

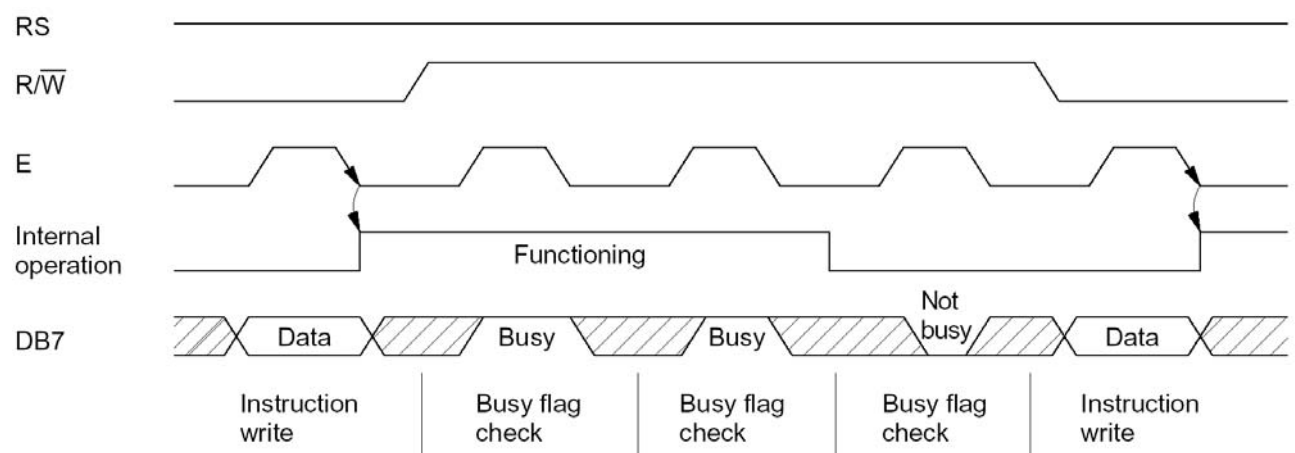
### 3.3.1 Ansteuerung einer LCD mit 2 Zeilen und 8 Zeichen auf der Basis von [HD44780](#)

Für diese Aufgabe wurde eine LCD von Conrad Elektronik mit der Bestellnummer 18 35 12 eingesetzt. 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 das Port C 8 Bit bidirektional arbeiten kann, soll auch diese schnellere Variante zum Einsatz kommen. Die 3 Steuersignale sollen mit Ausgabesignalen am Port A verbunden werden. Es wäre aber auch das Port B denkbar. In der folgenden Tabelle sind alle Verbindungen zur LCD zusammengestellt.

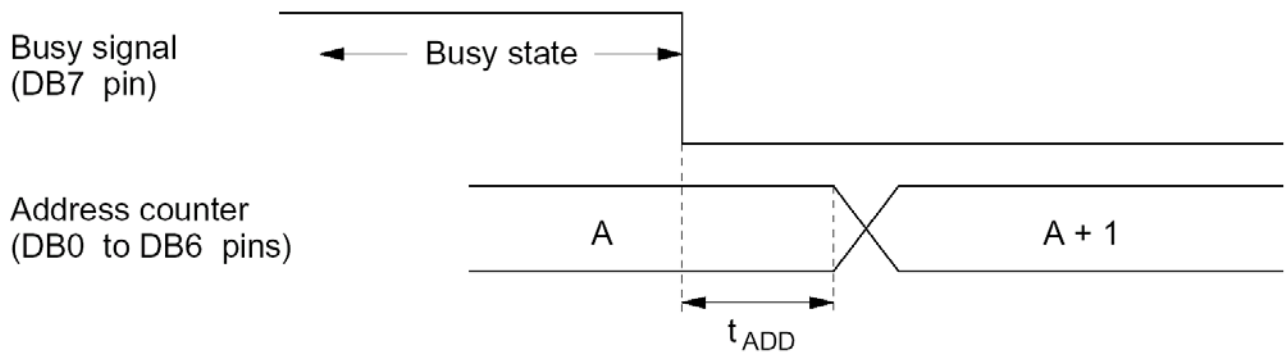
LCD Pin	Bedeutung	Hc11 Port + Bitnummer
01	<b>Vss</b> Masse	
02	<b>Vdd</b> 5V	
03	<b>Vo</b> Kontrast (10 ... 20K Einsteller zwischen GND und 5V. Spannung zwischen Pin 02 und Pin 03 auf 4,5V einstellen!)	
04	<b>RS=0</b> CMD, <b>RS=1</b> Daten	A4
05	<b>R/W=0</b> Write to LCD <b>R/W=1</b> Read from LCD	A5
06	<b>E</b> Enable ("H" to "L")	A6
07	<b>DB 0</b> Daten Input/Output	C0
08	<b>DB 1</b> Daten Input/Output	C1
09	<b>DB 2</b> Daten Input/Output	C2
10	<b>DB 3</b> Daten Input/Output	C3
11	<b>DB 4</b> Daten Input/Output	C4
12	<b>DB 5</b> Daten Input/Output	C5
13	<b>DB 6</b> Daten Input/Output	C6
14	<b>DB 7</b> Daten Input/Output	C7
15	<b>Anode</b> oder Kathode, je nach Jumper auf der Rückseite	
16	<b>Kathode</b> oder Anode, je nach Jumper auf der Rückseite <b>der LED für die Beleuchtung der Anzeige.</b> 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

Nach dem Senden von Daten oder Kommandos an die LCD muß vor dem Senden von weiteren Daten oder Kommandos auf die Rücknahme des Busy Flag der LCD gewartet werden. Das dazu notwendige Signalspiel ist in dem Dokument von Hitachi auf dem Blatt 33 (ist die Seite 199) zu finden.

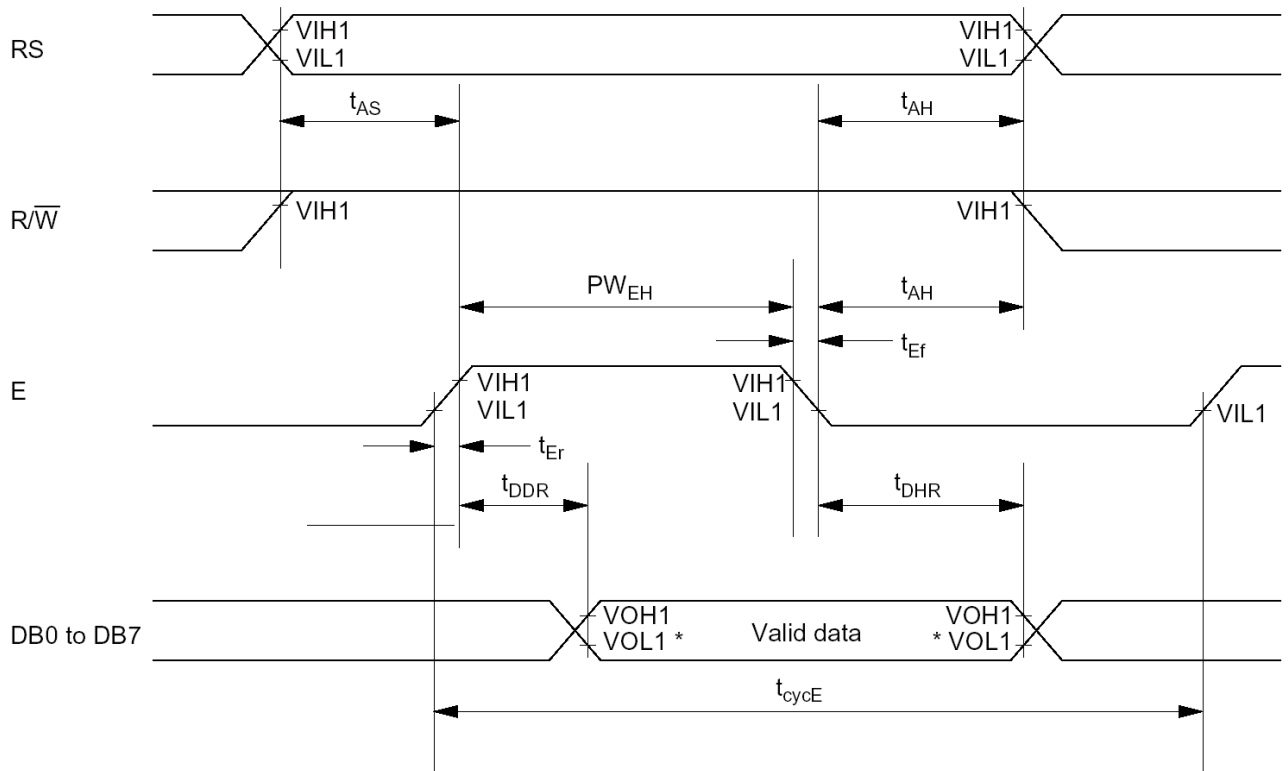


Wird bei der Ausgabe von Daten auf die LCD der Autoinkrement oder Autodekrement Mode für den Schreibpointer verwendet, so ist der zusammen mit dem Busy Flag gesendete Pointer, **erst 4µs nach dem Busy auf 0 gegangen ist, gültig!** Dies ist in dem Dokument von Hitachi auf Blatt 26 (Seite 192) zu finden.



Gültigkeit des Pointers erst  $t_{ADD} = 4\mu\text{s}$  nach der Rückflanke von Busy

Die Signalspiele für das Lesen von Daten und Statusinformationen der LCD sowie für das Schreiben von Daten und Kommandos zur LCD ist in dem Dokument von Hitachi auf Blatt 59 (Seite 225) zu sehen.

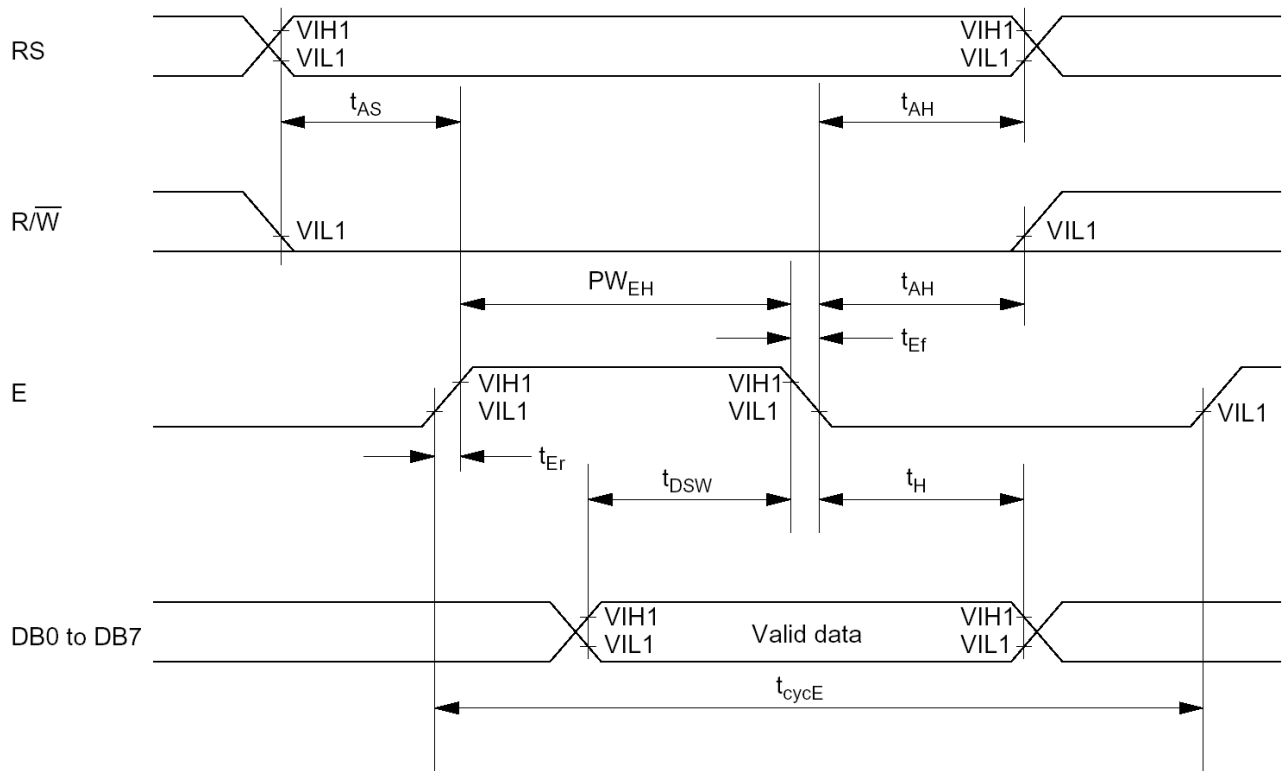


Note: \* VOL1 is assumed to be 0.8 V at 2 MHz operation.

### Signalspiele für das Lesen von Daten und Statusinformationen

Die Einhaltung der Flankenzeiten von  $t_{Er}$  und  $t_{Ef} \leq 25\text{ns}$  ist für die gängigen Mikrokontroller kein Problem. Auch die Einhaltung der Impulsbreite  $PW_{EH} \geq 450\text{ns}$  muß nur bei Prozessoren mit Taktfrequenzen größer 20MHz beachtet werden. Die Zeiten  $T_{AS} \geq 60\text{ns}$  und  $T_{AH} \geq 20\text{ns}$  sind jeder Zeit einzuhalten. Das gilt auch für die Zeit  $t_{cycE} \geq 1000\text{ns}$ .

Für das Schreiben von Daten und Kommandos gilt ein ähnliches Signalspiel, das im folgenden Bild zu sehen ist.

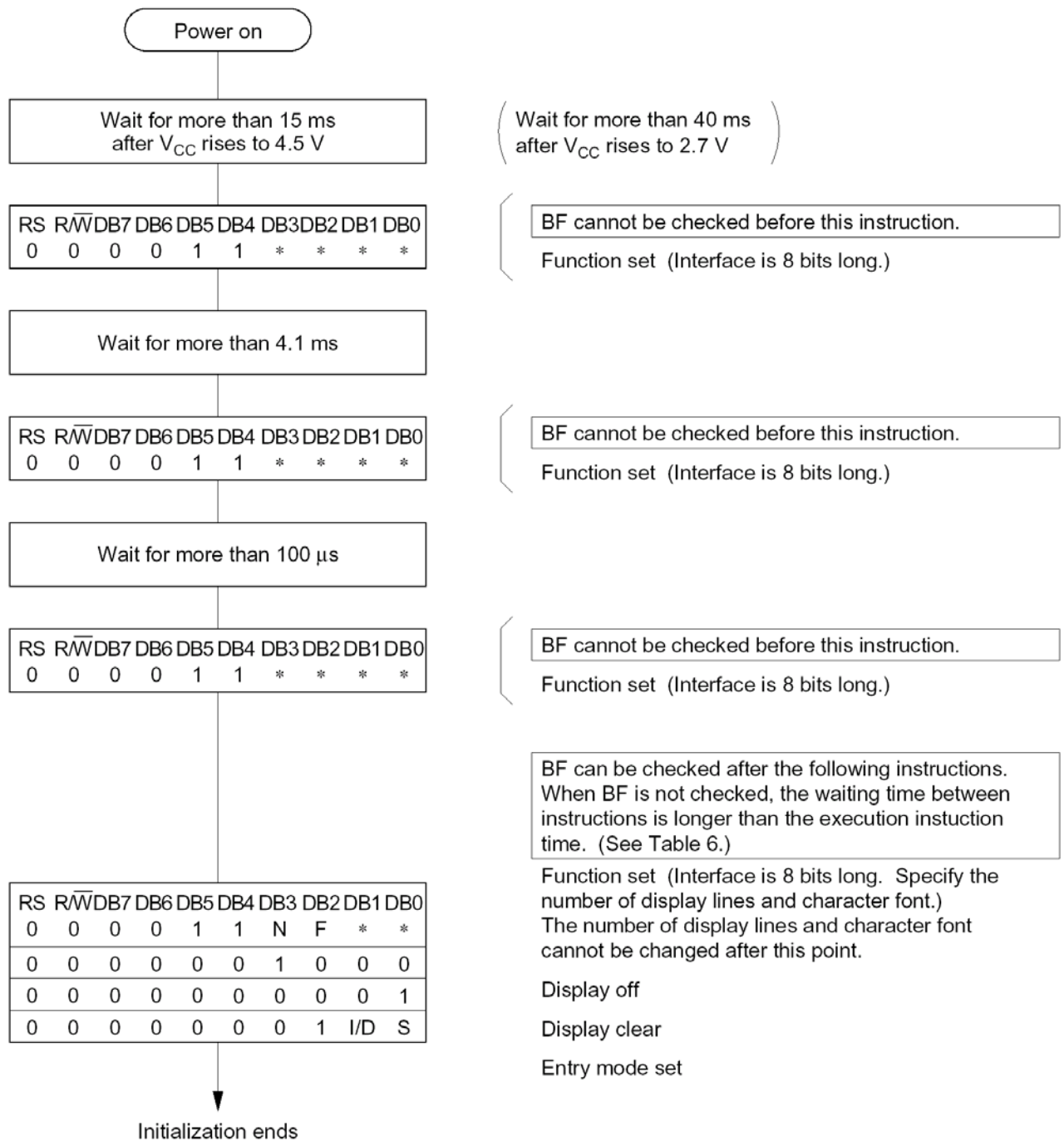


Signalspiele für das Schreiben von Daten und Kommandos

Beim Schreiben gelten die gleichen zeitlichen Bedingungen wie beim Lesen. Zusätzlich müssen die Zeiten  $t_{DSW} \geq 195\text{ns}$  und  $t_H \geq 10\text{ns}$  eingehalten werden. Beide Bedingungen sind wie die vorigen leicht einzuhalten, da es keinen Maximalwert gibt. Die Zeiten sind in dem Dokument von Hitachi auf dem Blatt 50 (Seite 216) zu finden.

Genauere Angaben zum Timing der LCD Module von Conrad Elektronik sind in dem bei Conrad Elektronik zum Download bereitgestellten [Dokument](#) vom 06.02.2003 nicht zu finden! Insbesondere fehlt ein Hinweis auf die Zeit  $t_{ADD} = 4\mu\text{s}$  für die Gültigkeit des Adresspointers nach dem Erkennen von Busy = 0. Da beim 68hc11 und bei den meisten anderen Mikrocontrollern die Befehlsausführungszeiten bei 2 MHz Systemtakt (8 MHz Quarz) schon für die meisten Befehle  $< 4\mu\text{s}$  ist, muß diese Zeit berücksichtigt werden, um die aktuelle Pointerposition auswerten zu können.

Für die Initialisierung der LCD steht in der 1. Phase die Abfrage des Busy Flag noch nicht zur Verfügung. Für die ersten 3 Kommandos müssen daher die vom Hersteller festgelegten Wartezeiten eingehalten werden. Das folgende Bild zeigt den Ablauf der Initialisierung aus dem Dokument von Hitachi auf Blatt 46 (Seite 212).



Ablauf der Initialisierung des LCD Moduls.

Die folgenden 68HC11 Programme sind für die Ansteuerung der LCD am Port A und C des Prozessors ausgelegt. Für die 3 Signale am Port A könnten auch andere Ausgangssignale verwendet werden.

### 1. Warten auf das Busy Flag der LCD

Das Unterprogramm gibt im Akkumulator B den aktuellen Pointer zurück.

```
* __   Warten bis LCD wieder Ready
* __   Zeiten nach "hd44780.pdf" Seite 199 und 225 von Hitachi
*
busy   psha
       clr   ddrc           Port C alles Input
       bclr  $10,oporta,y   RS von LCD auf 0 = CMD Mode
       bset  $20,oporta,y   R/W von LCD auf 1 = Lesen
```

```

busy1    bset    $40,oporta,y    E von LCD 0 --> 1
         ldaa   portc            Bit 7 von A ist Busy Flag
         bclr   $40,oporta,y    E von LCD 1 --> 0
         tsta
         bmi    busy1           N Flag = Busy testen
*
*__      Noch einmal lesen, da ADR+1 erst 4µs nachdem BUSY=0 ist
*__      gültig wird ("hd44780.pdf" Seite 192)
*
         bset   $40,oporta,y    E von LCD 0 --> 1
         ldab   portc            Cu Position nach b
         bclr   $40,oporta,y    E von LCD 1 --> 0

         bclr   $20,oporta,y    R/W von LCD auf 0 = Schreiben
         dec    ddrC            port C alles Output
         pula
         rts                    CU-Position in ACCU B
*

```

## 2. Senden von Daten und Kommandos zur LCD

```

*
*__      Kommando in A an LCD senden
*
lcd3cmd bsr    busy
         bra    lcd3cd
*
*__      Daten in A an LCD senden und wenn CU am sichtbaren Zeilen-
*__      ende ist, an den Anfang der anderen Zeile setzen
*
lcd3dat bsr    busy
*
*__      Steuerzeichen auswerten
*
         cmpa   #$0d            CR zum Zeilenanfang
         beq    zanf
*
         cmpa   #$0a            LF Zeile nach oben rollen wenn CU
*__      in der unterer Zeile
         beq    roll
*
         cmpa   #$0c            FF = CLS
         beq    lcd3cls
*
*__      Zeilenende abtesten
*
         psha
         tba
         anda   #$3f
         cmpa   #$08            nur 8 Zeichen pro Zeile sichtbar
         bne    no_zend
         tba
         anda   #$40
         eora   #$40
         oraa   #$80

```

```

        bsr    lcd3cd          CU an Zeilenanfang auf der anderen
*__      Zeile setzen
        bsr    busy
no_zend pula

        bset   $10,oporta,y    RS von LCD auf 1 für Daten
*
lcd3cd bset   $40,oporta,y    E von LCD 0 --> 1
        staa  portc
        bclr  $40,oporta,y    E von LCD 1 --> 0
        bclr  $10,oporta,y    RS von LCD auf 0 = CMD Mode
        rts
*

```

### 3. Behandlung der Steuerzeichen im UP LCD3DAT

```

*
*__      LF Zeile nach oben rollen, wenn CU in unterer Zeile
*
roll     bitb   #$40
        bne    zeil2          CU ist in Zeile 2
*
*__      CU ist noch in der 1. Zeile
*
        tba
        oraa  #$c0
        bra   lcd3cmd
*
*__      CU ist in Zeile 2
*
zeil2    jmp    z2to1          Zeile 2 nach Zeile 1 umladen
*
*__      CLS auf LCD3
*
lcd3cls ldaa  #$01
        bra   lcd3cmd
*
*__      CR zum Zeilenanfang
*
zanf     tba
        anda  #$c0
        oraa  #$80
        bra   lcd3cd
*

```

Zur Realisierung des Rollens der Zeilen nach oben werden noch 2 weitere Unterprogramme benötigt. Ein UP zum Zurücklesen von Daten auf der LCD und ein 2. zum Senden von Zeichenketten, die zum Löschen der unteren Zeile verwendet werden sollen.

```

*
*__      Zeichenkette an LCD senden
*
*__      Beginn bei Adresse in ix
*__      Ende bei $00 in der Zeichenkette
*

```

```

lcd3s   ldaa   0,x
          inx
          tsta
          beq   s_end           Zeichenkette ist zu Ende!
          bsr   lcd3dat
          bra   lcd3s
s_end    rts
*
*__      Daten von LCD zurücklesen
*
lcd3rd bsr    busy
          clr   ddrc           Port C alles Input
          bset  $10,oporta,y   RS von LCD auf 1 für Daten
          bset  $20,oporta,y   R/W von LCD auf 1 = Lesen
*
          bset  $40,oporta,y   E von LCD 0 --> 1
          ldaa  portc          Daten zurücklesen
          bclr  $40,oporta,y   E von LCD 1 --> 0
*
          bclr  $20,oporta,y   R/W von LCD auf 0 = Schreiben
          bclr  $10,oporta,y   RS von LCD auf 0 = CMD Mode
          dec   ddrc          port C alles Output
          rts
*
*__      Zeile 2 nach Zeile 1 umladen und Zeile 2 löschen
*
z2tol    pshx
          ldaa  #$c0           Adresse für Zeile2
          ldab  #$80           Adresse für Zeile1
          ldx   #8             8 Zeichen umladen
umla     psha
          pshb
          jsr   lcd3cmd        CU in Zeile 2 setzen
          bsr   lcd3rd        Daten lesen nach A
          pulb
          pshb
          psha
          tba
          jsr   lcd3cmd        CU in Zeile 1 setzen
          pula
          jsr   lcd3dat        Daten in Zeile 1 schreiben
          pulb
          pula
          inca
          incb
          dex
          bne   umla
*
*__      Umladen beendet → untere Zeile löschen
*
          ldx   #tx_cl2
          bsr   lcd3s
          pulx
          rts
*
*__      Zeichenkette zum Löschen der Zeile 2 nach dem Umladen

```

```

*
tx_cl2  dcc   "\r\n          \r\00"
*

```

Bei der Initialisierung der LCD ist in den 1. Schritten noch keine Abfrage des Busy Flags möglich. Die Ausführung der Kommandos muß durch vom Hersteller festgelegte Zeiten abgewartet werden. Zur Einhaltung dieser Zeiten soll das RTI-Signal abgefragt werden. Das RTI-Signal wird beim 68hc11 nach dem Einschalten bei 8MHz Quarz aller 4,1ms aktiviert. Da das Programm die Abfrage des RTI-Signals asynchron beginnt, wird zur Sicherheit für eine Wartezeit von 4,1ms das RTI-Flag 2 mal abgefragt. Die folgenden Unterprogramme werden zur Erzeugung von Zeitverzögerungen mit dem RTI-Flag bereitgestellt.

```

*
*__   Zeitverzögerung N * RTI
*__   N wird in A übergeben
*
delay  pshy
      ldy   #REG
wait   BRCLR $40,otflg2,Y,wait      RTI Impuls abwarten
      BSET  $40,otflg2,Y           RTI wieder freigeben
      deca
      bne  wait
      puly
      rts
*
*__   2 * RTI warten
*
wait2  psha
      ldaa  #2
      bsr   delay
      pula
      rts
*

```

Nun fehlt nur noch ein Programm zur Anfangsinitialisierung der LCD und der Ports des 68hc11 an denen die LCD angeschlossen ist. In der Initialisierung werden die Unterprogramme zur Zeitverzögerung und zur Kommandoausgabe verwendet.

```

*
*__   E/A Initialisierungen für LCD3
*
lcd3ini  clr   ddrc           Port C alles Input
        clr   portc        Ausgabebits von Port C auf 0
        clr   porta       Ausgabebits von Port A auf 0
        ldy   #REG        Adresse des E/A-Registerblock
        dec   ddrc        Port C alles Output
*
        ldaa  #5           5 * RTI warten (>15ms)
        jsr   delay
*
        ldaa  #$3f
        jsr   lcd3cd
*
        jsr   wait2        2 * RTI warten (>4,1ms)

```



```

*
    ldaa    #$3f
    jsr     lcd3cd
*
    jsr     wait2
*
    ldaa    #$3f
    jsr     lcd3cd
*
*___  Ab jetzt ist Busyabfrage möglich!
*
    ldaa    #$3f                8 Bit, 2 Zeilen und 5*10 Font
    jsr     lcd3cmd
*
    ldaa    #$3c                Display An
    jsr     lcd3cmd
*
    ldaa    #1                  Display löschen
    jsr     lcd3cmd
*
    ldaa    #6                  ADR+1, No Shift einstellen
    jsr     lcd3cmd
*
    ldaa    #$03                Return Home
    jsr     lcd3cmd
*
    ldaa    #$0d                Display-On, Cu-Off, Blinken-On
    jsr     lcd3cmd
*
    rts
*

```

Die Kommandos zur Initialisierung der LCD sind in dem Dokument von Hitachi auf den Blättern 25 und 26 (Seite 191 und 192) zu finden. Alle vorgestellten Programme sind in dem File „[LCD3.ICL](#)“ zu finden. Dieses Programm wird in dem Testprogramm „[LCD3TE3.A](#)“ als Include gerufen. Eine Übersetzungsliste ist in dem File „[LCD3TE3.LST](#)“ und das ausführbare Programm im S-Record Format im File „[LCD3TE3.SR](#)“ zu finden. „LCD3TE3.SR“ ist eine RAM Version die durch die Einfügung von Testhilfen nur mit dem Testmonitor „[moni11m](#)“ zusammen arbeitet. Die Testhilfen werden durch bedingte Übersetzungen nicht eingebunden, wenn das Programm nicht auf Adresse 2000 Hex übersetzt wird.