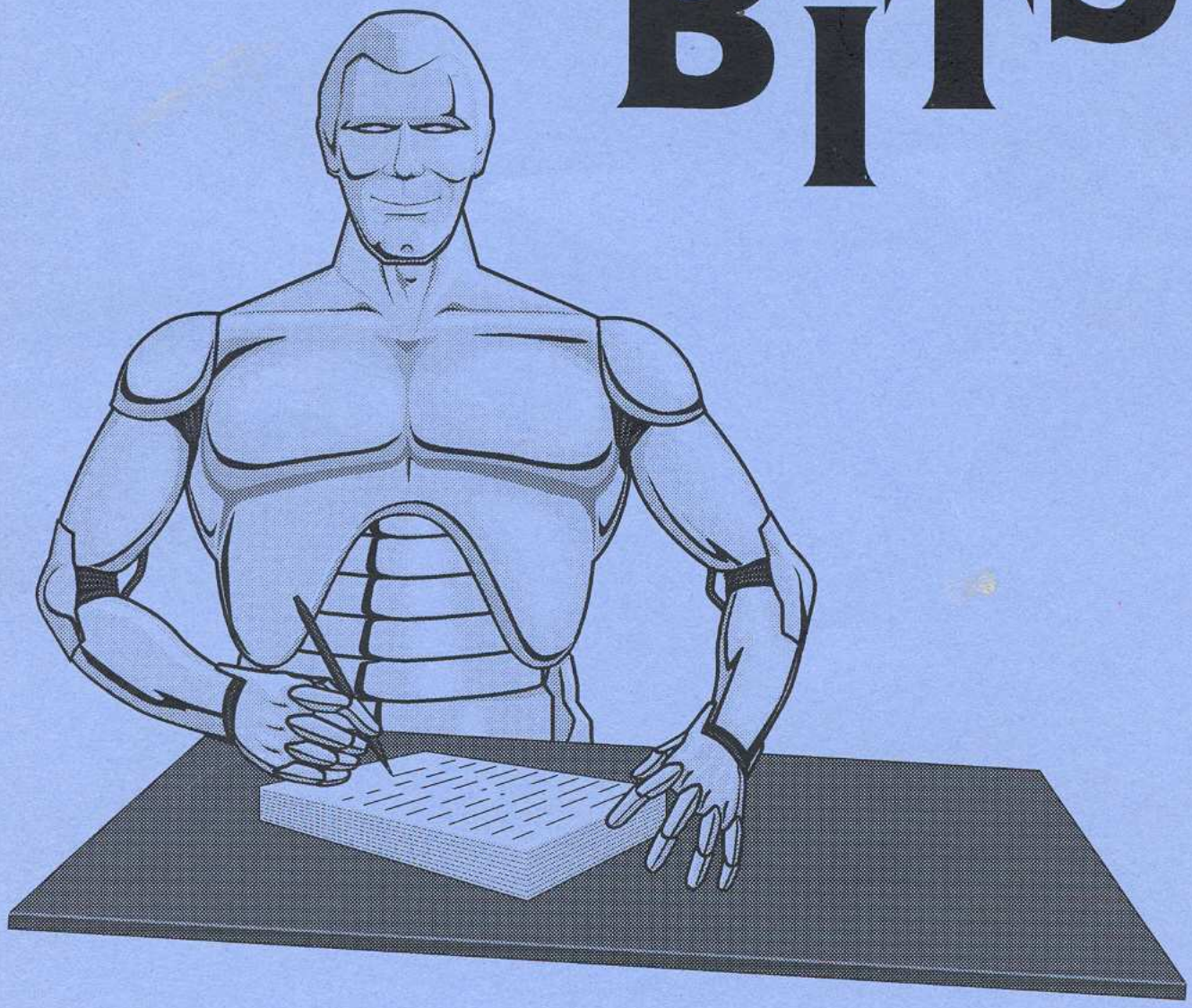
PORT BETAALD
GOUDA

Een uitgaven van de HCC Robotica GG
Jaargang 1, nummer 3 november 1998

ROBOBITS

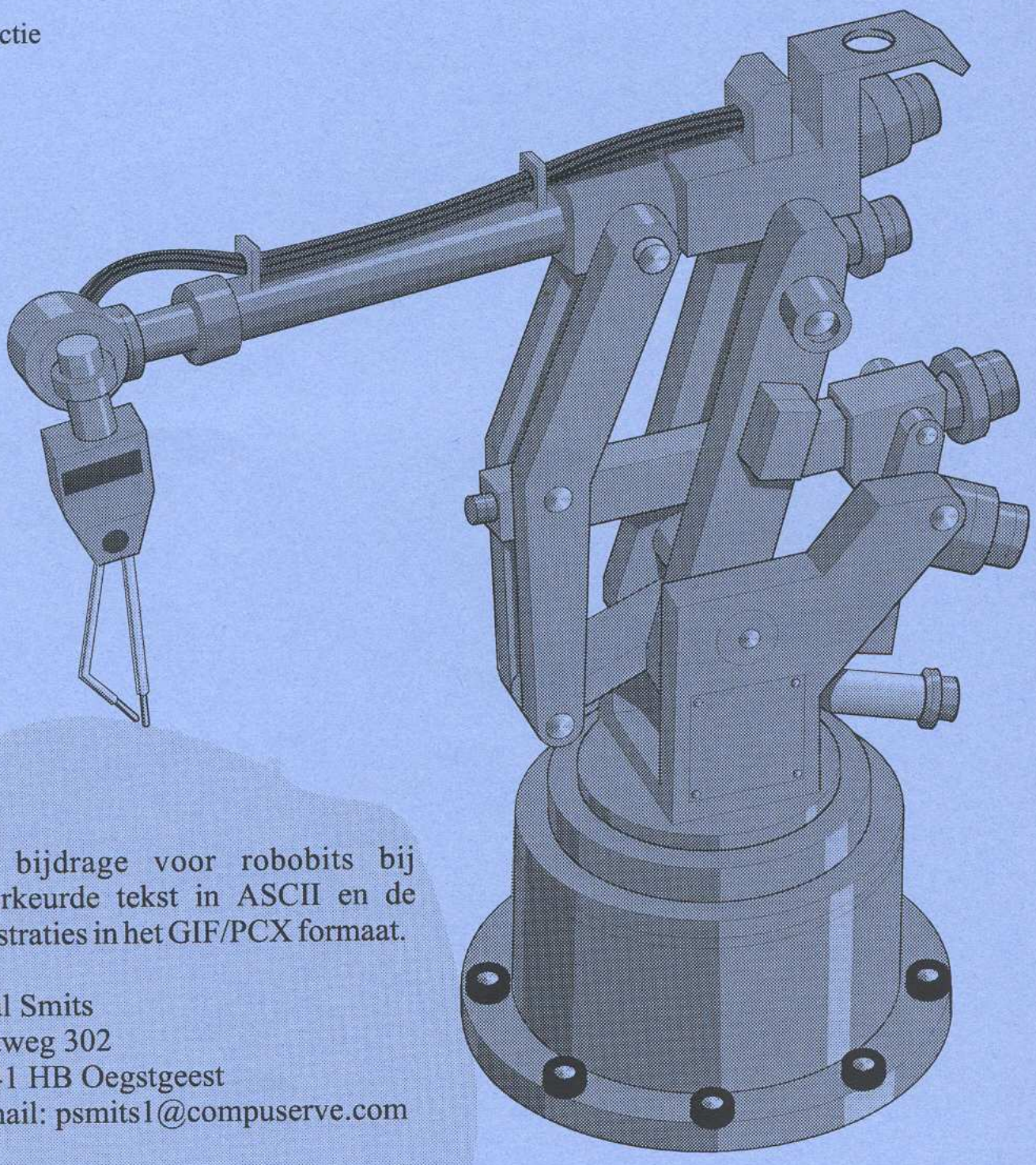


Afzender: HCC Robitica GG, p/a J.W. Ligthelm, Koekoekplein 13 2802 AD Gouda

Van de redactie

In deze door de tijdsdruk van de HCC-dagen verlate Robobits een special over LCD aansturing. Met dank aan Abraham Vreugdehil en onder vermelding van "New Brain Online" waar dit artikel in hun editie 22 heeft gestaan

De redactie



De bijdrage voor robobits bij voorkeurde tekst in ASCII en de illustraties in het GIF/PCX formaat.

Paul Smits
Lijtweg 302
2341 HB Oegstgeest
e-mail: psmits1@compuserve.com

lcd-aansturing

Als uitbreiding op het B+-bordje is een van de mogelijkheden een LCD-schermpje. Deze schermpjes zijn er in vele soorten en maten. De bekendste zijn de schermpjes van 1 regel met 16 karakters en de schermpjes van 2 regels met 16 karakters. Maar er zijn ook schermpjes van 8 en 40 karakters en andere van 4 regels. Daarnaast zijn er uitvoeringen met en zonder backlight. Al deze dingen komen wel tot uitdrukking in de prijs.

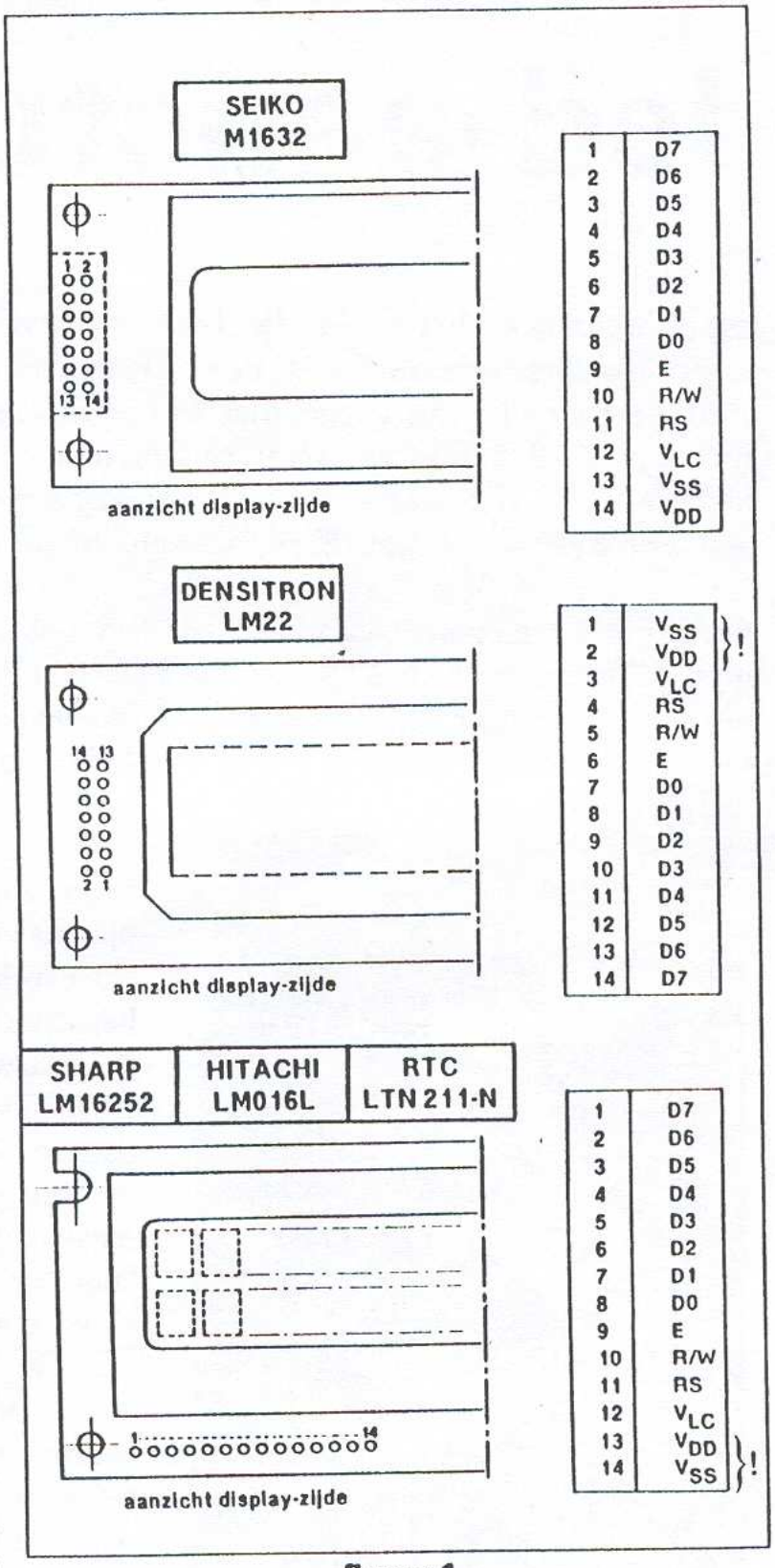
Binnen de B+-club in Naaldwijk wordt sinds het najaar van 1996 gewerkt met schermpjes van 2 × 16 karakters. Hier worden de meest uiteenlopende zaken mee gedaan. Van spelletjes tot lopende en vaste wisselende teksten tot meetwaarden van de A/D-meting en de 10 bits A/D-meting.

In principe werken de schermpjes als volgt: het schermpje bevat een klein geheugen, dat op het scherm afgedrukt wordt. Op de plaats waar de cursor staat, verschijnt de tekst. Op de 8 bits databus wordt de ascii-waarde van een getal of letter gezet, dan maak je de Enable hoog en weer laag en dan staat die letter of dat cijfer op het scherm en de cursor staat op de volgende positie. Als je dan weer een waarde op de databus zet en weer de Enable hoog en laag maakt, staat die waarde op die positie. Naast data kun je ook commando's naar het

scherm sturen. Deze kunnen informatie bevatten, hoe het scherm aangestuurd moet worden, hoeveel regels het scherm bevat, hoe de cursor moet zijn, en je kunt de cursor op een bepaalde positie zetten.

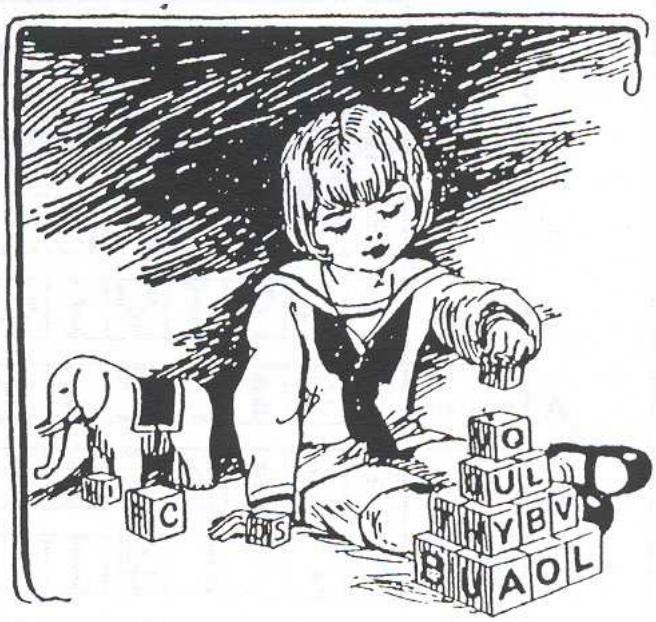
De meeste schermpjes bevatten 14 aansluitingen. Dit zijn 8 databusaansluitingen, Enable, RS (Register Select), R/W (Read / Write), 5 volt, massa en de schermintensiteitaansluiting. Daarnaast zijn er drie verschillende type schermpjes met elk hun eigen manier van aansluiten. De verschillende aansluitingen zijn in figuur 1 afgebeeld.

Nu is de vraag, welke schermaansluiting correspondeert met het schermpje dat je in je handen hebt. Ten eerste zoek je met een ohmmeter de massa-

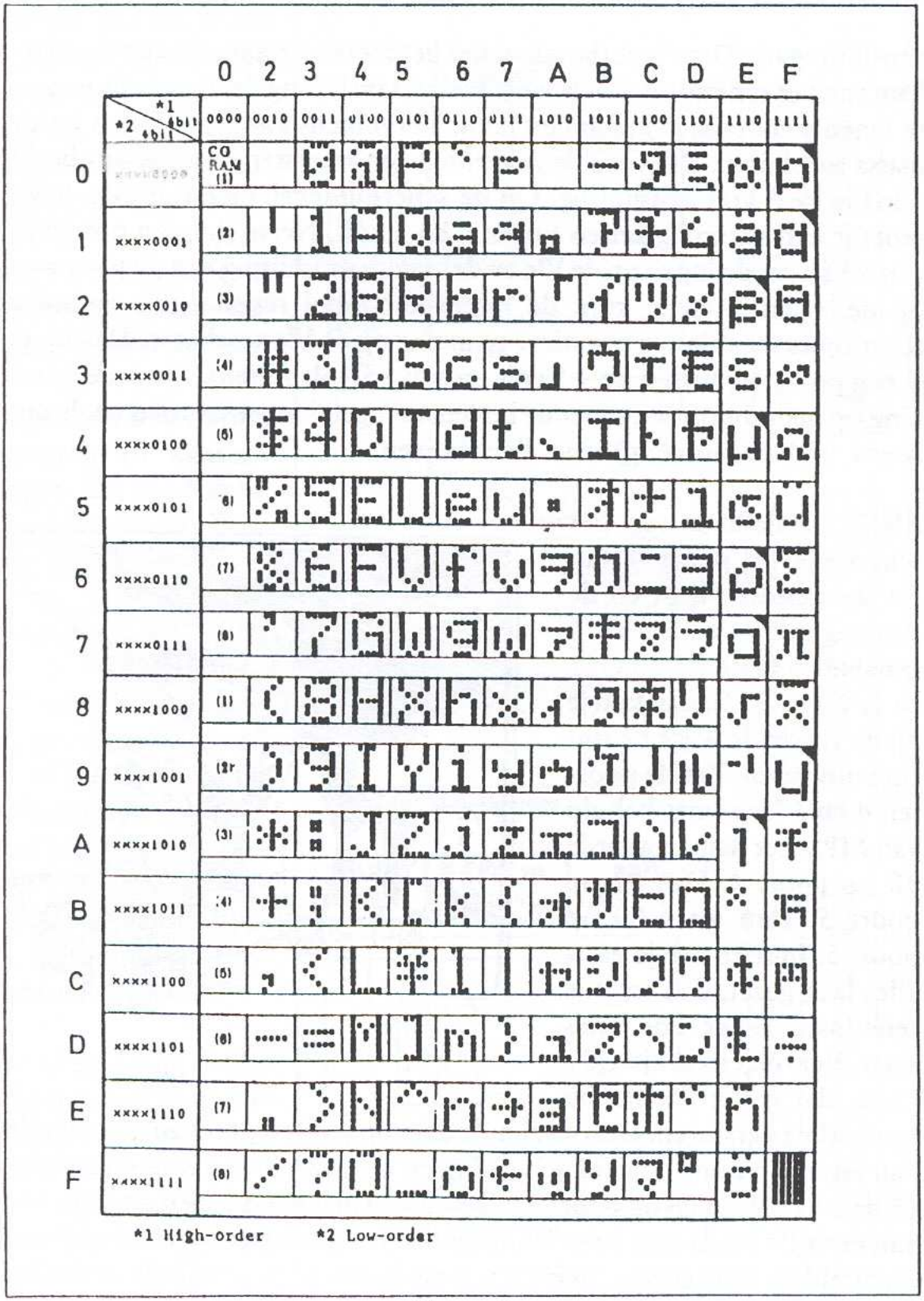


figuur 1

aansluiting op. Deze is verbonden met het metalen frame dat aan de voorkant van het scherm zit. Als je weet welke aansluiting dat is, weet je in grote lijnen welk type je in handen hebt. Vervolgens kun je de 5 volt en de massa aansluiten. Zet voor de zekerheid een weerstand van bijvoorbeeld 1 kΩ in de 5 volt aansluiting. Op de schermintensiteitsaansluiting (Vlc) moet je een spanning zetten tussen 0 en 5 volt. Neem dus een potmeter van 5 KΩ, zet de lopen op de Vlc en de andere aansluiting van de potmeter op de + en - en je kunt de intensiteit mooi regelen (bij sommige schermpjes moet je een negatieve spanning op de Vlc aansluiten. Dan moet je een potmeter tussen de +5 volt en een -5 volt zetten). Als je de spanning op het scherm zet, moet de bovenste regel donker worden en de onderste (indien aanwezig) moet licht blijven.



Als je de databus van het schermpje op poort 4 van B+ aangesloten hebt en de stuursignalen op poort 5 (Enable op Q50, RS op Q51 en R/W op Q52), dan kun je op de volgende wijze de sturing uitvoeren. Zet de poorten 4 en 5 laag (met behulp van MP). Zet dan de waarde 38 op poort 4. Dan 01 op poort 5. Dan weer 00 op poort 5. In feite heb je eerst alles laag gezet, dan weet je tenminste zeker hoe alles staat. Vervolgens zet je Q51 hoog, dat is het RS bit, nu kun je data versturen. Daarna zijn de data op poort 4 gezet en met behulp van bit 1 van poort 5 actief gemaakt. En op deze wijze zet je vervolgens de data 01, 0E en 06 op poort 4, afgewisseld met het Enablen door middel van eerst 01 en daarna weer 00 op poort 5 te zetten. Nu staat het scherm ingesteld op 2 regels van 16 karakters en de cursor en een lijntje onder het karakter. Ook staat de cursor vooraan op de eerste regel.


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

figuur 2

Dan moet je de RS hoog zetten om data te gaan versturen door poort 5 02 te maken. Zet dan de waarde 42 op poort 4, en dan 03 op poort 5 en weer 02 op poort 5. Je ziet dan de letter 'B' op het scherm verschijnen en de cursor staat een positie verder. Door nu weer een andere waarde op poort 4 te zetten en weer de Enable uit te voeren, zie je de volgende waarde verschijnen.

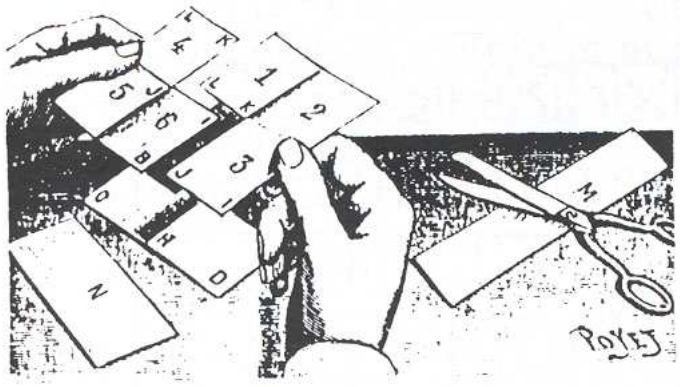
Een ding weet ik zeker, het geeft voldoening, als de eerste karakters op het LCD-scherm verschenen zijn.

Om via het monitorcommando MP het LCD-scherm te sturen is niet echt de opzet. Hier kunnen we een leuk programmaatje voor maken. Dat doen we als volgt.

```
IF
X
K1="38,01,0E, 06,D0"
K2="48,43,43,D1,20,20,20,52,4F,42,4F,54,49,43,41,D0"
K3="20,20,20,D1,20,20,20,20,20,20,20,20,20,20,20,D0"

B
P4=00 P5=00

L1  M10=K1
    ,!C2

L2  M10=K2
    ,!C2

L3  M10=K3
    ,!C2
    ,!L2

C1  V5-1,#$ V6-80,#$ ,!C0      ; wachtlusje.

C2  ; stuur gegevens naar het LCD-scherm.
```

```
V3=M10 _ V3:D0,=@ V3:D1,=C3 V3:D1,=$ P4=V3 . Q50+ ...
                Q50- ,!C1 ,!$
V5-1,#$ V6-1,#$ V7-20,#$
,!C0

C3  ; zet cursor op 2e regel.
    Q51- .. P40=C0 .. Q50+ ... Q50- .. Q51+ ..
    ,!C0

X
```

In de K1-file staan de initialiseringsdata, en in K2 en K3 staan de eigenlijke data. In L1, L2 en L3 wordt de K-file toegekend aan het memorygebied. In C2 worden de data uitgelezen via het commando V3=M10. Vervolgens wordt er gekeken, of de data D0 of D1 zijn. D0 betekent, dat het het einde van de file is. D1 betekent dat er naar de 2e regel van het lcd- scherm gesprongen moet worden. Dan wordt via het commando P4=V3 de waarde op de databus gezet. Met Q50+ en Q50- wordt de Enable uitgevoerd en in C1 wordt de nodige pauze in acht genomen. In C3 wordt eerst het Register Select bit laag gezet, zodat er commando's naar het scherm kunnen, en dan wordt de waarde C0 weggestuurd, wat betekent dat de cursor op positie 40 (hexadecimaal) gezet moet worden (het 7e bit hoog betekent, dat het een cursorpositie is en alleen het 6e bit hoog is de waarde 40 (hex); samen is dat dus C0). Na de bekende Q50+ .. Q50- (oftewel Enable) staat dit commando in de LCD-controller en door dan de RS-aansluiting weer hoog te maken door middel van Q51+ kun je weer data naar het nieuwe schermadres sturen. Wil je niet naar de eerste positie op de tweede regel toe maar naar de 4e, dan neem je dus het getal 43 in plaats van 40 (dus 3 hoger). Op deze wijze kun je



kommando	RS	R/W	databus D7	D6	D5	D4	D3	D2	D1	D0	omschrijving
display wissen	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 00 (linksboven).
cursor home	0	0	0	0	0	0	0	0	1	x	Zet de cursor op home-positie. Verschuivingen van de tekst worden ongedaan gemaakt (adres 00 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)
karakter-generator 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

figuur 3

naar elke positie op het scherm springen, zonder dat je het hele scherm telkens hoeft te wissen en opnieuw te beschrijven.

Aan de hand van de tabel met commando's (figuur 3) kun je zelf op de door jou gewenste wijze het schermpje laten functioneren. De betekenis van de bits is als volgt:

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

de adressen op het scherm

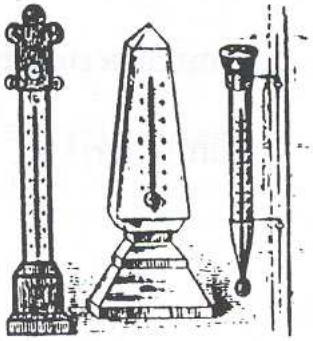
	zichtbaar links . . . rechts	onzichtbaar
boven	00 . . . 0F	10 . . . 27
onder	40 . . . 4F	50 . . . 67

deze adressen gelden alleen, als het scherm niet verschoven is

Er zijn nog veel meer mogelijkheden met het LCD-schermpje. Wil je alle mogelijkheden benutten, dan moet je de datasheets opvragen over bijvoorbeeld de Hitachi of de Sharp LCD-schermserie. In het boek *8052 AH Basic* van *electuur* staan ook de belangrijkste dingen en instellingen die met het LCD-scherm te doen zijn.

A. Vreugdenhil

omzetting



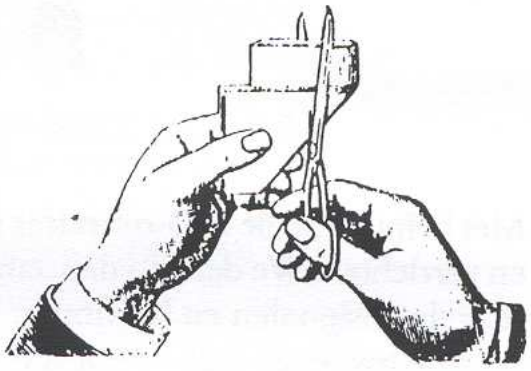
A/D-omzetting naar LCD-scherm

Met behulp van de A/D-omzetter in de 80C535 kunnen we allerlei metingen verrichten. We denken dan aan metingen zoals temperatuur, lichtsterkte, geluidssignalen en bijvoorbeeld potmeterstanden. We zetten dan een waarde tussen de 0 en 5 volt om in een waarde tussen de 00 en FF (hex), oftewel tussen de 0 en 255 decimaal.

De inlezing van de analoge signalen met behulp van B+ is zeer eenvoudig. Hiervoor hebben we het commando *U*. Er zijn 8 analoge ingangen, die hebben de naam *U0* tot *U7*. De waarde die achter dit commando zit, moeten we eerst in een variabele zetten om er mee te kunnen werken. Dus *V1=U5* betekent: zet de waarde van de analoge ingang 6 in variabele 1.

Als we nu deze waarde als data op het LCD-scherm willen zetten kunnen we dat niet direct doen. Zouden we bijvoorbeeld de waarde 32(decimaal, dat is 20 hexadecimaal) op het scherm zetten dan sturen we 20 naar het LCD-scherm. We krijgen dan een spatie op het scherm! Wat hebben we fout gedaan – wat moeten we eerst doen. We moeten het getal dat we op het scherm willen krijgen, eerst ontleden in afzonderlijke stukjes met betrekking tot de honderdtallen, tientallen en eenheden, vervolgens moeten we daar de ASCII-waarden van nemen en die moeten we achtereenvolgens naar het scherm sturen. Dan pas staan de decimale waarden op het scherm die we er willen hebben.

```
IF
X
; dit progje laat de 8 bits waarde van P6.7
; op het LCD-display in decimale vorm zien

K1="38,01,0E,06,D0"      ; initialiseringsfile

P4=00 P5=00 ,!C3         ; init scherm

L1
V1=U7 V1/64 V3/0A

L3
V2+30
V4+30
V5+30

L5
,!C4 ,!C2

L6
P4=20 Q50+..Q50-.. ,!C2
P4=20 Q50+..Q50-.. ,!C2   ; 2 spaties

L7
P4=V2 .. Q50+ .. Q50- ,!C2
P4=V4 .. Q50+ .. Q50- ,!C2
P4=V5 .. Q50+ .. Q50- ,!C2

L8
VB-1,#$  VC-2,#$ ,!L1
```



```
C2      VB-1,#$ ,!C0    ; kort wachtlusje
```

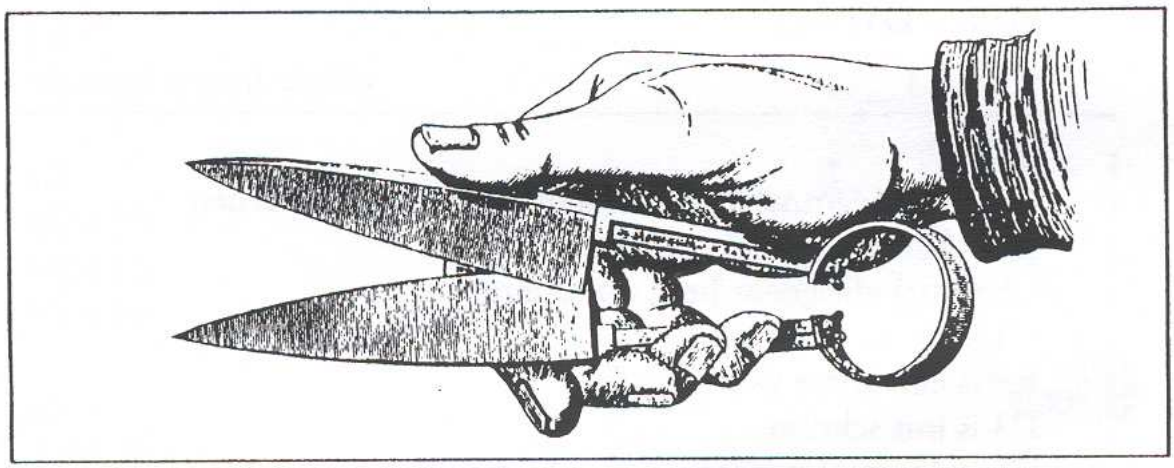
```
C3   ; initialiseringsroutine
     M10=K1
[    VD=M10 _ VD:D0,=@ P4=VD ...
     Q50+ ... Q50- ... ,!C2 ,!$   ]
     ,!C2
     ,!C0
```

```
C4   ; zet cursor op de 1e regel vooraan
     Q51- .. P4=80 .. Q50+ ...
     Q50- .. Q51+ ..
     ,!C0
```

```
;    uitleg:
;    L1 is het begin waar de variabelen 'gemaakt' worden
;    L3 is het ASCII-maakgedeelte
;    L6 stuurt alles naar het LCD-scherm toe
;    C1 is de ASCII-routine
;    C2 is een korte wachtlus.
;    C3 is init scherm
;    C4 is 1e regel vooraan

;    aansluiting LCD-scherm:

;    P4  = DATA
;    Q50 = ENABLE
;    Q51 = RS
;    Q52 = R/W

;    Q67 = analoge ingang (potmeter of sensor)

X
```

Wat gebeurt er in dit programma: in Label 1 wordt de waarde van analoge ingang 8 in Variabele 1 gezet. Deze wordt door 64 (hex) gedeeld. Dat is 100 decimaal. Het resultaat van deze deling (het honderdtal dus) staat in Variabele 2 en de rest in Variabele 3. Deze laatste wordt weer gedeeld

door 0A (hex) wat 10 decimaal is. Het resultaat hiervan (het tiental dus) staat in V4 en de rest (het 1 tal) in V5. Uiteindelijk staan dus de honderdtallen in V2, de tientallen in V4 en de eenheden in V5. In Label 3 wordt bij deze getallen 30 (hex) opgeteld om de juiste ASCII-waarden te krijgen. In Label 5 wordt naar de eerste regel op het scherm gesprongen. In Label 6 worden er twee spaties afgedrukt en in Label 7 worden de drie ASCII-waarden gezet die het getal waar het om gaat op het LCD-scherm schrijven. Na een wachtlusje in Label 8 wordt er weer naar Label 1 gesprongen.



Samengevat werkt het als volgt: het hexadecimale getal wordt gesplitst in drie getallen, te weten de honderdtallen, de tientallen en de eenheden. Dan worden de bijbehorende ASCII-waarden gemaakt en achtereenvolgens op het LCD-scherm gezet.

In plaats van de twee spaties in Label 6 kun je ook een tekst met TEMP of MAX of iets anders zetten. Hier moet je zelf een beetje fantasie voor hebben om er wat van te maken. Die fantasie hebben we wel in Naaldwijk als B+-*crew*, nu de tijd nog. Heb je nog ideeën of suggesties, dan horen we het wel.

A. Vreugdenhil

karakter

eigen karakters definiëren voor een LCD-scherm

Na een tijdje bezig te zijn met het LCD-scherm op je B+-bordje kom je tot de conclusie dat er bepaalde karakters of figuurtjes zijn, die niet in de standaard set van het LCD-scherm zitten. Een streepje boven de *e* bijvoorbeeld, of dat leuke tekentje, dat je in je gedachte had voor dat kleine spelletje. Dan gaan we deze toch zelf definiëren in het LCD-microcontroller geheugen. In dat geheugen is namelijk plaats voor 8 zelf te definiëren karakters in 64 geheugenplaatsen.

Hoe werkt het? We zetten de RS *(register select)* op 0, dat wil zeggen er komt een instructie aan, en geven dan de waarde 01xxxxxx, op de x'en zetten we het adres van de geheugenplaats waar we willen beginnen, en dat is 'bijna' altijd nul, dus schrijven we 01000000. Dan gaan we data naar het scherm sturen, dus we zetten de RS weer op 1. Nu kunnen we de data naar het *Character Generator RAM* sturen. De vraag is, welke data?

Eerst maken we een tekening van 5 × 8 blokjes, en dan zetten we de hexadecimale waarde achter elk regeltje. De laatste regel is altijd leeg, want die is voor de cursor. De zeven regels erboven zijn dus vervangen door 7 getallen. Als je deze getallen plus die 8e als zijnde 00 in het CG-RAM zet, heb je een karakter gedefinieerd. Als je in de ascii-tabel nu 00 opvraagt, krijg je jouw zelfgedefinieerde karakter. Zet je 16 of 24 getallen erin, in plaats van 8, dan staan er dus meerdere symbolen in het CG-RAM, die je aan kunt roepen in de ascii-tabel op de locaties 00, 01, 02 enz.

L2

X

In plaats van zelf 8 karakters te definieren in die 64 byte geheugen, kunnen we dit natuurlijk ook als gewoon extern geheugen gebruiken. Bij B+ hebben we dit niet echt nodig, maar er zijn natuurlijk microcontrollers als *PIC* en *basic-stamps* en andere systemen, die wel woekeren met geheugen. Daar zou je op deze wijze toch de beschikking hebben over 64 byte extra geheugen. Op de 128 die in sommige systemen zit, is dit toch een aanzienlijke uitbreiding.

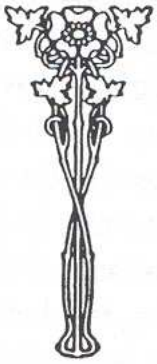
We zien aan het tweede karaktertje, dat we een matrixje van 5 × 8 geprogrammeerd hebben in plaats van 5 × 7. Dat kan dus ook. Als we namelijk de cursor uitzetten, hebben we ook de beschikking over het onderste 8e regeltje.

Character Codes (DD RAM Data)								CG RAM Address						Character Patterns (CG RAM DATA)							
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0
← Higher					Lower →			← Higher			Lower →			← Higher			Lower →				
0	0	0	0	*	0	0	0	0	0	0	0	0	0	*	*	*	1	1	1	1	0
											0	0	1				1	0	0	0	1
											0	1	0				1	0	0	0	1
											0	1	1				1	1	1	1	0
											1	0	0				1	0	1	0	0
											1	0	1				1	0	0	1	0
											1	1	0				1	0	0	1	1
											1	1	1	*	*	*	0	0	0	0	0
0	0	0	0	*	0	0	1	0	0	1	0	0	0	*	*	*	1	0	0	0	1
											0	0	1				0	1	0	1	0
											0	1	0				1	1	1	1	1
											0	1	1				0	0	1	0	0
											1	0	0				1	1	1	1	1
											1	0	1				0	0	1	0	0
											1	1	0				0	0	1	0	0
											1	1	1	*	*	*	0	0	0	0	0
											0	0	0	*	*	*					
											0	0	1								
											0	1	0								
0	0	0	0	*	1	1	1	1	1	1	0	1	1	*	*	*					
											1	0	0								
											1	0	1								
											1	1	0								
											1	1	1	*	*	*					

* don't care

Succes een ieder met zijn LCD-scherm.

A. Vreugdenhil



```
C4   ; zet de eigen karakters in CG-RAM adres 00
     Q51- Q52- Q50-
     P4=40 . Q50+ ,!C1 Q50-      ; zet CG-adres op 00
     Q51+

       ; 1e karakter
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=00 . Q50+ ,!C1 Q50-   ; x x x . . . . .
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=11 . Q50+ ,!C1 Q50-   ; x x x * . . . *
       P4=1F . Q50+ ,!C1 Q50-   ; x x x * * * * *
       P4=10 . Q50+ ,!C1 Q50-   ; x x x * . . . .
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=00 . Q50+ ,!C1 Q50-   ; x x x . . . . .

       ; 2e karakter
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=04 . Q50+ ,!C1 Q50-   ; x x x . . * . .
       P4=1F . Q50+ ,!C1 Q50-   ; x x x * * * * *
       P4=04 . Q50+ ,!C1 Q50-   ; x x x . . * . .
       P4=0E . Q50+ ,!C1 Q50-   ; x x x . * * * .
       P4=0A . Q50+ ,!C1 Q50-   ; x x x . * . * .
       P4=1B . Q50+ ,!C1 Q50-   ; x x x * * . * *

       Q51- ,!C0      ; die x'en worden niet gebruikt
*************************************************************

; K-file string:      D0 = einde
;                     D1 = 2e regel vooraan

; LCD-schermaansluitingen op B+

; P4  = DATA
; Q50 = ENABLE
; Q51 = RS (register select)
; Q52 = R/W
```

Op deze wijze kun je maximaal 8 karakters zelf definiëren. Naast de hele set die al in het geheugen van de LCD-controller zit, is dit meestal wel genoeg.

Hieronder staat een programma waarmee we twee karakters, die zelf gemaakt zijn, op het schermpje weergeven. Probeer het eens te doorgronden en zelf een karaktertje te maken door de getallen in CALL 4 te veranderen. Of teken de waarden die in CALL 4 staan eens uit op ruitjespapier, dan kun je alvast zien, welke tekens er gevormd gaan worden.

```
IF
X

K1="38,01,0E,06,D0"
K2="20,00,20,01,20,01,01,01,00,00,00;01,01,D1,01,01,01,00,DO"

P4=00 P5=00
,!C2    ; init scherm
,!C4    ; zet extra karakters in CG-adres 00
P4=00 P5=00
M10=K2 Q51+
V3=M10 _ V3:D0,=@ V3:D1,=C3 V3:D1,=$ P4=V3 . Q50+ ...Q50-
,!L2
*****************************************************************
C1  V5-1,#$ V6-80,#$ ; wachtlusje
    ,!C0
*****************************************************************
C2  ; initialiserings-CALL. Zet scherm goed
    M1=K1
    V3=M1 _ V3:D0,=@ P4=V3 . Q50+ ... Q50- ,!C1 ,!$
    ,!C0
*****************************************************************
C3  ; zet cursor op de 2e regel vooraan
    Q51- .. P4=C0 .. Q50+ ... Q50- .. Q51+ ..
    ,!C0
*****************************************************************
```

Het Bestuur

Voorzitter:.....	J.W.(Hans) Ligthelm Koekoekplein 13 2802 AD Gouda 0182-51 66 97
Secretaris:.....	L (Lex) Jansen Galjoenstraat 65 3334 PD Utrecht 030-2448714
Penningmeester:..	A (Abraham) Vreugdenhil Noordlandsweg 102 2691 KN 'S-Gravenzande 0174 420361
Lid:.....	R(onald) Bons Galjoenstraat 47 3534 PC Utrecht 030-2 44 79 29
Lid:.....	D(onald) Roganti
Adviseur:	A.G.M. Goossens Sheridanzijde 79 Zoetermee 079-3 31 08 93

REDACTIE

Redacteur:	P (Paul) Smits Lijtweg 302 2341 HB Oegstgeest 071-5156090