

LM-16251

De firma Sharp produceert een hele reeks LCD's, met afmetingen van klein tot groot (1 tot 4 regels met 6 tot 80 karakters), waarbij de gebruiker ook nog kan kiezen tussen verschillende uitvoeringen en voedingsspanningen. Enkele displays worden met een karaktergenerator geleverd. Sommige hebben bovendien een ingebouwde data-controller.

Het hier besproken display wordt toegepast in de DCF-referentie-tijd-klok en heeft een aantal voordelen opzichte van andere typen:

- ingebouwde karaktergenerator
- ingebouwde controller (voor eenvoudige data-overdracht)
- slechts twee voedingsspanningen
- goede prijs/prestatie-verhouding

Het display heeft 2 regels met 16 karakters. Ieder karakter wordt in een 5x8-matrix weergegeven.

Het eigenlijke display is samen met twee IC's en enkele andere onderdelen op een print gemonteerd. De schakeling is CMOS- en TTL-kompatibel en heeft met dioden beveiligde ingangen. Het aansluiten op een microprocessorsysteem is dan ook een eenvoudige zaak. Het energieverbruik bedraagt ongeveer 7 mW. De print meet 84 mm ± 44 mm, de twee regels 56,7 mm x 11,5 mm en een karakter 5,55 mm x 2,95 mm.

De karakterset

De karaktergenerator kent 200 karakters:

- 00 - 07: vrij te programmeren karakters
 - 20 - 7F: 96 ASCII-karakters
 - A0 - DF: 64 Japanse schrifttekens (Kana)
 - E0 - FF: 32 bijzondere tekens, waaronder buitenlandse karakters en griekse letters
- In tabel 2 vindt u een overzicht van de moge-

P. Lenz

lijkheden.

Naast de karakterset kent de controller ook een aantal kommando's, zoals wissen, positioneren van de cursor, het verschuiven van de tekst op het display, in- en uitschakelen van de cursor en het definiëren van eigen karakters. De karakters verschijnen altijd op de cursorpositie.

Verschillende gegevens en adressen kunnen ook gelezen worden. Tevens kan met behulp van de BUSY-FLAG worden vastgesteld of de vorige opdracht is afgerond. Het negeren van de BUSY-FLAG kan in het ergste geval het einde van het display betekenen.

Kommando-overzicht

In tabel 1 is de samenstelling van de kommando's gegeven. De verschillende bits hebben de volgende betekenis:

- ID = 0 cursor verschuift naar links
- ID = 1 cursor verschuift naar rechts
- S = 0 de tekst verschuift met de cursor mee
- S = 1 de tekst verschuift niet met de cursor mee
- D = 0 display uit
- D = 1 display aan
- C = 0 cursor uit
- C = 1 cursor aan
- B = 0 cursor knippert niet
- B = 1 cursor knippert
- SC = 0 verschuif cursor
- SC = 1 verschuif scherm
- RL = 0 verschuiven naar links
- RL = 1 verschuiven naar rechts
- DL = 0 4-bits-databus
- DL = 1 8-bits-databus
- N = 0 alleen de bovenste regel
- N = 1 beide regels
- BF = 1 busy, niet schrijven!

APPLIKATOR

kommando	RS	R/W	databus								omschrijving	
			D7	D6	D5	D4	D3	D2	D1	D0		
display wissen	0	0	0	0	0	0	0	0	0	0	1	Wist het geheugen, niet dat van de karaktergenerator. Zet de cursor op home-positie, adres 0'0' (linksboven).
cursor home	0	0	0	0	0	0	0	0	0	1	x	Zet de cursor op home-positie. Verschuivingen van de tekst worden ongedaan gemaakt (adres 0'0' is weer linksboven).
verschuif tekst en/of cursor	0	0	0	0	0	0	0	1	ID	S		Geeft aan in welke richting de cursor verschuift (ID) en of de tekst gelijktijdig ook moet verschuiven (S).
display aan/uit cursor aan/uit/ knipperen	0	0	0	0	0	0	1	D	C	B		Display aan/uit (D) Cursor aan/uit (C) cursor knipperen ja/nee (B)
cursor of display verschuiven	0	0	0	0	0	1	SC	RL	x	x		Verschuift cursor (SC=0) of tekst (SC=1) naar rechts/links (RL).
initialisatie	0	0	0	0	1	DL	N	x	x	x		Breedte data-bus (DL) 1 of 2 regels gebruiken (N)
karaktergenerator adres	0	0	0	1	karakter				rij			Geeft aan welke rij (000...111) van welk karakter (000...111) gedefinieerd moet worden met het volgende data-byte.
geheugenadres	0	0	1	adres								Zet de adresteller op "adres". De volgende data zijn ASCII-tekens.
BUSY-FLAG, adres lezen	0	1	BF	adres								Leest de BUSY-FLAG en het cursor-adres.
data schrijven	1	0	data								Schrijf data.	
data lezen	1	1	data								Lees data.	

x = don't care

Adressen:

adressen:	zichtbaar		onzichtbaar
	links	rechts	
boven	00...0F	10...27	
onder	40...4F	50...67	

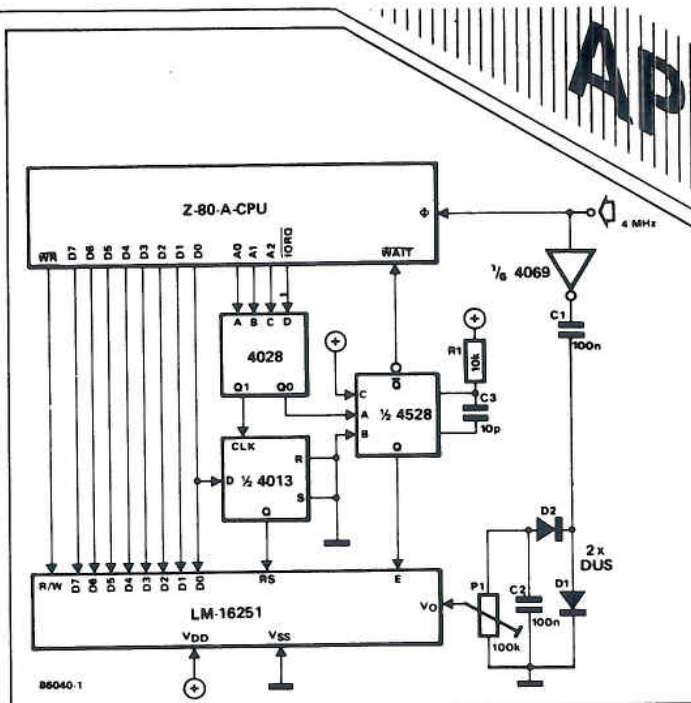
Deze adressen gelden alleen als er niet met de tekst is geschoven. Bij iedere verschuiving ($S=1$ of $SC=1$) verschuiven ook de adressen. Om vertrouwd te raken met de werking is het aan te bevelen om in het begin niet met de tekst te schuiven ($S=0$ en $SC=0$) en de cursor in te schakelen ($C=1$). Houd in de gaten dat het lezen en schrijven van data en kommando's de cursorpositie beïnvloedt.

De aansluitingen

- 1...8: bidirectionele databus D0...D7 (tri-state)
- 9: E = Enable. Hier moet rekening worden gehouden met enige belangrijke schakeltijden. RS en R/W moeten minstens 140 ns stabiel zijn, voordat E "1" mag worden. E moet dan minstens 450 ns "1" blijven, terwijl de datalijnen gedurende die tijd stabiel moeten zijn.
- 10: R/W, 0 = schrijven
1 = lezen
- 11: Register Select (RS).
Hiermee wordt onderscheid gemaakt tussen data en kommando's. RS kan men bijvoorbeeld sturen met behulp van een flipflop.
0 = commando
1 = data
- 12: V_o = Dit is de displayspanning. Ze dient circa -2 V te bedragen. De juiste spanning kan alleen door proberen worden gevonden; ze bepaalt het gewenste contrast en de afleeshoek.
- 13: $V_{DD} = +5$ V
- 14: $V_{SS} =$ massa

Een voorbeeld

Figuur 1 geeft een praktijkvoorbeeld waarbij het display is aangesloten op een Z-80-A-microprocessor-systeem. Voor de duidelijkheid is alleen de proces-



sor getekend. Deze processor heeft als voordeel dat de snelheid van de processor door middel van de WAIT-ingang, kan worden aangepast aan de snelheid van het display. De adresdekodeer 4028 is alleen met IORQ en niet met WR of RD verbonden om onderdelen te sparen.

Door uitgang Q1 van de 4028 wordt een flipflop (4013) geklokt, zodat de waarde van D0 wordt onthouden en aan RS wordt doorgegeven. Via Q0 wordt de monoflop (4528) gestart. De interne vertraging van de adresdekodeer en de monoflop zijn met zekerheid langer dan de vereiste 140 ns. De pulsduur van de monoflop bedraagt ongeveer 700ns. Het verkorten van deze tijd tot 450 ns heeft nauwelijks zin, want u kunt toch niet zo snel kijken (het display staat vol in circa 23 μ s). De

monoflop stopt ook de processor, zodat de datalijnen voldoende lang stabiel blijven. Dynamische RAM's hebben van het kortstondige stoppen geen last. Uit het kloksignaal van het processor-systeem wordt met C1/C2 en D1/D2 een negatieve spanning gemaakt, die met P1 op ongeveer -2 V wordt ingesteld.

Deze schakeling funktioneert tot een klokfrequentie van 4 MHz. Voor systemen met een klokfrequentie tot 2 MHz kan voor de adresdekodeer een 4514 worden gebruikt.

De praktijk

Na het inschakelen moet de bovenste regel volledig donker worden, terwijl de onderste regel helder blijft. Is dit niet zo, dan moet V negatiever worden gemaakt. Nu kunnen de



APPLIKATOR

eerste bevelen verstuurd worden. Voor iedere schrijfoopdracht moet de BUSY-FLAG gecontroleerd worden! Eerst moet het display worden geïnitieerd: maak RS 0 en stuur dan DL=1 en N=1, ofwel 38Hex, naar het display. De bovenste regel wordt nu ook helder. Vanaf nu kan men de gewenste kommando's versturen. Bijvoorbeeld 01Hex (display wissen) en daarna D=1, C=1, B=1 ofwel 0EHex (cursor aan) en, omdat de cursor net zo moet werken als op een beeldscherm, 06Hex (ID=1, S=0). Maken we RS nu "1" dan kunnen de eerste karakters worden verstuurd. Als die niet verschijnen moet V_o waarschijnlijk bijgesteld worden (met P1).

Nu zetten we de cursor in het midden van de tweede regel (op adres 48Hex), door C8Hex te versturen. Uiteraard moet eerst RS "0" worden gemaakt. Hier plaatsen we een zelf gedefinieerd karakter (nr. 03): dus RS=1 en 03Hex versturen. Regel 1 van dit karakter wordt nu geprogrammeerd met RS=0 en dan versturen we 01 + 011 (teken) + 001 (regel), ofwel 59Hex. Op regel moeten drie punten komen te staan, dus RS=1 en dan 00010101=15Hex. Daarna kan de tekst verschoven worden: RS=0 en 1CHex (SC=1, RL=1). De tekst verschuift nu naar rechts. Dat de adressen nu ook verschuiven ziet u als de cursor op adres 00 wordt gezet (80HexVersturen). De cursor verschijnt dan onder het tweede karakter van rechts. Met het kommando "home" (02Hex) komen alle karakters weer op de oorspronkelijke plaats terug.

APPLIKATOR

0 2 3 4 5 6 7 A B C D E F

	*1 *2 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0	xxxx0000	CG RAM (1)												
1	xxxx0001	(2)												
2	xxxx0010	(3)												
3	xxxx0011	(4)												
4	xxxx0100	(5)												
5	xxxx0101	(6)												
6	xxxx0110	(7)												
7	xxxx0111	(8)												
8	xxxx1000	(1)												
9	xxxx1001	(2)												
A	xxxx1010	(3)												
B	xxxx1011	(4)												
C	xxxx1100	(5)												
D	xxxx1101	(6)												
E	xxxx1110	(7)												
F	xxxx1111	(8)												

*1 High-order *2 Low-order

86040-T2

Opmerking: CG-RAM is het geheugen voor de zelf te definiëren karakters.