



Vejledning til LCD Karakter Modul, 2 x 16 karakter.

På min hjemmeside findes en kodegenerator. Den er gafflet fra en tysk side. Generatoren kan generere koder til 8051-kompatible, og heriblandt koder til at styre LCD-modulet.

Generatoren har været inspiration til materialet, og subrutinerne er taget herfra ! Endvidere har forskellige datablade været grundlag, og kontakt til POWER TIP.

Displaymodulerne laves af mange fabrikanter, og er ret kompatible med hinanden. Så det er næsten underordnet, om fabrikanten er fx SHARP eller POWER TIP. Der kan dog fx være forskel på, om der skal en negativ kontrast-spænding til for at få displayet til at vise karakterer.

Initiering af LCD displayet !!

Ved Power-on skal LCD-displayet initieres, og evt. indstilles til 4-bit kommunikation.

Det kan gøres med flg. program !

LCD_init:

```
mov a, #50 ;
```

LCD_initpause:

```
cal LCD_ws ; kald pauserutine, ca. 1,6 ms. Se hjemmeside !  
djnz acc, LCD_initpause ; 50 gange
```

```
clr LCD_RS ; Clr styresignaler  
clr LCD_RW  
clr LCD_ENABLE
```

```
mov a, #00101000b ; Load koden for 4-bit kommunikation  
call LCD_send_b ; Send ordre, ( Befehl )  
setb LCD_ENABLE  
clr LCD_ENABLE  
mov a, #00101000b ; Do, skal gores 2 gange !!  
call LCD_send_b ; Send ordre, ( Befehl )
```

```
mov a, #1100b ; Load 0000 1DCB, kode til indstilling af display  
; D=1: Display on  
; C=1: Cursor on  
; B= 0: Cursor blinker ikke.
```

```
call LCD_send_b  
call LCD_clear ; Alle karakterer = 20h, cursor pos. 1,1
```



LCD MODUL

Oversigt over Styrekoder, der kan sendes til LCD-MODULET.

Ved hjælp af forskellige ordrer kan man styre displayet. Her følger en oversigt:

Instruktion	RS, RW	Data og kommentarer
Clear display (Skriv 20h i Dataram) og sætter Dataram adresse pointer DDRAM = 00h, = Cursor home = 1. linie 1. karakter	00	0000 0001 Clear all display and return cursor to home position, adress 00h Eks: mov a, #01h call LCD_send_b
Cursor home, Ram adresse pointer = 0, Display sat til shifted = Scroll hvis man kommer uden for skærmen, Adresse > 15h DataRAM forbliver uændret.	00	0000 001* Returns cursor to home position. Also returns the display skiftet to original position DD RAM indhold unaffected.
Entry mode set Sætter cursorens bevægelses- retning. SH afgør, om displayet skal skoppes, dvs. cursoren står fast, og teksten skoppes ved hver bogstav der skrives til skærmen !!	00	0000 01,I/D,Sh I/D = 0, decrement cursor, mod venstre efter hver skrivning til RAM I/D = 1, increment cursor, mod højre. efter hver skrivning til Ram Sh = 0, specificerer ingen display shift. Sh = 1 display shift. Display scroller I retningen specificeret I bit 0 ? når cursor er nået til kanten af vinduet. eks: 0111 sætter display til at skifte efter hver data write. Dvs. at cursoren bliver stående, men at teksten rykker 1 mod venstre efter hver data write.
Display controlbit on / off	00	0000 1,D,C,B, D: All Display on/off, 0 = off, 1 = On C: Cursor, 0 = off, 1 = On B: Blinkende solid cursor, 0 = off, 1 = on. eks: Indstil LCD til display On, cursor On, og ikke blinkende cursor



LCD MODUL

		<pre>mov a, #00001110b call LCD_send_b</pre>
<p>Cursor eller Display shift uden ændring af DDRAM-indhold</p> <p>Sætter cursor moving og display shift kontrol bit, og retningen, display data RAM</p>	00	<p>0001,S/C,R/L,**,</p> <p>S/C: cursor moving eller scroll display. 0 = move Cursor, 1 = Scroll display.</p> <p>R/L:Direction, Right / Left. 0 = Shift Left, 1 = Right * = don't care</p>
<p>Function set.</p> <p>Sætter interface data længde</p>	00	<p>001,DL,N,F,**,</p> <p>DL: 4 bit / 8 bit, 0 = 4 bit, 1 = 8 bit. N: 1 linie, / 2 linier, 0 = 1 linies display F: display font, 5*8 dot / 5*11 dots, 0 = 5*8 dot</p> <p>eks: indstil LCD til 4-bit, 2 linies display, og 5x8 dots.</p> <pre>mov a, #00101000b call LCD_send_b</pre>
<p>CGRAM = Character Generator Ram</p> <p>Bringer LCD-en til at "pege på" adresse, hvor der efterfølgende kan gemmes data til at definere "selvdefinerbare" karakterer. Sætter CGRAM adresse i tæller dataram adr, selvdefinerede tegn.</p>	00	<p>01,AC5,AC4,AC3,AC2,AC1,AC0</p> <p>CGRAM data sendes efter denne ordre. Adressepointer incrementes efter hver write så successive bytes (rows) kan sendes</p> <p>Se eksempel efter skemaet !</p>
<p>Positioner cursor</p> <p>Sætter DDRAM Adresse.</p>	00	<p>1,AC6,AC5,AC4,AC3,AC2,AC1,AC0</p> <p>Det første 1-tal = 80h indstiller LCD-en til at "Pege på" en karakter i displayet som angivet ved de efterfølgende bit.</p> <p>Karaktererne i displayet har adresse 00h til 0Fh i øverste linie. 2. linie starter i adr. 40h til 4Fh.</p> <p>DDRAM data sendes / modtages efter denne ordre. Adressepointer incrementes automatisk</p> <p>Eks: Positioner cursor i 1. kar, 2. linie Mov a, #0C0h ; 80h + 40h = C0h = 1. kar I 2. linie call LCD_send_b</p>



LCD MODUL

Læs Busy flag og adresse	01	BF,AC6,AC5,AC4,AC3,AC2,AC1,AC0 Bit BF angiver om LCD er busy. Adresstælleren kan ses i bit AC6 til AC0 BF = 1 => Busy
Skriv i display data til internal LCD RAM, til DDRAM eller CGRAM	10	D7 til D0 Den værdi, der sendes til displayets karakterRAM, "peger på" en adresse i displayets ROM. Heri findes en indprogrammeret kode, der tænder respektive mønster af dots i displayets dotmatrix, så fx et "a" vises. Der anvendes en standard ASCII kode, suppleret med specialtegn, og evt. selvdefinerede tegn.
Læs Data fra RAM, CGRAM eller DDRAM	11	D7 til D0

Det display, vi arbejder med, (POWER TIP) har 4*20 karakterer i displayet. Fra fabrikanternes side er det defineret at 1. linie har adressen 00h og opad. Det vil her sige 00h til 0fh. 2. linie starter i adresse 40h.

Når cursoren er positioneret, og der evt. er indstillet cursor type osv. kan der skrives i displayet.

eks: Send værdien 53h som data. Call LCD_send_d.

```
Mov a, #53h  
call LCD_send_d
```

Sendes værdien 53 vil der i displayet blive skrevet et stort S. Dot-mønsteret, der giver et "S", er defineret i LCD'en på adresse 53h. Der er for hver type LCD-karakter-display fra fabrikantens side defineret hvad hver værdi, der sendes til LCD'en vil give af mønster i Dot-matrixen i displayet. For de normale karakterer følges ASCII systemet, der stammer fra teletype – og matrix-printernes tid. ASCII står for American Standard Code for Information Interchange. Fabrikanten kan for andre LCD-typer have defineret fx japanske tegn på andre pladser.

Definering af selvdefinerbare karakterer i LCD-displayet.

Vores LCD-display har ud over de fast indlagte tegn 8 selvdefinerbare karakterer. Dvs. at man selv i ASCII-tabellen kan "indlæse" et mønster, der skal frem i en karakter i displayet når pågældende karakters ASCII-værdi sendes til displayet.

Flg. program kan bruges til at definere selvdefinerbare karakterer.

```
mov a, #64 ; Send 64 Hex, = 01xx xxxx b, hvor xx xxxx angiver  
; rækkenummeret, man vil gemme. Der er plads til 8  
; karakterer a` 8 rækker. Her indstilles til række 0, som
```



LCD MODUL

```

; svarer til karakter 0, øverste række.
; 0100 0001 ville "pege på" 1, karakter 2. række fra
; oven.

call LCD_send_b

```

```

mov DPTR, #LCD_CGRAM ; adressen på de data, der ønskes lagt ned som
; brugerdefinerbare karakterer kan være defineret
; i en tabel.
mov R0, #64 ; antal rækker, 8x8 !

```

L1:

```

clr a
movc a, @A+DTPR ; hent tabelværdi
inc DTPR
call LCD_send_d
djnz R0, L1 ; efter sending til en adresse incrementes en tæller
; automatisk, så næste værdi gemmes i næste
; ram-lokation.

```

```

call LCD_returnhome ; positioner cursor i 1. linie 1. plads.
ret

```

LCD_CGRAM:

```

db 0,1,5,6,2,5,6,0 ; Her defineres de 8 rækker i første karakter, = kar 0 !
db 0,1,5,6,4,5,6,0 ; Og de 8 rækker i karakter 1 !
db 0,1,5,6,8,5,6,0 ; Værdierne angiver værdien af 5 bit. Se nedenfor!
db 0,1,5,6,2,5,6,0
db 0,1,5,6,16,5,6,0
db 0,1,5,6,2,24,6,0
db 0,1,5,6,2,5,6,0
db 0,1,5,6,2,5,6,0

```

Karaktererne består af en matrix med 8 rækker a` 5 punkter. Hver af disse er repræsenteret af en bit. Derfor kan værdierne ovenfor være fra 0 til 31 !

Følgende bit-kombination sendt til LCD`ens ram ville kunne skrive et 8-tal. Et 1-tal vil tænde pågældende dot.

11111	= 31
10001	= 17
10001	= 17
11111	= 31
10001	= 17
10001	= 17
11111	= 31
00000	= 0



LCD MODUL

Nederste række bruges normalt ikke, hvis der bruges cursor, og skal så være 00000 !

De definerbare karakterer ligger i LCD-en i RAM på adresse 00h til 07h

Eks. på selvdefineret tegn:

1	0	0	0	1	= 11h
0	1	0	1	0	= 0Ah
1	1	1	1	1	= 1Fh
0	0	1	0	0	= 04h
0	1	1	1	0	= 0Eh
0	0	1	0	0	= 04h
0	0	1	0	0	= 04h
0	0	0	0	0	= 00h

Række 8 (7) bør være = 00h pga. cursor.

Ønsker man at adressere en bestemt karakter, og ikke alle 8 fra begyndelsen, kan flg. måske være til hjælp.

Med de første 3 nuller vælges karakter 0 til 7. Med de sidste 3 vælges række 0 til 7 i karaktermønsteret.

**	000	000
	karakter 0-7	række 0-7

De respektive definerede karakterer vises på skærmen ved at vælge placering, og sende ASCII 00h til 08h

På følgende side er vist nogle flowcharts, der viser forløbet af kommunikationen med et LCD-modul, med koden fra "kodegeneratoren":



LCD MODUL

Flowchart over LCD-displayet

