

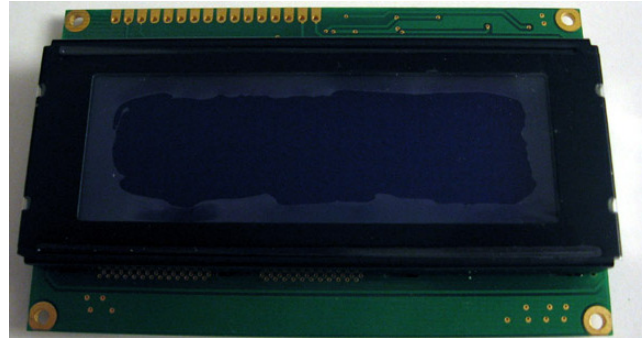


LCD Character display Intro

Der findes flere typer af LCD karakter-displays, fra forskellige firmaer.

Her er vist en type, der er blå.

Pins: Nummer 1 fra venstre



Her er vist en nærmere beskrivelse af de forskellige ben.

På det printudlæg, der kan downloades til uC og LCD-display, er der udlagt, således, at ledningerne blot skal forbindes "lige over".

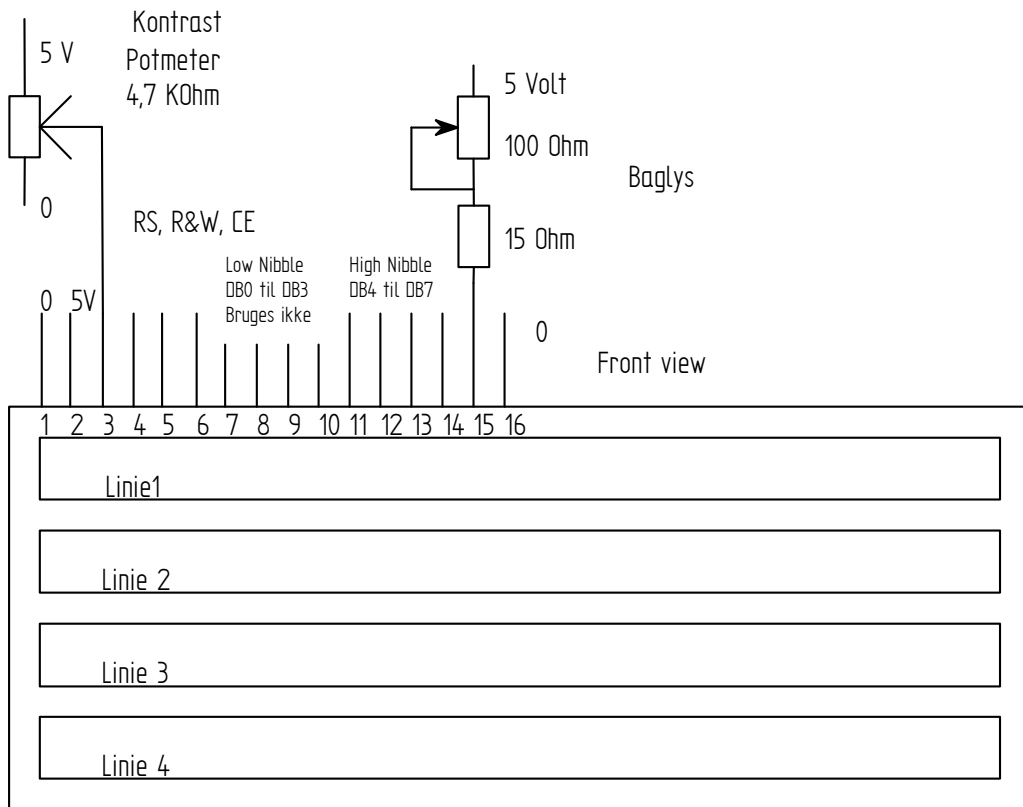
Pin 3 er til at justere kontrast i skærbilledet. Skal justeres med et potentiometer på fx 4,7 KOhm, mellem 0 og 5 Volt.

I 4 bit mode, bruges low nibble ikke, pin 7 – 10.

Mellem 15 og 16 er der forbundet lysdioder, der oplyser LCD'en bagfra. (Baglys) Tilsluttes 5 Volt via modstand.

Pin No.	Symbol	Level	Description
1	V _{SS}	0V	GND
2	V _{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Contrast Adjustment
4	RS	H/L	Register select signal
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A/Vee	–	+4.2V for LED (RA=0ohm)/Negative Voltage output
16	K	–	Power supply for B/L(0V)

På næste skitse ses et forbindelsesdiagram:

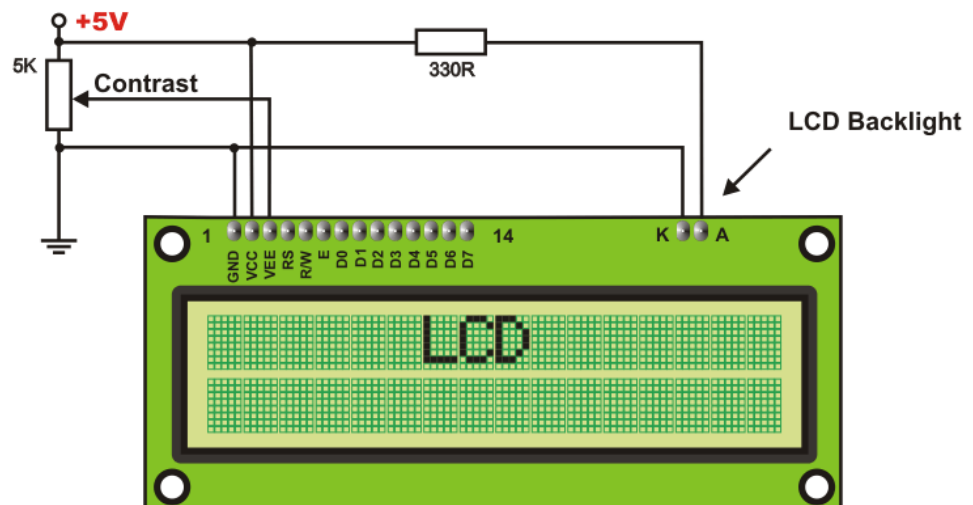


Winstar WH2004A



Her er vist et andet eksempel:

K og A er vist byttet om på denne skitse:

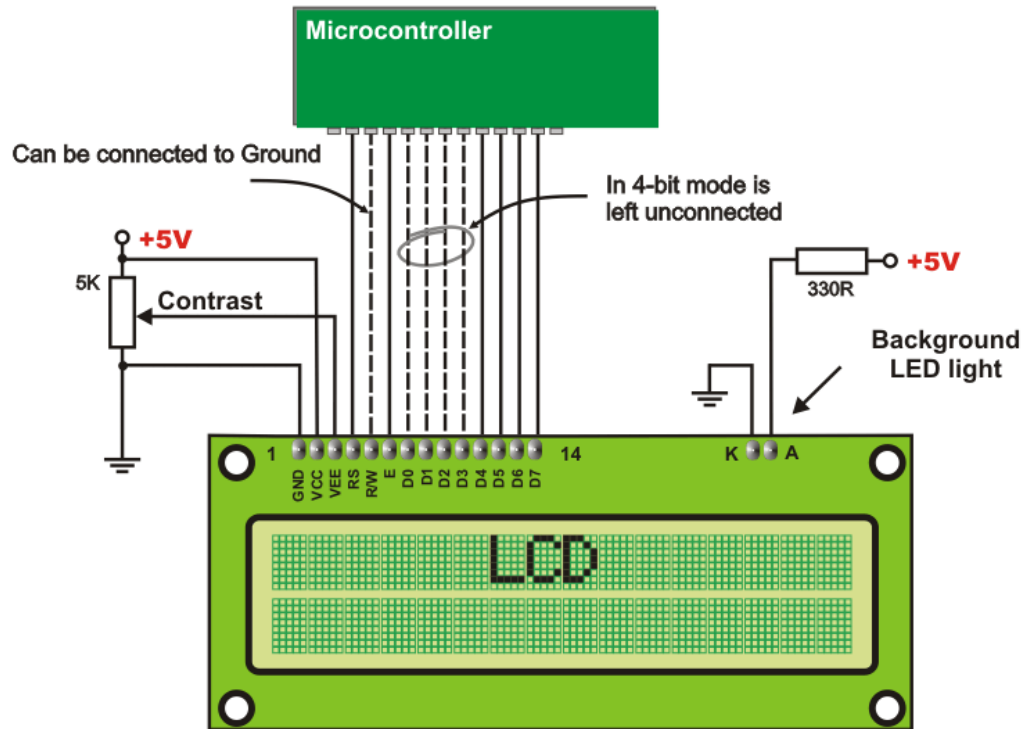




Et andet skitse.

Her forbundet med en uC.

K og A på baglys kan muligvis være ombyttet !



RAM / ROM-struktur i LCD-Panelet.

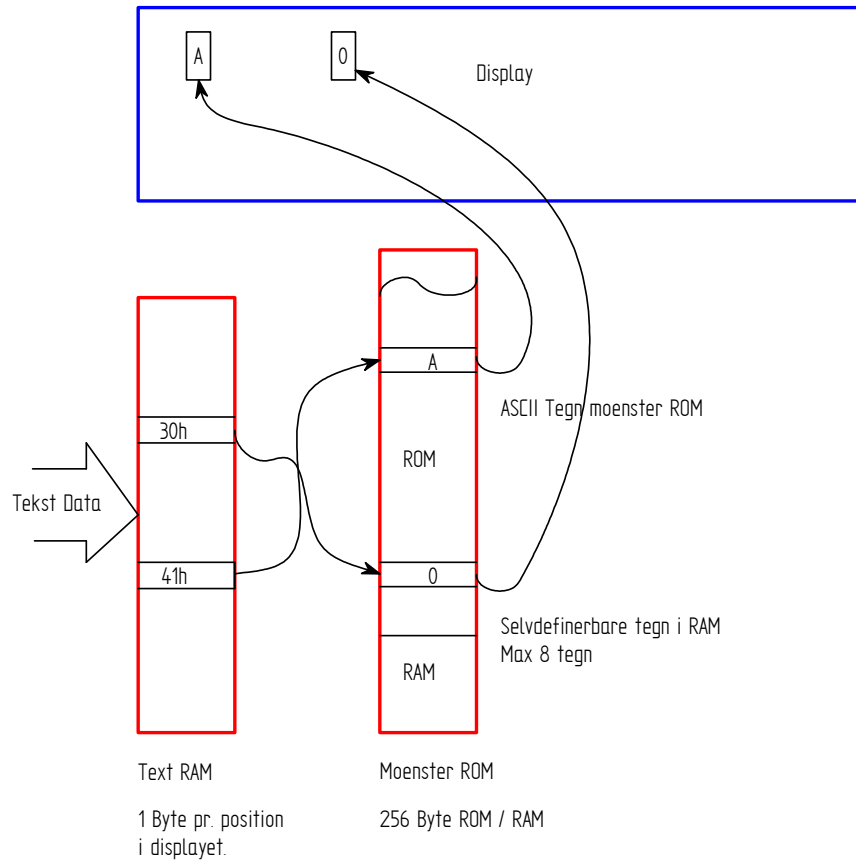
Når der skrives data til et LCD-display, gemmes de i en RAM i LCD'en.

Hver plads i displayet har sin egen RAM.

Værdien der gemmes i de forskellige RAM-adresser, refererer til et tegn i ASCII-tabellen, som så vises i displayet.

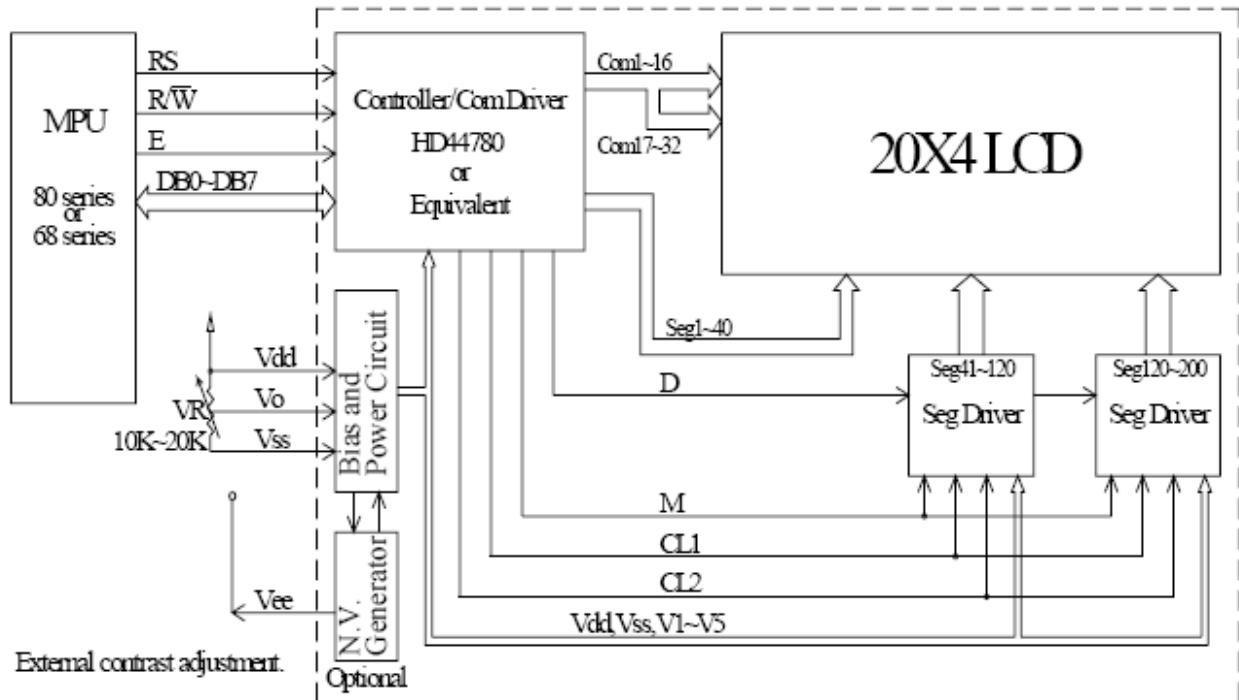
Det er muligt, selv at definere nogle tegn.

I min kildetekst til LCD sendes automatisk de danske æ, ø og å + Æ, Ø, Å til LCD'en.





Her ses et blokdiagram over selve LCD-displayet.



Beskrivelse:

Det LCD-display-modul, jeg har valgt, har 4 rækker a' 20 karakterer.

I mine programmer, har jeg valgt at der bruges 4 bit drift, for at spare på ben på uC'en.

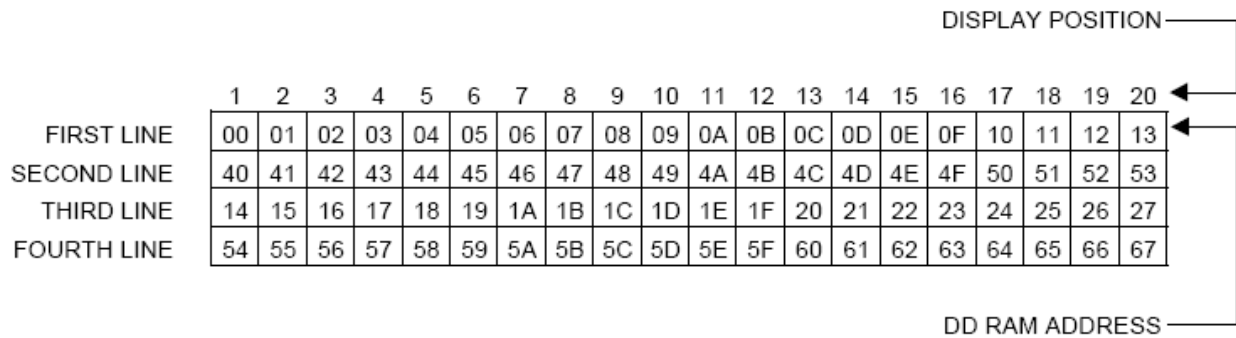
Der er indbygget baggrundslys: Dvs. det kan aflæses i mørke.

I 4 bit mode skal der sendes 2 gange 4 bit for at udgøre en hel byte.

Displayet har indbygget en ROM, hvori der er fast ind-programmeret et mønster, så fx 30h sendt til den viser et nul-tal. 31h viser et 1-tal. Som angivet i ASCII-tabellen.

Hver karakter plads i displayet har en RAM-adresse. Den HEX-værdi, der lægges ind i RAM-adressen, peger på en plads i den indbyggede ROM, hvor der er et fast mønster til bogstaver og tegn, jfr. ASCII tabellen.

Næste skema viser, hvilke ram-adresser, der refererer til hver karakter på de 4 linjer har.



LCD'en har en Cursor, der viser, hvilken Karakterplads med tilhørende RAM, der er selected. Når data sendes til Displayet, placeres de automatisk i RAM'en for cursorens position.

Når der er sendt en byte til displayet, flytter cursoren automatisk 1 fremad.

Ud over data til at skrive tekst på displayet, kan der sendes kontrolkoder til at styre displayets opførsel. Det styres af bittet Registerselect, RS.

Herudover kan der indlæses 8 selvdefinerede tegn i displayet. Dvs. der kan uploades et mønster, så de danske æ, ø, å, Æ, Ø og Å kan vises. De kommer til at ligge i de laveste 8 byte i karaktermønster-ROM'en. Dvs. de er lavet som RAM.

Eksempler på kontrolkoder:

- Clear Display
- Cursor home, Ram adresse pointer = 0
- Flyt Cursor
- Entry mode set (Cursor shift højre eller venstre etc.)
- Tænd / sluk for display og cursor
- Osv.

Næste billede viser ASCII Koderne: ASCII står for American Standard Code for Information Interchange. Det var de koder, der "i gamle dage" blev sendt til fx nåle-printere.



ASCII - The American Standard Code for Information Interchange

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@	96	60	140	##96;	`
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A	97	61	141	##97;	a
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B	98	62	142	##98;	b
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C	99	63	143	##99;	c
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D	100	64	144	##100;	d
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EM (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Extended ASCII Code. Dvs. de øverste 128 bytes.

128	Ç	144	É	161	í	177	⌘	193	⊥	209	⌘	225	β	241	±
129	ù	145	æ	162	ó	178	⌘	194	⌘	210	⌘	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⌘	211	⌘	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	-	212	⌘	228	Σ	244	∫
132	ã	148	ö	165	Ñ	181	‡	197	†	213	⌘	229	σ	245	∫
133	ä	149	ò	166	ª	182	‡	198	‡	214	⌘	230	μ	246	+
134	å	150	û	167	º	183	⌘	199	‡	215	⌘	231	τ	247	≈
135	ç	151	ù	168	¸	184	⌘	200	⌘	216	⌘	232	Φ	248	°
136	ê	152	-	169	-	185	‡	201	⌘	217	⌘	233	Θ	249	.
137	ë	153	Ö	170	¬	186		202	⌘	218	⌘	234	Ω	250	.
138	è	154	Û	171	½	187	⌘	203	⌘	219	■	235	δ	251	√
139	ï	156	£	172	¾	188	⌘	204	‡	220	■	236	∞	252	-
140	î	157	¥	173	¡	189	⌘	205	=	221	■	237	φ	253	²
141	ì	158	-	174	«	190	⌘	206	‡	222	■	238	e	254	■
142	Ä	159	f	175	»	191	⌘	207	⌘	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	L	208	⌘	224	α	240	≡		

Selvdefinerbare tegn:

Bogstavs-mønstrene, displayet viser, ligger i ROM.



Men der er plads til 8 stk. selvdefinerbare tegn. De skal sendes til LCD'en til ROM-adresse 0 til 7.

Dette kan dog kun lade sig gøre fordi, disse er lavet med RAM i stedet for fast programmeret ROM.

Et tegn på displayet er opbygget som en 5 x 8 matrix.

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

Nederste række bør være = 00h pga. at cursoren normalt er her.

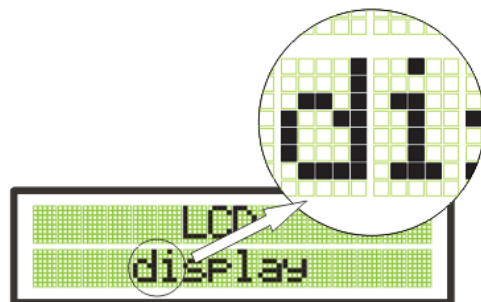
Jeg har lavet et mønster til de danske karakterer, og i opstarts-proceduren LCD_init lægges de ind i LCD-en.

Vha. en kodegenerator, se min hjemmeside!, kan man selv definere sine tegn!

Koder til at downloade danske karakterer til LCD-en

Et eksempel på, hvordan mønsteret ser ud:

```
LCD_cgram:
db 0,0,0,0,0,0,0,0      ;      ; 0
db 0,0,26,5,15,20,31,0  ;æ    1
db 0,1,14,21,21,14,16,0 ;ø    2
db 4,0,14,1,15,17,15,0  ;å    3
db 15,20,20,30,20,20,23,0 ;Æ    4
db 1,14,19,21,25,14,16,0 ;Ø    5
db 4,0,14,17,31,17,17,0 ;Å    6
db 31,21,21,14,14,4,4,0 ;VT   7
```



Opstarts – Procedure for LCD'displayet:

Efter Power ON skal man vente til LCD er vågnet op, dvs. spændinger er nået op på 5 Volt, osv.

Der skal sendes koder til displayet, for at initiere LCD'en og indstil den til 2 x 4 bit osv.



Herefter skal der uploades de danske karakterer.

Til sidst cleares displayet, og cursoren placeres i 1. linje 1. plads.

Subrutiner til at skrive tekst til LCD.:

På min hjemmeside under download / AT89C4051 /LCD kan findes en kildetekst, en ".A51" fil med en række subrutiner, der kan kopieres direkte ind i kildeteksten i Keil.

Hovedprogrammet, man selv skal skrive, skal så "kun" kalde de ønskede subrutiner.

Der er subrutiner til at positionere cursoren, til at skrive tekst-strings, slukke og tænde cursoren, osv.

Eksempel på at positionere cursoren:

Mov A, #0h ; I A indlæses værdien, hvor langt henne på linjen, cursoren skal
; placeres.

Call PCL1 ; Subrutinen Positioner Cursor Linie 1 kaldes. 1'-tallet kan også
; være 2, 3 eller 4 for de øvrige linjer.

Eksempel på at skrive tekst-streng.

Mov DPTR, #Txt1 ; Datapointeren peger på en tekst-string.

Call LCD_Printstring ; Skriv tekst.

Tekst defineres som en Tabel, og kan skrives direkte som tekst. Keil oversætter så teksten til de rigtige ASCII-koder.

Txt1: dB 'HTX Elektronik',0

Txt2: dB 'S', 02h, 'nderborg',0

Tekststrings skal afsluttes med et '0'. Dette får Subrutinen LCD_Printstring til at stoppe og returnere.

Når de specielle danske karakterer skal bruges, skal sendes adressen på den RAM, hvori mønsteret for ønskede bogstav er defineret. I initierings-rutinen uploades de danske karakterer til RAM-adresserne 0 til 7, som vist i følgende skema:

Adresse	Mønster
0	
1	æ
2	ø
3	å



Se evt. efter kodegenerator. Min hjemmeside, /
Atmel AT89C4051 / kodegenerator

4	Æ
5	Ø
6	Å
7	VT

Skrivning af tal fra RAM i uC'en:

Hvis der ønskes at skrive en talværdi, fra 0 til 9, der måtte befinde sig i en RAM-adresse i uC'en, skal den først laves om til ASCII kode.

Mov A, Sec100 ; Ramadressen med navnet Sec100 kopieres til A.

ORL A, #30h ; I Ascii-tabellen findes tallene fra 30h til 39h. Ved at or'e
; A med 30h, fås ASCII-koden for tallet.

Call LCD_Printc ; Subrutinen LCD_Printc sender karakteren i A.

: Pins-definitioner

I kildeteksten skal placeres følgende ben-navne:

LCD_RS EQU P1.7
LCD_RW EQU P1.6
LCD_ENABLE EQU P1.5
LCD_D4 EQU P1.4
LCD_D5 EQU P1.3
LCD_D6 EQU P1.2
LCD_D7 EQU P1.1

Eksempler på færdige subrutiner:

LCD_Init (der også sender danske karakterer til LCD)

LCD_returnhome:

LCD_clear:

curser_on:

curser_off:



LCD_printc: ; sender en karakter i Acc (Ascii)
LCD_printstring: ; Sender en String, der termineres med et 0, Bruger datapointeren DPTR
LCD_send_b: ;sender styrekode i ACC til LCD, Til at indstille displayet !
PLC1 ; Subrutiner til positionering af cursoren i linje 1, 2, 3 eller 4.
PLC2 ; Input: Indhold i Acc er X-værdi på linien (0 til 19) !!
PLC3
PLC4

Eks. på tekst: (afsluttes med 0)

Db 30h, 4ah,0 ; svært, idet man selv skal omregne fra ASCII til hex

Db 'Hej alle – God Jul',0

Db 'GI', 01h, 'delig Jul',0 ; Eksempel med danske karakterer !!