

Liquid crystal display

René Derkx

MSX Computer & Club Magazine nummer 85 - september / oktober 1996

Scanned, ocr'ed and converted to PDF by HansO, 2001

Hitachi scherm op MSX

Dit hardware-project maakt het mogelijk vanuit de MSX een liquid crystal display of LCD aan te sturen. Naast de hardware is echter ook in software voorzien, die in tegenstelling tot de hardware volledig door René Derkx is ontwikkeld. De schakeling is in feite gebaseerd op een schema dat ooit in *Elektuur* is gepubliceerd.

Het is reeds drie jaren geleden dat ik van een medestudent een liquid crystal display kon kopen voor een klein prijsje. Deze display, gebaseerd op een Hitachi HD44780 LCD-II controller, heeft twee rijen van veertig kolommen. Na de datasheets van de HD44780 bekeken te hebben, bleek dat de display erg eenvoudig aan te sturen is met behulp van een 8-bits microcomputer. Daarnaast beschikt de HD44780 over een vrij groot aantal instructiefuncties, waardoor niet op een laag niveau het display hoeft te worden aangestuurd. Hierna heb ik getracht om met behulp van de I/O-aansturing op de cartridge poort van de MSX computer de LCD aan te sturen. Dit bleek een succes te zijn, zodat ik daarna ook nog een driver voor mijn LCD module heb geschreven. Omdat de HD44780 vaak wordt gebruikt op LCD modules, leek het me aardig een artikel te schrijven voor mensen die zo'n LCD module zouden willen aansturen met de MSX.

Mogelijkheden

Wanneer we de datasheets van de Hitachi HD44780 bekijken, zien we dat de volgende mogelijkheden beschikbaar zijn:

- 80x8 bits display data ram;
- karaktergenerator-rom;
- karaktergenerator-ram;
- interne aansturing voor LCD;
- reset bij inschakeling van de LCD.

Het blokdiagram van de interne deelfuncties van de HD44780 staat in figuur 1. De HD44780 controller is ondergebracht in een complete LCD module, waar de controller aparte LCD drivers bestuurt. De controller en deze drivers samen kunnen de LCD aansturen.

De interne karaktergenerator-rom heeft 160 karakters van 5x7 pixels en 32 karakters van 5x10 pixels. De karakters 32 tot en met 127 in het rom komen op een paar karakters na overeen met de ascii-set. De karakters 128 tot en met 159 zijn niet gedefinieerd. Verder zitten er Kana karakters op de plaatsen 160 tot en met 223. Op de plaatsen 224 tot en met 255 zitten de karakters met een grootte van 5x10 pixels. Ten slotte kan de gebruiker zelf karakters maken op de plaatsen 0 tot en met 15. De

komplete karakter-rom/ram staat in figuur 2.

De belangrijkste commando's die men kan gebruiken om de display aan te sturen, zijn de volgende:

- wissen van de display;
- cursor naar linker bovenhoek;
- display aan/uit;
- cursor aan/uit;
- knipperende cursor aan/uit;
- besturing van de cursor;
- schuiven van de karakters.

Het gaat te ver om alle commando's te bespreken. De complete commandolijst staat in de tabel, zodat men zelf kan experimenteren met de display. Men kan met het softwarematig aansturen van de display niets kapot maken.

Hardware

De HD44780 controller heeft twee besturingslijnen, namelijk RS (register select) en R/W (read/write). Daarnaast heeft de controller acht datalijnen. Tenslotte is er een enable-ingang, zodat de controller 'weet' wanneer de data en besturingslijnen geldig zijn.

Omdat de meeste aansturingen van LCD-modules met behulp van de bovenstaande lijnen plaatsvinden, kunnen we een universele LCD-aansturing op de MSX gebruiken. Zo'n aansturing heeft ooit eens in het tijdschrift Elektuur gestaan en ik heb dit principe gebruikt, zij het in een gewijzigde vorm.

De I/O-adressen 0 tot en met 3 worden gebruikt voor de aansturing van RS en R/W. Misschien dat deze adressen voor andere MSX uitbreidingen worden gebruikt, zodat men het I/O-gebied zal moeten wijzigen. Dit heeft als consequentie dat men in plaats van IC1 een of meer IC's moet gaan gebruiken, waardoor de lay-out gewijzigd zal moeten worden.

Instruction	Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display	0	0	0	0	0	0	0	0	0	1
Return home	0	0	0	0	0	0	0	0	1	
Entry mode set	0	0	0	0	0	0	0	1	I/D	S
Display on /off control	0	0	0	0	0	0	1	D	C	B
Cursor/display shift	0	0	0	0	0	1	S/C	R/L		
Function set	0	0	0	0	1	DL	N	F		
Set CG ram address	0	0	0	1	Acg					
Set DD ram address	0	0	1	Add						
Read busy flag and address	0	1	AC							
Write CG/DD	1	0	Write							

ram data			data							
Read CG/DD ram data	1	1	Read data							

Tabel 1: HD44780 controller instruction set

Voor mensen die verder geen andere uitbreidingen op de MSX hebben, behoeven het schema en de lay-out geen wijzigingen. ..

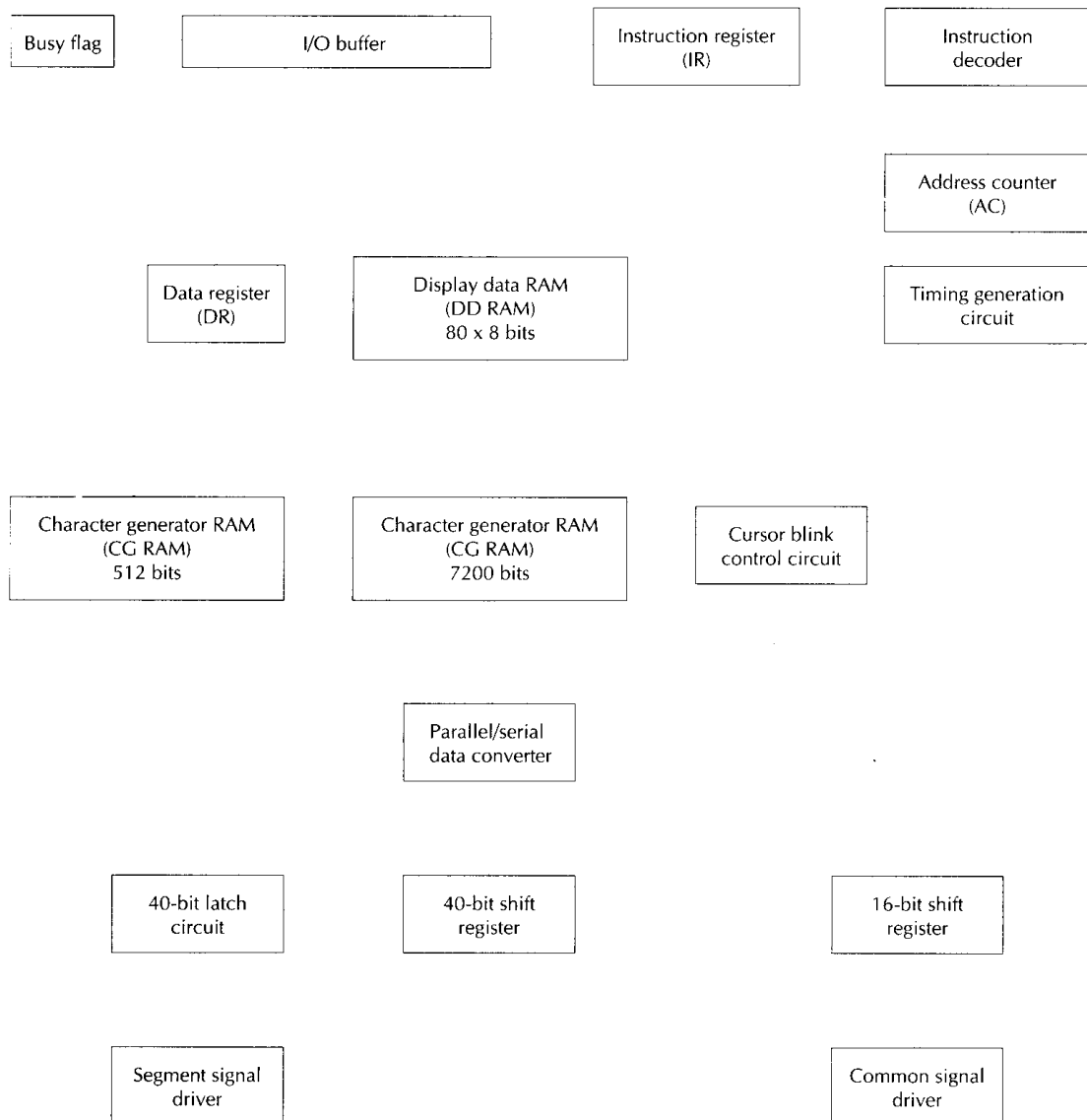
Voor de hardwarehobbyist behoeft het schema dat staat afgebeeld als figuur 3 geen verdere uitleg. Het enige wat nog vermeld dient te worden, is dat de potmeter dient voor de contrast-instelling van de LCD.

Zelfbouw

Men kan met de lay-out die is afgebeeld als figuur 4 gemakkelijk zelf de print etsen, wanneer men de middelen hiertoe heeft. De print zal dubbelzijdig geëtsd moeten worden, zodat men bij een normaal etsproces nog de doorverbindingen van de ene naar de andere zijde van de print zal moeten maken. Dit kan door middel van een simpel draadje dat men boven en onder soldeert, maar het kan ook met de daarvoor in de handel zijnde holnieten. Verder zullen sommige componenten aan de boven- én onderzijde moeten worden gesoldeerd, omdat op die plaatsen boven en onder op de print sporen naar die component lopen. Ook de connector voor de aansluiting op de LCD moet boven en onder gesoldeerd worden; dit kan voor sommige soorten connectors problemen opleveren. Tevens moet er gelet worden op het feit dat de dikte van de print in de buurt van de 1,5 millimeter moet blijven. Dit is gestandaardiseerd in de MSX norm. Wanneer men de mogelijkheid heeft de print te vertinnen, verdient dit zeker de aanbeveling. Een vertinde print zal minder snel oxideren dan een onbeschermd print.

I/D	Increment cursor address (1)	Decrement cursor address (0)
S	Accompanies display shift (1)	
S/C	Display shift (1)	Cursor move (0)
R/L	Shift right(1)	Shift left (0)
DL	8-bits interface (1)	4-bits interface (0)
N	2-line display (1)	1-line display (0)
F	5x10 pixels (1)	5x7 pixels (0)
BF	Internally operating (1)	Can accept instruction (0)
CGram	Character generator ram	
DDram	Display data ram	
Acg	CG ram address	
Add	DD ram (cursor) address	
AC	Address counter for DD/CG-ram address	

Tabel 2: Instruction set symbols



Figuur 1: Blokschema van de controller HD44780

Een nog mooiere — en duurdere — oplossing is het vergulden van de connectoruiteinden op de print.

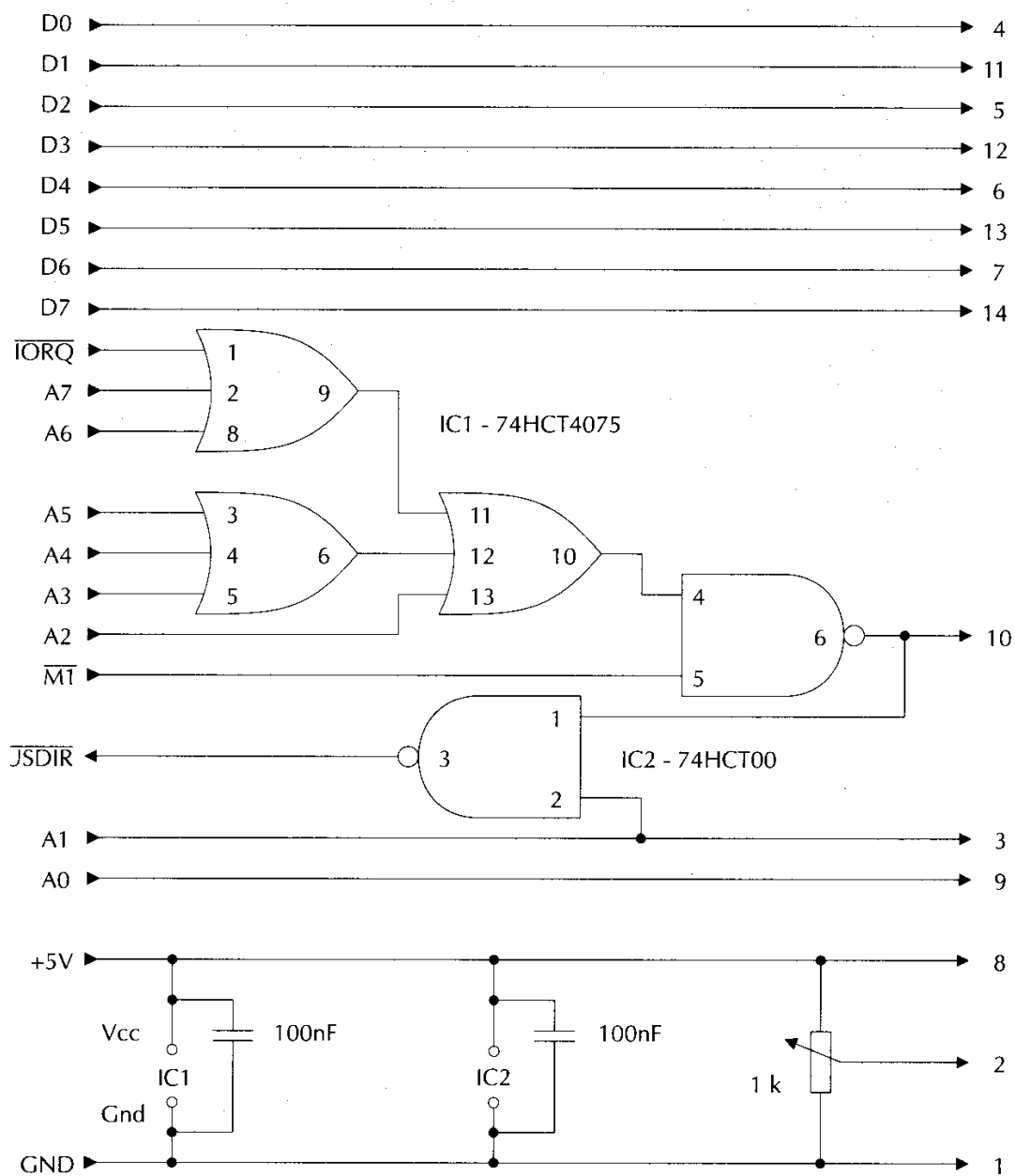
Het overzetten van de lay-out op een transparant kan gebeuren met behulp van een kopieerproces of reproductie. Men kan ook gebruik maken van de postscript file. Men kan dan met behulp van een laserprinter de lay-out rechtstreeks op kalkpapier zetten. Pas wel op dat de printer geschikt is voor het papier! De postscript file is qua afmetingen correct, zodat men hier niet mee moet knoeien. Voor het uitprinten moet

een daarvoor geschikt programma gebruikt worden.

	00	20	30	40	50	60	70	A0	B0	C0	D0	E0	F0
00	CG RAM		0	1	2	3	4	5	6	7	8	9	a
01	(2)	.	1	A	a	2	3	4	5	6	7	8	9
02	(3)	"	2	B	b	3	4	5	6	7	8	9	a
03	(4)	#	3	C	c	4	5	6	7	8	9	a	b
04	(5)	\$	4	D	d	5	6	7	8	9	a	b	c
05	(6)	%	5	E	e	6	7	8	9	a	b	c	d
06	(7)	&	6	F	f	7	8	9	a	b	c	d	e
07	(8)	'	7	G	g	8	9	a	b	c	d	e	f
08	(1)	(8	H	h	9	a	b	c	d	e	f	g
09	(2))	9	I	i	a	b	c	d	e	f	g	h
0A	(3)	*	0	J	j	b	c	d	e	f	g	h	i
0B	(4)	+	1	K	k	c	d	e	f	g	h	i	j
0C	(5)	,	2	L	l	d	e	f	g	h	i	j	k
0D	(6)	-	3	M	m	e	f	g	h	i	j	k	l
0E	(7)	.	4	N	n	f	g	h	i	j	k	l	m
0F	(8)	/	5	O	o	g	h	i	j	k	l	m	n

Figuur 2: Karakterset

Bij het belichtingsproces voor kalk-papier moet de belichtingstijd langer zijn dan bij transparanten. Ik hoop hiermee genoeg informatie en tips te hebben gegeven over het zelf bouwen van de print.



Figuur 3 Schema

Software

Met de instructies voor de Hitachi display controller als basis, heb ik een universele driver geschreven. Deze driver kan gebruikt worden in de schermmodes 0 en 1; dit zijn immers de enige schermmodes die opgebouwd worden uit een reeks ascii's. Omdat mijn display 2x40 karakters bezit, kan ik deze twee regels op mijn display

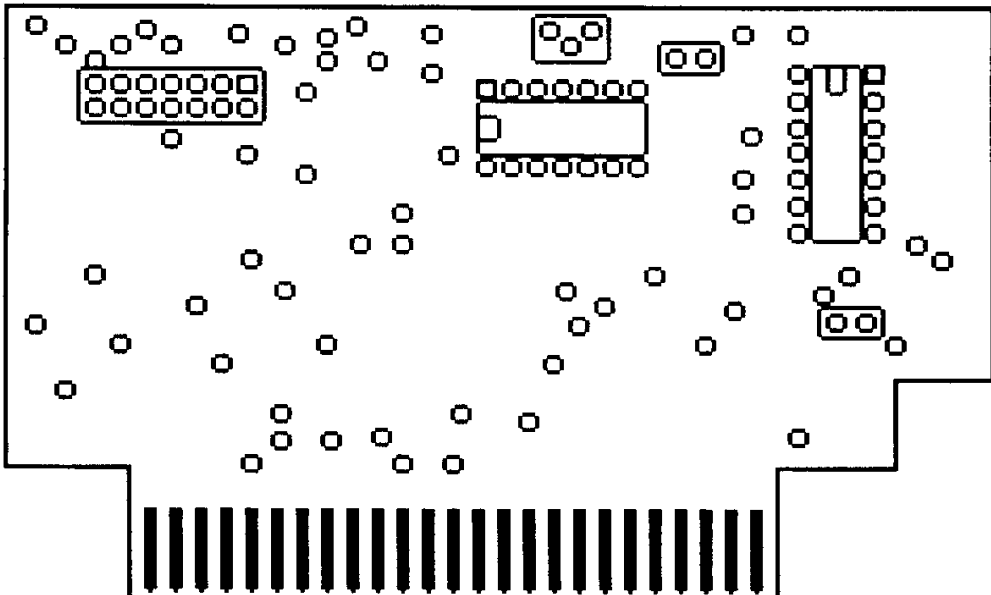
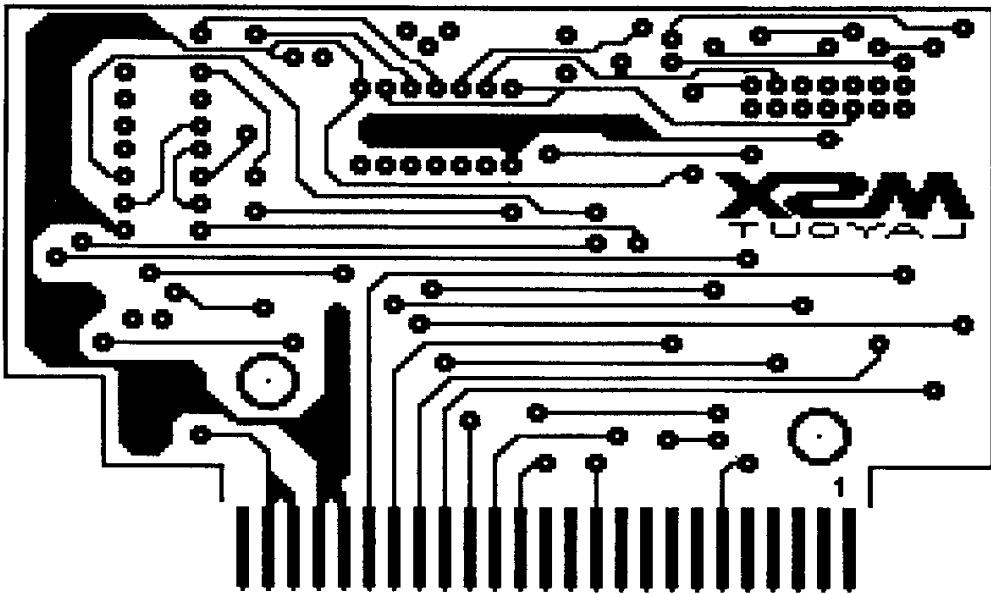
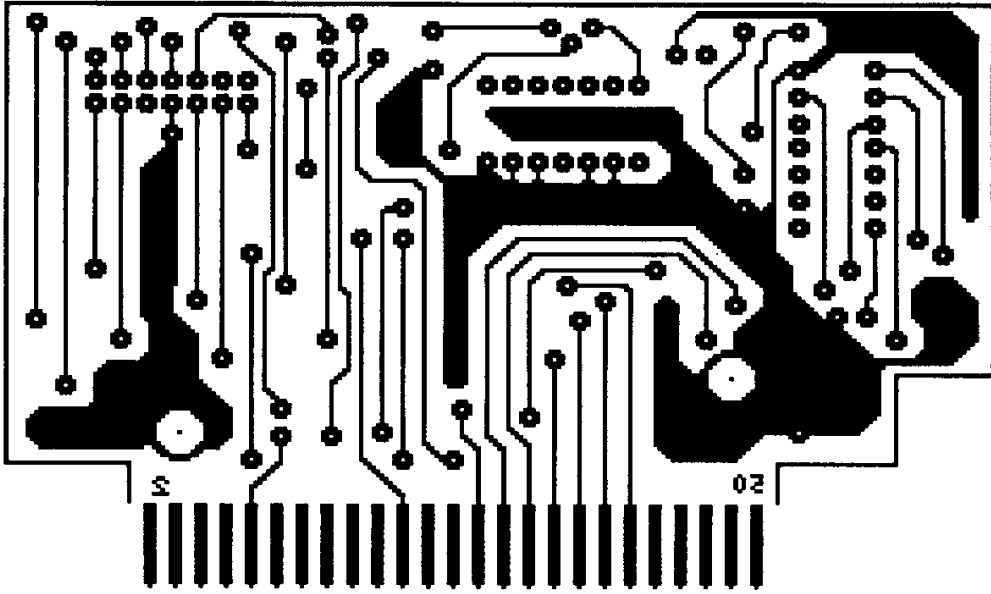
overeen laten komen met twee regels van de normale schermopbouw. Om de 80 kolommen die er kunnen zijn bij scherm-mode 0 zichtbaar te maken op de LCD, heb ik gebruik gemaakt van scrolling. Wanneer men een display gebruikt met 4x20 kolommen, dan zou men dus vier regels onder elkaar kunnen zetten, die overeenkomen met vier regels van het normale scherm. Wanneer men echter de door mij geschreven driver zal gaan gebruiken, zullen de vier regels van de display twee regels van de normale schermopbouw weergeven. Het komt er in feite op neer, dat de driver geschreven is voor 2x40 display's. Men kan ook andere display's gebruiken, maar dit zal een minder mooi resultaat geven. Dit is op te lossen door zelf de driver te wijzigen; voor de ervaren programmeur zal dat niet veel problemen opleveren. De driver wordt ondergebracht in de play buffer. Deze buffer wordt normaal gesproken gebruikt om de PLAY-commando's onder basic tijdelijk op te slaan. Wanneer men PLAY niet gebruikt, kan deze plek zonder problemen gebruikt worden. Wil men het commando PLAY echter wel gebruiken, dan kan men zonder problemen het adresgebied van de driver veranderen. Wanneer men de driver aanroept, wordt allereerst gecontroleerd of de LCD daadwerkelijk aangesloten is. Zo niet, dan zal daarvan melding gemaakt worden. Is er wel een LCD herkend, dan zal de display worden geïnitieerd. Ten slotte wordt de interrupt hook afgebogen, zodat na terugkeer de LCD continu wordt aangestuurd. Een uitvoerige bespreking van de software gaat te ver. Maar met behulp van de commentaarregels en de instructieset uit tabel 1, zou men er toch wel uit moeten kunnen komen.

Tenslotte

Een liquid crystal display aansluiten op een MSX blijkt in sommige gevallen voordelen op te leveren ten opzichte van een monitor. Er is minder plaats nodig en er is minder vermogensconsumptie.

Een nadeel daarentegen is wel dat de LCD alleen onder de schermmodi 0 en 1 bruikbaar is. De display voldoet goed bij gebruik onder basic, dos en vele andere programma's onder tekstmode. Ook Ted is bijvoorbeeld bruikbaar, maar men heeft nu niet het overzicht over het hele scherm en men zal ook de toetscombinaties uit het hoofd moeten kennen.

Ik wens ten slotte iedereen die aan de slag wil gaan met de hardware en/of software erg veel succes.



Figuur 4: Printlay-out en componentenopstelling

```
ORG      &HF975

; Hitachi display controller (HD44780) driver
; Geschreven door R.Derkx, Ysselsteyn

        LD      B,255                ; Display aangesloten?
NOTBEG: LD      C,2
        IN      D,(C)
        LD      A,D
        AND     &B10000000
        OR      A
        JR      Z,BEGIN              ; Display aangesloten
        DJNZ   NOTBEG              ; Display is afwezig
        LD      DE,NOTCON
        LD      C,9
        CALL   &HF37D              ; Afdrukken 'absent'
        RET

NOTCON:  DEFM   "HD44780 absent!$"
CONN:    DEFM   "HD44780 driver, "
        DEFM   "(c)1996 by R.Derkx$"

PRINT:   LD      DE,CONN
        LD      C,9
        CALL   &HF37D              ; Afdrukken tekst
        RET

BEGIN:   CALL   FUNCT              ; Function set
        CALL   ONOFF              ; Display ON/OFF control
        CALL   MODE                ; Entry mode set
        CALL   CLRDIS             ; Clear display

; initialisatie Hitachi display

        LD      A,12
        RST    &H18                ; Wissen van beeldscherm

        DI
        LD      HL,HOOK
        LD      DE,&HFDA4
        LD      BC,5
        LDIR                   ; MSXIO CHPUT hook wijzigen
        LD      HL,HOOK
        LD      DE,&HFDA9
        LD      BC,5
        LDIR                   ; MSXIO DSPCSR hook wijzigen
        EI
        CALL   PRINT              ; Tekst afdrukken
        RET                      ; Terug naar BASIC

SCMODE:  DEFB   0                  ; 40 of 80 kolommen

ROUTIN:  DI                      ; Interrupts uitzetten
        PUSH   AF
        PUSH   BC
        PUSH   DE
        PUSH   HL                ; Bewaren registers
        LD      A,&B10000000
        OUT    (0),A            ; Zet DDRAM adres op 0
        LD      B,40             ; 40 kolommen
        LD      A,(&HF3AE)       ; A := Aantal kolommen
        CP     41
        JR      C,MODE40
        LD      B,80             ; 80 kolommen
MODE40:  LD      A,B
        LD      (SCMODE),A      ; (SCMODE) := 40 of 80 kolommen
        LD      E,A
```

```

LD      D,0                ; VRAM adres (DE) := (SCMODE)
LD      A,0
SUB     E
LD      L,A
SLA    L                    ; Twee regels terug
LD      H,&HFF
CALL   READ                ; VRAM adres aanpassen
LD      HL,0
ROUT2:  EX   DE,HL
        CALL &H4A          ; A := VRAM adres (HL)
        DI
        EX   DE,HL
        CALL WRITE
        INC  L              ; ophogen DDRAM teller
        INC  DE             ; ophogen VRAM adres
        LD  A,40
        CP  L
        JR  NZ,ROUT2       ; Schrijven van eerste regel

LD      HL,0
LD      A,(SCMODE)
LD      E,A
LD      D,0
LD      A,0
SUB     E                    ; Een regel terug
LD      L,A
LD      H,&HFF
CALL   READ                ; VRAM adres aanpassen
LD      HL,0
ROUT3:  EX   DE,HL
        CALL &H4A          ; A := VRAM adres (HL)
        DI
        EX   DE,HL
        CALL WRITE
        INC  L              ; ophogen DDRAM teller
        INC  DE             ; ophogen VRAM adres
        LD  A,40
        CP  L
        JR  NZ,ROUT3       ; Schrijven van tweede regel
CALL   CURSOR
CP     40
JR     C,ROUT8
LD     A,39
ROUT8: LD  B,A
        LD  A,&H40          ; Cursor op tweede regel
ADD   A,B
ROUT7: SET  7,A
        OUT (0),A          ; Cursor plaatsen
EINDE: POP  HL
        POP  DE
        POP  BC
        POP  AF            ; Registers redden
        EI                ; Interrupts activeren
        RET               ; Terug naar hook

CURSOR: LD  C,19
        LD  A,&HF3AE)      ; A := Aantal kolommen
        CP  41
        JR  C,ROUT9
LD     C,39
ROUT9: SRL  A
        LD  B,A
        LD  A,C
        SUB A,B
        LD  B,A
        LD  A,&HF3DD)      ; A := X cursor positie
        ADD A,B            ; A := VRAM adres van de cursor
        RET

```

```

READ:   CALL  CURSOR
        CP    39
        JR    C,READ9
        JR    Z,READ9
        LD    B,39
        SUB   A,B
        LD    C,A
        LD    B,0
        ADD   HL,BC                ; Aanpassing X positie cursor
READ9:  LD    A,(&HF3DC)
        LD    B,A                ; B := Aantal regels
READ2:  ADD   HL,DE
        DJNZ READ2                ; DE := DE + (B * kolommen)
        EX   DE,HL
        RET

HOOK:   CALL  ROUTIN                ; Hook routine aanroep
        RET                        ; Verlaten van hook

CLRDIS: LD    A,&H01                ; Clear display
        OUT   (0),A
        CALL  CHECK
        RET

MODE:   LD    A,&B00000110          ; Entry mode set
        ; Ophogen cursorpositie en geen shift
        OUT   (0),A
        CALL  CHECK
        RET

ONOFF:  LD    A,&B00001111          ; Display ON/OFF control
        ; Display, cursor en blink aan
        OUT   (0),A
        CALL  CHECK
        RET

FUNCT:  LD    A,&B00111100          ; Function set
        ; 8 bits, 5 x 7 dots karakters, 2 lijnen
        OUT   (0),A
        CALL  CHECK
        RET

CHECK:   PUSH  DE                ; Wacht op 'LCD gereed'
CHECK2:  LD    C,2
        IN    D,(C)
        LD    A,D
        AND   &B10000000
        JR    NZ,CHECK2
        POP   DE
        RET

WRITE:  OUT   (1),A                ; Schrijven van karakter
        CALL  CHECK
        RET

```