ein Aufruf mit CALL 9. Das kann auch während des Programmlaufs durchgeführt werden. Nach dem CALL 9-Aufruf werden die Argumente der PRINT@, LIST@ usw. Kommandos an die serielle Druckerschnittstelle übergeben (*Bild 8*). Eine Überprüfung der TIMEOUT-Funktion ist aus dem Basic möglich durch Zugriff auf die interne Speicherstelle 1AH. Dieser Zustand kann mit „DBY(1AH).and.80H" geprüft werden. Ist dieser Wert<>0, liegt ein TIMEOUT-Fehler vor.

Unterprogramm-Aufruf

Der Aufruf der Unterprogramme erfolgt jeweils mit einem CALL N-Statement. N ist dabei die jeweilige Funktionsnummer (Bild 5). Eventuell notwendige Daten werden dabei auf den Argument-Stack übergeben. [TOS] ist hierbei die Zahl auf dem Argument-Stack, [TOS-1] ... [TOS-n] die darunterliegenden Zahlen.

```
Bit  7    6    5    4    3    2    1    0
Abk. TO   X    BSY  S    P1   P0   D1   D0

Bedeutung:
TO    TimeOut-Flag     Wird beim Ansprechen der Routine
                       das BUSY-Signal innerhalb von etwa 10 s nicht aktiv,
                       wird dieses Bit gesetzt und die weitere Ausgabe
                       unterdruckt, bis das Bit wieder geloscht ist.
X                      z.Zt. nicht benutzt
BSY                    0=aktiv low Busy / 1=aktiv High Busy
S                      0=1 Stopbit / 1=2 Stopbit
P1,P0                  00=keine 01=ODD 10=EVEN 11=MARK Paritat
D1,D0                  00=5 01=6 10=7 11=8 Datenbits
```

Bild 7. Die Bedeutung der Bits im Schnittstellensteuer-Byte

```
Datenformat  =  4800 Baud,8 Bit,  keine Paritat,  1 Stopbit,
                Busy uber  CTS Leitung (Busy=aktiv high)

          TO   X    BSY  S    P1   P0   D1   D0
FORMAT =  0    0    1    0    0    0    1    1    = 023H

Eingabe im Kommando-Modus:
>PUSH 23H
>CALL 9
>BAUD 4800
>LIST@
READY
>
```

Bild 8. Ein Listing wird mit 4800 Baud abgerufen

```
********************************* ! WICHTIG ********************************
Die Dioden und Taster sind von der Lotseite her zu bestucken.  Dioden zuerst
einloten.
********************************************************************************

Stuckliste
DISP      SHARP 16255, HITACHI LM016L
IC1       8255, 71055 o.a.
D         16 Dioden 1N4148 o.a.
T         4 Siemens-Tastenstreifen (je 4 Taster)
RNW1      Netzwerk 9pol, SIL 8#10 kOhm (22 kOhm)
P1        Poti 20 kOhm bis 50 kOhm
C1        10 mikroF/10 V, Tantal
C2        100 nF Vielschicht
ST2       Pfostensteckerleiste 2reihig 40pol.

Steckerbelegung

ST1 Buchsenleiste fur LC-Modul
1    GND
2    +5 V
3    KONTRAST-SPANNUNG 0-5 V
4    RS
5    R/W
6    E
7    D0
8    D1
9    D2
10   D3
11   D4
12   D5
13   D6
14   D7


ST2 Anschluß an Basic-Emuf PIO-Stecker

1                          2
3                          4
5                          6
7                          8
9    /WR                   10   /RD
11   RESET                 12   /PIOST
13   D 0                   14   GND
15   D 1                   16   A1
17   D 2                   18   A0
19   D 3                   20
21   D 4                   22
23   D 5                   24
25   D 6                   26
27   D 7                   28
29   +5 V                   30
31                         32
33                         34
35                         36
37                         38
39                         40
```

Bild 9. Das ist die Stückliste