

3.3.2 Einsatz einer 3*4 Folientastaturmatrix und einer LCD

Als Tastatur soll eine Folientastatur von Conrad Elektronik mit der Bestellnummer 709930 – 62 zum Einsatz kommen. Im Testprogramm soll die Tastatur zusammen mit einer LCD von Conrad Elektronik mit der Bestellnummer 18 35 12 eingesetzt werden. Die LCD - Anzeigen mit dem HD44780 können zum Datentransfer mit einem Rechner über ein 4 Bit oder 8 Bit bidirektionales Interface arbeiten. Da beim 68hc11 der Port C 8 Bit bidirektional arbeiten kann, soll auch diese schnellere Variante zum Einsatz kommen. Die 3 Steuerungssignale sollen mit Ausgabesignalen am Port A verbunden werden. Es wäre aber auch der Port B denkbar. In der folgenden Tabelle sind alle Verbindungen zur LCD und Tastatur zusammengestellt.

Tast Pin	Bedeutung (Tastaturanschlüsse von links nach rechts)	Hc11 Port + Bitnummer
S3	Spalte 3 (1, 4, 7, *)	D4
S2	Spalte 2 (2, 5, 8, 0)	D3
S1	Spalte 1 (3, 6, 9, #)	D2
Z4	Zeile 4 (*, 0, #) und Widerstand von 56K an 5V	E5
Z3	Zeile 3 (7, 8, 9) und Widerstand von 56K an 5V	E4
Z2	Zeile 2 (4, 5, 6) und Widerstand von 56K an 5V	E3
Z1	Zeile 1 (1, 2, 3) und Widerstand von 56K an 5V	E2
LCD Pin	Bedeutung	Hc11 Port + Bitnummer
01	Vss Masse	
02	Vdd 5V	
03	Vo Kontrast (10 ... 20K Einsteller zwischen GND und 5V. Spannung zwischen Pin 02 und Pin 03 auf 4,5V einstellen!)	
04	RS=0 CMD, RS=1 Daten	A4
05	R/W=0 Write to LCD R/W=1 Read from LCD	A5
06	E Enable ("H" to "L")	A6
07	DB 0 Daten Input/Output	C0
08	DB 1 Daten Input/Output	C1
09	DB 2 Daten Input/Output	C2
10	DB 3 Daten Input/Output	C3
11	DB 4 Daten Input/Output	C4
12	DB 5 Daten Input/Output	C5
13	DB 6 Daten Input/Output	C6
14	DB 7 Daten Input/Output	C7
15	Anode oder Kathode, je nach Jumper auf der Rückseite	
16	Kathode oder Anode, je nach Jumper auf der Rückseite der LED für die Beleuchtung der Anzeige. Ein Widerstand von 5R ist bereits intern vorhanden und kann damit an 5V angeschlossen werden. (Zum Strom sparen noch externen Widerstand vorsehen.)	

Tabelle der Signale des LCD Modul und der Tastatur

Die Ansteuerung der LCD soll wie im Abschnitt 3.3.1 erfolgen. In diesem Abschnitt wird nur das Tastaturprogramm erläutert. Für jede Taste oder erlaubte Tastenkombination soll ein Platz in einer Tabelle angesprochen werden, auf dem dann der Zielcode abgelegt

werden kann. Erlaubt sollen alle 12 Einzeltasten und alle Doppeltastenbetätigungen in den Spalten sein. Damit ergeben sich $12 + 9 = 21$ verschiedene Eingaben, denen je ein anderer Kode zugeordnet werden soll. Alle 21 Eingaben sollen in einen Wert von 0 bis 31 umgerechnet werden, welcher als Offset zur Anfangsadresse einer Tabelle von 32 Byte verwendet wird.

In der Initialisierung werden die 3 Ausgabelleitungen für die Spalten (D2, D3 und D4) auf „0“ gesetzt. Alle 4 Zeilenleitungen werden durch je einen Widerstand von 56K an 5V auf „1“ gelegt. Wird nun eine Taste oder Tastenkombination betätigt, dann werden durch den Kurzschluß der entsprechenden Zeilen und Spalten eine oder zwei Zeilen von „1“ auf „0“ wechseln. In dem daraufhin immer nur eine Spaltenleitung auf „1“ gesetzt wird, kann die betätigte Taste oder Tastenkombination lokalisiert werden. Erhalten die Zeilen Z1 die Wertigkeit $2^2 = 4$, Z2 die Wertigkeit $2^3 = 8$, Z3 die Wertigkeit $2^4 = 16$ und Z4 die Wertigkeit $2^5 = 32$ und man zieht davon die Wertigkeit der Spalten S1 mit $2^0 = 1$, S2 mit $2^1 = 2$ und S3 mit $2^2 = 4$ ab, dann ergibt sich folgende Wertzuordnung zu den Tasten oder Tastenkombinationen bei einer Betätigung. Bei allen Tastenkombinationen die dabei einen Wert größer als 31 ergeben, soll zum Schluß noch 20 subtrahiert werden.

Tasten	Lokalisiert durch	Wertberechnung	Offset für Tabelle
1	Z1 S3	4 - 4	0
2	Z1 S2	4 - 2	2
3	Z1 S1	4 - 1	3
4	Z2 S3	8 - 4	4
5	Z2 S2	8 - 2	6
6	Z2 S1	8 - 1	7
7	Z3 S3	16 - 4	12
8	Z3 S2	16 - 2	14
9	Z3 S1	16 - 1	15
0	Z4 S2	32 - 2	30
*	Z4 S3	32 - 4	28
#	Z4 S1	32 - 1	31
1 + 4	Z2 Z1 S3	8 + 4 - 4	8
2 + 5	Z2 Z1 S2	8 + 4 - 2	10
3 + 6	Z2 Z1 S1	8 + 4 - 1	11
4 + 7	Z3 Z2 S3	16 + 8 - 4	20
5 + 8	Z3 Z2 S2	16 + 8 - 2	22
6 + 9	Z3 Z2 S1	16 + 8 - 1	23
7 + *	Z4 Z3 S3	32 + 16 - 4	44 - 20 = 24
8 + 0	Z4 Z3 S2	32 + 16 - 2	46 - 20 = 26
9 + #	Z4 Z3 S1	32 + 16 - 1	47 - 20 = 27

Die Tabelle im 68hc11 Programm kann dann wie folgt aussehen:

```

*
*__  Tabelle für das Kypad
*
keytab  equ  *
*
*__  Zeilenwert - Spalte/4 = Offset
*__  -> if Offset>31 then Offset-20

```

```

*
    dc.b  "1"          1
    dc.b  $01
    dc.b  "2"          2
    dc.b  "3"          3

    dc.b  "4"          4
    dc.b  $02
    dc.b  "5"          5
    dc.b  "6"          6

    dc.b  "A"          14 -> "A"
    dc.b  $03
    dc.b  "B"          25 -> "B"
    dc.b  "C"          36 -> "C"

    dc.b  "7"          7
    dc.b  $04
    dc.b  "8"          8
    dc.b  "9"          9
*

    dc.b  $05
    dc.b  $06
    dc.b  $07
    dc.b  $08

    dc.b  "D"          47 -> "D"
    dc.b  $09
    dc.b  "E"          58 -> "E"
    dc.b  "F"          69 -> "F"

    dc.b  $0a          7* -> LF
    dc.b  $0b
    dc.b  $1b          80 -> ESC
    dc.b  $0d          9# -> CR = ENTER

    dc.b  "*"          *
    dc.b  $0c
    dc.b  "0"          0
    dc.b  "#"          #
*
*___ Ende der Tabellen
*

```

In der Tabelle sind an den entsprechenden Stellen die ASCII-Kodes der Zeichen auf den Tasten eingetragen, da auch die LCD mit diesen Kodierungen arbeitet. Für die Tastenkombinationen sind die Buchstaben „A“ bis „F“ und die wichtigsten Steuerzeichen eingetragen. Die Tastenkombinationen sind für eine hexadezimale Eingabe ausgewählt worden. Das Tastaturprogramm beginnt mit der Abfrage ob noch eine Taste betätigt ist. Wenn dies nicht der Fall, dann wechselt das Programm in eine Schleife in der auf die Betätigung einer Taste gewartet wird. Folgendes Programm realisiert diese Funktion:

```

*
*___ Tastaturprogramm
*___ Zeichen in A

```

```

*
*__   Warten bis keine Taste betätigt
*
inkey   clr   portd
no_key  ldaa  #$3c
        anda  porte
        eora  #$3c
        bne  no_key

*
*__   Warten auf eine Taste
*
key_no  ldaa  #$3c
        anda  porte
        eora  #$3c           4 Bit negieren
        beq  key_no         keine Taste betätigt
*

```

Sobald eine Taste erkannt wird, muß solange gewartet werden, bis die Kontakte nicht mehr prellen und bei einer Kombination auch die 2. Taste sicher erkannt wird. Dabei wird immer wieder überprüft ob noch eine Taste betätigt ist. Zu kurze Betätigungen werden dadurch als Fehleingabe ignoriert.

```

*
*__   ausreichend für Doppeltasten entprellen
*
        ldaa  #$28           $28 ist getesteter Wert
        staa  <mem
        ldy   #REG
waitta  BRCLR $40,otflg2,Y,waitta   RTI Impuls abwarten
        BSET  $40,otflg2,Y         RTI wieder freigeben
        dec  mem
        bne  waitta

        ldaa  #$3c
        anda  porte
        eora  #$3c           4 Bit negieren
        beq  key_no         keine Taste betätigt
*

```

Nun kann die Auswertung durch Aktivierung der 3 Spalten nacheinander beginnen. Nur wenn die richtige Spalte aktiviert ist werden die Zeilenleitungen wieder „1“ sein. Die beim Eintritt in das folgende Programm bereits ermittelte Zeileninformation muß daher gespeichert werden.

```

*
*__   Auswertung der Taste
*
        xgdx           Zeilen merken
        ldab  #$04     Spalte 1
        stab  portd
        nop           Laufzeit durch
        nop           die Matrix abwarten
        ldaa  #$3c
        anda  porte
        eora  #$3c     4 Bit negieren

```

```

        beq     spalte
*
        LSLB
        stab   portd           Spalte 2
        nop
        nop
        ldaa  #$3c
        anda  porte
        eora  #$3c           4 Bit negieren
        beq   spalte
*
        LSLB
        stab   portd           Spalte 3
        nop
        nop
        ldaa  #$3c
        anda  porte
        eora  #$3c           4 Bit negieren
        beq   spalte
*
*__   nicht gefunden (Störimpuls)
*
        clr   portd
        jmp   inkey
*
*__   Taste gefunden
*
spalte equ   *
*

```

Nun müssen die Zeilen- und Spalten- Werte so aufbereitet werden, daß damit die Adressierung der obigen Tabelle erfolgen kann. Die Zeileninformation muß nicht verändert werden, da durch die Verwendung der Signale E2 bis E5 die richtige Wertigkeit schon vorhanden ist. Da für die Spalten die Signale D2 bis D4 verwendet werden, muß diese Information 2 mal nach rechts verschoben werden! Danach kann die Berechnung des Offset entsprechend obiger Vorschrift erfolgen und der Wert aus der Tabelle ausgelesen werden.

```

*
        xgdx
        ldab  portd           Zeilen wieder in A
                                Spalten nach B laden
        lsrb
        lsrb
                                Spalte/4
        clr   portd
        stab  mem             mem=Spalte/4
        suba  mem
        cmpa  #31
        ble   pos_io
        suba  #20             Tasten der Zeilen 3+4
pos_io  equ   *
*
*__   Wert für die Taste aus der Tabelle lesen
*
        ldx  #keytab
        tab
                                offset nach b

```

```

        abx                x:=x+offset
        ldaa $00,x
*
*   Tastencode ist im Register A
*
        rts
*

```

Damit ist das Unterprogramm zur Tastaturabfrage beendet. Auch für die Tastatur sind beim Systemstart einige Einstellungen notwendig. Diese Einstellungen werden von einem Unterprogramm zur Initialisierung bereitgestellt.

```

*
*   E/A Initialisierungen für die Tastatur
*
fotaini  ldaa  ddrd
          oraa  #$1c          Port D Bits 4, 3 und 2 für
          staa ddrd          Spalten auf Output schalten.
          rts
*

```

Zum Test der Tastatur soll ein Programm die vorgestellten Unterprogramme verwenden und die eingegebenen Zeichen auf der LCD anzeigen und die Steuerzeichen ausführen. Da das LCD Programm das Steuerzeichen ESC nicht auswertet, dafür aber das Steuerzeichen Form Feed (FF), wird im Programm ESC in FF gewandelt. Die Tastaturprogramme sind in dem File „[FOLTAST.ICL](#)“ enthalten. Es wird von dem Programm „[LCD3Fol.a](#)“ als Include gerufen. Dieses Programm ruft auch „[LCD3.ICL](#)“. Im File „[LCD3FOL.LST](#)“ ist das Übersetzungslisting zu finden. Das File „[LCD3FOL.SR](#)“ enthält den Programmcode im S-Record Format.

Ein zweites Testprogramm arbeitet mit dem seriellen Interface zusammen. Zum Test kann auf einem PC das Hyper-Terminal gestartet und direkt mit einer seriellen Schnittstelle verbunden werden. Das Programm verwendet dazu Unterprogramme aus dem Testmonitor „[moni11m](#)“ und ist damit nur zusammen mit dem Monitor lauffähig. Für die oben bereitgestellte RAM Version „[LCD3FOL.SR](#)“ gilt dies übrigens auch, da in der RAM Version mit bedingter Übersetzung Testhilfsprogramme eingefügt sind. Das zweite Testprogramm ist unter „[FOLTAST.A](#)“, das Listing unter „[FOLTAST.LST](#)“ und das Programm im S-Record Format unter „[FOLTAST.SR](#)“ gespeichert. Unter dem Namen „[FOLTAST.TXT](#)“ ist ein Protokoll der Abarbeitung mit dem Hyper-Terminal zu finden.